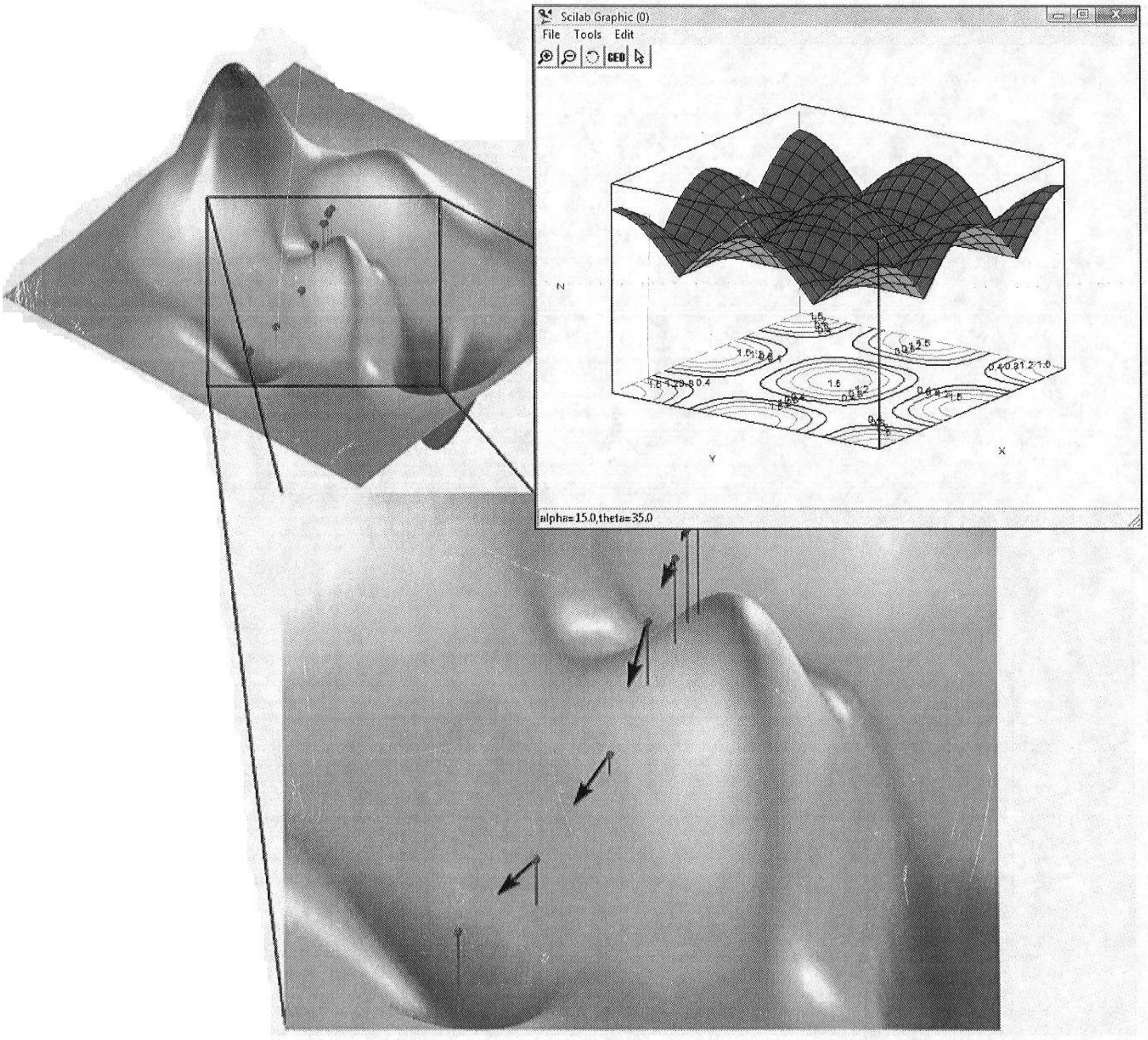
เทคนิคการหาค่าเหมาะที่สุด

Optimization Techniques

SUT - SECOND EDITION



50242961201

ธนัดชัย กุลวรวานิชพงษ์

195.-

เทคนิคการหาค่าเหมาะที่สุด
Optimization Techniques

SUT 2st Edition

ธนัดชัย กุลวรวานิชพงษ์
หน่วยวิจัยไฟฟ้ากำลังและพลังงาน
สาขาวิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเทคโนโลยีสุรนารี

สาขาวิชาวิศวกรรมไฟฟ้า
สำนักวิชาวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีสุรนารี

คำนำจากผู้เขียน

ในปัจจุบัน เทคโนโลยีแขนงต่าง ๆ มีความเจริญก้าวหน้าไปมาก โดยเฉพาะอย่างยิ่ง เทคโนโลยีด้านคอมพิวเตอร์และอุปกรณ์ช่วยคำนวณในเชิงวิทยาศาสตร์ ทำให้การออกแบบในงานวิศวกรรมมีความท้าทายและซับซ้อนมากยิ่งขึ้น การแก้ปัญหาด้วยสูตรคำนวณอย่างง่ายต่อสถานการณ์ในปัจจุบัน ซึ่งไม่สามารถตอบสนองต่อความต้องการให้ครอบคลุมในทุกด้าน การออกแบบโดยการปรับแต่งอย่างละเอียดและประณีต ในอดีตมีข้อจำกัดที่เวลาการออกแบบที่ใช้เวลานาน เนื่องจากเครื่องคำนวณมีสมรรถนะต่ำ ทำให้การออกแบบด้วยการลองผิดลองถูก (trial & error) จากประสบการณ์ของผู้ออกแบบเอง เข้ามามีบทบาทสำคัญ ทำให้เกิดปัญหาที่ตามมา 2 ประเด็นหลัก ประการแรก คือ การออกแบบที่ได้นำมาใช้งานได้ดี แต่ยังไม่ดีที่สุด ผู้ออกแบบสามารถลดต้นทุน ลดวัสดุที่ใช้ ปริมาณเชื้อเพลิง ค่าตอบแทนหรืออื่น ๆ ที่เกี่ยวข้องกับการออกแบบได้อีก บางครั้งความแตกต่างอาจจะอยู่ในปริมาณที่สูงในหลักล้านบาทต่อปีของต้นทุนการผลิตก็เป็นได้ การนำเทคนิคการหาค่าเหมาะที่สุดมาใช้งานช่วยให้การออกแบบมุ่งประเด็นไปสู่เป้าหมายที่ตั้งไว้ได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม ระเบียบวิธีการแก้ปัญหามีมากมาย ผู้ออกแบบจำเป็นต้องเข้าใจหลักการ และครอบเขตที่ใช้งานได้ของระเบียบวิธีนั้นให้ถ่องแท้เสียก่อน

เอกสารเล่มนี้ แบ่งเนื้อหาออกเป็น 9 บท ประกอบไปด้วยเทคนิคทางคณิตศาสตร์เพื่อแก้ปัญหาค่าเหมาะที่สุดในรูปแบบต่าง ๆ บทที่ 1 นำเสนอหลักการโดยทั่วไป ชี้ประเด็นให้เห็นถึงกลยุทธ์ของระเบียบวิธีการค้นหาต่าง ๆ ได้แก่ กลยุทธ์การค้นหาตามเส้น และกลยุทธ์ขอบเขตความเชื่อมั่น บทที่ 2 กล่าวถึงเทคนิคการหาค่าต่ำที่สุดของปัญหาตัวแปรหนึ่งมิติ ซึ่งถือว่าเป็นพื้นฐานที่สำคัญในการแก้ปัญหาย่อยการค้นหาตามเส้นในบทถัดไป ในบทนี้ ได้นำเสนอเทคนิคที่สำคัญและใช้งานบ่อยทั้งสิ้น 4 วิธี บทที่ 3 เป็นการแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับ โดยไม่จำกัดรูปแบบของฟังก์ชันวัตถุประสงค์ อย่างไรก็ตาม ข้อกำหนดที่ต้องคำนึงถึงก่อนการแก้ปัญหา คือ ความราบเรียบของฟังก์ชันวัตถุประสงค์และการหาค่าอนุพันธ์ได้อย่างน้อยหนึ่งอันดับ บทที่ 4 แนะนำการแก้ปัญหากำหนดการเชิงเส้น

อย่างง่าย เพื่อให้เข้าใจในหลักการ เนื่องจากปัญหาหลายรูปแบบอยู่ในรูปปัญหากำหนดการเชิงเส้นนี้ โดยเฉพาะงานด้านการวิเคราะห์ต้นทุนโดยรวมของระบบเพื่อใช้ในการออกแบบ บทที่ 5 เป็นปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับ ซึ่งครอบคลุมปัญหาในบริบทที่ไม่จำกัด ถือเป็นระเบียบวิธีการแก้ปัญหาขั้นสุดท้ายที่ประยุกต์เข้ากับปัญหาการออกแบบในงานวิศวกรรมได้ทุกประเภท ถึงกระนั้น ระเบียบวิธีเหล่านี้มีความไม่เป็นเชิงเส้น และยุ่งยากซับซ้อน ถ้ารูปแบบของปัญหาสามารถจัดรูปให้เป็นรูปแบบอื่น ๆ เช่น การแก้ปัญหาแบบไม่มีเงื่อนไข หรือปัญหากำหนดการเชิงเส้น ก็เป็นสิ่งที่น่าสนใจมากกว่า เนื่องจากความง่ายในการคำนวณหาผลเฉลย บทที่ 6 เป็นส่วนที่ชี้ให้เห็นการวิเคราะห์ปัญหาในด้านต่าง ๆ เพื่อดำเนินการขั้นตอนที่สำคัญที่สุด คือ กระบวนการสร้างปัญหาค่าเหมาะที่สุด ถ้าปัญหาที่สร้างขึ้นชัดเจนและครอบคลุมปัจจัยทุกด้าน ผลเฉลยที่ได้ย่อมมีความสมเหตุสมผลและใช้งานได้จริงในทางปฏิบัติ บทที่ 7 เป็นส่วนของการนำเทคนิคการคำนวณอย่างชาญฉลาดที่อ้างอิงเทคนิคปัญญาประดิษฐ์มาใช้งาน การค้นหาด้วยวิธีการเหล่านี้ช้ากว่ามาก แต่ได้ผลเฉลยหรือคำตอบสำหรับปัญหาที่ซับซ้อนมาก ๆ ได้ดี ซึ่งมาช่วยเติมเต็มการแก้ปัญหาในกรณีที่ปัญหาที่พิจารณามีความไม่เชิงเส้นสูง ทำให้ระเบียบวิธีทางคณิตศาสตร์ที่นำเสนอไว้ข้างต้น ไม่สามารถแก้ปัญหาได้อย่างมีประสิทธิภาพเท่าที่ควร บทที่ 8 และ 9 ถือว่าเป็นการรวบรวมปัญหาค่าเหมาะที่สุดจากงานวิจัย ที่มีการประยุกต์ใช้เทคนิคการหาค่าเหมาะที่สุด ทั้งรูปแบบทางคณิตศาสตร์และการใช้เทคนิคชาญฉลาดที่หลากหลาย เพื่อนำมาใช้เป็นแนวทางในการประยุกต์ใช้สำหรับนักศึกษา นักวิจัย และผู้สนใจ ให้เข้ากับระบบที่พิจารณามากที่สุด นอกจากนี้ เอกสารนี้ประกอบด้วยภาคผนวกรวม 3 ตอน นำเสนอองค์ความรู้พื้นฐานที่จำเป็น การใช้งานโปรแกรม SCILAB เบื้องต้น และซอร์สโคดของกล่องเครื่องมือที่ผู้เขียนได้พัฒนาขึ้นประกอบการใช้เอกสารเล่มนี้ ผู้เขียนหวังไว้เป็นอย่างยิ่งว่า จะสามารถทำให้นักศึกษา นักวิจัย และผู้สนใจทั่วไป นำเทคนิคการหาค่าเหมาะที่สุดไปใช้ในงานวิศวกรรมอย่างแพร่หลายต่อไป

ธนัดชัย กุลวรวานิชพงษ์ (14 สิงหาคม 2550)

สารบัญ

บทที่ 1 บทนำสู่ปัญหาค่าเหมาะที่สุด
(Introduction to Optimization Problems)

"An engineering work is intrinsically an optimization problem"

1.1 เกริ่นนำ (Introduction)

กระบวนการทางวิศวกรรมศาสตร์มุ่งเน้นการนำองค์ความรู้ทางด้านต่างๆ เพื่อนำมาใช้ให้เกิดประโยชน์ต่อสภาพความเป็นอยู่ การทำงานและสร้างสิ่งอำนวยความสะดวกต่างๆ เพื่อตอบสนองความต้องการของมนุษย์ตามปัจจัยสี่ เมื่อการรวมกลุ่มทางสังคมมีขนาดใหญ่ขึ้น การบริโภคทรัพยากรที่มีจำกัดทำให้เกิดผลกระทบต่าง ๆ ตามมามากมาย ทั้งทางด้านค่าใช้จ่าย ทางด้านสิ่งแวดล้อม หรืออื่น ๆ ในโลกยุคปัจจุบันที่มีการแข่งขันสูง การผลิตสินค้าหรือเครื่องอำนวยความสะดวกต่าง ๆ ต้องถูกออกแบบให้รัดกุม เหมาะสม และราคาย่อมเยา ส่งผลให้การทำงานวิศวกรรมถูกท้าทายด้วยเงื่อนไขดังกล่าว การหาค่าเหมาะที่สุดจึงเป็นสิ่งที่วิศวกรต้องศึกษา เพื่อนำแนวคิดและวิธีการแก้ปัญหาไปประยุกต์ใช้ให้เข้ากับงานที่ได้รับมอบหมาย ซึ่งอาจจะกล่าวได้ว่า งานด้านวิศวกรรมศาสตร์ คือ งานการหาค่าเหมาะที่สุดนั่นเอง

การศึกษาปัญหาค่าเหมาะที่สุดไม่สามารถดำเนินการได้โดยตรง มีความจำเป็นอย่างยิ่งที่จะต้องสร้างแบบจำลองทางคณิตศาสตร์ของปัญหาขึ้นมา ภายใต้ฟังก์ชันทางคณิตศาสตร์ที่มีคุณสมบัติที่ชัดเจน เช่น ความสามารถในการคำนวณอนุพันธ์ หรือความต่อเนื่องของฟังก์ชัน เป็นต้น ดังนั้น การศึกษาปัญหาค่าเหมาะที่สุดส่วนใหญ่จะเน้นไปที่เทคนิคการคำนวณบนพื้นฐานของฟังก์ชันคณิตศาสตร์ โดยมีข้อนิยามต่าง ๆ ดังนี้

รูปแบบของปัญหาจะถูกกำหนดผ่านปริมาณทางกายภาพใดๆ ที่ต้องการหาค่าต่ำสุดหรือสูงสุด เรียกปริมาณที่ต้องการนี้ว่า ฟังก์ชันวัตถุประสงค์ (objective function) เรียกตัวแปรหรือพารามิเตอร์ที่ใช้ในการปรับค่าว่า ตัวแปรควบคุมหรือตัวแปรออกแบบ (control variables, design variables) ซึ่งสามารถนำมาเขียนเป็นความสัมพันธ์ทางคณิตศาสตร์ในรูปแบบของปัญหาค่าเหมาะที่สุดได้ดังนี้

Minimize $f(\mathbf{x})$ // กรณีปัญหาค่าต่ำที่สุด

$\mathbf{x} \in S$ S: แทนเซตหรือปริภูมิการค้นหา

หรือ

Maximize $f(\mathbf{x})$ // กรณีปัญหาค่าสูงที่สุด

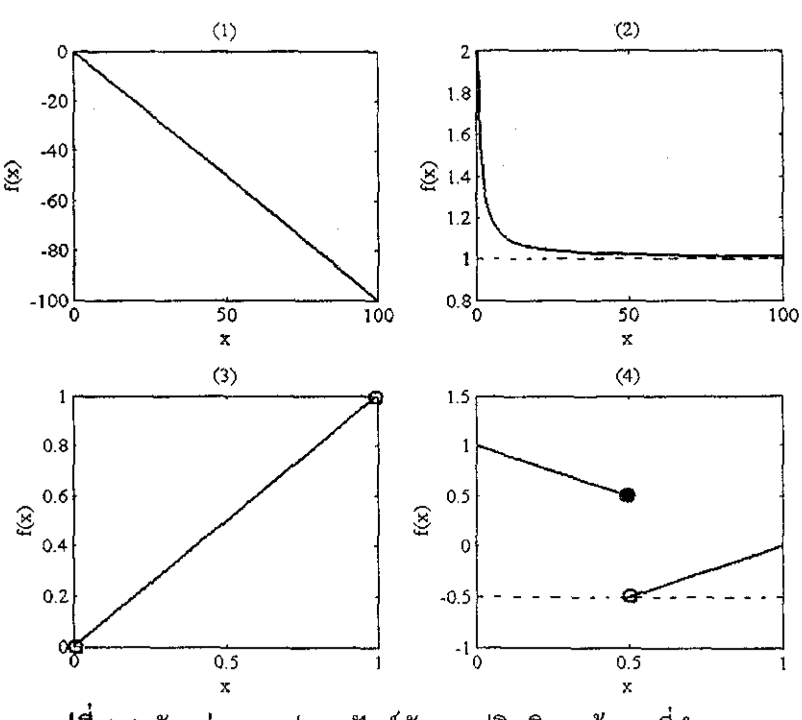
$\mathbf{x} \in S$

เพื่อให้การค้นหาจุดคำตอบของตัวแปรควบคุมเป็นไปอย่างถูกต้องและแม่นยำ การศึกษาทฤษฎีบททางคณิตศาสตร์ที่เกี่ยวพันกับการมีอยู่ของจุดต่ำสุดหรือจุดสูงสุดมีความสำคัญเป็นอันดับแรก ดังรายละเอียดต่อไปนี้

1.2 การมีอยู่ของผลเฉลยที่เหมาะที่สุด (Existence of an Optimum)

การกำหนดปัญหาค่าเหมาะที่สุดกับการมีอยู่ของผลเฉลยเป็นคนละประเด็น ปัญหาค่าเหมาะที่สุดบางอย่างที่มีข้อจำกัดมากจนเกินไปอาจจะส่งผลให้เป็นปัญหาที่ไม่มีคำตอบได้ ดังนั้นการศึกษาทฤษฎีบทพื้นฐานเพื่อตรวจสอบการมีอยู่ของผลเฉลยที่อาจจะเป็นค่าที่เหมาะสมมีจำเป็น โดยพิจารณาจากตัวอย่างต่อไปนี้

1. $f(x) = -x$, $S = [0,\infty)$ ในกรณีนี้ ฟังก์ชัน f ลดลงตามค่า x ที่เพิ่มขึ้นโดยไม่มีขอบเขตจำกัด ค่า x สามารถเพิ่มได้เรื่อยๆ และค่าของฟังก์ชันจะไม่เข้าสู่ค่าต่ำสุดใดๆ เลย
2. $f(x) = 1 + 1/x$, $S = [1,\infty)$ในกรณีนี้ ฟังก์ชัน f ลดลงตามค่า x ที่เพิ่มขึ้น เมื่อค่า x เพิ่มจากขอบเขตล่าง ค่าของฟังก์ชันจะลดลงจาก 2 ลู่เข้าสู่ 1 แต่ถึงแม้จะเพิ่ม x มากขึ้นเท่าใดก็ตาม ค่าของฟังก์ชันจะไม่มีค่าเท่ากับ 1
3. $f(x) = x$, $S = (0,1)$ ในกรณีนี้ เมื่อ x มีค่าเข้าใกล้ 0 ฟังก์ชัน f จะลดลงเข้าสู่ 0 เช่นกัน เนื่องจาก 0 อยู่นอกขอบเขตของเซต ดังนั้น ค่าของฟังก์ชันจะไม่มีค่าเป็น 0
4. $f(x) = 1 - x$ ถ้า $x \le ½$ และ $f(x) = x - 1$ ถ้า $x > ½$ โดยที่ $S = [0, 1]$



รูปที่ 1.1 ตัวอย่างกราฟของฟังก์ชันบนปริภูมิการค้นหาที่กำหนด

จากตัวอย่างดังกล่าว จะพบว่า ในข้อ 1 และ 2 เซต S ไม่มีขอบเขต ทำให้ไม่สามารถระบุจุดต่ำสุดของฟังก์ชันภายใต้การกำหนดเซตดังกล่าวได้ ตัวอย่างที่ 3 กำหนดเซต S เป็นเซตเปิด ไม่สามารถระบุจุดต่ำสุดได้เช่นกัน ในขณะที่ตัวอย่างสุดท้ายเซต S มีความไม่ต่อเนื่อง ดังนั้น ในการแก้ปัญหาค่าเหมาะที่สุดหลายตัวแปร มีความจำเป็นอย่างยิ่งที่เซต S หรือที่เรียกว่า ปริภูมิการค้นหา (searching space) ต้องมีคุณสมบัติของ***การมีขอบเขต เป็นเซตปิด หรือมีความต่อเนื่อง***เป็นลำดับแรก

การพิจารณาจุดคำตอบหรือผลเฉลยของปัญหาค่าเหมาะที่สุดนี้ ต้องอยู่ภายใต้กรอบของการกำหนดขอบเขตหรือปริภูมิในการค้นหาควบคู่กันด้วย ดังนั้น ถ้าสนใจให้ปัญหาค่าเหมาะที่สุดอยู่ในรูปการแก้ปัญหาค่าต่ำที่สุดแล้ว จุดต่ำสุดของปัญหาต้องมีคุณสมบัติดังนี้

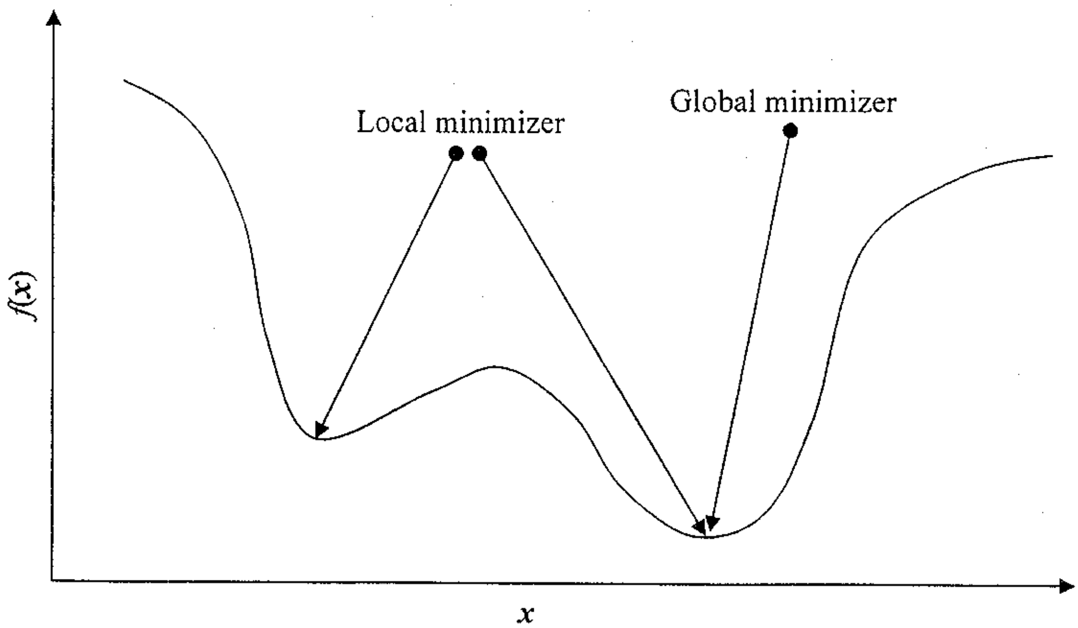
ถ้ากำหนดให้ S แทนปริภูมิการค้นหา

- ผลเฉลย x^* จะเป็นจุดต่ำสุดโดยรวมของปัญหา (global minimizer) ก็ต่อเมื่อ

  $f(x^*) \le f(x)$ สำหรับ $\forall x \in S$

- ถ้ากำหนดให้ $\aleph(x^*)$ แทนเซตข้างเคียง (neighborhood set) ของผลเฉลย x^* แล้ว ผลเฉลย x^* จะเป็นจุดต่ำสุดเฉพาะถิ่นของปัญหา (local minimizer) ก็ต่อเมื่อ

  $f(x^*) \le f(x)$ สำหรับ $\forall x \in \aleph(x^*)$



รูปที่ 1.2 จุดต่ำสุดโดยรวมและจุดต่ำสุดเฉพาะถิ่น

1.3 การหาจุดต่ำสุดเฉพาะถิ่น (Recognizing a Local Minimum)

ถ้าฟังก์ชันใดๆ นิยามขึ้นภายใต้ปริภูมิการค้นหา S ที่มีความต่อเนื่อง และเป็นเซตปิด การระบุตำแหน่งของจุดต่ำสุดเฉพาะถิ่นสามารถทำได้โดยใช้การประเมินค่าเกรเดียน (gradient) และ

เฮสเซียน (hessian) ซึ่งอาศัยข้อมูลการคำนวณอนุพันธ์ของฟังก์ชัน ดังนั้น ฟังก์ชันที่กำหนดให้เป็นฟังก์ชันวัตถุประสงค์อย่างน้อยที่สุดต้องสามารถหาค่าอนุพันธ์ได้สองอันดับ

ถ้ากำหนดให้ จุดคำตอบที่เหมาะที่สุดของปัญหามีค่าเป็น x^* โดยใช้การประมาณค่าด้วยอนุกรมเทย์เลอร์ของฟังก์ชันที่พิจารณารอบๆ จุด x^* จะได้ว่า

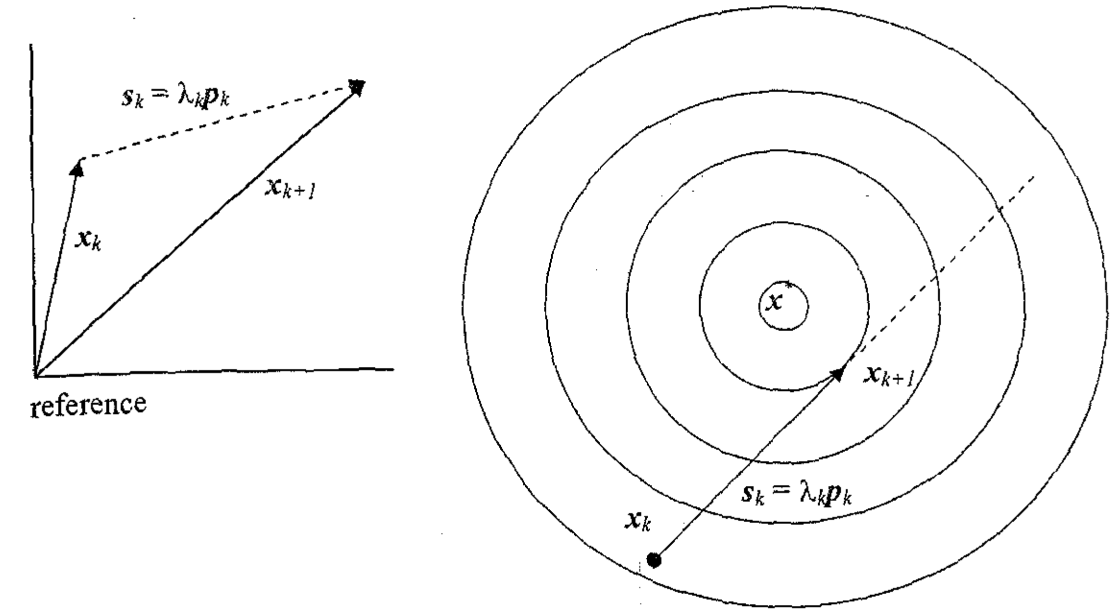
$$f(x^*) < f(x^* + s) = f(x^*) + s^T \nabla f(x^*) + \frac{1}{2} s^T \nabla^2 f(\xi) s \tag{1.1}$$

จุดต่ำสุดมีคุณสมบัติที่สำคัญคือ $\nabla f(x^*) = 0$ เงื่อนไขนี้ เป็นเงื่อนไขแรกที่จำเป็นต้องกำหนดไว้ในการแก้ปัญหาการหาค่าเหมาะที่สุดทุกประเภท เรียกว่า เงื่อนไขที่จำเป็นอันดับหนึ่ง (first-order necessary condition) แต่การใช้เงื่อนไขนี้เพียงอย่างเดียวไม่เพียงพอที่จะระบุว่าจุดดังกล่าวเป็นจุดต่ำสุดได้ เนื่องจาก $\nabla f(x^*) = 0$ บอกได้เพียงว่าจุดดังกล่าวเป็นจุดหยุดนิ่ง (stationary point) อาจจะเป็นจุดต่ำสุด จุดสูงสุด หรือจุดเปลี่ยนโค้ง ก็ได้ ดังนั้น ต้องการเงื่อนไขเพิ่มเติมในการแก้ปัญหา

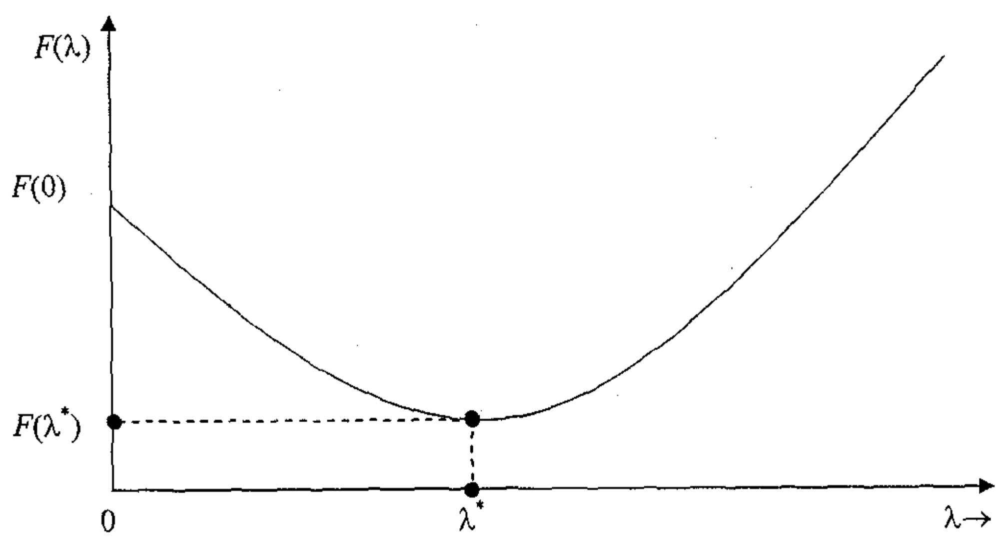
จากคุณสมบัติของจุดต่ำสุด $f(x^*) < f(x^* + s)$ ทำให้พจน์ที่สองทางด้านขวามือ ในสมการข้างต้นต้องมีค่ามากกว่าศูนย์จึงจะสอดคล้องกับเงื่อนไขนี้ นั่นคือ $\nabla^2 f(x^*)$ ต้องเป็น *psdf* (positive semi-definite matrix) นั่นคือ $v^T \nabla^2 f(x^*) v \geq 0$ และเป็นจริงสำหรับ $v^T \nabla^2 f(\xi) v \geq 0$ ด้วย ถ้า $\|\xi - x^*\|$ มีค่าน้อยๆ

ดังนั้น ถ้าเลือก s ให้เหมาะสมแล้ว ค่า ξ จะมีค่าเข้าใกล้ x^* มากเพียงพอที่จะรับประกันได้ว่า $f(x^*) < f(x)$ สำหรับทุกๆ ค่าของ x ซึ่งเงื่อนไขนี้เรียกว่า เงื่อนไขที่เพียงพออันดับสอง (second-order sufficient condition)

ในช่วง 40 ปี ที่ผ่านมานี้ ได้มีการพัฒนาอัลกอริทึมสำหรับการหาค่าเหมาะที่สุดสำหรับฟังก์ชันที่ต่อเนื่องและมีความราบเรียบ อัลกอริทึมทั้งหมดอาจจะเขียนในรูปของการหาค่าสูงที่สุดหรือการหาค่าต่ำที่สุดก็ได้ อย่างไรก็ตาม ปัญหาค่าเหมาะที่สุดในรูปแบบของปัญหาค่าต่ำสุดได้รับความนิยมกันอย่างแพร่หลาย และในที่นี้จะอ้างอิงรูปแบบของปัญหาเป็นแบบการหาค่าต่ำที่สุด อัลกอริทึมการค้นหาจุดคำตอบเริ่มต้นด้วยการกำหนดจุดเริ่มต้น x_0 จากนั้น การประมาณค่าผลเฉลยด้วยเทคนิคต่าง ๆ จะถูกนำมาใช้ ด้วยการสร้างลำดับของผลเฉลย $\{x_0, x_1, \ldots, x_k, \ldots, x_\infty\}$ ลำดับที่สร้างขึ้นนี้ ต้องมีคุณสมบัติให้ค่าของฟังก์ชันวัตถุประสงค์ลดลงในทุก ๆ รอบการคำนวณที่ใช้ไป รูปแบบของอัลกอริทึมทั้งหมดอาจจะพิจารณาได้เป็นสองแนวทางเพื่อสร้างลำดับที่มีคุณสมบัติดังกล่าว ได้แก่ กลยุทธ์การค้นหาตามเส้น (line search strategy) และกลยุทธ์ขอบเขตความเชื่อมั่น (trust region strategy) ดังนี้



รูปที่ 1.3 กลยุทธ์การค้นหาตามเส้น



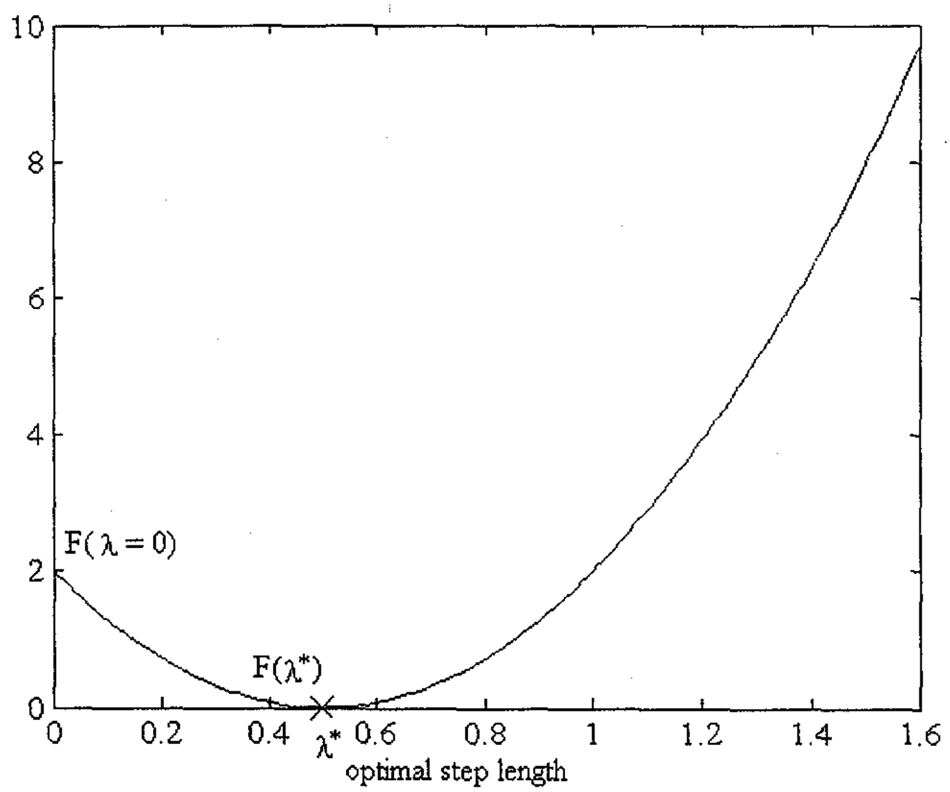
รูปที่ 1.4 ปัญหาย่อยการหาช่วงก้าวที่เหมาะสม

○ **กลยุทธ์การค้นหาตามเส้น (line search strategy)**

กลยุทธ์การค้นหาตามเส้นนี้ อาศัยหลักการสร้างทิศทางการค้นหา (search direction: p_k) ที่เหมาะสม โดยทิศทางการค้นหานี้ต้องมีคุณสมบัติลาดลง (descent direction หรือ down-hill direction) กำหนดให้ทิศทางการค้นหาดังกล่าวมีค่าเป็น p_k ผลเฉลยในรอบถัดไปคำนวณได้จากความสัมพันธ์ต่อไปนี้ $x_{k+1} = x_k + \lambda_k p_k$ พิจารณาได้ดังรูปที่ 1.3 ถ้ากำหนดทิศทางการค้นหาได้อย่างเหมาะสม ปัญหาที่เหลืออยู่จะถูกลดรูปลงเป็นปัญหาย่อยการหาช่วงก้าวที่เหมาะสม (optimal step length: λ) เพื่อให้การลดค่าลงของฟังก์ชันมีค่ามากที่สุดในทิศทางการค้นหา p_k ดังนั้นเพื่อให้สะดวกในการกำหนดสัญลักษณ์ กำหนดให้ $\lambda = \lambda_k$ จะได้ปัญหาย่อยการค้นหาตามเส้นดังนี้

$$\text{Minimize} \qquad F(\lambda) \equiv f(\boldsymbol{x}_k + \lambda \boldsymbol{p}_k) \qquad\qquad (1.2)$$
$$\lambda > 0$$

ในรอบการคำนวณที่ $k+1$ ใดๆ สิ่งที่ทราบค่า ได้แก่ ผลเฉลยในรอบก่อนหน้า $\boldsymbol{x}_k$ และทิศทางการค้นหา $\boldsymbol{p}_k$ จะเหลือตัวแปรที่ไม่ทราบค่าเพียงตัวเดียว คือ λ เท่านั้น เรียกว่า ปัญหาการหาช่วงก้าวที่เหมาะสม เมื่อปรับค่า λ โดยเริ่มต้นจาก 0 ให้มีค่าเป็นบวกและเพิ่มขึ้นอย่างต่อเนื่อง ภายใต้ทิศทางการค้นหา $\boldsymbol{p}_k$ ที่กำหนด จะทำให้ค่าของฟังก์ชัน $F(\lambda)$ ลดค่าลงอย่างต่อเนื่องจนถึงค่า $\lambda = \lambda^*$ ณ จุดนี้ ค่าของฟังก์ชันวัตถุประสงค์จะลดค่าลงมากที่สุดในทิศทางการค้นหา $\boldsymbol{p}_k$ ถ้าเพิ่มค่า λ ให้มากกว่านี้ ค่าของฟังก์ชันวัตถุประสงค์จะมีค่าเพิ่มขึ้น ดังรูปที่ 1.4 เพื่อให้มองเห็นภาพที่ชัดเจน ศึกษาเพิ่มเติมจากตัวอย่างต่อไปนี้



รูปที่ 1.5 ช่วงก้าวที่เหมาะสมสำหรับตัวอย่างที่ 1.1

ตัวอย่างที่ 1.1 กำหนดฟังก์ชันวัตถุประสงค์เป็น $f(x_1, x_2) = x_1^2 + x_2^2$ และทิศทางการค้นหากำหนดให้มีค่าเป็น $\boldsymbol{p} = [-2\ -2]^T$ พิจารณาจากจุดเริ่มต้น $\boldsymbol{x}_0 = [1.0 \quad 1.0]^T$; $\boldsymbol{x} = [x_1 \quad x_2]^T$

วิธีทำ ณ ผลเฉลยเริ่มต้น จะได้ฟังก์ชันวัตถุประสงค์เป็น

$$f(\boldsymbol{x}_0) = (1.0)^2 + (1.0)^2 = 2.0$$

ปัญหาค่าเหมาะที่สุดนี้ เริ่มต้นจากการปรับปรุงผลเฉลยจากจุดเริ่มต้น โดย

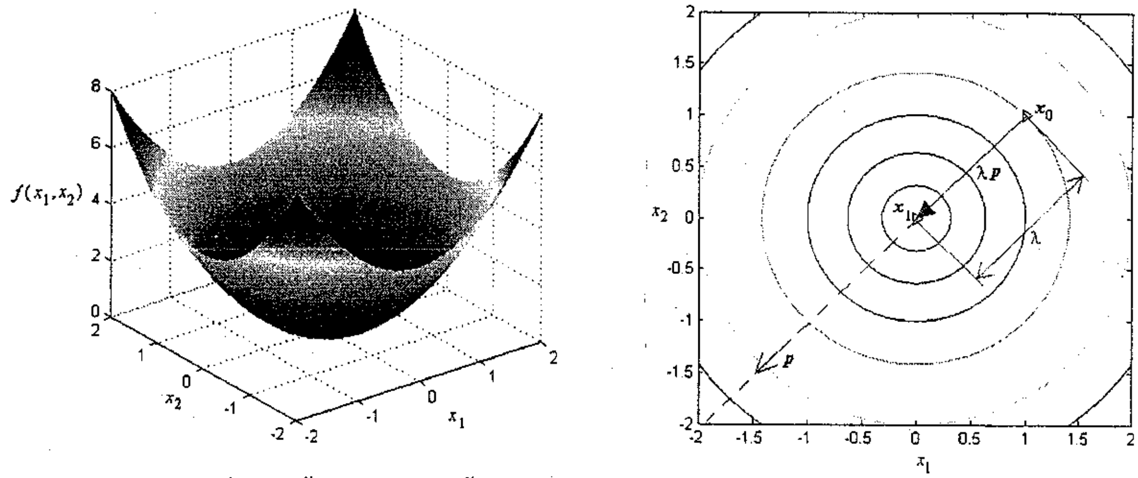
$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \lambda \boldsymbol{p} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \lambda \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \lambda \begin{bmatrix} -2 \\ -2 \end{bmatrix} = \begin{bmatrix} 1-2\lambda \\ 1-2\lambda \end{bmatrix}$$

แทนค่า ในสมการของปัญหาย่อยการหาช่วงก้าวที่เหมาะสมจะได้ว่า

$$\text{Minimise} \qquad F(\lambda) = f(\boldsymbol{x_0} + \lambda\boldsymbol{p}) = f\left(\begin{bmatrix}1-2\lambda\\1-2\lambda\end{bmatrix}\right) = (1 - 2\lambda)^2 + (1 - 2\lambda)^2$$

$$\lambda > 0$$

จะพบว่า $F(\lambda)$ ที่ได้ เป็นฟังก์ชันกำลังสอง ดังนั้น การหาค่าช่วงก้าวที่เหมาะสมทำได้ไม่ยากนัก พิจารณาจากกราฟในรูปที่ 1.5 จะได้ค่า $\lambda^* = 0.5$ ส่งผลให้ $\boldsymbol{x_1} = [0.0 \;\; 0.0]^T = \boldsymbol{x}^*$ ผลเฉลยที่ถูกปรับปรุงนี้ให้ค่า $f(\boldsymbol{x_1}) = (0.0)^2 + (0.0)^2 = 0.0$ จะเห็นได้ว่า ค่าของฟังก์ชันลดลงจาก $f(\boldsymbol{x_0}) \to f(\boldsymbol{x_1})$ นั่นคือ ลดลงจาก 2.0 → 0.0 ดังรูปที่ 1.6



รูปที่ 1.6 พื้นผิวและเส้นชั้นความสูง (contour) ของตัวอย่างที่ 1.1

อย่างไรก็ตาม ตัวอย่างนี้ต้องการนำเสนอภาพโดยรวมของกลยุทธ์การค้นหาตามเส้น ซึ่งก็คือ เทคนิคการหาช่วงก้าวที่เหมาะสมตามทิศทางการค้นหาที่ถูกกำหนดไว้นั่นเอง ทำให้การคำนวณ λ^* ทำได้ง่าย แต่ปัญหาค่าเหมาะที่สุดโดยทั่วไป สมการ $F(\lambda)$ ไม่จำเป็นต้องอยู่ในรูปของฟังก์ชันกำลังสองเสมอไป ทำให้การหาค่ายุ่งยากขึ้นไปอีก

ตัวอย่างที่ 1.2 กำหนดฟังก์ชันวัตถุประสงค์เป็น $f(x_1,x_2) = (x_1 - 1.5x_2)^2 + (x_2 - 1)^2$ และทิศทางการค้นหาเป็น $\boldsymbol{p} = [0 \;\; 2]^T$ พิจารณาจากจุดเริ่มต้น $\boldsymbol{x_0} = [0.0 \;\; 0.0]^T$;

วิธีทำ ณ ผลเฉลยเริ่มต้น จะได้ฟังก์ชันวัตถุประสงค์เป็น $f(\boldsymbol{x_0}) = 1.0$

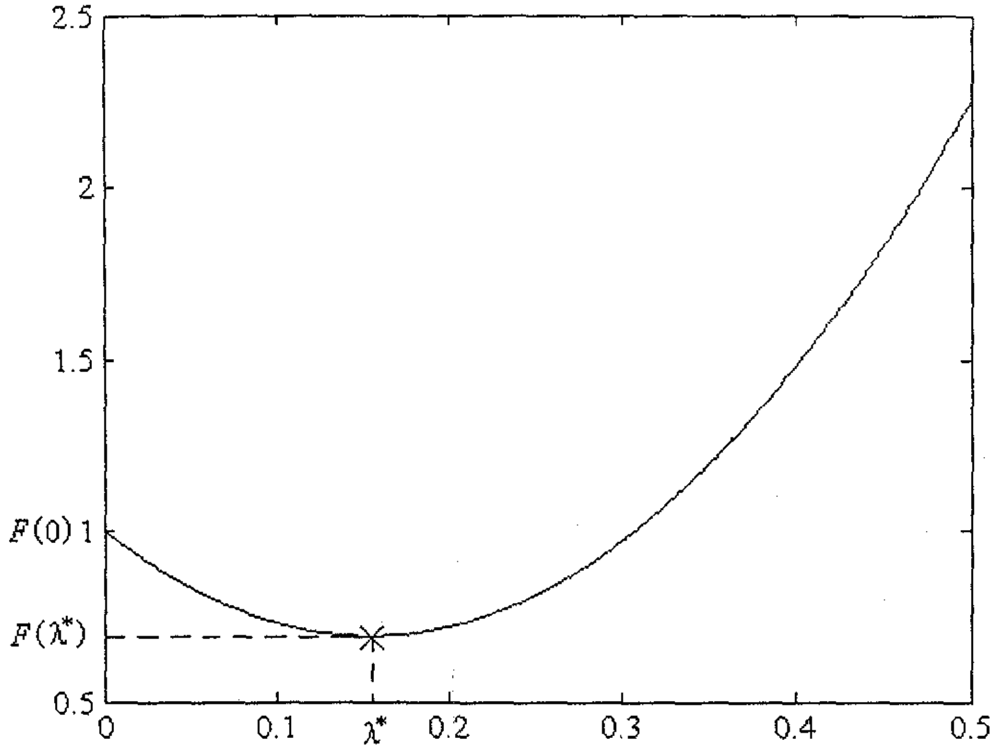
ปัญหาค่าเหมาะที่สุดนี้ เริ่มต้นจากการปรับปรุงผลเฉลยจากจุดเริ่มต้น โดย

$$\boldsymbol{x_1} = \boldsymbol{x_0} + \lambda\boldsymbol{p} = \begin{bmatrix}x_1\\x_2\end{bmatrix} + \lambda\begin{bmatrix}p_1\\p_2\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix} + \lambda\begin{bmatrix}0\\2\end{bmatrix} = \begin{bmatrix}0\\2\lambda\end{bmatrix}$$

แทนค่า ในสมการของปัญหาย่อยการหาช่วงก้าวที่เหมาะสมจะได้ว่า

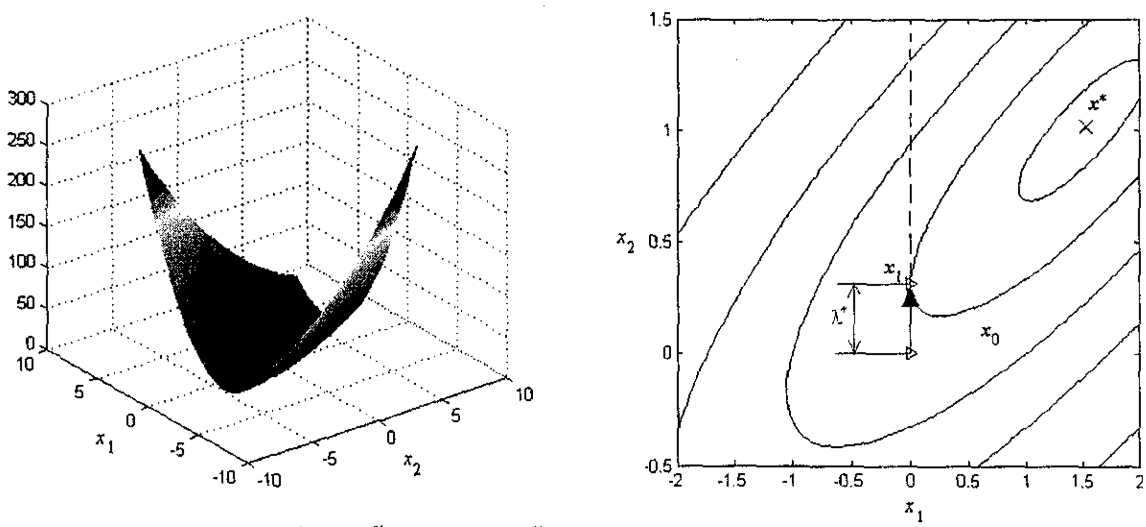
$$\text{Minimise} \qquad F(\lambda) = f(\boldsymbol{x_0} + \lambda\boldsymbol{p}) = f\left(\begin{bmatrix}0\\2\lambda\end{bmatrix}\right) = 3\lambda^2 + (2\lambda - 1)^2$$

$$\lambda > 0$$

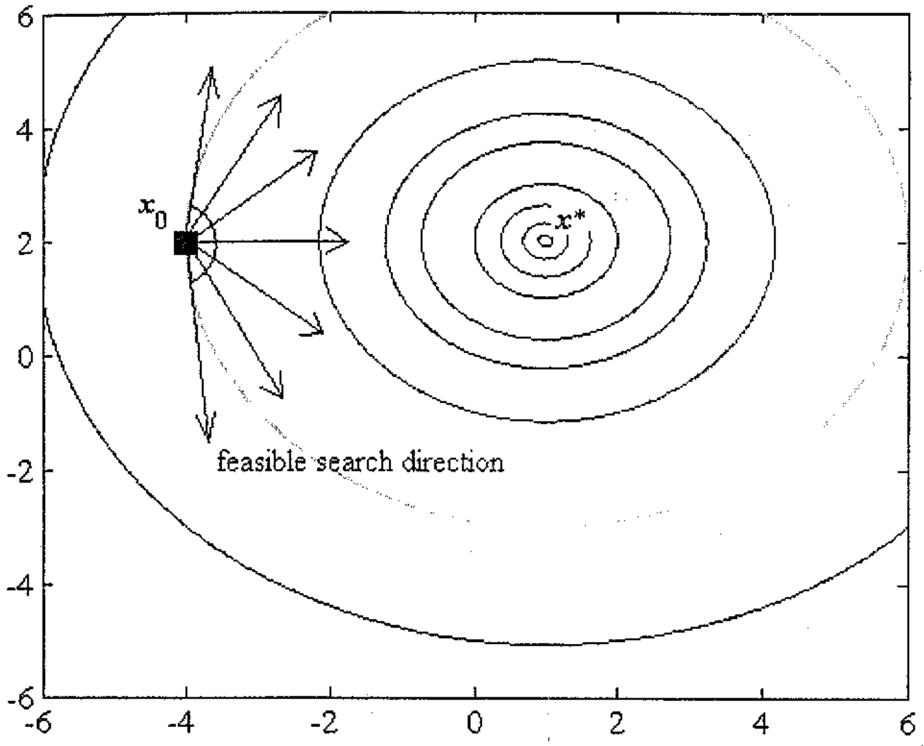


รูปที่ 1.7 ช่วงก้าวที่เหมาะสมสำหรับตัวอย่างที่ 1.2

พิจารณาจากกราฟในรูปที่ 1.7 จะได้ค่า $\lambda^* = 2/13 = 0.1538$ ส่งผลให้ $\boldsymbol{x}_1 = [0.0\ \ 0.3076]^T$ ผลเฉลยที่ถูกปรับปรุงนี้ให้ค่า $f(\boldsymbol{x}_1) = 0.6923$ จะเห็นได้ว่า ค่าของฟังก์ชันลดลงจาก $f(\boldsymbol{x}_0) \rightarrow f(\boldsymbol{x}_1)$ นั่นคือ ลดลงจาก $1.0 \rightarrow 0.6923$ ดังรูปที่ 1.8



รูปที่ 1.8 พื้นผิวและเส้นชั้นความสูงของตัวอย่างที่ 1.2



รูปที่ 1.9 ทิศทางการค้นหาที่เป็นไปได้

จะพบว่า การแก้ปัญหาย่อยช่วงก้าวที่เหมาะสมนี้จะมีผลเฉลย ($\lambda^* > 0$) ก็ต่อเมื่อ ทิศทางการค้นหามีคุณสมบัติลาดลงเท่านั้น ทิศทางการค้นหาที่มีคุณสมบัติดังกล่าว จะเรียกว่า ทิศทางที่เป็นไปได้ (feasible search direction) ในการค้นพบจุดต่ำสุดนั่นเอง พิจารณาทิศทางที่เป็นไปได้ดังรูปที่ 1.9

○ **กลยุทธ์ขอบเขตความเชื่อมั่น (trust region strategy)**

กลยุทธ์นี้ อาศัยหลักการสร้างแบบจำลองทางคณิตศาสตร์ของฟังก์ชันวัตถุประสงค์ในรูปแบบของฟังก์ชันกำลังสองรอบๆ จุดที่พิจารณา x_k ใด ๆ นั่นคือ

$$f(x) \approx q(x)$$

$$f(x_k + s_k) = f(x_k) + s_k^T \nabla f(x_k) + \frac{1}{2} s_k^T \nabla^2 f(x_k) s_k \tag{1.3}$$

ถ้าจุดเริ่มต้น x_k ที่พิจารณาอยู่ใกล้จุดต่ำสุด x^* มากพอ อนุกรมเทย์เลอร์สามารถนำมาใช้เป็นตัวประมาณค่าฟังก์ชันต่อเนื่องดังกล่าวได้ โดยการพิจารณาถึงพจน์อนุพันธ์อันดับที่สองจะได้แบบจำลองของฟังก์ชันกำลังสอง เรียก $\nabla f(x_k)$ ว่า เกรเดียนต์ (gradient) ใช้สัญลักษณ์ $\boldsymbol{g}(x_k)$ และเรียก $\nabla^2 f(x_k)$ ว่า เฮสเซียน (Hessian) ใช้สัญลักษณ์ $\boldsymbol{H}(x_k)$

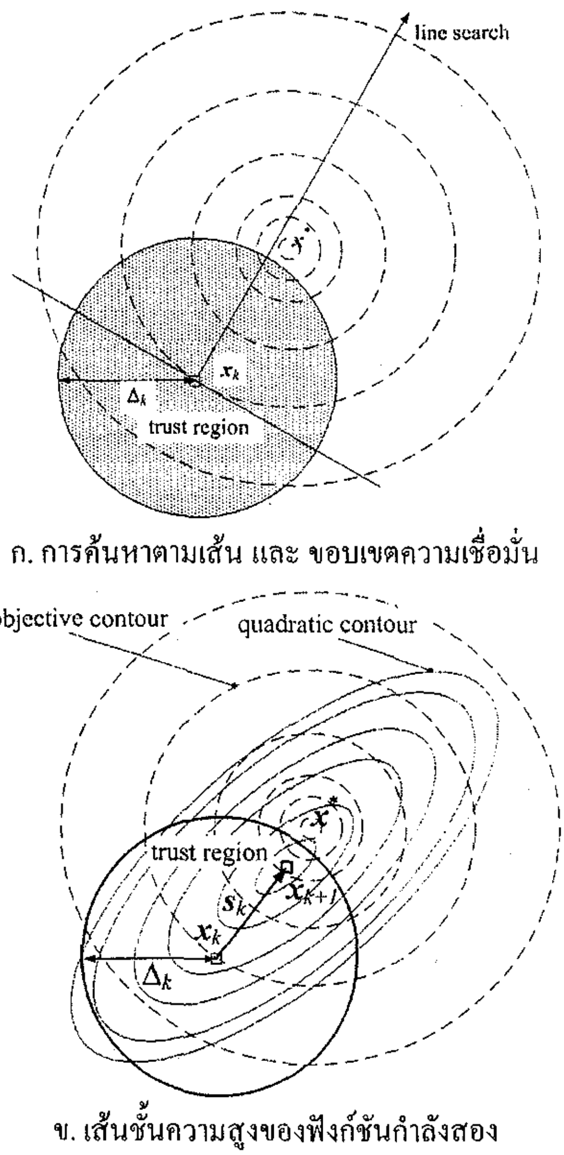
$$f(x_k + s_k) = f(x_k) + s_k^T \boldsymbol{g}(x_k) + \frac{1}{2} s_k^T \boldsymbol{H}(x_k) s_k \tag{1.4}$$

อย่างไรก็ตาม $f(x_k)$ ถือเป็นค่าคงที่ ไม่จำเป็นต้องนำมาคิดในสมการจะได้ปัญหาขอบเขตความเชื่อมั่นดังนี้

$$\text{Minimize} \qquad q(s_k) = s_k^T g(x_k) + \frac{1}{2} s_k^T H(x_k) s_k \tag{1.5}$$

$$\|s_k\| \le \Delta_k$$

โดยที่ Δ_k แทนขอบเขตความเชื่อมั่นของผลเฉลยที่จะเป็นจุดคำตอบ เนื่องจากที่ระยะห่างมากกว่านี้ ส่งผลให้ความคลาดเคลื่อนจากการประมาณค่ามากเกินไป ไม่เป็นผลดีต่อการปรับปรุงผลเฉลย พิจารณาจากรูปที่ 1.10



ก. การค้นหาตามเส้น และ ขอบเขตความเชื่อมั่น

ข. เส้นชั้นความสูงของฟังก์ชันกำลังสอง

รูปที่ 1.10 หลักการของกลยุทธ์ขอบเขตความเชื่อมั่น

รูปที่ 1.10-ข แสดงเส้นชั้นความสูงของฟังก์ชันกำลังสองที่ใช้สำหรับประมาณค่าฟังก์ชันวัตถุประสงค์ จะพบว่า เส้นชั้นความสูงของฟังก์ชันกำลังสองนี้จะให้จุดต่ำสุดออกมาค่าหนึ่งตาม

หลักการหาจุดต่ำสุดของฟังก์ชันกำลังสอง ถ้าจุดต่ำสุดที่ได้จากการประมาณค่ากำลังสองนี้ อยู่**ภายในขอบเขตความเชื่อมั่น** $\|s_k\| \le \Delta_k$ จะกำหนดให้เป็นผลเฉลยที่ได้รับการปรับปรุง ในทาง**กลับกัน** ถ้าจุดต่ำสุดนี้อยู่นอกขอบเขตความเชื่อมั่น $\|s_k\| > \Delta_k$ เวกเตอร์ผลต่าง s_k จะถูกกำหนดโดย**การลดขนาด**ลงให้มีค่าอยู่ในขอบเขตความเชื่อมั่น $\|s_k\| = \Delta_k$ พิจารณาจากตัวอย่างต่อไปนี้

ตัวอย่างที่ 1.3 กำหนดฟังก์ชันวัตถุประสงค์เป็น $f(x_1, x_2) = x_1^2 + x_2^2$ เริ่มต้นกระบวนการค้นหา**จากจุดเริ่มต้น** $x_0 = [1.0 \quad 1.0]^T$ ประมาณค่าฟังก์ชันวัตถุประสงค์ในรูปของฟังก์ชันกำลังสองโดย**ใช้การคำนวณเกรเดียนต์และเฮสเซียน** จะได้ว่า

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}; \boldsymbol{H}(\boldsymbol{x}) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

นั่นคือ

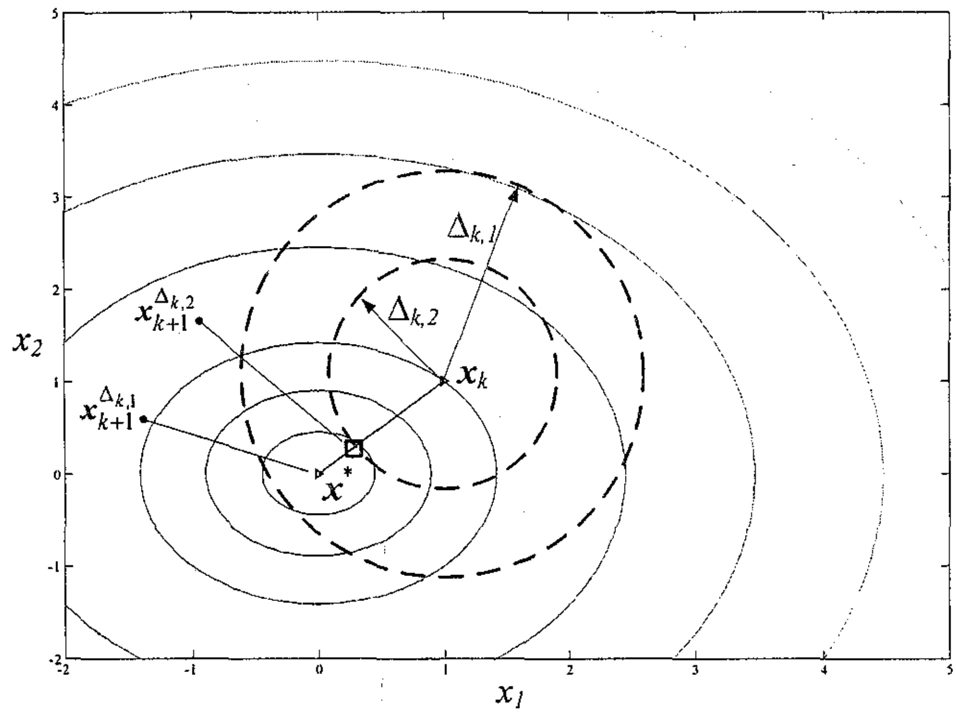
$$\begin{aligned} q(\boldsymbol{s}_k) &= f(\boldsymbol{x}_k) + \boldsymbol{s}_k^T \boldsymbol{g}(\boldsymbol{x}_k) + \frac{1}{2}\boldsymbol{s}_k^T \boldsymbol{H}(\boldsymbol{x}_k)\boldsymbol{s}_k \\ &= f\left(\begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}\right) + \boldsymbol{s}_k^T \boldsymbol{g}\left(\begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}\right) + \frac{1}{2}\boldsymbol{s}_k^T \boldsymbol{H}\left(\begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}\right)\boldsymbol{s}_k \\ &= (2.0) + \boldsymbol{s}_k^T \begin{bmatrix} 2.0 \\ 2.0 \end{bmatrix} + \frac{1}{2}\boldsymbol{s}_k^T \begin{bmatrix} 2.0 & 0.0 \\ 0.0 & 2.0 \end{bmatrix}\boldsymbol{s}_k \\ &= s_1^2 + s_2^2 + 2s_1 + 2s_2 + 2 \end{aligned}$$

จากการคำนวณจุดต่ำสุดของฟังก์ชันกำลังสอง จะได้ว่า

$$\frac{\partial}{\partial \boldsymbol{s}_k} q(\boldsymbol{s}_k) = \boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{H}(\boldsymbol{x}_k)\boldsymbol{s}_k = 0$$

$$\therefore \boldsymbol{s}_k = -\{\boldsymbol{H}(\boldsymbol{x}_k)\}^{-1} \boldsymbol{g}(\boldsymbol{x}_k) = -\begin{bmatrix} 2.0 & 0.0 \\ 0.0 & 2.0 \end{bmatrix}^{-1} \begin{bmatrix} 2.0 \\ 2.0 \end{bmatrix} = \begin{bmatrix} -1.0 \\ -1.0 \end{bmatrix}$$

จากรูปที่ 1.11 ถ้ากำหนดขอบเขตความเชื่อมั่นมีค่ากว้างมากพอ จุดต่ำสุดของฟังก์ชันกำลังสองที่ประมาณได้จะอยู่ภายในขอบเขตที่กำหนด แต่ถ้ากำหนดขอบเขตความเชื่อมั่นแคบเกินไปย่อมเสียโอกาสการได้มาซึ่งการประมาณจุดคำตอบที่ดีกว่า ในตัวอย่างนี้ จะเห็นได้ชัดเจนว่าฟังก์ชันวัตถุประสงค์ที่กำหนดเป็นฟังก์ชันกำลังสอง ทำให้การประมาณค่าของฟังก์ชันวัตถุประสงค์ดังกล่าวด้วยฟังก์ชันกำลังสองมีความแม่นตรง และจะได้ผลเฉลยเหมาะที่สุดโดยใช้การวนรอบเพียงรอบเดียวเท่านั้น



รูปที่ 1.11 แผนภาพขอบเขตความเชื่อมั่นของตัวอย่างที่ 1.3

ตัวอย่างที่ 1.4 กำหนดฟังก์ชันวัตถุประสงค์เป็น $f(x_1, x_2) = x_1^2 + x_2^2 - sin(x_1 x_2)$ เริ่มต้นการค้นหาจากจุดเริ่มต้น x_0 = $[1.0 \ 1.0]^T$ ประมาณค่าฟังก์ชันวัตถุประสงค์ในรูปของฟังก์ชันกำลังสองโดยใช้การคำนวณเกรเดียนต์และเฮสเซียน จะได้ว่า

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} 2x_1 - x_2 \cos(x_1 x_2) \\ 2x_2 - x_1 \cos(x_1 x_2) \end{bmatrix};$$

$$\boldsymbol{H}(\boldsymbol{x}) = \begin{bmatrix} 2 + x_2^2 \sin(x_1 x_2) & x_1 x_2 \sin(x_1 x_2) - \cos(x_1 x_2) \\ x_1 x_2 \sin(x_1 x_2) - \cos(x_1 x_2) & 2 + x_1^2 \sin(x_1 x_2) \end{bmatrix}$$

นั่นคือ

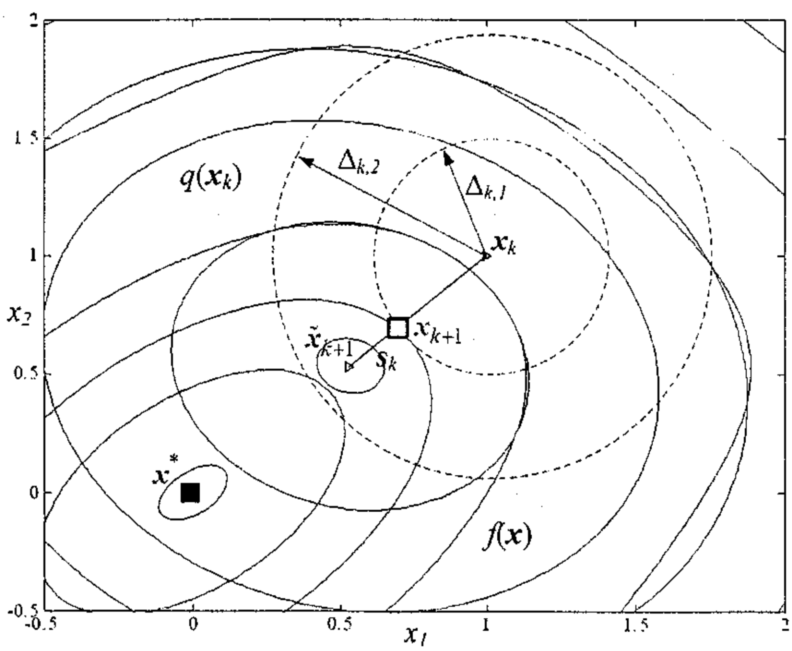
$$\begin{aligned} q(\boldsymbol{s}_k) &= f(\boldsymbol{x}_k) + \boldsymbol{s}_k^T \boldsymbol{g}(\boldsymbol{x}_k) + \frac{1}{2}\boldsymbol{s}_k^T \boldsymbol{H}(\boldsymbol{x}_k)\boldsymbol{s}_k \\ &= f\left(\begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}\right) + \boldsymbol{s}_k^T \boldsymbol{g}\left(\begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}\right) + \frac{1}{2}\boldsymbol{s}_k^T \boldsymbol{H}\left(\begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}\right)\boldsymbol{s}_k \\ &= (1.16) + \boldsymbol{s}_k^T \begin{bmatrix} 1.46 \\ 1.46 \end{bmatrix} + \frac{1}{2}\boldsymbol{s}_k^T \begin{bmatrix} 2.84 & 0.30 \\ 0.30 & 2.84 \end{bmatrix}\boldsymbol{s}_k \\ &= 1.42 s_1^2 + 1.42 s_2^2 + 0.3 s_1 s_2 + 1.46 s_1 + 1.46 s_2 + 1.16 \end{aligned}$$

จากการคำนวณจุดต่ำสุดของฟังก์ชันกำลังสอง จะได้ว่า

$$\frac{\partial}{\partial \boldsymbol{s}_k} q(\boldsymbol{s}_k) = \boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{H}(\boldsymbol{x}_k)\boldsymbol{s}_k = 0$$

$$\therefore \boldsymbol{s}_k = -\{\boldsymbol{H}(\boldsymbol{x}_k)\}^{-1}\boldsymbol{g}(\boldsymbol{x}_k) = -\begin{bmatrix} 2.84 & 0.30 \\ 0.30 & 2.84 \end{bmatrix}^{-1}\begin{bmatrix} 1.46 \\ 1.46 \end{bmatrix} = \begin{bmatrix} -0.47 \\ -0.47 \end{bmatrix}$$

จากกราฟที่แสดงในรูปที่ 1.12 จะพบว่า ในกรณีนี้ เส้นโครงร่างของฟังก์ชันวัตถุประสงค์และเส้นโครงร่างของฟังก์ชันกำลังสองที่ใช้ประมาณค่ามีความแตกต่างกัน จุดต่ำสุดของฟังก์ชันกำลังสอง $\tilde{x}_{k+1}$ อยู่ห่างจากผลเฉลยเหมาะที่สุด x^* อย่างชัดเจน เมื่อกำหนด Δ_k แคบเกินไป จะทำให้เสียโอกาสในการปรับปรุงจุดคำตอบให้ใกล้ผลเฉลยเหมาะที่สุด แต่การปรับให้ Δ_k มีค่ามากเกินไป ก็ไม่ส่งผลดีแต่อย่างใด การประมาณค่าฟังก์ชันวัตถุประสงค์ที่มีความไม่เป็นเชิงเส้นสูงด้วยฟังก์ชันกำลังสองนั้น ไม่สามารถทำได้อย่างแม่นยำถ้าจุดที่พิจารณา x_k มีระยะห่างจาก x^* มากเกินไป ดังนั้น ความเชื่อมั่นในแบบจำลองฟังก์ชันกำลังสองต้องมีขอบเขตจำกัด การกำหนด ได้อย่างเหมาะสมย่อมทำให้ผลเฉลยลู่เข้าสู่ผลเฉลยเหมาะที่สุดอย่างรวดเร็ว ในส่วนนี้จะไม่พิจารณารายละเอียดการกำหนด Δ_k แต่อย่างใด จะนำเสนอเฉพาะหลักการของกลยุทธ์ขอบเขตความเชื่อมั่นเท่านั้น



รูปที่ 1.12 แผนภาพขอบเขตความเชื่อมั่นfh;pc[[0e]v'de]y'lv'

1.4 อัตราการลู่เข้า

ระเบียบวิธีต่าง ๆ ที่จะนำเสนอในเอกสารฉบับนี้มีประสิทธิภาพในการค้นหาจุดคำตอบได้แตกต่างกัน อัตราการลู่เข้าเป็นตัวบ่งชี้ถึงประสิทธิภาพดังกล่าว โดยระเบียบวิธีต่าง ๆ จะมีรูปแบบหรืออาศัยหลักการสร้างชุดของลำดับผลเฉลยเพื่อประมาณจุดคำตอบ ถ้าลำดับดังกล่าวที่ถูกสร้างขึ้นลู่เข้าสู่ค่า ๆ หนึ่งซึ่งใกล้กับจุดคำตอบมากเพียงพอ ระเบียบวิธีนั้นจะมีคุณสมบัติลู่เข้า อย่างไรก็ตาม

ถึงแม้ว่าแต่ละวิธีจะมีคุณสมบัติลู่เข้า จำนวนของลำดับที่ถูกสร้างขึ้นจะบ่งชี้ถึงสมรรถนะของระเบียบวิธีนั้น ๆ โดยระเบียบวิธีที่ลู่เข้าสู่จุดคำตอบด้วยจำนวนลำดับที่น้อยกว่าย่อมมีอัตราการลู่เข้าที่รวดเร็วกว่า ในส่วนนี้ จะนำเสนอตัวชี้วัดอัตราการลู่เข้าดังกล่าว

กำหนดให้ $\{x_k\}$ เป็นลำดับของผลเฉลยที่ใช้ประมาณจุดคำตอบ $e_k = x_k - x^*$ แทนค่าความคลาดเคลื่อนของผลเฉลยในรอบการคำนวณที่ k ใด ๆ และ $\underset{k\to\infty}{limit} \quad e_k = 0$

จะกล่าวว่า ลำดับ $\{x_k\}$ มีคุณสมบัติลู่เข้าสู่จุดคำตอบ x^* ด้วยอัตราการลู่เข้า r ด้วยค่าคงตัวการลู่เข้า C โดยที่

$$\underset{k\to\infty}{limit} \quad \frac{\|e_{k+1}\|}{\|e_k\|^r} = C < \infty \tag{1.6}$$

เพื่อให้ง่ายในการทำความเข้าใจ ความสัมพันธ์ดังกล่าวอาจจะเขียนในรูปอย่างง่ายได้ดังต่อไปนี้

$$\|e_{k+1}\| = C\|e_k\|^r \tag{1.7}$$

ระเบียบวิธีที่พิจารณาจะมีคุณสมบัติลู่เข้าก็ต่อเมื่อ $0 < C < 1$, ถ้า $r = 1$ จะเรียกว่า ระเบียบวิธีนั้นว่า มีอัตราการลู่เข้าเชิงเส้น (linear convergence property) ถ้า $C > 1$ ระเบียบวิธีดังกล่าวจะไม่ลู่เข้าสู่จุดคำตอบ ระเบียบวิธีที่มีคุณสมบัติการลู่เข้าแบบนี้ ได้แก่ ระเบียบวิธีชันที่สุด (steepest descent method)

จะพบว่า ถ้า $C = 0.1$ เริ่มต้นการคำนวณที่ $||e_0|| = 1.0$ จะได้ $||e_k||$ ดังต่อไปนี้

1.0 0.1 0.01 0.001 1×10^{-4} 1×10^{-5} 1×10^{-6} 1×10^{-7} 1×10^{-8}

ปัญหานี้ลู่เข้าสู่ค่าความคลาดเคลื่อนเท่ากับ 1×10^{-8} ในการคำนวณรอบที่ 8

ถ้ากำหนดให้ $C = 0.8$ เริ่มต้นการคำนวณที่ $||e_0|| = 1.0$ จะได้ $||e_k||$ ดังต่อไปนี้

1.0 0.8 0.64 0.512 0.410 0.328 ....

สำหรับกรณีที่กำหนดให้ความคลาดเคลื่อนน้อยกว่า 1×10^{-8} ต้องใช้การคำนวณถึง 83 รอบ ดังนั้น ระเบียบวิธีที่แตกต่างกัน อัตราการลู่เข้าและค่าคงที่การลู่เข้าจะแตกต่างกัน การเลือกใช้ระเบียบวิธีที่เหมาะสมจะช่วยให้การแก้ปัญหาทำได้รวดเร็ว

ในกรณีที่ $r = 1$, $C = 0$ จะเรียกว่า มีคุณสมบัติการลู่เข้าแบบเชิงเส้นยิ่งยวด (superlinear convergence property) ซึ่งเป็นคุณสมบัติสำคัญ อัลกอริทึมที่มีคุณสมบัติการลู่เข้าแบบนี้ ได้แก่ ระเบียบวิธีเกรเดียนต์สังยุค (conjugate gradient method) หรือระเบียบวิธีคล้ายนิวตัน (quasi-Newton method)

ในกรณีที่ $r = 2$ จะเรียกว่า มีคุณสมบัติการลู่เข้ากำลังสอง (quadratic convergence property) ในปัจจุบัน ระเบียบวิธีที่มีคุณสมบัติการลู่เข้ากำลังสองที่นักคณิตศาสตร์คิดค้นขึ้นมีเพียงระเบียบวิธีนิวตัน (Newton method) เท่านั้น อย่างไรก็ตาม ระเบียบวิธีนี้ มีเงื่อนไขการลู่เข้า นั่นคือ เมื่อเมตริกซ์เฮสเซียนของปัญหาต้องเป็น *pdf* ระเบียบวิธีนิวตันจึงไม่เป็นที่นิยม เนื่องจากไม่สามารถรับประกันการลู่เข้าได้กับปัญหาทุกประเภท

ตัวอย่างที่ 1.5 กำหนดให้ระเบียบวิธีที่พิจารณาสร้างจุดคำตอบของปัญหาด้วยลำดับ $x_k = 1 + 10^{-k}$ ปัญหาดังกล่าวลู่เข้าสู่ $x^* = 1.0$

จะได้ว่า

$$e_k = x_k - x^* = (1 + 10^{-k}) - 1.0 = 10^{-k}$$
$$e_{k+1} = x_{k+1} - x^* = (1 + 10^{-(k+1)}) - 1.0 = 10^{-(k+1)} = 0.1\times 10^{-k}$$

นั่นคือ $e_{k+1} = 0.1\times e_k \quad \Rightarrow \quad \|e_{k+1}\| = 0.1\|e_k\|$

$C = 0.1$, $r = 1$ มีคุณสมบัติการลู่เข้าเชิงเส้น ด้วยค่าคงตัวการลู่เข้า $C = 0.1$

ตัวอย่างที่ 1.6 กำหนดให้ระเบียบวิธีที่พิจารณาสร้างผลเฉลยของปัญหาด้วยลำดับ $x_{k+1} = x_k + \dfrac{1}{x_k}$ ปัญหาดังกล่าวลู่เข้าสู่ $x^* = 2.0$ จงคำนวณอัตราการลู่เข้า และค่าคงตัวการลู่เข้า

วิธีทำ จะได้ว่า

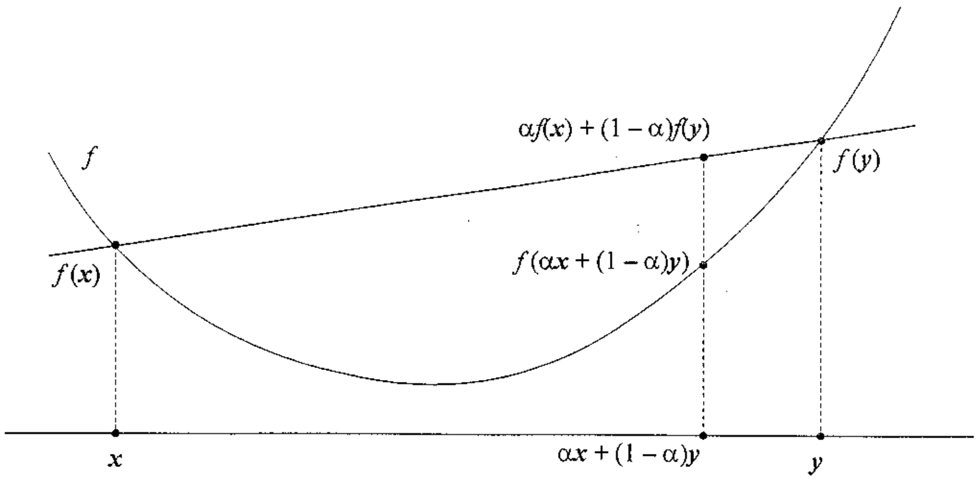
$$e_{k+1} = x_{k+1} - x^* = \frac{1}{2}\left(x_k + \frac{4}{x_k}\right) - 2 = \frac{1}{2x_k}\left(x_k^2 - 4x_k + 4\right)$$
$$= \frac{1}{2x_k}(x_k - 2)^2 = \frac{1}{2x_k}e_k^2$$
$$\therefore \quad \lim_{k\to\infty} \frac{\|e_k\|}{\|e_{k+1}\|^2} = \frac{1}{2x_k}\bigg|_{x_\infty \triangleq x^*} = \frac{1}{2x^*} = \frac{1}{2(2)} = \frac{1}{4} = C$$

นั่นคือ $e_{k+1} = 0.25\times e_k^2 \quad \Rightarrow \quad \|e_{k+1}\| = 0.25\|e_k\|^2$

$C = 0.25$, $r = 2$ มีคุณสมบัติการลู่เข้ากำลังสอง ด้วยค่าคงตัวการลู่เข้า $C = 0.25$

1.5 ความโค้งชันหรือความเป็นคอนเวกซ์ (convexity)

ระเบียบวิธีที่จะนำเสนอในเอกสารนี้ ทั้งหมดตั้งอยู่บนข้อกำหนดของการค้นหาจุดที่สอดคล้องกับเงื่อนไขที่ทำให้เกรเดียนต์ของฟังก์ชันวัตถุประสงค์หรือเกรเดียนต์ของฟังก์ชันลากรองจ์ (Lagrange function) มีค่าเป็นศูนย์ เงื่อนไขดังกล่าวเพียงพอที่จะระบุจุดต่ำสุดเฉพาะถิ่น แต่ไม่อาจจะรับประกันจุดต่ำสุดโดยรวมของปัญหาได้ อย่างไรก็ตาม สำหรับปัญหาในบางรูปแบบที่มีฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับที่มีคุณสมบัติความเป็นคอนเวกซ์แล้ว ปัญหานั้น ๆ จะมีจุดต่ำสุดเพียงจุดเดียวเท่านั้นในปริภูมิของการค้นหา นั่นคือ จุดต่ำสุดเฉพาะถิ่นที่ค้นพบจะเป็นจุดต่ำสุดโดยรวมโดยปริยาย ดังนั้น การศึกษาทฤษฎีความเป็นคอนเวกซ์ของฟังก์ชันมีความสำคัญมากในการแก้ปัญหาค่าเหมาะที่สุด



รูปที่ 1.13 ฟังก์ชันคอนเวกซ์

กำหนดเซต S โดยที่ $\boldsymbol{x}, \boldsymbol{y} \in S$ จะมีคุณสมบัติความเป็นคอนเวกซ์ ก็ต่อเมื่อ

$$\alpha \boldsymbol{x} + (1 - \alpha)\boldsymbol{y} \in S \text{ สำหรับ } \quad 0 \le \alpha \le 1.0$$

เรียกฟังก์ชันที่มีคุณสมบัติต่อไปนี้ว่าเป็นฟังก์ชันคอนเว็กซ์ ที่นิยามบนเซตคอนเวกซ์ S ก็ต่อเมื่อ

$$f(\alpha \boldsymbol{x} + (1 - \alpha)\boldsymbol{y}) \le \alpha f(\boldsymbol{x}) + (1 - \alpha)f(\boldsymbol{y}) \text{ สำหรับ } \forall \boldsymbol{x}, \forall \boldsymbol{y} \in S \text{ และ } 0 \le \alpha \le 1.0$$

พิจารณาจากกราฟในรูปที่ 1.13

ตัวอย่างที่ 1.7 จงแสดงว่าฟังก์ชันต่อไปนี้เป็นฟังก์ชันคอนเวกซ์หรือไม่

1. $f(x) = 2x - 1$
2. $f(x) = x^2$

<u>กรณีที่ 1</u>: จะได้

$$f(\alpha x + (1 - \alpha)y) = 2(\alpha x + (1 - \alpha)y) - 1 = \alpha(2x) + (1 - \alpha)(2y)$$
$$= \alpha f(x) + (1 - \alpha)f(y)$$ ดังนั้น $f(x) = 2x - 1$ เป็นฟังก์ชันคอนเวกซ์

<u>กรณีที่ 2</u>: จะได้

$$f(\alpha x + (1 - \alpha)y) = (\alpha x + (1 - \alpha)y)^2$$
$$= \alpha^2 x^2 + 2\alpha(1 - \alpha)xy + (1 - \alpha)^2 y^2$$

และ $\alpha f(x) + (1 - \alpha)f(y) = \alpha x^2 + (1 - \alpha)y^2$

ถ้า $f(x)$ เป็นฟังก์ชันคอนเวกซ์แล้ว $f(\alpha \boldsymbol{x} + (1 - \alpha)\boldsymbol{y}) \le \alpha f(\boldsymbol{x}) + (1 - \alpha)f(\boldsymbol{y})$ นั่นคือ

$$f(\alpha \boldsymbol{x} + (1 - \alpha)y) - \{\alpha f(\boldsymbol{x}) + (1 - \alpha)f(y)\} \le 0$$

จะได้ว่า
$$\alpha^2 x^2 + 2\alpha(1 - \alpha)xy + (1 - \alpha)^2 y^2 - \alpha x^2 - (1 - \alpha)y^2 \le 0$$
$$\alpha(\alpha - 1)x^2 - 2\alpha(\alpha - 1)xy + (1 - 2\alpha + \alpha^2)y^2 - (1 - \alpha)y^2 \le 0$$
$$\alpha(\alpha - 1)x^2 - 2\alpha(\alpha - 1)xy + (-\alpha + \alpha^2)y^2 \le 0$$
$$\alpha(\alpha - 1)x^2 - 2\alpha(\alpha - 1)xy + \alpha(\alpha - 1)y^2 \le 0$$
$$\alpha(\alpha - 1)\{x^2 - 2xy + y^2\} \le 0$$
$$\alpha(\alpha - 1)(x - y)^2 \le 0$$

จะพบว่า จากอสมการที่ได้ $0 \le \alpha \le 1$ ดังนั้น $(\alpha - 1) \le 0$ และ $(x - y)^2 \ge 0$ นั่นคือ อสมการที่ได้เป็นจริงเสมอ ดังนั้น ฟังก์ชัน $f(x) = x^2$ เป็นฟังก์ชันคอนเวกซ์

จากตัวอย่าง จะเห็นได้ว่า การพิสูจน์ค่อนข้างยุ่งยากมาก อย่างไรก็ตาม การประเมินความเป็นคอนเวกซ์ของฟังก์ชันใด ๆ อาจจะทำได้โดยการพิจารณาค่าอนุพันธ์อันดับสองในกรณีของฟังก์ชันมิติเดียว หรือจากเมตริกซ์เฮสเซียนในกรณีของฟังก์ชันหลายมิติ

ถ้าฟังก์ชัน $f(x)$ ที่พิจารณาบนเซตที่มีคุณสมบัติการเว้า S แล้ว

<u>กรณีหนึ่งมิติ</u> $f''(x) \ge 0$ สำหรับ $\forall x \in S$

<u>กรณีหลายมิติ</u> $y^T \nabla^2 f(x) y \ge 0$ สำหรับ $\forall y$ และ $\forall x \in S$

นั่นคือ ในกรณีของฟังก์ชันหลายมิติ เมตริกซ์เฮสเซียนต้องมีคุณสมบัติเป็น *psdf* นั่นเอง

ตัวอย่างที่ 1.8 จงแสดงว่าฟังก์ชันต่อไปนี้เป็นฟังก์ชันคอนเวกซ์หรือไม่

1. $f(x) = x^2$
2. $f(x_1, x_2) = 2x_1^2 - 2x_1x_2 + 4x_2^2$

วิธีทำ

<u>กรณีที่ 1</u>: $f''(x) = 2 \ge 0 \Rightarrow f(x)$ เป็นฟังก์ชันคอนเวกซ์

<u>กรณีที่ 2</u>:

$$\nabla^2 f(x_1, x_2) = \begin{bmatrix} 4 & -2 \\ -2 & 8 \end{bmatrix}$$

ตรวจสอบเมตริกซ์เฮสเซียนว่าเป็น *psdf* หรือไม่ โดยกำหนดเวกเตอร์ $y = [y_1\ y_2]^T$ ใด ๆ จะได้

$$\begin{aligned} y^T \nabla^2 f(x_1, x_2) y &= [y_1 \quad y_2]\begin{bmatrix} 4 & -2 \\ -2 & 8 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ &= [y_1 \quad y_2]\begin{bmatrix} 4y_1 - 2y_2 \\ -2y_1 + 8y_2 \end{bmatrix} \\ &= y_1(4y_1 - 2y_2) + y_2(-2y_1 + 8y_2) \\ &= 4y_1^2 - 2y_1y_2 - 2y_1y_2 + 8y_2^2 \\ &= 4\left(y_1^2 - y_1y_2 + 2y_2^2\right) = 4\left(y_1^2 - 2\left(\tfrac{1}{2}\right)y_1y_2 + \tfrac{1}{4}y_2^2 + \tfrac{7}{4}y_2^2\right) \\ &= 4\left(y_1 - \tfrac{1}{2}y_2\right)^2 + 7y_2^2 \ge 0 \quad \Rightarrow f(x_1, x_2) \end{aligned}$$

ดังนั้น $f(x_1, x_2) = 2x_1^2 - 2x_1x_2 + 4x_2^2$ เป็นฟังก์ชันคอนเวกซ์

1.6 สรุป

ในบทนี้ หลักการค้นหาผลเฉลยเหมาะที่สุดได้ถูกนำเสนอพร้อมตัวอย่างการคำนวณ อย่างไรก็ตาม การได้มาซึ่งค่าตัวแปรหรือพารามิเตอร์ต่างๆ ยังมิได้กล่าวไว้ในบทนี้ เนื่องจากต้องใช้เทคนิคการคำนวณเชิงตัวเลขเข้าช่วย โดยหลักการแล้ว กลยุทธ์ที่ใช้ในการค้นหาผลเฉลยเหมาะที่สุด

มีสองกลยุทธ์ ได้แก่ กลยุทธ์การค้นหาตามเส้น (line search strategy) และกลยุทธ์ขอบเขตความเชื่อมั่น (trust region strategy) ซึ่งเนื้อหาโดยละเอียดจะกล่าวถึงในบทถัดไป

1.7 แบบฝึกหัดท้ายบท

1. กำหนดฟังก์ชันวัตถุประสงค์ $f(\boldsymbol{x})$ ทิศทางการค้นหา $\boldsymbol{p}$ และจุดเริ่มต้น $\boldsymbol{x_0}$ ดังต่อไปนี้ จงคำนวณช่วงก้าวที่เหมาะสม (optimal step length: λ) ผลเฉลยที่ถูกปรับปรุงในรอบถัดไปและการลดค่าลงของฟังก์ชันวัตถุประสงค์ที่ได้

   ก. $f(x_1,x_2)=(x_1+1)^2+4x_1(x_2-1)+(x_2-2)^2$, $\boldsymbol{p}=[2.0\ \ 4.0]^T$, $\boldsymbol{x_0}=[0.0\ \ 0.0]^T$

   ข. $f(x_1,x_2,x_3)=x_1^2+2x_2^2+(x_3-1)^2$, $\boldsymbol{p}=[0.0\ \ 0.0\ \ \text{-}2.0]^T$, $\boldsymbol{x_0}=[0.0\ \ 0.0\ \ 0.0]^T$

   ค. $f(x_1,x_2)=(x_1+x_2)^4+x_2^2$, $\boldsymbol{p}=[0.0\ \ \text{-}4.0]^T$, $\boldsymbol{x_0}=[2.0\ \ \text{-}2.0]^T$

2. จงประมาณค่าฟังก์ชัน $f(\boldsymbol{x})$ ต่อไปนี้ ด้วยฟังก์ชันกำลังสองโดยใช้เทคนิคการคำนวณเกรเดียนต์และเฮสเซียน ณ จุด $\boldsymbol{x_0}$ ที่กำหนด

   ก. $f(x_1,x_2)=(x_1+1)^2+4x_1(x_2-1)+(x_2-2)^2$, $\boldsymbol{x_0}=[0.0\ \ 0.0]^T$

   ข. $f(x_1,x_2,x_3)=x_1^2+2x_2^2+(x_3-1)^2$, $\boldsymbol{x_0}=[0.0\ \ 0.0\ \ 0.0]^T$

   ค. $f(x_1,x_2)=(x_1+x_2)^4+x_2^2$, $\boldsymbol{x_0}=[2.0\ \ \text{-}2.0]^T$

3. กำหนดให้ระเบียบวิธีที่พิจารณาประมาณจุดคำตอบของปัญหาด้วยลำดับที่กำหนดให้ดังต่อไปนี้ จงคำนวณอัตราการลู่เข้าและค่าคงที่การลู่เข้า

   ก. $\boldsymbol{x}_k = 4 + 5^{-k}$

   ข. $\boldsymbol{x}_k = 2^{-2^k}$

   ค. $\boldsymbol{x}_k = 5^{-k^2}$

4. จงแสดงว่าฟังก์ชันต่อไปนี้เป็นฟังก์ชันความเว้าหรือไม่

   ก. $f(\boldsymbol{x}) = 3x^2 + 4x - 5$

   ข. $f(\boldsymbol{x})=\sqrt{1+x^2}$

   ค. $f(\boldsymbol{x})=2x^4+3x^3+8x^2$

   ง. $f(\boldsymbol{x}) = x/(1 + x^4)$

   จ. $f(\boldsymbol{x})=2x_1^2+3x_1(x_2-2)+x_2^2$

   ฉ. $f(\boldsymbol{x})=2x_1^2+3x_1x_2+5x_2^2-2x_1+6x_2$

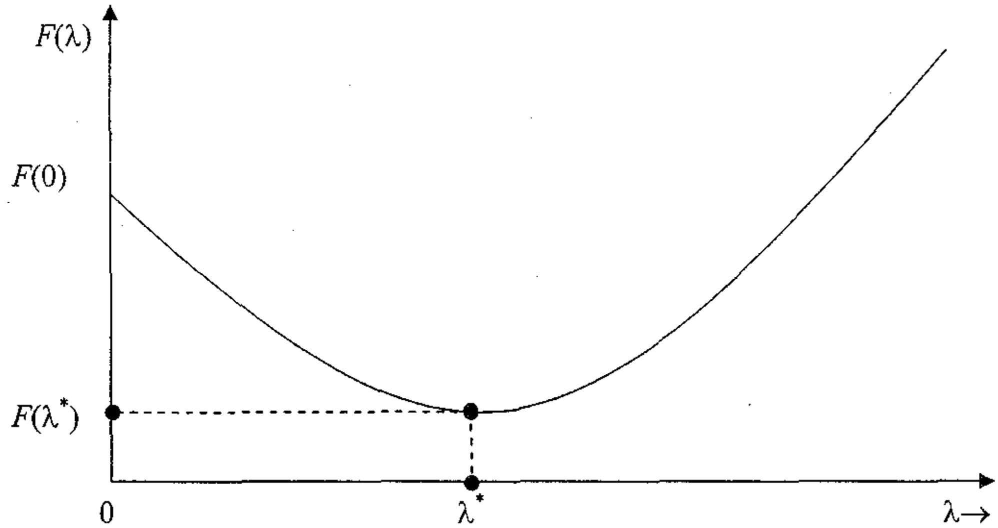
   ช. $f(\boldsymbol{x})=2x_1^2+3x_2^2+x_3^2$

บทที่ 2 การค้นหาผลเฉลยต่ำที่สุดของฟังก์ชันตัวแปรเดี่ยว (Minimization of a Univariate Function)

"Example is always more efficacious than precept"

2.1 เกริ่นนำ

ปัญหาค่าเหมาะที่สุดของฟังก์ชันหลายตัวแปรโดยใช้กลยุทธ์การค้นหาตามเส้นนั้น ในแต่ละรอบการคำนวณ ปัญหาดังกล่าวจะถูกแปลงไปเป็นปัญหาย่อยการค้นหาผลเฉลยต่ำที่สุดของฟังก์ชันตัวแปรเดี่ยว ซึ่งเรียกว่า ปัญหาการค้นหาช่วงก้าวที่เหมาะสม (optimal step length sub-problem) ถ้าปัญหาดังกล่าวอยู่ในรูปฟังก์ชันกำลังสอง การแก้ปัญหาจะทำได้อย่างรวดเร็ว โดยทั่วไปแล้ว ฟังก์ชันวัตถุประสงค์ของปัญหาการค้นหาช่วงก้าวที่เหมาะสมไม่จำเป็นต้องเป็นฟังก์ชันกำลังสอง การค้นหาจุดต่ำสุดด้วยระเบียบวิธีเชิงตัวเลขที่มีประสิทธิภาพจะต้องถูกนำมาใช้ ในบทนี้ ระเบียบวิธีเชิงตัวเลขดังกล่าวจะถูกนำเสนอเพื่อเน้นการนำไปแก้ปัญหาการค้นหาช่วงก้าวที่เหมาะสมได้อย่างมีประสิทธิภาพ ดังรายละเอียดต่อไปนี้



รูปที่ 2.1 ปัญหาย่อยการหาช่วงก้าวที่เหมาะสม

2.2 ปัญหาการค้นหาช่วงก้าวที่เหมาะสม

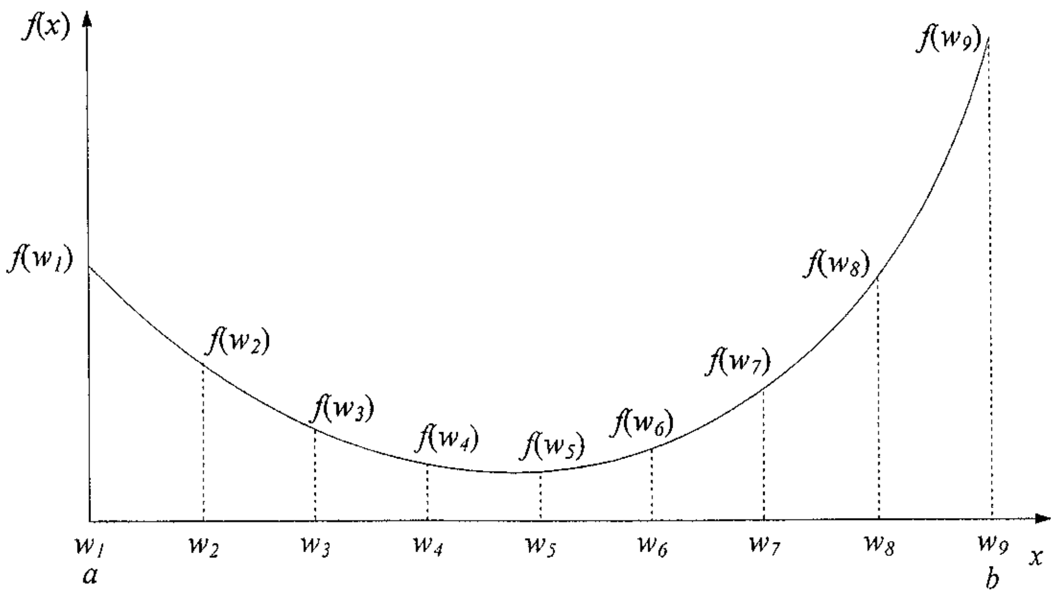
กลยุทธ์การค้นหาตามเส้นอาศัยหลักการสร้างทิศทางการค้นหา (search direction: $\boldsymbol{p}_k$) ที่เหมาะสม โดยทิศทางการค้นหานี้ต้องมีคุณสมบัติลาดลง (descent direction หรือ down-hill direction) กำหนดให้ทิศทางการค้นหาดังกล่าวมีค่าเป็น $\boldsymbol{p}_k$ ผลเฉลยในรอบถัดไปคำนวณได้จากความสัมพันธ์ต่อไปนี้ $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \boldsymbol{p}_k$ พิจารณาได้ดังรูปที่ 2.1 ถ้ากำหนดทิศทางการค้นหาได้อย่างเหมาะสม ปัญหาที่เหลืออยู่จะถูกลดรูปลงเป็นปัญหาย่อยการหาช่วงก้าวที่เหมาะสม (optimal tep length: λ) ซึ่งเป็นปัญหาการค้นหาผลเฉลยต่ำที่สุดของฟังก์ชันตัวแปรเดี่ยว เพื่อให้การลดค่า

ลงของฟังก์ชันมีค่ามากที่สุดในทิศทางการค้นหา p_k ดังนั้น เพื่อให้สะดวกในการกำหนดสัญลักษณ์ กำหนดให้ $\lambda = \lambda_k$ จะได้ปัญหาย่อยการค้นหาตามเส้นดังนี้

$$\underset{\lambda > 0}{\text{Minimize}} \quad F(\lambda) \equiv f(x_k + \lambda p_k) \tag{2.1}$$

ในรอบการคำนวณที่ $k+1$ ใด ๆ สิ่งที่ทราบค่า ได้แก่ ผลเฉลยในรอบก่อนหน้า x_k และทิศทางการค้นหา p_k จะเหลือตัวแปรที่ไม่ทราบค่าเพียงตัวเดียว คือ λ เท่านั้น เรียกว่า ปัญหาการหาช่วงก้าวที่เหมาะสม เมื่อปรับค่า λ โดยเริ่มต้นจาก 0 ให้มีค่าเป็นบวกและเพิ่มขึ้นอย่างต่อเนื่อง ภายใต้ทิศทางการค้นหา p_k ที่กำหนด จะทำให้ค่าของฟังก์ชัน $F(\lambda)$ ลดค่าลงอย่างต่อเนื่องจนถึงค่า $\lambda = \lambda^*$ ณ จุดนี้ ค่าของฟังก์ชันวัตถุประสงค์จะลดค่าลงมากที่สุดในทิศทางการค้นหา p_k ถ้าเพิ่มค่า λ ให้มากกว่านี้ ค่าของฟังก์ชันวัตถุประสงค์จะมีค่าเพิ่มขึ้น ดังรูปที่ 2.1 จะเห็นได้ว่า การค้นหาผลเฉลยต่ำที่สุดของฟังก์ชันตัวแปรเดี่ยวนี้มีความสำคัญอย่างมากต่อกลยุทธ์การค้นหาตามเส้น ระเบียบวิธีเชิงตัวเลขที่มีประสิทธิภาพสูงย่อมทำให้การค้นหาผลเฉลยต่ำสุดทำได้รวดเร็ว ดังนั้น ก่อนที่จะศึกษาการแก้ปัญหาค่าเหมาะที่สุด จำเป็นอย่างที่จะต้องศึกษาการแก้ปัญหาการค้นหาผลเฉลยต่ำสุดของฟังก์ชันตัวแปรเดี่ยวเสียก่อน

ภายใต้ข้อสมมติที่ว่า จุดเริ่มต้นของกระบวนการค้นหา หรือผลเฉลยที่ได้รับการปรับปรุงในรอบการคำนวณล่าสุดอยู่ใกล้กับผลเฉลยต่ำที่สุดมากพอ ร่วมกับการเลือกทิศทางการค้นหาที่เหมาะสม มีทิศทางลาดลง จะทำให้ปัญหาการค้นหาช่วงก้าวที่เหมาะสมมีคุณสมบัติที่เรียกว่า ฟังก์ชันที่มีจุดต่ำสุดเพียงจุดเดียว (unimodal function) ภายในปริภูมิการค้นหาที่กำหนด ระเบียบวิธีเชิงตัวเลขที่ใช้มีหลากหลาย ในชั้นนี้จะนำเสนอวิธีพื้นฐานเพื่อสร้างความเข้าใจในหลักการวิธีการที่มีประสิทธิภาพสูงและเป็นที่นิยมใช้กับปัญหาช่วงก้าวที่เหมาะสมดังต่อไปนี้



รูปที่ 2.2 การค้นหาตามกริด

2.3 ระเบียบวิธีการค้นหาตามกริด (Grid Search Method)

วิธีนี้น่าจะเป็นวิธีที่ง่ายที่สุด โดยอาศัยการสร้างกริด (grid) ขึ้นมา และทำการหาค่าฟังก์ชันที่ทุก ๆ จุดตามกริดที่กำหนด และหาค่าช่วงที่จุดต่ำสุดจะถูกบรรจุอยู่ ถ้ารู้ในเบื้องต้นว่าจุดต่ำสุดต้องอยู่ในช่วง $[a,b]$ อย่างแน่นอน ดังแสดงในรูปที่ 2 วิธีการนี้เหมาะที่จะใช้ในกรณีที่ช่วงดังกล่าวมีค่าไม่กว้างนัก มิฉะนั้นแล้ว จะทำให้ใช้เวลาในการคำนวณนานเกินไป วิธีการนี้จะทำการกำหนดช่วงการค้นหาซ้ำไปเรื่อยๆ จนกว่าช่วงการค้นหาจะมีค่าน้อยกว่าความคลาดเคลื่อนสูงสุดที่ยอมรับได้

กำหนดให้แบ่งช่วงการค้นหา $[a,b]$ ออกเป็น $n-1$ ช่วงเท่า ๆ กัน จะได้จำนวนจุดทั้งสิ้น n จุด การค้นหาจะดำเนินการโดยคำนวณค่าฟังก์ชันที่ตำแหน่งกริดทั้ง n จุด หลักการค้นหาจะพิจารณาลดช่วงการค้นหาจากช่วง $[a,b]$ ให้มีขนาดเล็กลง โดยพิจารณาคู่ของจุดที่อยู่ติดกันที่มีโอกาสบรรจุค่าต่ำสุด ดังอัลกอริทึมต่อไปนี้

ขั้นที่ 1: เตรียมข้อมูล กำหนดตัวนับและค่าเริ่มต้นช่วงการค้นหา $[a,b]$ ใด ๆ
iter_no = 1; $f_{min} = \infty$;

ขั้นที่ 2: แบ่งช่วงการค้นหา $[a,b]$ ออกเป็น $n-1$ ช่วงย่อย กำหนดตัวนับ $i = 1$

ขั้นที่ 3: ตรวจสอบค่าฟังก์ชันของจุด w_i
ถ้า $f(w_i) < f_{\min}$, ให้ $f_{\min} = f(w_i)$ และ idmin = i

ขั้นที่ 4: ถ้า $i \le n-1$ เพิ่มตัวนับ $i = i+1$ ทำซ้ำขั้นที่ 3

ขั้นที่ 5: ให้ k = idmin กำหนดช่วงการค้นหาในรอบต่อไปเป็น $[a,b] = [w_{k-1}, w_{k+1}]$

ขั้นที่ 6: ถ้า $\varepsilon = b - a > \varepsilon_{allow}$ ทำซ้ำขั้นที่ 2 และ iter_no = iter_no + 1,
ถ้าไม่ใช่ จุดคำตอบมีค่าเป็น $x_{opt} = \dfrac{a+b}{2}$

ตัวอย่างที่ 2.1 กำหนดฟังก์ชันวัตถุประสงค์เป็น $f(x) = x^2 - 3x \cdot exp(-x)$ ในช่วงการค้นหา [0,1] โดยใช้ระเบียบวิธีการค้นหาตามกริดด้วยความคลาดเคลื่อนที่ยอมรับได้เท่ากับ 0.0001

วิธีทำ ดำเนินการแบ่งจุดออกเป็น 8 ช่วง [a,b] = [0,1]

<u>รอบการคำนวณที่ 1:</u>

คำนวณค่าฟังก์ชันวัตถุประสงค์ประจำกริดทั้ง 9 จุด สำหรับรอบการคำนวณที่ 1

grid	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
value	0.000	0.125	0.250	0.375	0.500	0.625	0.750	0.875	1.000
$f(x)$	0.000	-0.3153	-0.5216	-0.6326	-0.6598	-0.6130	-0.5003	-0.3286	-0.1036

จะพบว่า $f_{min} = -0.6598$, idmin = 5

จะได้ว่า $[a,b] = [0.375, 0.625] \quad \Rightarrow \quad \varepsilon = 0.25 > 0.0001$

รอบการคำนวณที่ 2:

คำนวณค่าฟังก์ชันวัตถุประสงค์ประจำกริดทั้ง 9 จุด สำหรับรอบการคำนวณที่ 2

grid	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
value	0.375	0.4063	0.4375	0.4688	0.5000	0.5313	0.5625	0.5938	0.625
$f(x)$	-0.6325	-0.6468	-0.6560	-0.6603	-0.6598	-0.6547	-0.6451	-0.6312	-0.6130

จะพบว่า f_{min} = -0.6603, idmin = 4

จะได้ว่า $[a,b]$ = [0.4375,0.5000] $\Rightarrow$ $\varepsilon = 0.0625 > 0.0001$

รอบการคำนวณที่ 3:

คำนวณค่าฟังก์ชันวัตถุประสงค์ประจำกริดทั้ง 9 จุด สำหรับรอบการคำนวณที่ 3

grid	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
value	0.4375	0.4453	0.4531	0.4609	0.4688	0.4766	0.4844	0.4922	0.5000
$f(x)$	-0.6560	-0.6575	-0.6587	-0.6597	-0.6603	-0.66060	-0.66063	-0.6604	-0.6598

จะพบว่า f_{min} = -0.66063, idmin = 7

จะได้ว่า $[a,b]$ = [0.4766,0.4922] $\Rightarrow$ $\varepsilon = 0.0156 > 0.0001$

รอบการคำนวณที่ 4:

คำนวณค่าฟังก์ชันวัตถุประสงค์ประจำกริดทั้ง 9 จุด สำหรับรอบการคำนวณที่ 4

grid	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
value	0.4766	0.4786	0.4805	0.4825	0.4844	0.4864	0.4883	0.4903	0.4922
$f(x)$	-0.6606	-0.6606	-0.6607	-0.6606	-0.6606	-0.6606	-0.6605	-0.6604	-0.6604

จะพบว่า f_{min} = -0.6607, idmin = 3

จะได้ว่า $[a,b]$ = [0.4786,0.4825] $\Rightarrow$ $\varepsilon = 0.0039 > 0.0001$

ดำเนินการคำนวณวนรอบต่อไป จะได้ผลการคำนวณดังต่อไปนี้

รอบที่	a	b	f_{min}
0	0.0000	1.00000	--
1	0.3750	0.62500	-0.65980
2	0.4375	0.50000	-0.66028
3	0.47656	0.49219	-0.66063
4	0.47852	0.48242	-0.66065
5	0.48047	0.48145	-0.66065
6	0.48096	0.48120	-0.66065
7	0.48108	0.48114	-0.66065

จะผลเฉลย

$$x_{opt} = \frac{0.48108 + 0.48114}{2} = 0.48112$$

โปรแกรม SCILAB สำหรับตัวอย่างที่ 2.1

// SCILAB source program

```
// กำหนดฟังก์ชันวัตถุประสงค์                f_univar01.sce
========================
        f_univar01.sce
========================
deff('[f]=f_univar01(x)','f = x^2 - 3*x*exp(-x)');
```

```
// ฟังก์ชันสำหรับคำนวณการค้นหาตามกริด        gridsearch.sci
========================
        gridsearch.sci
========================
function [xmin,fmin] = gridsearch(func,interval,Npnt,tolerance,MaxIter)
Interval_Err = interval(2) - interval(1);
fpnt(1) = func(interval(1));
fpnt(Npnt) = func(interval(2));
count = 0;
xmin = (interval(1) + interval(2))/2;
fmin = func(xmin);
disp('Iter  XL   XU    Xmin   Fmin   Err');
disp([count interval xmin fmin Interval_Err]);
while Interval_Err > tolerance
   dx = Interval_Err/(Npnt-1);
   for k = 2:Npnt-1
      fpnt(k) = func(interval(1) + dx*(k-1));
   end
   [fmin,idmin] = min(fpnt);
   xmin = interval(1) + dx*(idmin-1);
   interval(1) = interval(1) + dx*(idmin-2);
   interval(2) = interval(1) + 2*dx;
   fmin = func(xmin);
   Interval_Err = interval(2) - interval(1);
   count = count + 1;
   disp([count interval xmin fmin Interval_Err]);
```

```
    if count > MaxIter
        break;
    end
end
xmin = (interval(1) + interval(2))/2;
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('f_univar01.sce');                 ประกาศฟังก์ชันวัตถุประสงค์
--> exec('gridsearch.sci');                 ประกาศฟังก์ชันเพื่อการคำนวณ
--> [xmin,fmin] = gridsearch(f_univar01,[0 1],9,1e-4,100);
 Iter  XL          XU          Xmin        Fmin         Err
  0.   0.          1.          0.5         -0.6597960   1.
  1.   0.375       0.625       0.5         -0.6597960   0.25
  2.   0.4375      0.5         0.46875     -0.6602822   0.0625
  3.   0.4765625   0.4921875   0.484375    -0.6606252   0.015625
  4.   0.4785156   0.4824219   0.4804688   -0.6606500   0.0039062
  5.   0.4804688   0.4814453   0.4809570   -0.6606510   0.0009766
  6.   0.4809570   0.4812012   0.4810791   -0.660651    0.0002441
  7.   0.4810791   0.4811401   0.4811096   -0.660651    0.0000610
```

2.4 ระเบียบวิธีการค้นหาของฟีโบนาซซี (Fibonacci Search Method)

ระเบียบวิธีการค้นหาตามกริดสามารถค้นหาค่าต่ำสุดได้ โดยอาศัยการคำนวณค่าฟังก์ชันที่จุดต่าง ๆ จากการสร้างกริดขึ้นมาแล้วทำการเปรียบเทียบค่าฟังก์ชันทุกจุดบนกริดที่สร้างขึ้น ทำให้การคำนวณไม่มีประสิทธิภาพเท่าที่ควร เช่น ถ้าทำการสร้างกริดโดยแบ่งเป็น 4 ช่วง แล้ว ในแต่ละรอบการคำนวณ จะต้องคำนวณค่าฟังก์ชันทั้งสิ้น 5 จุด และถ้าใช้จำนวนรอบทั้งหมด 20 รอบ จะใช้การคำนวณ ค่าฟังก์ชันทั้งหมด 100 จุด ถ้าแบ่ง 8 ช่วง จะต้องใช้การคำนวณ 9 จุดต่อรอบ ในจำนวน 20 รอบนั้นจะใช้การคำนวณค่าฟังก์ชันทั้งหมด 180 จุด เป็นต้น ทำให้เกิดคำถามว่า

1) ถ้ากำหนดจำนวนครั้งของการคำนวณค่าฟังก์ชันในแต่ละรอบของการค้นหา ทำอย่างไรจึงจะสามารถระบุช่วงที่มีค่าต่ำสุดอยู่ได้ โดยทำให้เกิดค่าความผิดพลาดของจุดต่ำสุดที่ประมาณได้กับจุดต่ำสุดที่แท้จริงน้อยที่สุด

2) เมื่อกำหนดค่าความคลาดเคลื่อนระหว่างจุดคำตอบจากจุดต่ำสุดแล้ว ทำอย่างไรจึงจะทำให้จำนวนการคำนวณค่าของฟังก์ชันที่ใช้น้อยที่สุด คำถามเหล่านี้สามารถตอบได้จากการพิจารณาคำถาม ต่อไปนี้

3) เมื่อกำหนดค่าความผิดพลาดที่ ยอมรับได้ของการประมาณค่าต่ำสุดในช่วงการค้นหา $[a,b]$ และกำหนดจำนวนครั้งในการคำนวณค่าของฟังก์ชันแล้ว ค่าสูงสุดของ $b - a$ ต้องมีค่าเท่าใดสำหรับจำนวนการคำนวณค่าฟังก์ชัน ดังกล่าว

ในปี 1953 ไคเฟอร์ (Kiefer) นักคณิตศาสตร์ชาวอเมริกันได้พิสูจน์ให้เห็นว่าคำตอบของปัญหาดังกล่าวมีความสัมพันธ์กับลำดับตัวเลขฟีโบนาซซี (Fibonacci number) ที่นิยามโดย

$F_0 = F_1 = 1; F_N = F_{N-1} + F_{N-2}$ สำหรับ $N \geq 2$

ลีโอนาโด (Leonardo) แห่งปีซา (Pisa) ค.ศ. 1175 – 1230 ค้นพบลำดับตัวเลขนี้จากการสำรวจปัญหาจำนวนประชา กรของกระต่าย ตัวอย่างของลำดับนี้ 16 อันดับแรกเป็นดังนี้

1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987

โดยการแก้ปัญหาสมการผลต่างข้างต้น จะได้ว่า

$$F_N = \frac{\sqrt{5}}{5(2^{N+1})}\left[\left(\sqrt{5}+1\right)^{N+1} + (-1)^N\left(\sqrt{5}-1\right)^{N+1}\right] \text{ โดยที่ } N = 0, 1, 2, \ldots$$

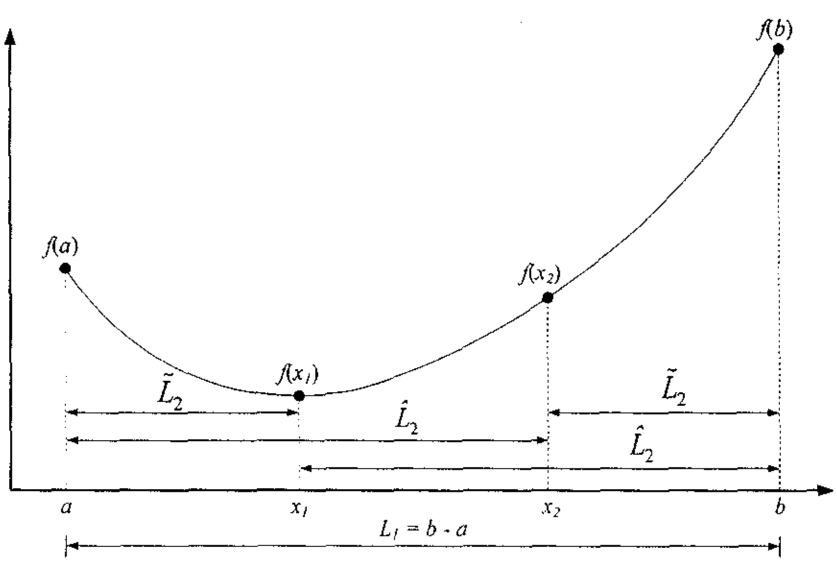
หรืออยู่ในรูปลิมิตต่อไปนี้

$$\lim_{N\to\infty} \frac{F_N}{F_{N-1}} = \frac{\sqrt{5}-1}{2}$$

วิธีการนี้กำหนดให้ช่วงของการค้นหามีค่าดังนี้

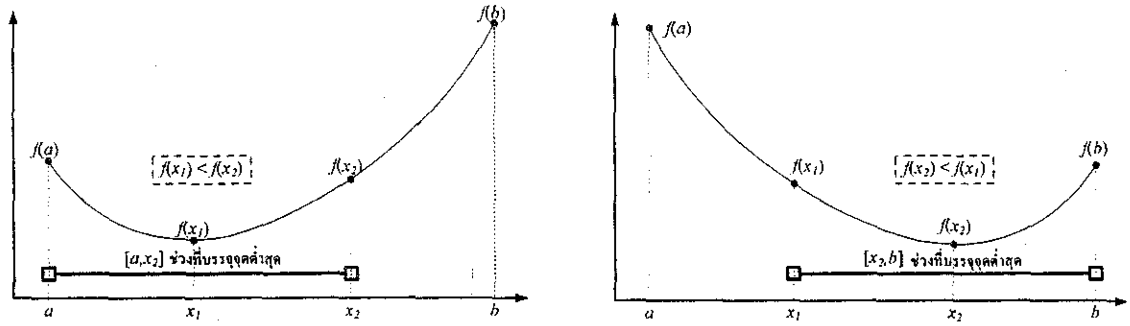
$$F_N \times \delta = b - a \quad \text{นั่นคือ} \quad \delta = \frac{b-a}{F_N} \leq \varepsilon$$

เมื่อกำหนดค่า a b ε และ $L_1 = b - a$ จำนวนช่วงที่แบ่งหาได้จาก การคำนวณค่าตัวเลขลำดับฟีโบนาซซี F_N ที่มีค่าน้อยที่สุดที่ทำให้เงื่อนไข $\delta \leq \varepsilon$ เป็นจริง โดยที่ ε คือ ค่าความคลาดเคลื่อนสูงที่สุดที่ยอมรับได้ ตัวเลข N ที่ได้จากสมการดังกล่าวเป็นตัวกำหนดจำนวนรอบในการคำนวณของระเบียบวิธีการค้นหาของฟีโบนาซซี วิธีการนี้ ใช้การคำนวณจุดภายในช่วง $[a,b]$ 2 จุด คือ x_1 และ x_2 โดยที่จุดทั้งสองอยู่ห่างจากขอบการค้นหาเป็นระยะ L_2 เท่ากัน ดังรูป



รูปที่ 2.3 การแบ่งจุดเพื่อคำนวณสำหรับระเบียบวิธีฟีโบนาซซี

จากรูป ระยะ L_2 อาจจะเป็น $\tilde{L}_2$ หรือ $\hat{L}_2$ ก็ได้ ทำให้ลำดับของจุด x_1 และ x_2 อาจจะสลับกันได้ในการคำนวณวนรอบ ปัญหานี้ไม่ใช่เรื่องใหญ่ ด้วยอัลกอริทึมการตรวจสอบที่รัดกุมแก้ปัญหานี้ได้ ปัญหาที่สำคัญกว่าก็คือ จะลดขอบเขตของช่วงการค้นหา $[a,b]$ ได้อย่างไร โดยอาศัยการใช้ตัวเลขลำดับฟีโบนาซซี จะได้ความสัมพันธ์ $L_2 = \frac{F_{k-2}}{F_k} L_1$ ถ้ากำหนดให้ $a < x_1 < x_2 < b$ แล้ว ช่วงการค้นหาจะถูกลดรูปได้สองกรณี ได้แก่ $[a,x_2]$ หรือ $[x_1,b]$ เท่านั้น ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 การลดขอบเขตช่วงการค้นหา

อัลกอริทึมการค้นหาจุดต่ำสุดด้วยระเบียบวิธีฟีโบนาซซี สามารถสรุปได้ดังนี้

ขั้นที่ 1: เตรียมข้อมูล กำหนดตัวนับและค่าเริ่มต้นช่วงการค้นหา $[a,b]$ ใด ๆ

ขั้นที่ 2: จากค่าความคลาดเคลื่อนสูงสุดที่ยอมรับได้ ε คำนวณ N จาก $\delta = \frac{b-a}{F_N} \le \varepsilon$

กำหนดให้ $i = 2$ และ คำนวณ $L_2 = \frac{F_{N-2}}{F_N} \times (b-a)$

ขั้นที่ 3: คำนวณ $L_1 = b - a$

ถ้า $L_2 < 0.5 \times L_1$, ให้คำนวณ $x_1 = a + L_2$, $x_2 = b - L_2$

ถ้า $L_2 > 0.5 \times L_1$, ให้คำนวณ $x_1 = b - L_2$, $x_2 = a + L_2$

ขั้นที่ 4: ตรวจสอบค่าฟังก์ชันของจุด x_1 และ x_2

ถ้า $f(x_1) < f(x_2)$, ให้ $[a,b] = [a,x_2]$

ถ้า $f(x_1) > f(x_2)$, ให้ $[a,b] = [x_1,b]$

ถ้า $f(x_1) = f(x_2)$, ให้ $[a,b] = [x_1,x_2]$ และ $i = i + 1$

คำนวณ $L_2 = \frac{F_{N-i}}{F_{N-i+2}} L_1$

ขั้นที่ 5: เพิ่มตัวนับ $i = i + 1$ ถ้า $i < n$, ทำซ้ำขั้นที่ 3

ขั้นที่ 6: จุดคำตอบมีค่าเป็น $x_{opt} = \frac{a+b}{2}$

ตัวอย่างที่ 2.2 กำหนดฟังก์ชันวัตถุประสงค์เป็น $f(x) = x^2 - 3x \cdot exp(-x)$ ในช่วงการค้นหา $[0,1]$ โดยใช้ระเบียบวิธีการค้นหาของฟีโบนาซซีด้วยความคลาดเคลื่อนที่ยอมรับได้เท่ากับ 0.0001

วิธีทำ คำนวณจำนวนรอบสูงสุดที่ต้องใช้

จากการแปรค่า N จาก 1 ให้เพิ่มขึ้นเรื่อยๆ จะพบว่า ที่ค่า $F_{N=21} = 10946$

$$\delta = \frac{b-a}{F_N} = \frac{1-0}{10946} = 9.14\times10^{-5} \le \varepsilon \ \left(1\times10^{-4}\right)$$

กำหนดให้ $i = 2$ และ คำนวณ $L_2 = \frac{F_{N-2}}{F_N}\times(b-a) = \frac{F_{19}}{F_{22}}\times(1-0) = 0.382$

<u>รอบการคำนวณที่ 1</u>: $L_1 = 1 - 0 = 1.0$

$L_2 = 0.382 < 0.5\times1.0$

ดังนั้น $x_1 = a + L_2 = 0 + 0.382 = 0.382 \quad \Rightarrow \quad f(x_1) = -0.6362$

$x_2 = b - L_2 = 1 - 0.382 = 0.618 \quad \Rightarrow \quad f(x_2) = -0.6174$

จะพบว่า $f(x_1) < f(x_2)$ ดังนั้น $[a,b] = [a,x_2] = [0,0.618]$

$$L_2 = \frac{F_{N-i}}{F_{N-i+2}}\times L_1 = \frac{F_{19}}{F_{21}}\times1.0 = \frac{4181}{10946}\times1.0 = 0.382$$

$i = i + 1 = 3$

ในรอบการคำนวณที่ 1 นี้ ประมาณจุดคำตอบได้เป็น $\tilde{x} = \frac{a+b}{2} = \frac{0+0.618}{2} = 0.309$

<u>รอบการคำนวณที่ 2</u>: $L_1 = 0.618 - 0 = 0.618$

$L_2 = 0.382 > 0.5\times L_1 = 0.309$

ดังนั้น $x_1 = b - L_2 = 0.618 - 0.382 = 0.236 \quad \Rightarrow \quad f(x_1) = -0.5035$

$x_2 = a + L_2 = 0 + 0.382 = 0.382 \quad \Rightarrow \quad f(x_2) = -0.6362$

จะพบว่า $f(x_1) > f(x_2)$ ดังนั้น $[a,b] = [x_1,b] = [0.236,0.618]$

$$L_2 = \frac{F_{N-i}}{F_{N-i+2}}\times L_1 = \frac{F_{18}}{F_{20}}\times0.382 = \frac{2584}{6765}\times0.618 = 0.2361$$

$i = i + 1 = 4$

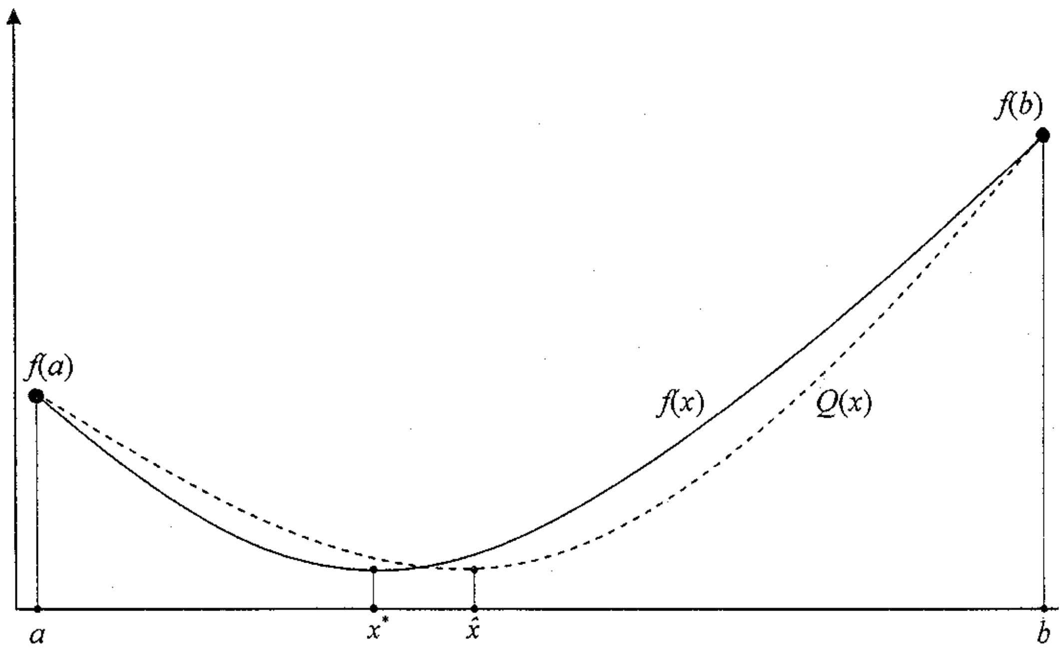
ในรอบการคำนวณที่ 2 นี้ ประมาณจุดคำตอบได้เป็น $\tilde{x} = \frac{a+b}{2} = \frac{0.236+0.618}{2} = 0.4275$

ดำเนินการคำนวณวนรอบต่อไปรวมทั้งสิ้น 19 รอบการคำนวณจะได้ผลดังนี้

รอบที่	$\tilde{x} = \frac{a+b}{2}$	$b-a$	$f(\tilde{x})$
1	0.30902	1.00000	-0.58512
2	0.42705	0.61803	-0.65349
3	0.50000	0.38197	-0.6598
4	0.45492	0.23607	-0.65899
5	0.48278	0.14590	-0.66064
6	0.46556	0.09017	-0.66007
7	0.4762	0.055728	-0.66059
8	0.48278	0.034442	-0.66064
9	0.47871	0.021286	-0.66064
10	0.48123	0.013155	-0.66065
11	0.47967	0.0081308	-0.66065
12	0.48063	0.0050247	-0.66065
13	0.48123	0.0031062	-0.66065
14	0.48086	0.0019185	-0.66065
15	0.48109	0.0011876	-0.66065

16	0.48123	0.00073086	-0.66065
17	0.48113	0.00045679	-0.66065
18	0.48109	0.00027407	-0.66065
19	0.48109	0.00018272	-0.66065

เนื่องจากสองรอบการคำนวณแรก ($i = 1$, $i = 2$) ต้องกำหนดเป็นค่าเริ่มต้น ไม่นำมาใช้ในการคำนวณ ดังนั้นจำนวนรอบที่ต้องใช้จะมีค่าเท่ากับ $N - 2 = 19$



รูปที่ 2.5 การประมาณค่าฟังก์ชันวัตถุประสงค์ด้วยสมการกำลังสอง

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 2.2</u>

// SCILAB source program

```
// ฟังก์ชันสำหรับคำนวณการค้นหาฟีโบนาซซี     fibonacci.sci
=========================
        fibonacci.sci
=========================
function [xmin,fmin] = fibonacci(fobj,xinterval,vtol,ftol)
k=0;
f0=%inf;
ferr=%inf;
delta=%inf;
F(1)=1;F(2)=1;
while delta>vtol
   if k>1 then
```

```
        F(k+1)=F(k)+F(k-1);
    end
        delta=(xinterval(2)-xinterval(1))/F(k+1);
    k=k+1;
end
xl=xinterval(1);
xu=xinterval(2);
disp('Iter  Xmin  Fmin   XU-XL   Ferr');
while (k>2)
    L1 = xu - xl;
    L2 = F(k-2)/F(k)*L1;
    if L2<0.5*L1 then
        x01 = xl + L2;
        x02 = xu - L2;
    else
        x01 = xu - L2;
        x02 = xl + L2;
    end
    if fobj(x01)<fobj(x02) then
        xu = x02;
    elseif fobj(x01)>fobj(x02) then
        xl = x01;
    else
        xl = x01;
        xu = x02;
    end
    xmin = (xl + xu)/2;
    fmin = fobj(xmin);
    ferr=abs((f0-fmin));
    f0=fmin;
    disp([k xmin xu-xl fmin ferr]);
    k=k-1;
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('f_univar01.sce');                          ประกาศฟังก์ชันวัตถุประสงค์
--> exec('fibonacci.sci');                           ประกาศฟังก์ชันเพื่อการคำนวณ
--> [xmin,fmin] = fibonacci(f_univar01,[0 1],1e-4,1e-4);
```

Iter	XL	XU	Xmin	Fmin	XU-XL	Ferr
21.	0.0000000	0.6180340	0.3090170	-0.5851199	0.6180340	Inf
20.	0.2360680	0.6180340	0.4270510	-0.6534904	0.3819660	0.0683705
19.	0.3819660	0.6180340	0.5	-0.6597960	0.2360680	0.0063055
18.	0.3819660	0.5278641	0.4549150	-0.6589856	0.1458980	0.0008103
17.	0.4376941	0.5278641	0.4827791	-0.6606442	0.0901699	0.0016586
16.	0.4376941	0.4934223	0.4655582	-0.6600663	0.0557281	0.0005779
15.	0.4589804	0.4934223	0.4762014	-0.6605931	0.0344418	0.0005268
14.	0.4721359	0.4934223	0.4827791	-0.6606442	0.0212863	0.0000511
13.	0.4721359	0.4852914	0.4787137	-0.6606373	0.0131555	0.0000069
12.	0.4771606	0.4852914	0.4812260	-0.6606510	0.0081308	0.0000137
11.	0.4771606	0.4821853	0.4796729	-0.6606461	0.0050247	0.0000049
10.	0.4790791	0.4821853	0.4806322	-0.6606505	0.0031062	0.0000044
9.	0.4802668	0.4821853	0.4812260	-0.6606510	0.0019185	0.0000005
8.	0.4802668	0.4814544	0.4808606	-0.6606509	0.0011876	0.0000001
7.	0.4807236	0.4814544	0.4810890	-0.660651	0.0007309	0.0000001
6.	0.4809976	0.4814544	0.4812260	-0.6606510	0.0004568	3.780D-08
5.	0.4809976	0.4812717	0.4811347	-0.660651	0.0002741	3.525D-08
4.	0.4809976	0.4811803	0.4810890	-0.660651	0.0001827	2.551D-09
3.	0.4810890	0.4810890	0.4810890	-0.660651	0.0000000	0.0000000

2.5 ระเบียบวิธีการค้นหากำลังสอง (quadratic search method)

ระเบียบวิธีการค้นหากำลังสองนี้ ดำเนินการด้วยการประมาณค่าฟังก์ชันที่ต้องการหาค่าต่ำสุดด้วยฟังก์ชันกำลังสองในรูป $Q(x) = px^2 + qx + r$ โดยอาศัยการคำนวณค่าสัมประสิทธิ์ p q และ r จากข้อมูลที่มีอยู่ จากสมการวิธีการนี้ต้องการข้อมูลจำนวนทั้งสิ้น 3 จุด

ถ้าช่วง $[a,b]$ มีขนาดเล็กมากพอ การประมาณจุดต่ำสุด x^* ของฟังก์ชันวัตถุประสงค์ด้วยจุดต่ำสุด $\hat{x}$ ของฟังก์ชันกำลังสองจะมีค่าความคลาดเคลื่อนที่ยอมรับได้ โดยอาศัยหลักการประมาณฟังก์ชันกำลังสองในแต่ละรอบของการคำนวณ จะได้ช่วงการค้นหา $[a,b]$ มีขนาดเล็กลงเรื่อย ๆ วิธีการนี้ต้องการความรู้เรื่องอนุพันธ์ของฟังก์ชันพหุนามเพื่อนำมาใช้ประมาณค่าจุด $\hat{x}$ เท่านั้น ไม่

ต้องการข้อมูลของอนุพันธ์ที่จุดทั้ง 3 แต่อย่างใด วิธีนี้จึงเป็นวิธีที่มีประสิทธิภาพที่สุดในวิธีการที่ไม่ใช้ข้อมูลของอนุพันธ์ของฟังก์ชันวัตถุประสงค์

กำหนดให้จุด c เป็นจุดภายในใดๆ ($a < c < b$) ที่มีคุณสมบัติดังต่อไปนี้ คือ $f(c) < f(b)$ และ $f(c) < f(a)$ โดยทั่วไปแล้ว ถ้าช่วงการค้นหา $[a,b]$ มีค่าเหมาะสม สามารถใช้จุดกึ่งกลางของช่วงการค้นหาเป็นจุด c ได้ นั่นคือ $c = \frac{a+b}{2}$ แทนค่าจุด a b และ c พร้อมกับค่าของฟังก์ชัน $f(a)$ $f(b)$ และ $f(c)$ ลงในสมการกำลังสอง $Q(x) = px^2 + qx + r$ จะได้ระบบสมการเชิงเส้น 3 ตัวแปร ดังนี้

$$\begin{bmatrix} a^2 & a & 1 \\ b^2 & b & 1 \\ c^2 & c & 1 \end{bmatrix}\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} f(a) \\ f(b) \\ f(c) \end{bmatrix} \tag{2.2}$$

จุดต่ำสุดของสมการ $Q(x) = px^2 + qx + r$ คำนวณได้จากค่าอนุพันธ์อันดับที่หนึ่งที่จุดต่ำสุด มีค่าเท่ากับ 0 นั่นคือ $\hat{x} = -\frac{q}{2p}$ แทนค่า p และ q จากการแก้สมการเมตริกซ์ข้างบนจะได้ว่า

$$\hat{x} = \frac{1}{2}\frac{\left(b^2-c^2\right)f(a)+\left(c^2-a^2\right)f(b)+\left(a^2-b^2\right)f(c)}{(b-c)f(a)+(c-a)f(b)+(a-b)f(c)} \tag{2.3}$$

วิธีการค้นหานี้สามารถสรุปเป็นขั้นตอนได้ดังนี้

ขั้นที่ 1: เตรียมข้อมูล กำหนดตัวนับและค่าเริ่มต้นช่วงการค้นหา $[a,b]$ ใดๆ

ขั้นที่ 2: คำนวณค่า $f(a)$ $f(b)$ และ $f(c)$ โดยที่ $f(c) < f(a)$ และ $f(c) < f(b)$

ขั้นที่ 3: คำนวณค่า $\hat{x}$ และค่า $f(\hat{x})$

ขั้นที่ 4: ถ้า $\hat{x} < c$ และ $f(\hat{x}) < f(c)$ ให้ทำการเซตค่าดังนี้

$a = a; b = c;$

ถ้า $\hat{x} > c$ และ $f(\hat{x}) > f(c)$ ให้ทำการเซตค่าดังนี้

$a = a; b = \hat{x};$

ถ้า $\hat{x} < c$ และ $f(\hat{x}) > f(c)$ ให้ทำการเซตค่าดังนี้

$a = \hat{x}; b = b;$

ถ้า $\hat{x} > c$ และ $f(\hat{x}) < f(c)$ ให้ทำการเซตค่าดังนี้

$a = c; b = b;$

ขั้นที่ 5: สิ้นสุดการค้นหา ถ้า $b - a <$ ค่าผิดพลาดสูงสุดที่ยอมรับได้ จุดต่ำสุด คือ $\hat{x}$ ถ้าไม่ใช่ ให้ทำซ้ำขั้นตอนที่ 2

ตัวอย่างที่ 2.3 กำหนดฟังก์ชันวัตถุประสงค์เป็น $f(x) = x^2 - 3x \cdot exp(-x)$ ในช่วงการค้นหา $[0,1]$ โดยใช้ระเบียบวิธีการค้นหากำลังสองด้วยความคลาดเคลื่อนที่ยอมรับได้เท่ากับ 0.0001

วิธีทำ

รอบการคำนวณที่ 1:

$a = 0.0, \qquad \Rightarrow f(a) = 0.0$

$b = 1.0, \qquad \Rightarrow f(b) = -0.1036$

$c = \dfrac{a+b}{2} = 0.5\,, \qquad \Rightarrow f(c) = -0.6598$

$$\hat{x} = \frac{1}{2}\frac{\left(b^2-c^2\right)f(a)+\left(c^2-a^2\right)f(b)+\left(a^2-b^2\right)f(c)}{(b-c)f(a)+(c-a)f(b)+(a-b)f(c)} = 0.5213$$

$f(\hat{x}) = -0.6568$

ตรวจสอบเงื่อนไขจะพบว่า $\hat{x} > c$ และ $f(\hat{x}) > f(c)$

$a = a = 0.0;\ b = \hat{x} = 0.5213;$

$b - a = 0.5$ คำนวณต่อไป

รอบการคำนวณที่ 2:

$a = 0.0, \qquad \Rightarrow f(a) = 0.0$

$b = 0.5213, \qquad \Rightarrow f(b) = -0.6568$

$c = \dfrac{a+b}{2} = 0.2606\,, \qquad \Rightarrow f(c) = -0.5346$

$$\hat{x} = \frac{1}{2}\frac{\left(b^2-c^2\right)f(a)+\left(c^2-a^2\right)f(b)+\left(a^2-b^2\right)f(c)}{(b-c)f(a)+(c-a)f(b)+(a-b)f(c)} = 0.4681$$

$f(\hat{x}) = -0.6602$

ตรวจสอบเงื่อนไขจะพบว่า $\hat{x} > c$ และ $f(\hat{x}) < f(c)$

$a = c = 0.2606;\ b = b = 0.5213;$

$b - a = 0.5213 - 0.2606 = 0.2607$ คำนวณต่อไป

จากการคำนวณวนรอบต่อไปจะได้ว่า

รอบที่	a	b	$\hat{x}$	$f(\hat{x})$
1	0.00000	1.00000	0.52131	-0.65681
2	0.00000	0.52131	0.46822	-0.66025
3	0.26065	0.52131	0.47986	-0.66065
4	0.39098	0.52131	0.48148	-0.66065
5	0.45614	0.52131	0.48124	-0.66065
6	0.45614	0.48873	0.48111	-0.66065
7	0.47244	0.48873	0.48111	-0.66065
8	0.48058	0.48873	0.48110	-0.66065
9	0.48058	0.48465	0.48110	-0.66065
10	0.48058	0.48262	0.48110	-0.66065
11	0.48058	0.48160	0.48110	-0.66065
12	0.48109	0.48160	0.48110	-0.66065
13	0.48109	0.48134	0.48110	-0.66065
14	0.48109	0.48122	0.48110	-0.66065

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 2.3</u>

// SCILAB source program

```
// ฟังก์ชันสำหรับคำนวณการค้นหากำลังสอง      quadratic.sci
========================
        quadratic.sci
========================
function [xmin,fmin] = quadratic(fobj,xinterval,vtol,opt)
k=0;
a=xinterval(1);
b=xinterval(2);
while (b-a>vtol)
   k=k+1;
   c=(a+b)/2;
   fa=fobj(a);
   fb=fobj(b);
   fc=fobj(c);
   xs=0.5*((b^2-c^2)*fa+(c^2-a^2)*fb+(a^2-b^2)*fc)/((b-c)*fa+(c-a)*fb+(a-b)*fc);
   fs=fobj(xs);
   if opt==1 then
      disp([k  a  b  xs  fs  b-a]);
   end
   if (xs<c)&(fs<fc) then
      a=a;
      b=c;
   elseif (xs>c)&(fs>fc) then
      a=a;
      b=xs;
   elseif (xs<c)&(fs>fc) then
      a=xs;
      b=b;
   elseif (xs>c)&(fs<fc) then
      a=c;
      b=b;
```

```
  end
end
xmin=c;
fmin=fc;
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('f_univar01.sce');                    ประกาศฟังก์ชันวัตถุประสงค์
--> exec('quadratic.sci');                     ประกาศฟังก์ชันเพื่อการคำนวณ
--> [xmin,fmin] = quadratic(f_univar01,[0 1],1e-4,1);
 Iter   a          b          Xmin       Fmin        b – a
  1.  0.0000000  1.0000000  0.5213080  -0.6568076  1.0000000
  2.  0.0000000  0.5213080  0.4682244  -0.6602501  0.5213080
  3.  0.2606540  0.5213080  0.4798576  -0.6606473  0.2606540
  4.  0.3909810  0.5213080  0.4814801  -0.6606507  0.1303270
  5.  0.4561445  0.5213080  0.4812439  -0.6606510  0.0651635
  6.  0.4561445  0.4887263  0.4811066  -0.660651   0.0325818
  7.  0.4724354  0.4887263  0.4811108  -0.660651   0.0162909
  8.  0.4805808  0.4887263  0.4810968  -0.660651   0.0081454
  9.  0.4805808  0.4846536  0.4810998  -0.660651   0.0040727
 10.  0.4805808  0.4826172  0.4811003  -0.660651   0.0020364
 11.  0.4805808  0.4815990  0.4811003  -0.660651   0.0010182
 12.  0.4810899  0.4815990  0.4811002  -0.660651   0.0005091
 13.  0.4810899  0.4813445  0.4811002  -0.660651   0.0002545
 14.  0.4810899  0.4812172  0.4811002  -0.660651   0.0001273
```

2.6 ระเบียบวิธีการค้นหากำลังสาม (cubic search method)

วิธีการสุดท้ายสำหรับการหาค่าต่ำสุดขอฟังก์ชันตัวแปรเดี่ยว และได้รับการยอมรับว่าเป็น วิธีที่มีประสิทธิภาพมากที่สุดใช้หลักการเช่นเดียวกับระเบียบวิธีการค้นหากำลังสอง มีความแตกต่าง กันตรงที่ วิธีการนี้อาศัยการประมาณค่าฟังก์ชันด้วยฟังก์ชันกำลังสามหรือพหุนามอันดับที่สามใน รูป $Q(x) = px^3 + qx^2 + rx + s$ โดย ต้องการข้อมูลของจุดพร้อมอนุพันธ์ที่จุดดังกล่าวเพียง 2 จุด เท่านั้น จากเงื่อนไขของค่าอนุพันธ์ที่จุดต่ำสุดมีค่าเท่ากับ 0 นั่นคือ

$$\frac{d}{dx}Q(x) = 3px^2 + 2qx + r = 0 \tag{2.4}$$

จุดต่ำสุดจากเงื่อนไขดังกล่าวคำนวณได้ดังนี้

$$\hat{x} = \frac{-q \pm \sqrt{q^2 - 3pr}}{3p} \tag{2.5}$$

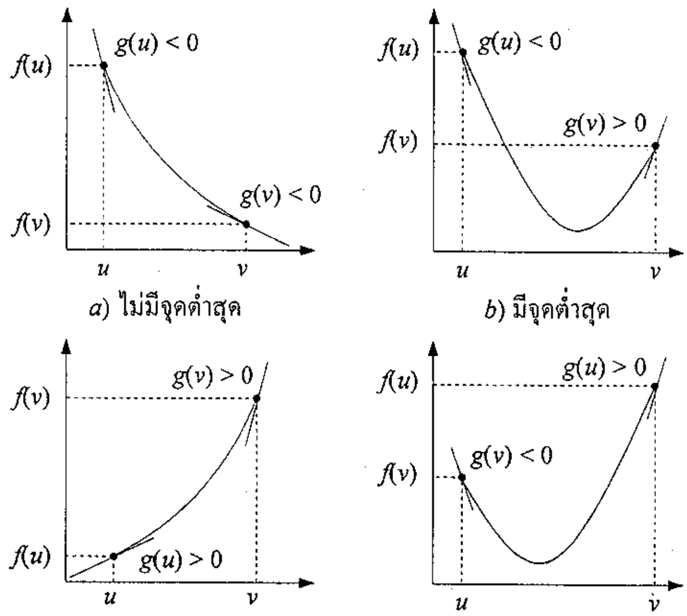
จากจุดที่ได้ จะมี 2 จุดด้วยกัน การหาจุดต่ำสุด ต้องการข้อมูลเพิ่มเติมจากอนุพันธ์อันดับสอง $\frac{d^2}{dx^2}Q(x) = 6px + 2q$ ซึ่งอนุพันธ์อันดับสองที่จุดต่ำสุดจะต้องมีค่าเป็นบวก

ค่า $\hat{x}$ สามารถคำนวณได้จากสมการต่อไปนี้ ซึ่งได้ทำการเสนอและพิสูจน์โดยเดวิดัน (Davidon) ในปี 1959 ในที่นี้จะข้ามบทพิสูจน์ของที่มาของสูตรดังกล่าว จะได้ว่า

$$\hat{x} = a + (b-a)\left\{1 - \frac{g(b) + V - W}{g(b) - g(a) + 2V}\right\} \tag{2.6}$$

โดยที่ $V = \sqrt{W^2 - g(a)g(b)}$

$$W = \frac{3}{b-a}\left[f(a) - f(b)\right] + g(a) + g(b)$$



รูปที่ 2.6 การตรวจสอบช่วงที่มีจุดต่ำสุดบรรจุอยู่

ในการพิจารณาว่าช่วงการค้นหาย่อยช่วงใดระหว่าง $[a, \hat{x}]$ และ $[\hat{x}, b]$ ที่มีจุดต่ำสุดบรรจุอยู่ให้ใช้หลักการดังนี้

ถ้าผลคูณของค่าอนุพันธ์ที่จุดขอบของช่วงที่พิจารณา $[u,v]$ น้อยกว่าศูนย์ $g(u)g(v) < 0$ ช่วงที่พิจารณานี้มีจุดต่ำสุดบรรจุอยู่ แต่ถ้า $g(u)g(v) > 0$ ช่วงดังกล่าวไม่มีจุดต่ำสุดบรรจุอยู่ พิจารณาได้จากรูปที่ 2.6 ดังนั้นในการพิจารณาด้วยวิธีการนี้ทำการประเมินว่า ช่วงใด $[a, \hat{x}]$ หรือ $[\hat{x}, b]$ ที่มีจุดต่ำสุดบรรจุอยู่ และจะทำการแบ่งช่วงย่อย ไปเรื่อย ๆ จนกระทั่งความกว้างของช่วงใน

การค้นหา มีค่าน้อยกว่าค่าที่ตั้งไว้ จุดต่ำสุดที่ประมาณได้ คือ จุดกึ่งกลางของช่วงการค้นหาสุดท้ายที่ได้จากกระบวนการข้างต้นนั่นเอง

ตัวอย่างที่ 2.4 กำหนดฟังก์ชันวัตถุประสงค์เป็น $f(x)=x^2-3x\cdot exp(-x)$ ในช่วงการค้นหา [0,1] โดยใช้ระเบียบวิธีการค้นหากำลังสามด้วยความคลาดเคลื่อนที่ยอมรับได้เท่ากับ 0.0001

วิธีทำ คำนวณอนุพันธ์ของฟังก์ชันได้

$$g(x)=\frac{d\,f(x)}{dx}=2x+3x\cdot exp(-x)-3exp(-x)$$

<u>รอบการคำนวณที่ 1</u>:

$$a=0, \qquad \Rightarrow f(a)=0, \qquad \Rightarrow g(a)=-3$$
$$b=1, \qquad \Rightarrow f(b)=-0.1036, \qquad \Rightarrow g(a)=2$$
$$W=\frac{3}{b-a}\left[f(a)-f(b)\right]+g(a)+g(b)=-0.6892$$
$$V=\sqrt{W^2-g(a)g(b)}=2.5446$$
$$\hat{x}=a+(b-a)\left\{1-\frac{g(b)+V-W}{g(b)-g(a)+2V}\right\}=0.4812$$

ดังนั้น $\hat{x}=0.4812, \quad \Rightarrow f(\hat{x})=-0.6607, \quad \Rightarrow g(\hat{x})=0.0007$

ตรวจสอบช่วง $[a,\hat{x}]$ = [0,0.4812]

$g(a)g(\hat{x})=-3\times 0.0007<0$ ช่วงนี้มีจุดต่ำสุดบรรจุอยู่

จะได้ว่า $[a,b]$ = [0.0.4812] เพื่อใช้ในรอบถัดไป

<u>รอบการคำนวณที่ 2</u>:

$$a=0, \qquad \Rightarrow f(a)=0, \qquad \Rightarrow g(a)=-3$$
$$b=0.4812, \qquad \Rightarrow f(b)=-0.6607, \qquad \Rightarrow g(a)=0.0007$$
$$W=\frac{3}{b-a}\left[f(a)-f(b)\right]+g(a)+g(b)=1.1198$$
$$V=\sqrt{W^2-g(a)g(b)}=1.1207$$
$$\hat{x}=a+(b-a)\left\{1-\frac{g(b)+V-W}{g(b)-g(a)+2V}\right\}=0.4810$$

ดังนั้น $\hat{x}=0.4810, \quad \Rightarrow f(\hat{x})=-0.6607, \quad \Rightarrow g(\hat{x})=-0.0002$

ตรวจสอบช่วง $[a,\hat{x}]$ = [0,0.4810]

$g(a)g(\hat{x})=-3\times -0.0002>0$ ช่วงนี้ไม่มีจุดต่ำสุดบรรจุอยู่

ตรวจสอบช่วง $[\hat{x},a]$ = [0.4810,0.4812]

$g(\hat{x})g(b)=-0.0002\times 0.0007<0$ ช่วงนี้มีจุดต่ำสุดบรรจุอยู่

จะได้ว่า $[a,b]$ = [0.4810,0.4812] เพื่อใช้ในรอบถัดไป

เนื่องจากความคลาดเคลื่อนมีค่าน้อย นั่นคือ $b-a$ = 0.4812 − 0.4810 = 0.0002

ในที่นี้จะสิ้นสุดการคำนวณ และจะได้ผลเฉลย $x^* = \dfrac{a+b}{2} = \dfrac{0.4810+0.4812}{2} = 0.4811$

โดยใช้ระเบียบวิธีการค้นหากำลังสาม ผลเฉลยจะถูกค้นพบภายใน 2 รอบการคำนวณเท่านั้น ถึงแม้จะมีระเบียบวิธีการอื่น ๆ ที่อาจจะนำมาใช้งานได้ แต่วิธีที่ได้รับความนิยมและมีประสิทธิภาพ ได้แก่ การค้นหากำลังสองและการค้นหากำลังสาม สำหรับระเบียบวิธีนิวตัน ถึงแม้จะมีประสิทธิภาพสูง ข้อด้อยที่สำคัญของวิธีการนี้ คือ การไม่อาจจะรับประกันจุดคำตอบได้ในทุกกรณี การแก้ปัญหาด้วยระเบียบวิธีนิวตันนั้นจะลู่เข้า ก็ต่อเมื่อ ทุก ๆ รอบการคำนวณค่าอนุพันธ์ของฟังก์ชันของผลเฉลยที่ถูกปรับปรุงมีค่าเป็นบวกทุกค่า ดังนั้น ในที่นี้จะไม่นำเสนอระเบียบวิธีนิวตันเพื่อใช้แก้ปัญหาการค้นหาช่วงก้าวที่เหมาะสม

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 2.4</u>

// SCILAB source program

```
// ฟังก์ชันสำหรับคำนวณการค้นหากำลังสอง      cubicsearch.sci
========================
        cubicsearch.sci
========================
function [xmin,fmin]=cubicsearch(fobj,fgrad,xinterval,vtol,gtol,opt)
k=0;
gmin=%inf;
f0=%inf;
xl=xinterval(1);
xu=xinterval(2);
fa=fobj(xl);
fb=fobj(xu);
ga=fgrad(xl);
gb=fgrad(xu);
while (xu-xl>vtol)&(gmin>gtol)
   k=k+1;
   W=3*(fa-fb)/(xu-xl)+ga+gb;
   V=sqrt(W*W-ga*gb);
   U=1-(gb+V-W)/(gb-ga+2*V);
   xmin=xl+(xu-xl)*U;
   fmin=fobj(xmin);
   gmin=fgrad(xmin);
   if ((ga<0)&((gmin>0)|(fmin>fa))) then
```

```
          xu=xmin;
      fb=fmin;
      gb=gmin;
   else
         xl=xmin;
      fa=fmin;
      ga=gmin;
   end
   if (opt==1) then
      disp([k xl xu xmin fmin xu-xl gmin]);
   end
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('f_univar01.sce');                    ประกาศฟังก์ชันวัตถุประสงค์
--> exec('g_univar01.sce');                    ประกาศเกรเดียนต์ของฟังก์ชันวัตถุประสงค์
--> exec('cubicsearch.sci');                   ประกาศฟังก์ชันเพื่อการคำนวณ
--> [xmin,fmin]=cubicsearch(f_univar01,g_univar01,[0 1],1e-4,1e-4,1);
  Iter    XL          XU          Xmin        Fmin        b – a       ||Gmin||
   1.     0.0000000   0.4812586   0.4812586   -0.6606509  0.4812586   0.0007625
   2.     0.4810947   0.4812586   0.4810947   -0.660651   0.0001638   0.0000265
```

2.7 การกำหนดช่วงที่มีค่าต่ำสุดบรรจุอยู่

วิธีการค้นหาที่กล่าวมาข้างต้นนั้น ตั้งอยู่บนพื้นฐานที่ทราบช่วงการค้นหา $[a,b]$ ที่มีจุดต่ำสุดอยู่ แต่คำถามคือ ในการแก้ปัญหาการค้นหาตามเส้นจะกำหนดช่วงการค้นหานี้ได้อย่างไร ในที่นี้จะนำเสนอวิธีการเลือกช่วงการค้นหาดังกล่าวอย่างง่ายดังนี้

ทำการคำนวณค่าฟังก์ชันที่จุดใด ๆ กำหนดให้เป็นจุด a แต่ปัญหาการค้นหาตามเส้นนั้น ค่า λ ต้องมีค่ามากกว่า 0 เสมอ ดังนั้น เลือก $a = 0$ จึงมีความเหมาะสมที่สุด เลือกจุดที่ 2 โดยการบวกค่า a ด้วย h ($h > 0$) ทำการประเมินค่า $f(a + h)$ ถ้า $f(a + h) > f(a)$ นั่นแสดงว่าระหว่างจุด a กับจุดดังกล่าวมีจุดต่ำสุดบรรจุอยู่ แต่ถ้าไม่ใช่ ให้ทำการเลื่อนจุดที่ 2 ออกไปอีกโดยการบวกด้วยค่า kh โดยที่ k เป็นจำนวนเต็มบวก สามารถกล่าวโดยสรุปเป็นขั้นตอนได้ดังนี้

<u>ขั้นที่ 1</u>: กำหนด $k = 1$ และ $a = 0$ เริ่มต้นการค้นหาจุด b

<u>ขั้นที่ 2</u>: ถ้า $f(a + kh) < f(a)$ ให้เซต $k = k + \Delta k$

ทำซ้ำขั้นตอนนี้ จนกว่า $f(a + kh) > f(a)$

จะได้จุด $b = a + kh$

ตัวอย่างที่ 2.5 กำหนดฟังก์ชันวัตถุประสงค์ของปัญหาการค้นหาตามเส้นเป็น

$$f(\lambda) = \lambda^2 - 12\lambda + 15$$

จงคำนวณขอบเขตการค้นหาที่มีจุดต่ำสุดบรรจุอยู่ และให้ค้นจุดต่ำสุดโดยใช้ระเบียบวิธีการคำนวณที่เหมาะสมด้วยขอบเขตการค้นหาที่ได้

วิธีทำ ใช้ค่า $h = 1.0$, $\Delta k = 2$ ในการคำนวณ

<u>รอบการคำนวณที่ 1</u>: $k = 1$, $a = 0$ และ $kh = 1.0$

จะได้ $f(a) = 15, f(a + kh) = 4$

$f(a + kh) < f(a)$ ให้เซต $k = k + \Delta k = 1 + 2 = 3$

<u>รอบการคำนวณที่ 2</u>: $k = 3$, $a = 0$ และ $kh = 3.0$

จะได้ $f(a) = 15, f(a + kh) = -12$

$f(a + kh) < f(a)$ ให้เซต $k = k + \Delta k = 3 + 2 = 5$

<u>รอบการคำนวณที่ 3</u>: $k = 5$, $a = 0$ และ $kh = 5.0$

จะได้ $f(a) = 15, f(a + kh) = -20$

$f(a + kh) < f(a)$ ให้เซต $k = k + \Delta k = 5 + 2 = 7$

<u>รอบการคำนวณที่ 4</u>: $k = 7$, $a = 0$ และ $kh = 7.0$

จะได้ $f(a) = 15, f(a + kh) = -20$

$f(a + kh) < f(a)$ ให้เซต $k = k + \Delta k = 7 + 2 = 9$

<u>รอบการคำนวณที่ 5</u>: $k = 9$, $a = 0$ และ $kh = 9.0$

จะได้ $f(a) = 15, f(a + kh) = -12$

$f(a + kh) < f(a)$ ให้เซต $k = k + \Delta k = 9 + 2 = 11$

<u>รอบการคำนวณที่ 6</u>: $k = 11$, $a = 0$ และ $kh = 11.0$

จะได้ $f(a) = 15, f(a + kh) = 4$

$f(a + kh) < f(a)$ ให้เซต $k = k + \Delta k = 11 + 2 = 13$

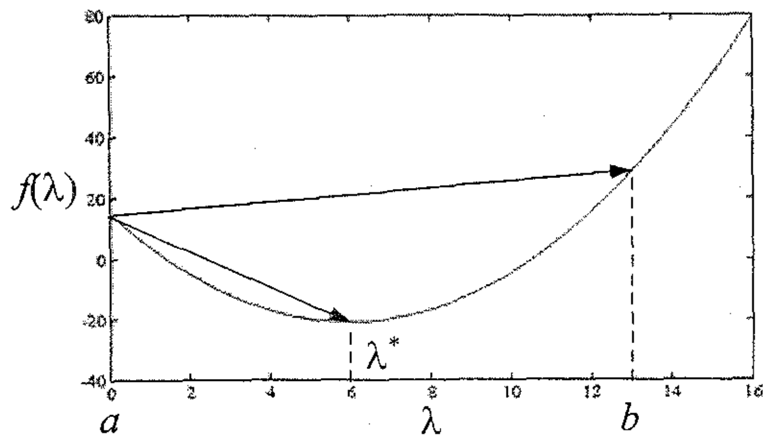
<u>รอบการคำนวณที่ 7</u>: $k = 13$, $a = 0$ และ $kh = 13.0$

จะได้ $f(a) = 15, f(a + kh) = 28$

$f(a + kh) > f(a)$ สิ้นสุดการคำนวณ จะได้ $b = a + kh = 13.0$

ดังนั้น จะได้ช่วงการค้นหาเป็น [0.0, 13.0]

ช่วงการค้นหานี้รับประกันการบรรจุต่ำสุด พิจารณาจากกราฟในรูปต่อไปนี้



ใช้ระเบียบวิธีการค้นหากำลังสาม

$$f(\lambda)=\lambda^2-12\lambda+15,\ g(\lambda)=\frac{d f(\lambda)}{d\lambda}=2\lambda-12$$

<u>รอบการคำนวณที่ 1</u>:

$$a = 0, \qquad \Rightarrow f(a) = 15, \qquad \Rightarrow g(a) = -12$$
$$b = 13, \qquad \Rightarrow f(b) = 28, \qquad \Rightarrow g(a) = 14$$
$$W=\frac{3}{b-a}\left[f(a)-f(b)\right]+g(a)+g(b)=-1.0$$
$$V=\sqrt{W^2-g(a)g(b)}=13.0$$
$$\hat{x}=a+(b-a)\left\{1-\frac{g(b)+V-W}{g(b)-g(a)+2V}\right\}=6.0$$

ดังนั้น $\hat{x}=6.0, \qquad \Rightarrow f(\hat{x})=-21, \qquad \Rightarrow g(\hat{x})=0.0$

จะพบว่า จุด $\hat{x}=6.0$ นี้ ให้ค่าเกรเดียนต์เป็นศูนย์ ดังนั้น จุดนี้เป็นจุดต่ำสุดนั่นเอง ในกรณีของฟังก์ชันกำลังสอง ระเบียบวิธีกำลังสามสามารถคำนวณจุดคำตอบได้อย่างรวดเร็ว

2.8 สรุป

บทนี้นำเสนอเทคนิคการหาค่าต่ำสุดของฟังก์ชันตัวแปรเดี่ยวที่เป็นพื้นฐานสำคัญสำหรับการแก้ปัญหาการค้นหาตามเส้น ระเบียบวิธีที่มีประสิทธิภาพและถูกนำมาใช้งานกันอย่างแพร่หลายในปัจจุบัน ได้แก่ ระเบียบวิธีกำลังสองสำหรับฟังก์ชันที่ไม่ต้องการคำนวณค่าอนุพันธ์ และระเบียบวิธีกำลังสามที่ใช้การคำนวณอนุพันธ์อันดับที่หนึ่งซึ่งมีอัตราการลู่เข้าที่รวดเร็วมาก ในเอกสารนี้ ถึงแม้จะมีการนำเสนอระเบียบวิธีการคำนวณที่หลากหลาย ระเบียบวิธีที่จะนำไปใช้ในการแก้ปัญหาการค้นหาตามเส้นในบทถัดไป ได้แก่ ระเบียบวิธีกำลังสองและระเบียบวิธีกำลังสาม

2.9 แบบฝึกหัดท้ายบท

1. จงหาจุดต่ำสุดของฟังก์ชันต่อไปนี้ในช่วงการค้นหา [0,3] ของฟังก์ชันต่อไปนี้ด้วยระเบียบวิธีการค้นหาตามกริด ระเบียบวิธีฟีโบนาชชี ระเบียบวิธีกำลังสอง และระเบียบวิธีกำลังสาม
   a. $F(\lambda) = \lambda^5 - 5\lambda^3 - 20\lambda + 5$

b. $F(\lambda)=\dfrac{0.75}{\lambda^2-2\lambda+2}+0.65(\lambda-1)tan^{-1}\left(\dfrac{1}{\lambda-1}\right)-0.65$

c. $F(\lambda) = \lambda^2 + 4\cos\lambda$

d. $F(\lambda) = -4\lambda\sin\lambda$

e. $F(\lambda)=2(\lambda-3)^2+e^{0.5\lambda^2}$

2. จงหาจุดต่ำสุดของฟังก์ชันต่อไปนี้ในช่วงการค้นหา [0,3] ของฟังก์ชันในข้อ 1. ด้วยระเบียบวิธีฟีโบนาชชี

3. จงหาจุดต่ำสุดของฟังก์ชันต่อไปนี้ในช่วงการค้นหา [0,3] ของฟังก์ชันในข้อ 1. ด้วยระเบียบวิธีกำลังสอง

4. จงหาจุดต่ำสุดของฟังก์ชันต่อไปนี้ในช่วงการค้นหา [0,3] ของฟังก์ชันในข้อ 1. ด้วยระเบียบวิธีกำลังสาม

5. จงคำนวณจุดต่ำสุดของฟังก์ชัน $f(x_1,x_2)=x_1-x_2+2x_1^2+2x_1x_2+x_2^2$ ในทิศทางการค้นหา $\boldsymbol{p} = [-1\ \ 0]^T$ จากจุด $\boldsymbol{x_0} = [0\ \ 0]^T$

6. จงคำนวณจุดต่ำสุดของฟังก์ชัน $f(x_1,x_2)=100\left(x_2-x_1^2\right)^2+(1-x_1)^2$ ในทิศทางการค้นหา $\boldsymbol{p} = [4\ \ 0]^T$ จากจุด $\boldsymbol{x_0} = [-1\ \ 1]^T$

บทที่ 3 ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไข
(Unconstrained Optimization Problems)

"Everything you know is easy"

3.1 เกริ่นนำ

บทนี้ นำเสนอเทคนิคการแก้ปัญหาค่าเหมาะที่สุดสำหรับฟังก์ชันที่ประกอบด้วยตัวแปรมากกว่าหนึ่งตัวแบบไม่มีเงื่อนไขบังคับ (unconstrained optimization problems) ภายใต้ข้อกำหนดดังนี้

$$\text{ค้นหา } \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \text{ที่ทำให้ } f(\boldsymbol{x}) \text{ มีค่าต่ำที่สุด}$$

เทคนิคดังกล่าวมีมากมาย อย่างไรก็ตาม อาจจะแบ่งออกได้เป็นสองกลุ่ม ได้แก่ ระเบียบวิธีการค้นหาโดยตรง (direct search method) หรือเรียกอีกอย่างหนึ่งว่า ระเบียบวิธีที่ไม่พึ่งพาค่าเกรเดียนต์ (non-gradient search method) และระเบียบวิธีพึ่งพาวิธีเกรเดียนต์ (gradient search method) ระเบียบวิธีการค้นหาโดยตรงนี้ ต้องการเพียงแค่ข้อมูลของฟังก์ชันวัตถุประสงค์ ณ จุดคำตอบที่สนใจ โดยไม่ต้องใช้ข้อมูลอนุพันธ์ของฟังก์ชันแต่อย่างใด แต่โดยทั่วไปมีประสิทธิภาพต่ำ การนำข้อมูลอนุพันธ์ของฟังก์ชันมาใช้งานทำให้การคำนวณมีประสิทธิภาพ แต่ปัญหาที่สำคัญของระเบียบวิธีนี้ คือ การคำนวณค่าอนุพันธ์นั่นเอง สำหรับปัญหาบางรูปแบบ อนุพันธ์ของฟังก์ชันคำนวณได้ยากหรือเสียเวลาคำนวณมากเกินไป

เพื่อให้มองเห็นภาพของระเบียบวิธีการค้นหาที่หลากหลาย ในบทนี้จะนำเสนอระเบียบวิธีการค้นหาโดยตรงทั้งสิ้น 4 วิธี ได้แก่ ระเบียบวิธีการเดินสุ่ม (random walk method) ระเบียบวิธีตัวแปรเดี่ยว (univariate method) และระเบียบวิธีทิศทางสังยุค (conjugate direction method)

3.2 ระเบียบวิธีการเดินสุ่ม (Random Walk Method)

วิธีการนี้อาศัยการสร้างตัวแปรสุ่มเข้ามาช่วยในกระบวนการกำหนดทิศทางการค้นหา จากการปรับปรุงจุดคำตอบ $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \boldsymbol{p}_k$ วิธีการนี้ จะกำหนดช่วงก้าวการค้นหาเริ่มต้น $\lambda_k = \tilde{\lambda}$ จากนั้นจะทำการสุ่มค่าทิศทางการค้นหา $\boldsymbol{p}_k$ ดังนี้

$$\boldsymbol{p}_k = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} = \frac{1}{\sqrt{u_1^2 + u_2^2 + \cdots + u_n^2}} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \qquad \text{โดยที่ } u_i \in [-1, +1] \qquad (3.1)$$

โดยการสุ่มสร้าง p_k จำนวนทั้งสิ้น N ชุดที่มากพอ จะทำให้ได้เซตของจุดที่ถูกสร้างขึ้นแบบสุ่ม N ชุด เรียกว่า เซตข้างเคียง (neighborhood set: $\aleph(x_k)$) จุดคำตอบที่ให้ค่าฟังก์ชันวัตถุประสงค์ต่ำที่สุด และต่ำกว่า $f(x_k)$ จะถูกเลือกให้เป็นผลเฉลยที่ได้รับการปรับปรุง x_{k+1} ดำเนินการคำนวณซ้ำจนกระทั่งความคลาดเคลื่อนของผลเฉลยอยู่ในเกณฑ์ที่พอใจ อาจจะสรุปการคำนวณเป็นขั้นตอนดังนี้

<u>ขั้นที่ 1</u>: กำหนดค่าเริ่มต้น $k = 1$, $i = 1$, $\lambda_k = \tilde{\lambda}$

<u>ขั้นที่ 2</u>: สุ่มทิศทางการค้นหา $p_{k,i}$, คำนวณ $x_{k+1,i} = x_k + \lambda_k p_{k,i}$

ถ้า $f(x_{k+1,i}) < f(x_k)$ ให้ $x_{k+1} = x_{k+1,i}$ เพิ่มตัวนับ $i = i + 1$

<u>ขั้นที่ 3</u>: ถ้า $i \le N$ ทำซ้ำขั้นที่ 2, ถ้าไม่ใช่ ให้ $\lambda_{k+1} = \dfrac{\lambda_k}{2}$, $k = k + 1$, $i = 1$

<u>ขั้นที่ 4</u>: ถ้า $k \le Max\_Iter$ ทำขั้นตอนที่ 2

<u>ขั้นที่ 5</u>: ค้นพบจุดคำตอบ

ตัวอย่างที่ 3.1 จงแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับต่อไปนี้

$$\text{Minimize} \qquad f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

กำหนดให้จุดเริ่มต้นการค้นหาเป็น $x_0 = [0.0 \quad 0.0]^T$

วิธีทำ

<u>รอบการคำนวณที่ 1</u>:

สร้างทิศทางการค้นหาแบบสุ่มจะได้

```
 0.0679   0.0679        0
 0.5053   0.5053  -0.1042

-0.6690  -0.6690        0
-0.5513  -0.5513   1.8189
```

$$p_{1,1} = \begin{bmatrix} 0.0679 \\ 0.5053 \end{bmatrix} \qquad \Rightarrow x_{2,1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 1.0 \times \begin{bmatrix} 0.0679 \\ 0.5053 \end{bmatrix} = \begin{bmatrix} 0.0679 \\ 0.5053 \end{bmatrix}$$

$$\Rightarrow f(x_{2,1}) = -0.1042$$

$$p_{1,2} = \begin{bmatrix} -0.6690 \\ -0.5513 \end{bmatrix} \qquad \Rightarrow x_{2,2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 1.0 \times \begin{bmatrix} -0.6690 \\ -0.5513 \end{bmatrix} = \begin{bmatrix} -0.6690 \\ -0.5513 \end{bmatrix}$$

$$\Rightarrow f(x_{2,1}) = 1.8189$$

ดำเนินการสุ่ม $p_{1,i}$ จนครบ $N = 10$ ชุด จะได้ผลเฉลยเป็น

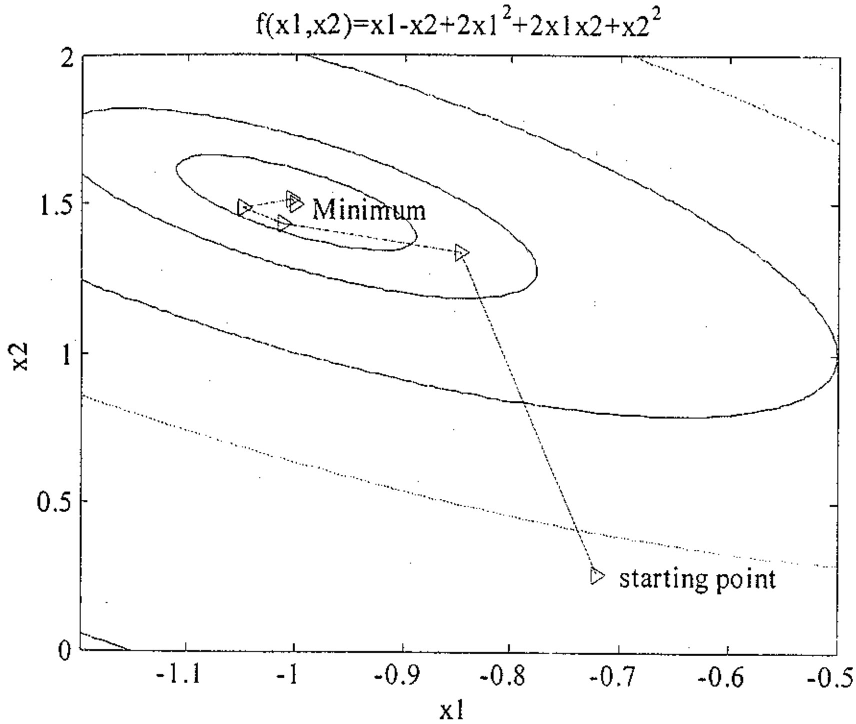
$x_2 = [-0.7207 \quad 0.2623]^T$

* <u>หมายเหตุ</u> อย่าลืมว่า ทิศทางการค้นหา $p_{1,i}$ เป็นตัวแปรสุ่ม

เข้าสู่รอบการคำนวณที่ 2

เนื่องจากการคำนวณยึดหลักของตัวแปรสุ่ม ในตัวอย่างนี้ จะนำเสนอผลเฉลยที่ได้จากการคำนวณและแผนภาพการลู่เข้าสู่จุดคำตอบดังต่อไปนี้

k	λ_k	x_1	x_2	$f(x_1,x_2)$	*error*
1.0000	1.0000	-0.7207	0.2623	-0.2534	- -
2.0000	0.5000	-0.8478	1.3353	-1.2267	0.1670
3.0000	0.2500	-1.0119	1.4279	-1.2428	0.0161
4.0000	0.1250	-1.0484	1.4841	-1.2435	0.0007
5.0000	0.0625	-1.0033	1.5114	-1.2499	0.0064
6.0000	0.0313	-1.0000	1.4968	-1.2500	0.0001



รูปที่ 3.1 รูปแบบการค้นหาของระเบียบวิธีการเดินสุ่มของตัวอย่างที่ 3.1

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 3.1</u>

// SCILAB source program

```
// กำหนดฟังก์ชันวัตถุประสงค์                    f_unc01.sci
=========================
          f_unc01.sci
=========================

function f = f_unc01(x)
x1=x(1);
x2=x(2);
f=x1-x2+2*x1^2+2*x1*x2+x2^2;
endfunction
```

```
// ฟังก์ชันสำหรับคำนวณด้วยระเบียบวิธีการเดินสุ่ม        randwalk.sci
========================
        randwalk.sci
========================
function [xmin,fmin,k] = randwalk(fobj,x0,N,lambda,max_iter,err_func)
n = length(x0);
xk = x0;
fk = fobj(x0);
fmin = fk;
xmin = xk;
ferr = %inf;
k=0;
disp([k lambda x0' fmin ferr]);
for k=1:max_iter
    for l=1:N
        pl = -1+2*rand(n,1);
        xii = xk+lambda*pl;
        fii = fobj(xii);
        if fii<fmin then
            fmin = fii;
            xmin = xii;
        end
    end
    ferr = abs(fk-fmin);
    fk = fmin;
    xk = xmin;
    disp([k lambda xmin' fmin ferr]);
    lambda = lambda/2;
    if ferr<err_func then
        break;
    end
end
endfunction
```

เนื่องจากระเบียบวิธีนี้ อ้างอิงกระบวนการสุ่มตัวเลข ดังนั้น การรันโปรแกรมแต่ละครั้งจะได้คำตอบที่แตกต่างกันเสมอนอกจากจะมีการกำหนดรูปแบบของตัวเลขสุ่มลำดับเดียวกัน ซึ่งจะไม่กล่าวถึงในที่นี้ โดยปัญหานี้มีจุดคำตอบอยู่ที่ $x^* = [-1.0 \;\; +1.5\,]^T, f(x^*) = -1.25$

```
// ผลการรันโปรแกรม
--> exec('f_unc01.sci');                     ประกาศฟังก์ชันวัตถุประสงค์
--> exec('randwalk.sci');                    ประกาศฟังก์ชันเพื่อการคำนวณ
--> [xmin,fmin,k]=randwalk(f_unc01,[0;0],10,1,100,1e-4);
// Iter  Lambda      X1           X2          F(x)          Ferr
   0.   1.0000000   0.0000000   0.0000000   0.0000000    Inf
   1.   1.0000000  - 0.7033515  0.7507754  - 0.9571745   0.9571745
   2.   0.5000000  - 0.7279479  1.0903110  - 1.1570438   0.1998692
   3.   0.2500000  - 0.7858892  1.2184339  - 1.1996064   0.0425626
   4.   0.1250000  - 0.8509493  1.3095801  - 1.2260725   0.0264661
   5.   0.0625000  - 0.9073220  1.3719191  - 1.2401574   0.0140849
   6.   0.0312500  - 0.9249430  1.3908877  - 1.2432067   0.0030493
   7.   0.0156250  - 0.9336469  1.4049021  - 1.2447710   0.0015643
   8.   0.0078125  - 0.9396921  1.4104166  - 1.2455059   0.0007349
   9.   0.0039062  - 0.9432074  1.4141506  - 1.2459303   0.0004244
   10.  0.0019531  - 0.9434345  1.4160249  - 1.2460491   0.0001188
   11.  0.0009766  - 0.9443364  1.4169980  - 1.2461542   0.0001051
   12.  0.0004883  - 0.9446223  1.4171668  - 1.2461795   0.0000253
-->[xmin,fmin,k]=randwalk(f_unc01,[0;0],10,1,100,1e-4);
   0.   1.000000    0.0000000   0.0000000   0.0000000    Inf
   1.   1.000000   - 0.7129772  0.9944052  - 1.1198442   1.1198442
   2.   0.500000   - 0.9204840  1.4090843  - 1.2435473   0.1237031
   3.   0.250000   - 0.9936519  1.4710514  - 1.2494489   0.0059017
   4.   0.125000   - 0.9936519  1.4710514  - 1.2494489   0.0000000
-->[xmin,fmin,k]=randwalk(f_unc01,[0;0],10,1,100,1e-4);
   0.   1.000000    0.0000000   0.0000000   0.0000000    Inf
   1.   1.000000   - 0.9905111  0.6992698  - 0.6238471   0.6238471
   2.   0.500000   - 0.6745843  1.1976575  - 1.1435723   0.5197252
   3.   0.250000   - 0.8847546  1.4449071  - 1.2331002   0.0895279
   4.   0.125000   - 0.9817979  1.4439648  - 1.2482373   0.0151372
   5.   0.062500   - 0.9927812  1.4601715  - 1.2488845   0.0006472
```

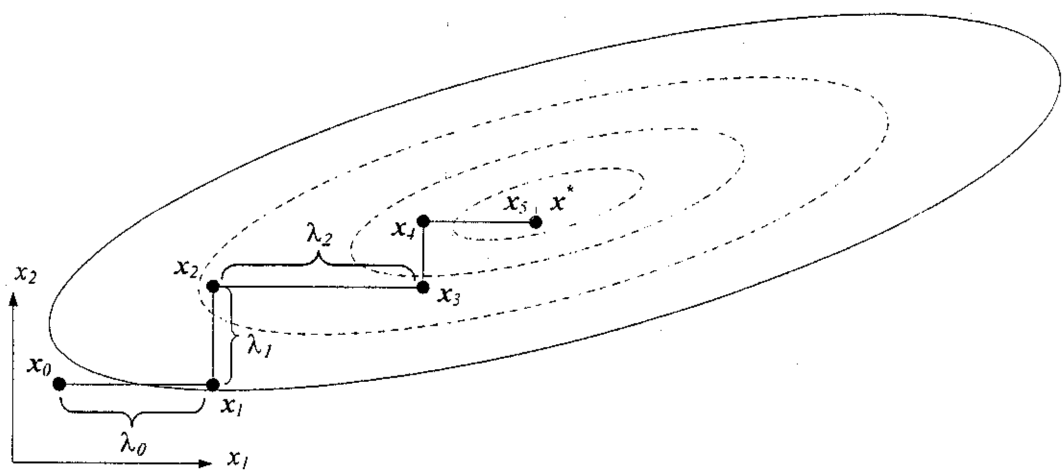
6.	0.031250	- 0.9935106	1.4860549	- 1.2499023	0.0010178
7.	0.015625	**- 0.9933617**	**1.4959031**	**- 1.2499495**	0.0000472

จากการรันโปรแกรมตัวอย่าง จำนวน 3 ครั้ง จะได้ผลเฉลยสรุปได้ดังตารางต่อไปนี้

ตารางที่ 3.1 ผลเฉลยของตัวอย่างที่ 3.1 เมื่อรันโปรแกรมจำนวน 3 ครั้ง

รันครั้งที่	x_1	x_2	$f(x^*)$	จำนวนรอบ
1	-0.9446	1.4172	-1.2462	12
2	-0.9937	1.4711	-1.2494	4
3	-0.9934	1.4602	-1.2489	7

3.3 ระเบียบวิธีตัวแปรเดี่ยว (Univariate Method)



รูปที่ 3.2 รูปแบบการค้นหาของระเบียบวิธีตัวแปรเดี่ยว

ระเบียบวิธีนี้ใช้การกำหนดทิศทางการค้นหาที่แน่นอน ได้แก่ ทิศทางที่ขนานกับแกนของตัวแปรแต่ละตัวทีละแกน ถ้ากำหนดปัญหาที่พิจารณามีตัวแปรควบคุมทั้งสิ้น n ตัวแปร การค้นหาจะแบ่งออกเป็นวัฏจักร หนึ่งวัฏจักรประกอบด้วย n การค้นหาย่อย โดยกำหนดทิศทางการค้นหาของการค้นหาย่อย ดังนี้

สำหรับรอบการค้นหาย่อยที่ k ใดๆ กำหนดให้ $\boldsymbol{p}_k = (p_1, p_2, \ldots, p_n)$ ซึ่ง $p_i = 0$ เมื่อ $i \neq k$ ถ้า $i = k$ แล้ว $p_i = 1.0$ จากทิศทางของตัวแปรหนึ่งมิติ มีความเป็นไปได้ 2 ทิศทางเท่านั้น ได้แก่ ทิศทาง $+\boldsymbol{p}_k$ และทิศทาง $-\boldsymbol{p}_k$ ดังนั้น ต้องทำการทดสอบหาทิศทางที่ถูกต้อง โดยคำนวณดังนี้

$$\begin{aligned} f^+ &= f\left(\boldsymbol{x}_k + \varepsilon \cdot \boldsymbol{p}_k\right) \\ f^- &= f\left(\boldsymbol{x}_k - \varepsilon \cdot \boldsymbol{p}_k\right) \end{aligned} \; ; \quad \text{เมื่อ } \varepsilon \text{ มีค่าเล็กๆ}$$

ถ้า $(f^+ < f_k < f^-)$ จะได้ทิศทางการค้นหาเป็น $+\boldsymbol{p}_k$

ถ้าไม่ใช่ $(f^- < f_k < f^+)$ จะได้ทิศทางการค้นหาเป็น $-\boldsymbol{p}_k$

จากนั้น ทำการแก้ปัญหาช่วงก้าวที่เหมาะสม

$$\underset{\lambda > 0}{\text{Minimize}} \quad F(\lambda) \equiv f(\boldsymbol{x}_k \pm \lambda \boldsymbol{p}_k) \tag{3.2}$$

ทำซ้ำจนกว่าจะไม่สามารถปรับปรุงผลเฉลยได้ หลักการนี้แสดงไว้ดังกราฟในรูปที่ 3.2

ตัวอย่างที่ 3.2 จงแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับโดยใช้ระเบียบวิธีตัวแปรเดี่ยว

$$\text{Minimize} \quad f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

กำหนดให้ $\boldsymbol{x}_0 = [0.0 \;\; 0.0]^T$

วิธีทำ ให้ $\varepsilon = 0.01$

<u>รอบการคำนวณที่ 1</u>:

$$f^+ = f(\boldsymbol{x}_k + \varepsilon \cdot \boldsymbol{p}_k) = f\left(\begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + 0.01 \times \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix}\right) = -0.0099$$

$$f^- = f(\boldsymbol{x}_k - \varepsilon \cdot \boldsymbol{p}_k) = f\left(\begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} - 0.01 \times \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix}\right) = 0.0101$$

จะได้ทิศทางการค้นหาเป็น $p = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}$ จากกลยุทธ์การค้นหาตามเส้น สร้างปัญหาการค้นหาช่วงก้าวที่เหมาะสม ได้เป็น

$$\underset{\lambda > 0}{\text{Minimize}} \quad F(\lambda) = f\left(\begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + \lambda \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}\right) = f\begin{pmatrix} 0 \\ \lambda \end{pmatrix} = \lambda^2 - \lambda$$

โดยใช้ระเบียบวิธีการค้นหาคิวบิก ใช้การคำนวณทั้งสิ้น 2 รอบ

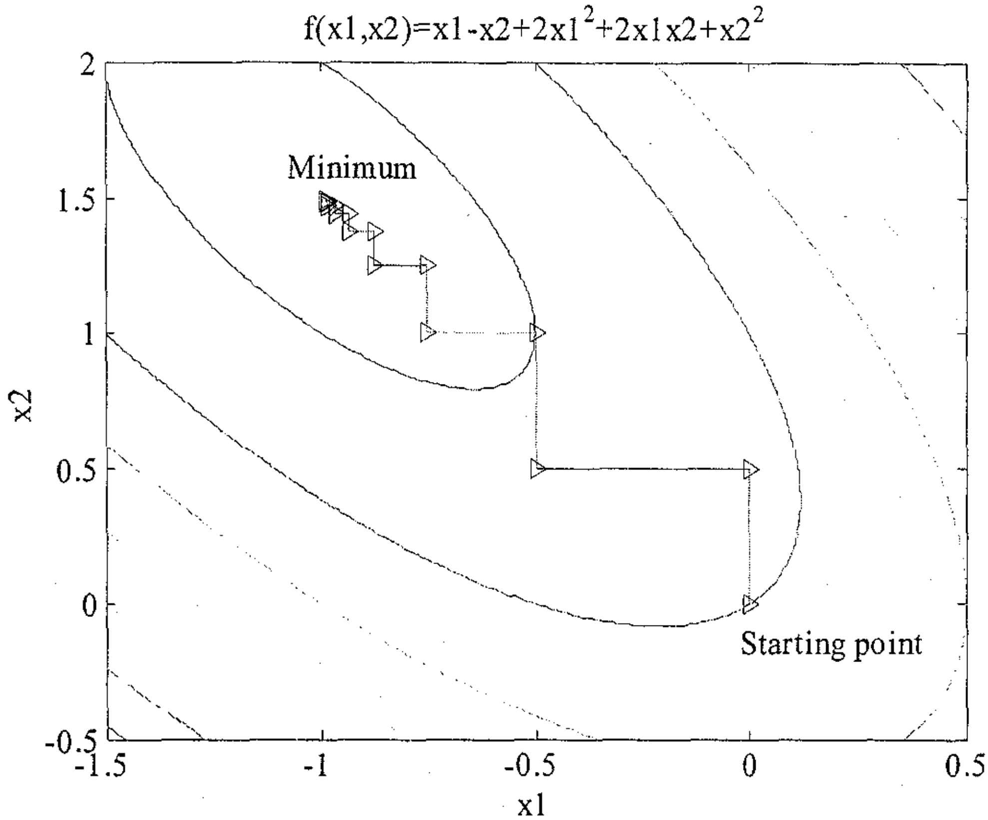
k	λ_L	λ_U	λ^*	$F(\lambda^*)$
0.0000	0.0000	1.0000	-	-
1.0000	0.0000	0.5000	0.5000	-0.2500
2.0000	0.5000	0.5000	0.5000	-0.2500

ได้ $\lambda_1 = 0.5$

ปรับปรุงจุดคำตอบได้เป็น

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \lambda_1 \boldsymbol{p}_1 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + 0.5 \times \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.5 \end{bmatrix}$$

ดำเนินการคำนวณในรอบถัดไป โดยเปลี่ยนไปใช้ $\boldsymbol{p}_2 = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}$ ทำซ้ำการคำนวณวนรอบ จะได้ผลเฉลยดังกราฟในรูปที่ 3.3



รูปที่ 3.3 การลู่เข้าของระเบียบวิธีตัวแปรเดี่ยวสำหรับตัวอย่างที่ 3.2

โปรแกรม SCILAB สำหรับตัวอย่างที่ 3.2

// SCILAB source programs

```
// กำหนดฟังก์ชันการค้นหาตามเส้น                    lsearch.sci
========================
       lsearch.sci
========================
function x=lsearch(fobj,x0,p)
global x_init s_dir fobj_g
f0=fobj(x0);
f1=fobj(x0+p);
L0=0;L1=1;
if f0>f1 then
  k=1;
  while f0>f1
    L1=L1+2*k*L1;
    f1=fobj(x0+L1*p);
    k=k+1;
```

```
  end
end
x_init=x0;
s_dir=p;
fobj_g=fobj;
x=cubicsearch1(flsearch,[L0 L1],1e-4,1e-4,0);
endfunction
```

```
// กำหนดฟังก์ชันวัตถุประสงค์ของฟังก์ชันการค้นหาตามเส้น          flsearch.sci
========================
        flsearch.sci
========================
function f=flsearch(z)
global x_init s_dir fobj_g
x=x_init+z*s_dir;
f=fobj_g(x);
endfunction
```

```
// กำหนดฟังก์ชันคำนวณค่าอนุพันธ์เชิงตัวเลข          fgradnum1.sci
========================
        fgradnum1.sci
========================
function g = fgradnum1(fname,x)
dx=1e-6;
x=x-dx;
f1=fname(x);
x=x+2*dx;
f2=fname(x);
g=(f2-f1)/(2*dx);
endfunction
```

```
// กำหนดฟังก์ชันคำนวณการค้นหากำลังสาม          cubicsearch1.sci
========================
        cubicsearch1.sci
========================
```

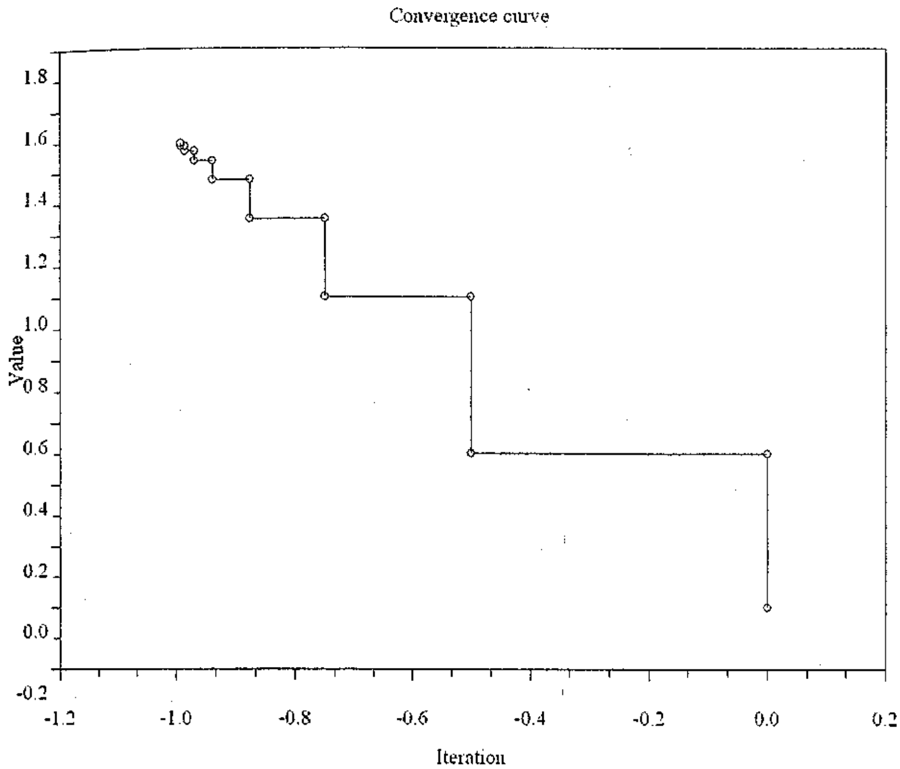
```
function [xmin,fmin]=cubicsearch1(fobj,xinterval,vtol,ftol,opt)
k=0;ferr=%inf;
f0=%inf;
xl=xinterval(1);
xu=xinterval(2);
fa=fobj(xl);
fb=fobj(xu);
ga=fgradnum1(fobj,xl);
gb=fgradnum1(fobj,xu);
while (xu-xl>vtol)&(ferr>ftol)
   k=k+1;
   W=3*(fa-fb)/(xu-xl)+ga+gb;
   V=sqrt(W*W-ga*gb);
   U=1-(gb+V-W)/(gb-ga+2*V);
   xmin=xl+(xu-xl)*U;
   fmin=fobj(xmin);
   gmin=fgradnum1(fobj,xmin);
   if ((ga<0)&((gmin>0)|(fmin>fa))) then
            xu=xmin;
      fb=fmin;
      gb=gmin;
   else
       xl=xmin;
      fa=fmin;
      ga=gmin;
   end
   ferr=abs(f0-fmin);
   f0=fmin;
   if (opt==1) then
      disp([k xl xu xmin fmin]);
   end
end
endfunction
```

```
//กำหนดฟังก์ชันสำหรับคำนวณด้วยระเบียบวิธีตัวแปรเดี่ยว                    univar.sci
========================
        univar.sci
========================
function [xmin,fmin,k]=univar(fobj,x0,epsilon,ftol,opt)
n=length(x0);
f0=fobj(x0);
k=0;
ferr=%inf;
if opt==1 then
   disp([k x0' f0 ferr])
end
while (ferr>ftol)
   k=k+1;
   m=modulo(k,n);
   p=zeros(n,1);
   p(m+1,1)=1.0;
   fp=fobj(x0+epsilon*p);
   fn=fobj(x0-epsilon*p);
   if fp<f0 then
      p=p;
   elseif fn<f0 then
      p=-p;
   else
      break;
   end
   lambda=lsearch(fobj,x0,p);
   xmin=x0+lambda*p;
   fmin=fobj(xmin)
   ferr=abs(fmin-f0)
   if opt==1 then
      disp([k xmin' fmin ferr])
   elseif opt==2
      plot2d([x0(1) xmin(1)],[x0(2) xmin(2)],style=[-9]);
```

```
      plot2d([x0(1) xmin(1)],[x0(2) xmin(2)]);
      xtitle('Convergence curve','Iteration','Value');
      xgrid(2);
    end
    f0=fmin;
    x0=xmin;
  end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('f_unc01.sci');                 ประกาศฟังก์ชันวัตถุประสงค์
--> exec('lsearch.sci');                 ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('flsearch.sci');                ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('fgradnum1.sci');               ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('cubicsearch1.sci');            ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('univar.sci');                  ประกาศฟังก์ชันหลักของการคำนวณ
-->[xmin,fmin,k]=univar(f_unc01,[0;0],0.01,1e-4,1);
// Iter      X1           X2           F(x)         Ferr
    0.    0.0000000    0.0000000    0.0000000    Inf
    1.    0.0000000    0.5000000  - 0.2500000    0.2500000
    2.  - 0.5000000    0.5000000  - 0.7500000    0.5000000
    3.  - 0.5000000    1.0000000  - 1.0000000    0.2500000
    4.  - 0.7500000    1.0000000  - 1.1250000    0.1250000
    5.  - 0.7500000    1.2500000  - 1.1875000    0.0625000
    6.  - 0.8750000    1.2500000  - 1.2187500    0.0312500
    7.  - 0.8750000    1.3750000  - 1.2343750    0.0156250
    8.  - 0.9375000    1.3750000  - 1.2421875    0.0078125
    9.  - 0.9375000    1.4375000  - 1.2460938    0.0039062
    10. - 0.9687500    1.4375000  - 1.2480469    0.0019531
    11. - 0.9687500    1.4687500  - 1.2490234    0.0009766
    12. - 0.9843750    1.4687500  - 1.2495117    0.0004883
    13. - 0.9843750    1.4843750  - 1.2497559    0.0002441
    14. - 0.9921875    1.4843750  - 1.2498779    0.0001221
    15. - 0.9921875    1.4921875  - 1.2499390    0.0000610
```

เนื่องจาก โปรแกรมนี้ ได้เขียนชุดคำสั่งวาดกราฟอัตราการลู่เข้าไว้ด้วย จะได้ผลดังกราฟต่อไปนี้



รูปที่ 3.4 กราฟการลู่เข้าของระเบียบวิธีตัวแปรเดี่ยวสำหรับตัวอย่างที่ 3.2

3.4 ระเบียบวิธีทิศทางสังยุค (Conjugate Direction Method)

ระเบียบวิธีทิศทางสังยุคนี้ สามารถค้นหาจุดต่ำสุดของฟังก์ชันกำลังสองได้โดยใช้จำนวนรอบการค้นหาจำกัด และเนื่องจากฟังก์ชันกำลังสองนำมาใช้ประมาณฟังก์ชันวัตถุประสงค์ไม่เชิงเส้นใด ๆ ได้ ทำให้การใช้เทคนิคนี้ช่วยแก้ปัญหาไม่เชิงเส้นค้นหาผลเฉลยได้รวดเร็ว

<u>นิยาม</u>: ทิศทางการค้นหาสองชุด p_i และ p_k ใด ๆ จะเรียกว่ามีคุณสมบัติการเป็นทิศทางสังยุคซึ่งกันและกัน ก็ต่อเมื่อ $p_i^T Q\, p_k = 0$

ในกรณีของระเบียบวิธีทิศทางสังยุค ทิศทางการค้นหาที่สร้างขึ้นในแต่ละรอบการคำนวณ $\{p_0, p_1, p_2, \ldots, p_{n-1}\}$ ต้องมีคุณสมบัติ $p_i^T Q\, p_k = 0$ เมื่อ $0 \le i \ne k \le n-1$ โดยที่ Q เป็นเมตริกซ์ *pdf* (positive definite matrix)

สำหรับปัญหาการหาค่าเหมาะที่สุดนี้ Q คือ เมตริกซ์เฮสเซียน H นั่นเอง ในกรณีของฟังก์ชันกำลังสอง เมตริกซ์เฮสเซียนมีค่าคงที่ ทำให้รับประกันได้ว่าจะค้นพบจุดคำนวณในการคำนวณวนรอบไม่เกิน n รอบ เมื่อ n คือ จำนวนมิติของปัญหา พิจารณาปัญหาที่มีฟังก์ชันวัตถุประสงค์ในรูปฟังก์ชันไม่เชิงเส้นใด ๆ ดังต่อไปนี้

$$\text{Minimize} \quad f(x) \tag{3.3}$$

โดยใช้อนุกรมเทย์เลอร์ประมาณค่าฟังก์ชันจะได้ว่า

$$f(\boldsymbol{x}_{k+1}) = f(\boldsymbol{x}_k + \Delta\boldsymbol{x}_k) \approx f(\boldsymbol{x}_k) + \nabla^T f(\boldsymbol{x}_k)\Delta\boldsymbol{x}_k + \tfrac{1}{2}(\Delta\boldsymbol{x}_k)^T H(\boldsymbol{x}_k)\Delta\boldsymbol{x}_k$$

$$\Delta\boldsymbol{x}_k = \lambda_k \boldsymbol{p}_k$$

$$f(\boldsymbol{x}_{k+1}) \approx f(\boldsymbol{x}_k) + \nabla^T f(\boldsymbol{x}_k)\lambda_k \boldsymbol{p}_k + \tfrac{1}{2}(\lambda_k \boldsymbol{p}_k)^T H(\boldsymbol{x}_k)\lambda_k \boldsymbol{p}_k$$

นั่นคือ ณ จุดต่ำสุดจะได้ว่า

$$\frac{\partial f(\boldsymbol{x}_k + \lambda_k \boldsymbol{p}_k)}{\partial \lambda_k} = \nabla^T f(\boldsymbol{x}_k)\boldsymbol{p}_k + \lambda_k(\boldsymbol{p}_k)^T H(\boldsymbol{x}_k)\boldsymbol{p}_k = 0$$

$$\therefore \qquad \lambda_k = -\frac{\nabla^T f(\boldsymbol{x}_k)\boldsymbol{p}_k}{(\boldsymbol{p}_k)^T H(\boldsymbol{x}_k)\boldsymbol{p}_k} \tag{3.4}$$

ตัวอย่างที่ 3.3 จงแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับโดยใช้ระเบียบวิธีทิศทางสังยุค

$$\text{Minimize} \qquad f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

กำหนดให้ $\boldsymbol{x}_0 = [0.0 \;\; 0.0]^T$ และ $\boldsymbol{p}_0 = [-1.0 \;\; 1.0]^T$

วิธีทำ คำนวณเกรเดียนต์และเมตริกซ์เฮสเซียน จะได้

$$\nabla f(\boldsymbol{x}_k) = \begin{bmatrix} 1+4x_1+2x_2 \\ -1+2x_1+2x_2 \end{bmatrix}; \quad \boldsymbol{H}(\boldsymbol{x}_k) = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}$$

<u>รอบการคำนวณที่ 1</u>:

จาก $\boldsymbol{p}_0 = [-1.0 \;\; 1.0]^T$ หา $\boldsymbol{p}_1$ ที่มีคุณสมบัติสังยุคซึ่งกันและกัน นั่นคือ

$$p_0^T \boldsymbol{H}\, p_1 = [-1.0 \quad 1.0]\begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} p_1^{(1)} \\ p_1^{(2)} \end{bmatrix} = 0$$

จะได้ว่า $\qquad -2p_1^{(1)} = 0 \qquad \Rightarrow p_1^{(1)} = 0 \qquad \Rightarrow p_1^{(2)} \in \Re$

เลือก $\quad p_1^{(2)} = 1.0$

นั่นคือ $\quad \boldsymbol{p}_1 = \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix}$

ดังนั้น $\quad \lambda_0 = -\dfrac{\nabla^T f(\boldsymbol{x}_0)\boldsymbol{p}_1}{(\boldsymbol{p}_1)^T \boldsymbol{H}(\boldsymbol{x}_0)\boldsymbol{p}_1} = -\dfrac{[1.0 \quad -1.0]\begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix}}{[0.0 \quad 1.0]\begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix}} = -\dfrac{-1.0}{2.0} = 0.5$

จะได้ $\quad \boldsymbol{x}_1 = \boldsymbol{x}_0 + \lambda_0 \boldsymbol{p}_1 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + (0.5)\begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.5 \end{bmatrix}$

คำนวณเกรเดียนต์ $\nabla f(\boldsymbol{x}_k) = \begin{bmatrix} 1+4x_1+2x_2 \\ -1+2x_1+2x_2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 0.0 \end{bmatrix}; \qquad \|\nabla f(\boldsymbol{x}_1)\| > 0.0001$

<u>รอบการคำนวณที่ 2</u>:

จาก $\boldsymbol{p}_1 = [0.0 \;\; 1.0]^T$ หา $\boldsymbol{p}_2$ ที่มีคุณสมบัติสังยุคซึ่งกันและกัน นั่นคือ

$$p_1^T H\, p_2 = \begin{bmatrix} 0.0 & 1.0 \end{bmatrix} \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} p_2^{(1)} \\ p_2^{(2)} \end{bmatrix} = 0$$

จะได้ว่า $2p_2^{(1)} + 2p_2^{(2)} = 0 \quad \Rightarrow p_2^{(1)} = 1.0 \quad \Rightarrow p_2^{(2)} = -1.0$

นั่นคือ $p_2 = \begin{bmatrix} 1.0 \\ -1.0 \end{bmatrix}$

ดังนั้น $\lambda_1 = -\dfrac{\nabla^T f(x_1)\, p_2}{(p_2)^T H(x_1)\, p_2} = -\dfrac{\begin{bmatrix} 2.0 & 0.0 \end{bmatrix}\begin{bmatrix} 1.0 \\ -1.0 \end{bmatrix}}{\begin{bmatrix} 1.0 & -1.0 \end{bmatrix}\begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} 1.0 \\ -1.0 \end{bmatrix}} = -\dfrac{2.0}{2.0} = -1.0$

จะได้ $x_2 = x_1 + \lambda_1 p_2 = \begin{bmatrix} 0.0 \\ 0.5 \end{bmatrix} + (-1.0)\begin{bmatrix} 1.0 \\ -1.0 \end{bmatrix} = \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}$

คำนวณเกรเดียนต์ $\nabla f(x_k) = \begin{bmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}; \qquad \left\| \nabla f(x_1) \right\| < 0.0001$

สิ้นสุดการคำนวณวนรอบ

ได้ผลเฉลยเป็น $x^* = \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}$ ใช้การคำนวณเพียงสองรอบเท่านั้น

3.5 ระเบียบวิธีชันที่สุด (Steepest Descent Method หรือ Cauchy Method)

ระเบียบวิธีชันที่สุดนี้ เป็นระเบียบวิธีขั้นพื้นฐานของการแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับ อาศัยการคำนวณเกรเดียนต์ของฟังก์ชันวัตถุประสงค์ เป็นคุณสมบัติที่สำคัญมาก ถ้าเคลื่อนจุดคำตอบไปในทิศทางของค่าเกรเดียนต์จากจุดเริ่มต้นใด ๆ การเคลื่อนที่ดังกล่าวจะเกิดขึ้นในทิศทางที่ให้ค่าฟังก์ชันเพิ่มขึ้นมากที่สุด ในทางตรงกันข้าม การเคลื่อนตำแหน่งของผลเฉลยในทิศทางลบเกรเดียนต์ จะส่งผลให้การลดลงของค่าฟังก์ชันวัตถุประสงค์มากที่สุดเช่นกัน โดยอาศัยหลักการดังกล่าว อาจจะเลือกกำหนดทิศทางการค้นหาให้มีค่าเป็นทิศทางลบเกรเดียนต์ได้ นั่นคือ $p_k = -\nabla f(x_k)$ ระเบียบวิธีนี้ให้คุณสมบัติการลู่เข้าแบบเชิงเส้น (linear convergence) และได้ผลเฉลยเฉพาะถิ่น (local minimum) เท่านั้น

พิจารณาจากจุดคำตอบในรอบการค้นหาที่ k ใด ๆ สามารถเขียนกระจายอนุกรมเทย์เลอร์สำหรับประมาณค่าฟังก์ชันในรอบการค้นหาถัดไป โดยใช้พจน์อนุพันธ์อันดับหนึ่งเท่านั้น จะได้

$$f(x_{k+1}) = f(x_k + \lambda p_k) \approx f(x_k) + \lambda p_k^T \nabla f(x_k) \tag{3.5}$$

เพื่อให้การปรับปรุงจุดคำตอบมีคุณสมบัติลาดลง ดังนั้น

$$f(x_{k+1}) < f(x_k) + \lambda p_k^T \nabla f(x_k)$$

$$\therefore \qquad \lambda p_k^T \nabla f(x_k) < 0 \tag{3.6}$$

การเลือกทิศทางลบเกรเดียนต์เป็นทิศทางการค้นหาจะได้ว่า

$$p_1^T \boldsymbol{H}\, p_2 = \begin{bmatrix}0.0 & 1.0\end{bmatrix}\begin{bmatrix}4 & 2\\ 2 & 2\end{bmatrix}\begin{bmatrix}p_2^{(1)}\\ p_2^{(2)}\end{bmatrix} = 0$$

จะได้ว่า $2p_2^{(1)} + 2p_2^{(2)} = 0 \quad \Rightarrow p_2^{(1)} = 1.0 \quad \Rightarrow p_2^{(2)} = -1.0$

นั่นคือ $\boldsymbol{p}_2 = \begin{bmatrix}1.0\\ -1.0\end{bmatrix}$

ดังนั้น $$\lambda_1 = -\frac{\nabla^T f(\boldsymbol{x}_1)\boldsymbol{p}_2}{(\boldsymbol{p}_2)^T \boldsymbol{H}(\boldsymbol{x}_1)\boldsymbol{p}_2} = -\frac{\begin{bmatrix}2.0 & 0.0\end{bmatrix}\begin{bmatrix}1.0\\ -1.0\end{bmatrix}}{\begin{bmatrix}1.0 & -1.0\end{bmatrix}\begin{bmatrix}4 & 2\\ 2 & 2\end{bmatrix}\begin{bmatrix}1.0\\ -1.0\end{bmatrix}} = -\frac{2.0}{2.0} = -1.0$$

จะได้ $\boldsymbol{x}_2 = \boldsymbol{x}_1 + \lambda_1 \boldsymbol{p}_2 = \begin{bmatrix}0.0\\ 0.5\end{bmatrix} + (-1.0)\begin{bmatrix}1.0\\ -1.0\end{bmatrix} = \begin{bmatrix}-1.0\\ 1.5\end{bmatrix}$

คำนวณเกรเดียนต์ $\nabla f(\boldsymbol{x}_k) = \begin{bmatrix}1+4x_1+2x_2\\ -1+2x_1+2x_2\end{bmatrix} = \begin{bmatrix}0.0\\ 0.0\end{bmatrix}$; $\quad \|\nabla f(\boldsymbol{x}_1)\| < 0.0001$

สิ้นสุดการคำนวณวนรอบ

ได้ผลเฉลยเป็น $\boldsymbol{x}^* = \begin{bmatrix}-1.0\\ 1.5\end{bmatrix}$ ใช้การคำนวณเพียงสองรอบเท่านั้น

3.5 ระเบียบวิธีชันที่สุด (Steepest Descent Method หรือ Cauchy Method)

ระเบียบวิธีชันที่สุดนี้ เป็นระเบียบวิธีขั้นพื้นฐานของการแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับ อาศัยการคำนวณเกรเดียนต์ของฟังก์ชันวัตถุประสงค์ เป็นคุณสมบัติที่สำคัญมาก ถ้าเคลื่อนจุดคำตอบไปในทิศทางของค่าเกรเดียนต์จากจุดเริ่มต้นใด ๆ การเคลื่อนที่ดังกล่าวจะเกิดขึ้นในทิศทางที่ให้ค่าฟังก์ชันเพิ่มขึ้นมากที่สุด ในทางตรงกันข้าม การเคลื่อนตำแหน่งของผลเฉลยในทิศทางลบเกรเดียนต์ จะส่งผลให้การลดลงของค่าฟังก์ชันวัตถุประสงค์มากที่สุดเช่นกัน โดยอาศัยหลักการดังกล่าว อาจจะเลือกกำหนดทิศทางการค้นหาให้มีค่าเป็นทิศทางลบเกรเดียนต์ได้ นั่นคือ $\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k)$ ระเบียบวิธีนี้ให้คุณสมบัติการลู่เข้าแบบเชิงเส้น (linear convergence) และได้ผลเฉลยเฉพาะถิ่น (local minimum) เท่านั้น

พิจารณาจากจุดคำตอบในรอบการค้นหาที่ k ใด ๆ สามารถเขียนกระจายอนุกรมเทย์เลอร์สำหรับประมาณค่าฟังก์ชันในรอบการค้นหาถัดไป โดยใช้พจน์อนุพันธ์อันดับหนึ่งเท่านั้น จะได้

$$f(\boldsymbol{x}_{k+1}) = f(\boldsymbol{x}_k + \lambda \boldsymbol{p}_k) \approx f(\boldsymbol{x}_k) + \lambda \boldsymbol{p}_k^T \nabla f(\boldsymbol{x}_k) \tag{3.5}$$

เพื่อให้การปรับปรุงจุดคำตอบมีคุณสมบัติลาดลง ดังนั้น

$$f(\boldsymbol{x}_{k+1}) < f(\boldsymbol{x}_k) + \lambda \boldsymbol{p}_k^T \nabla f(\boldsymbol{x}_k)$$

$$\therefore \quad \lambda \boldsymbol{p}_k^T \nabla f(\boldsymbol{x}_k) < 0 \tag{3.6}$$

การเลือกทิศทางลบเกรเดียนต์เป็นทิศทางการค้นหาจะได้ว่า

$$p_k = -\nabla f(x_k) \quad \Rightarrow \quad \lambda_k\left[-\nabla f(x_k)\right]^T \nabla f(x_k) = -\lambda_k \left\|\nabla f(x_k)\right\|^2 < 0 \qquad (3.7)$$

จากความสัมพันธ์ดังกล่าว จะเห็นได้ว่า ระเบียบวิธีชันที่สุด รับประกันคุณสมบัติการลู่เข้าในทุกๆ รอบการคำนวณ มีรายละเอียดอัลกอริทึมการคำนวณดังต่อไปนี้

ขั้นที่ 1: กำหนดค่าเริ่มต้น $k = 0$, เลือกจุดเริ่มต้นของการค้นหา x_k

ถ้า $\left\|\nabla f(x_k)\right\| < \varepsilon$ สิ้นสุดการคำนวณ ข้ามไปขั้นที่ 6

ขั้นที่ 2: คำนวณทิศทางการค้นหา $p_k = -\nabla f(x_k)$

ขั้นที่ 3: แก้ปัญหาย่อยการค้นหาตามเส้น (line search sub-problem)

$$\text{Minimize} \qquad F(\lambda) \equiv f(x_k + \lambda_k p_k)$$
$$\lambda_k > 0$$

ขั้นที่ 4: ปรับปรุงจุดคำตอบ $x_{k+1} = x_k + \lambda_k p_k$

ขั้นที่ 5: ถ้า $\left\|\nabla f(x_{k+1})\right\| < \varepsilon$ สิ้นสุดการคำนวณ ข้ามไปขั้นตอนที่ 6

ถ้าไม่ใช่ ให้เพิ่มตัวนับ $k = k+1$ ทำซ้ำขั้นที่ 2

ขั้นที่ 6: ได้ผลเฉลย $\qquad x^* = x_{k+1}$

ตัวอย่างที่ 3.4 จงแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับโดยใช้ระเบียบวิธีชันที่สุด

$$\text{Minimize} \qquad f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

กำหนดให้ $x_0 = [0.0 \;\; 0.0]^T$ ใช้ความคลาดเคลื่อนสูงสุดที่ยอมรับได้ $\varepsilon = 0.0001$

วิธีทำ คำนวณเกรเดียนต์ของฟังก์ชันวัตถุประสงค์ จะได้

$$\nabla f(x_k) = \begin{bmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{bmatrix};$$

จะได้ว่า $\nabla f(x_0) = \begin{bmatrix} 1.0 \\ -1.0 \end{bmatrix} \Rightarrow \left\|\nabla f(x_0)\right\| = \sqrt{2} < 0.0001$

รอบการคำนวณที่ 1:

จาก $p_0 = -\nabla f(x_0) = \begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix}$ จะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

$$\text{Minimize} \qquad F(\lambda) = f\left(\begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + \lambda \begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix}\right) = f\left(\begin{bmatrix} -\lambda \\ \lambda \end{bmatrix}\right) = \lambda^2 - 2\lambda$$
$$\lambda > 0$$

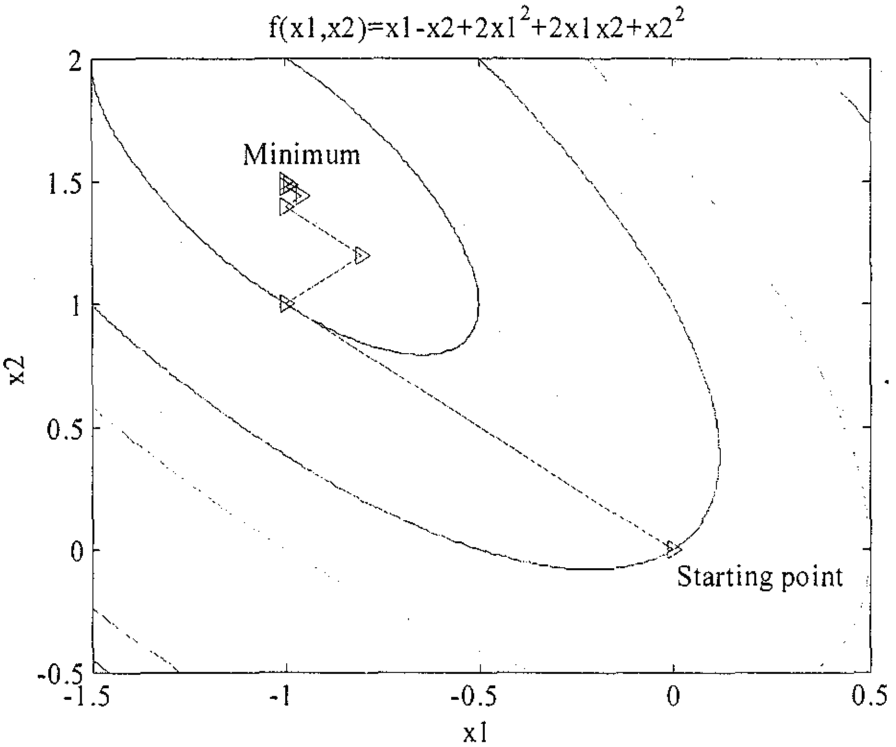
แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหาคิวบิก จะได้ (เพื่อให้คุ้นเคยกับการแก้ปัญหาย่อยการค้นหาตามเส้น ถึงแม้สมการจะอยู่ในรูปฟังก์ชันกำลังสองซึ่งมีผลเฉลยที่แม่นตรง การแก้ปัญหาจะใช้การคำนวณวนรอบ)

โดยใช้ระเบียบวิธีการค้นหาคิวบิก ใช้การคำนวณทั้งสิ้น 2 รอบ

k	λ_L	λ_U	λ^*	$F(\lambda^*)$
0.0000	0.0000	3.0000	-	-
1.0000	1.0000	3.0000	1.0000	-1.0000
2.0000	1.0000	1.0000	1.0000	-1.0000

ได้ $\lambda_0 = 1.0$ จะได้ $x_1 = x_0 + \lambda_0 p_1 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + (1.0)\begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix} = \begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix}$

จะได้ว่า $\nabla f(x_1) = \begin{bmatrix} -1.0 \\ -1.0 \end{bmatrix} \Rightarrow \|\nabla f(x_1)\| = \sqrt{2} < 0.0001$



รูปที่ 3.5 กราฟการลู่เข้าของระเบียบวิธีชันที่สุดสำหรับตัวอย่างที่ 3.4

ได้ผลการคำนวณวนรอบดังต่อไปนี้

k	x_1	x_2	$f(x_k)$	$\nabla f(x_k)$
0.0000	0.0000	0.0000	0.0000	1.4142
1.0000	-1.0000	1.0000	-1.0000	1.4142
2.0000	-0.8000	1.2000	-1.2000	0.2828
3.0000	-1.0000	1.4000	-1.2400	0.2828
4.0000	-0.9600	1.4400	-1.2480	0.0566
5.0000	-1.0000	1.4800	-1.2496	0.0566
6.0000	-0.9920	1.4880	-1.2499	0.0113
7.0000	-1.0000	1.4960	-1.2500	0.0113
8.0000	-0.9984	1.4976	-1.2500	0.0023
9.0000	-1.0000	1.4992	-1.2500	0.0023
10.000	-0.9997	1.4995	-1.2500	0.0005
11.000	-1.0000	1.4998	-1.2500	0.0005
12.000	-0.9999	1.4999	-1.2500	0.0001

ตัวอย่างที่ 3.5 จงแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับโดยใช้ระเบียบวิธีชันที่สุด

Minimize $\quad f(x_1, x_2) = x_1^2 + x_2^2$

กำหนดให้ $\boldsymbol{x_0} = [1.0 \;\; 1.0]^T$ ใช้ความคลาดเคลื่อนสูงสุดที่ยอมรับได้ $\varepsilon = 0.0001$

วิธีทำ คำนวณเกรเดียนต์ของฟังก์ชันวัตถุประสงค์ จะได้

$$\nabla f(\boldsymbol{x}_k) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix};$$

จะได้ว่า $\nabla f(\boldsymbol{x}_0) = \begin{bmatrix} 2.0 \\ 2.0 \end{bmatrix} \Rightarrow \|\nabla f(\boldsymbol{x}_0)\| = 2\sqrt{2} < 0.0001$

<u>รอบการคำนวณที่ 1</u>:

จาก $\boldsymbol{p}_0 = -\nabla f(\boldsymbol{x}_0) = \begin{bmatrix} -2.0 \\ -2.0 \end{bmatrix}$ จะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

Minimize $\quad F(\lambda) = f\left(\begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix} + \lambda \begin{bmatrix} -2.0 \\ -2.0 \end{bmatrix}\right) = f\left(\begin{bmatrix} 1-2\lambda \\ 1-2\lambda \end{bmatrix}\right) = 2(1-2\lambda)^2$

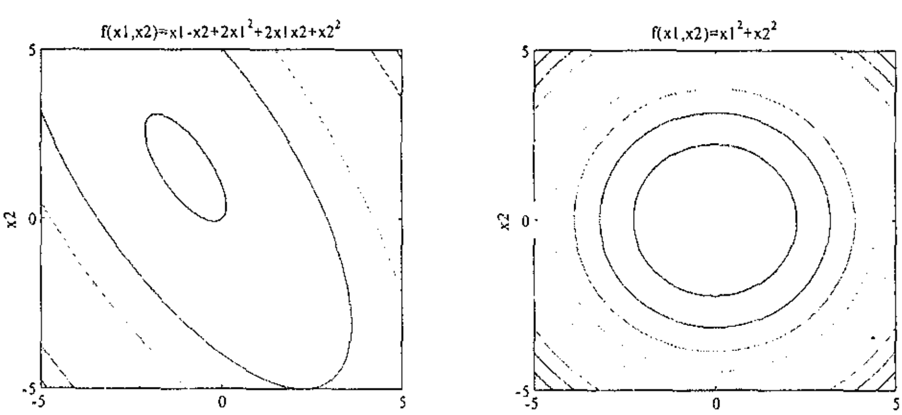
$\lambda > 0$

แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหาคิวบิก จะได้ $\lambda = ½ = 0.5$

ได้ $\quad \lambda_0 = 0.5$ จะได้ $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \lambda_0 \boldsymbol{p}_1 = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix} + (0.5)\begin{bmatrix} -2.0 \\ -2.0 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$

ตรวจสอบ $\nabla f(\boldsymbol{x}_1) = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} \Rightarrow \|\nabla f(\boldsymbol{x}_1)\| = 0 < 0.0001$ สิ้นสุดการคำนวณ

จะเห็นได้ว่า ตัวอย่างนี้ ถึงแม้ฟังก์ชันวัตถุประสงค์จะมีรูปแบบเป็นฟังก์ชันกำลังสองเช่นเดียวกันกับตัวอย่างที่ผ่านมา แต่ความสวยงามของฟังก์ชันมีความแตกต่างกัน ตัวอย่างนี้ค้นพบจุดคำตอบโดยใช้จำนวนรอบการค้นหาเพียง 1 รอบเท่านั้น เนื่องจากมีความสมมาตรของคอนทัวร์ ดังรูปที่ 3



รูปที่ 3.6 ลักษณะเส้นโครงร่างของฟังก์ชันวัตถุประสงค์

ระเบียบวิธีชันที่สุดนี้ไวต่อการเปลี่ยนแปลงค่าของตัวแปร ถ้าฟังก์ชันที่พิจารณามีความลาดชันของตัวแปรแต่ละตัวไม่เท่ากันดังตัวอย่างที่ 3.4 กระบวนการค้นหาจะลู่เข้าช้า ถ้าเส้นโครงร่างมี

ความสมมาตรหรือความลดชันของตัวแปรแต่ละตัวไม่ต่างกัน ดังรูปที่ 3.6 จะส่งผลให้ทิศทางลบเกรเดียนต์พาดผ่านจุดต่ำสุดพอดี และการค้นหาใช้เพียงรอบเดียวเท่านั้น อย่างไรก็ตาม ปัญหาในทางวิศวกรรมโดยทั่วไปไม่อาจจะคาดเดาได้ว่ารูปแบบของฟังก์ชันมีลักษณะเส้นโครงร่างเป็นอย่างไร ส่งผลให้ระเบียบวิธีชันที่สุดนี้เมื่อนำมาใช้แล้วมีผลการลู่เข้าที่ค่อนข้างช้า

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 3.4 และ 3.5</u>

// SCILAB source programs

```
// กำหนดฟังก์ชันระเบียบวิธีชันที่สุด                    steepest.sci
========================
        steepest.sci
========================
function [xmin,fmin,k]=steepest(fobj,fgrad,x0,ftol,vtol,opt)
f0=fobj(x0);
g0=fgrad(x0);
ferr=norm(g0);
k=0;
if opt==1 then
   disp([k x0' f0 ferr]);
end
while (ferr>ftol)
   k=k+1;
   p=-g0;
   lambda=lsearch(fobj,x0,p);
   xmin=x0+lambda*p;
   fmin=fobj(xmin);
   gmin=fgrad(xmin);
   ferr=norm(gmin);
   if opt==1 then
      disp([k xmin' fmin ferr]);
   end
   x0=xmin;f0=fmin;g0=gmin;
end
endfunction
```

```
// กำหนดเกรเดียนต์ของฟังก์ชันวัตถุประสงค์ที่กำหนดโดย f_unc01.sci    g_unc01.sci
========================
        g_unc01.sci
========================
function g = g_unc01(x)
x1=x(1);
x2=x(2);
g=[1+4*x1+2*x2;-1+2*x1+2*x2];
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('f_unc01.sci');              ประกาศฟังก์ชันวัตถุประสงค์
--> exec('g_unc01.sci');              ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('lsearch.sci');              ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('flsearch.sci');             ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('fgradnum1.sci');            ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('cubicsearch1.sci');         ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('steepest.sci');             ประกาศฟังก์ชันหลักของการคำนวณ
--> [xmin,fmin,k]=steepest(f_unc01,g_unc01,[0;0],1e-4,1e-4,1);
// Iter     X1          X2          F(x)         Ferr
   0.    0.0000000   0.0000000   0.0000000    1.4142136
   1.  - 1.0000000   1.0000000 - 1.0000000    1.4142136
   2.  - 0.8000000   1.2000000 - 1.2000000    0.2828427
   3.  - 1.0000000   1.4000000 - 1.2400000    0.2828427
   4.  - 0.9600000   1.4400000 - 1.2480000    0.0565685
   5.  - 1.0000000   1.4800000 - 1.2496000    0.0565685
   6.  - 0.9920000   1.4880000 - 1.2499200    0.0113137
   7.  - 1.0000000   1.4960000 - 1.2499840    0.0113137
   8.  - 0.9984000   1.4976000 - 1.2499968    0.0022627
   9.  - 1.0000001   1.4992001 - 1.2499994    0.0022629
   10. - 0.9996801   1.4995201 - 1.2499999    0.0004524
   11. - 0.9999999   1.4998400 - 1.2500000    0.0004523
   12. - 0.9999361   1.4999038 - 1.2500000    0.0000904
```

ทดสอบโปรแกรมด้วยฟังก์ชันวัตถุประสงค์ที่กำหนดในตัวอย่างที่ 3.5 จะได้ว่า

```
// กำหนดฟังก์ชันวัตถุประสงค์                    f_unc02.sci
=========================
        f_unc02.sci
=========================
function f = f_unc02(x)
x1=x(1);
x2=x(2);
f=x1^2+x2^2;
endfunction
```

```
// กำหนดเกรเดียนต์ของฟังก์ชันวัตถุประสงค์ที่กำหนดโดย f_unc02.sci    g_unc02.sci
=========================
        g_unc02.sci
=========================
function g = g_unc02(x)
x1=x(1);
x2=x(2);
g=[2*x1;2*x2];
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('f_unc02.sci');              ประกาศฟังก์ชันวัตถุประสงค์
--> exec('g_unc02.sci');              ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('lsearch.sci');              ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('flsearch.sci');             ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('fgradnum1.sci');            ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('cubicsearch1.sci');         ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('steepest.sci');             ประกาศฟังก์ชันหลักของการคำนวณ
--> [xmin,fmin,k]=steepest(f_unc02,g_unc02,[1;1],1e-4,1e-4,1);
// Iter    X1          X2          F(x)        Ferr
    0.   1.0000000   1.0000000   2.0000000   2.8284271
    1.   2.220D-16   2.220D-16   9.861D-32   6.280D-16
       ≈ 0.00000     0.0000000   0.0000000   0.0000000
```

3.6 ระเบียบวิธีเกรเดียนต์สังยุค (Conjugate Gradient Method)

คุณสมบัติการลู่เข้าของระเบียบวิธีชันที่สุดในหัวข้อที่ผ่านมา สามารถปรับปรุงให้ดีขึ้นได้ โดยการพัฒนาทิศทางการค้นหาในรูปของทิศทางสังยุค เมื่อนำมาใช้กับฟังก์ชันกำลังสองแล้วจะพบว่าสามารถค้นหาคำตอบได้โดยใช้การคำนวณ n รอบเท่านั้น แต่การกำหนดทิศทางสังยุคโดยทั่วไปทำได้ยาก อย่างไรก็ตาม ในกรณีของฟังก์ชันกำลังสองการกำหนดทิศทางสังยุคดังกล่าวไม่ยุ่งยากมากนัก ระเบียบวิธีเกรเดียนต์สังยุคนี้ ใช้การสร้างทิศทางการค้นหาที่มีคุณสมบัติการเป็นทิศทางสังยุค โดยทิศทางที่ถูกสร้างขึ้นมีความสัมพันธ์กับเกรเดียนต์ของฟังก์ชัน ดังนี้

กำหนดให้ฟังก์ชันวัตถุประสงค์ใดๆ ประมาณได้ด้วยฟังก์ชันกำลังสอง

$$\text{Minimize} \qquad f(x) \approx Q(x) = c + \boldsymbol{b}^T \boldsymbol{x} + \tfrac{1}{2}\boldsymbol{x}^T A \boldsymbol{x} \tag{3.8}$$

ในรอบการคำนวณแรก ใช้ $\boldsymbol{p}_0 = -\nabla f(\boldsymbol{x}_0)$ เมื่อทำการปรับปรุงจุดคำตอบ การกำหนดทิศทางการค้นหาในรอบถัดไปให้ใช้ความสัมพันธ์ของผลรวมเชิงเส้นระหว่างทิศทางลบเกรเดียนต์และทิศทางการค้นหาในรอบที่ผ่านมา ดังสมการ

$$\boldsymbol{p}_1 = -\nabla f(\boldsymbol{x}_1) + \beta_1 \boldsymbol{p}_0 \tag{3.9}$$

หรือในการคำนวณรอบการค้นหาที่ $k+1$ ใดๆ

$$\boldsymbol{p}_{k+1} = -\nabla f(\boldsymbol{x}_{k+1}) + \beta_{k+1} \boldsymbol{p}_k \tag{3.10}$$

โดยที่

$$\beta_{k+1} = \frac{\nabla^T f(\boldsymbol{x}_{k+1}) \cdot \nabla f(\boldsymbol{x}_{k+1})}{\nabla^T f(\boldsymbol{x}_k) \cdot \nabla f(\boldsymbol{x}_k)} = \left\| \frac{\nabla f(\boldsymbol{x}_{k+1})}{\nabla f(\boldsymbol{x}_k)} \right\|^2 \tag{3.11}$$

จากความสัมพันธ์ดังกล่าว ระเบียบวิธีเกรเดียนต์สังยุคมีรายละเอียดอัลกอริทึมดังต่อไปนี้

<u>ขั้นที่ 1</u>: กำหนดค่าเริ่มต้น $k = 0$, เลือกจุดเริ่มต้นของการค้นหา $\boldsymbol{x}_k$ และ $\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k)$

ถ้า $\|\nabla f(\boldsymbol{x}_k)\| < \varepsilon$ สิ้นสุดการคำนวณ ข้ามไปขั้นที่ 6

<u>ขั้นที่ 2</u>: แก้ปัญหาย่อยการค้นหาตามเส้น (line search sub-problem)

$$\text{Minimize} \qquad F(\lambda) \equiv f(\boldsymbol{x}_k + \lambda_k \boldsymbol{p}_k)$$
$$\lambda_k > 0$$

<u>ขั้นที่ 3</u>: ปรับปรุงจุดคำตอบ $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \boldsymbol{p}_k$

<u>ขั้นที่ 4</u>: ถ้า $\|\nabla f(\boldsymbol{x}_{k+1})\| < \varepsilon$ สิ้นสุดการคำนวณ ข้ามไปขั้นตอนที่ 6

<u>ขั้นที่ 5</u>: คำนวณ $\beta_{k+1} = \dfrac{\nabla^T f(\boldsymbol{x}_{k+1}) \cdot \nabla f(\boldsymbol{x}_{k+1})}{\nabla^T f(\boldsymbol{x}_k) \cdot \nabla f(\boldsymbol{x}_k)} = \left\| \dfrac{\nabla f(\boldsymbol{x}_{k+1})}{\nabla f(\boldsymbol{x}_k)} \right\|^2$

$\boldsymbol{p}_{k+1} = -\nabla f(\boldsymbol{x}_{k+1}) + \beta_{k+1} \boldsymbol{p}_k$

เพิ่มตัวนับ $k = k+1$ ทำซ้ำขั้นตอนที่ 2

<u>ขั้นที่ 6</u>: ได้ผลเฉลย $\boldsymbol{x}^* = \boldsymbol{x}_{k+1}$

ตัวอย่างที่ 3.6 จงแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับโดยใช้ระเบียบวิธีเกรเดียนต์สังยุค

Minimize $\quad f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$

กำหนดให้ $x_0 = [0.0 \;\; 0.0]^T$ ใช้ความคลาดเคลื่อนสูงสุดที่ยอมรับได้ $\varepsilon = 0.0001$

วิธีทำ คำนวณเกรเดียนต์ของฟังก์ชันวัตถุประสงค์ จะได้

$$\nabla f(x_k) = \begin{bmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{bmatrix};$$

จะได้ว่า $\nabla f(x_0) = \begin{bmatrix} 1.0 \\ -1.0 \end{bmatrix} \Rightarrow \|\nabla f(x_0)\| = \sqrt{2} < 0.0001$

<u>รอบการคำนวณที่ 1</u>:

จาก $p_0 = -\nabla f(x_0) = \begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix}$ จะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

Minimize $\quad F(\lambda) = f\left(\begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + \lambda \begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix}\right) = f\left(\begin{bmatrix} -\lambda \\ \lambda \end{bmatrix}\right) = \lambda^2 - 2\lambda$

$\lambda > 0$

แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหาคิวบิก จะได้ $\lambda_0 = 1.0$

จะได้ $\quad x_1 = x_0 + \lambda_0 p_0 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + (1.0)\begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix} = \begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix}$

จะได้ว่า $\nabla f(x_1) = \begin{bmatrix} -1.0 \\ -1.0 \end{bmatrix} \Rightarrow \|\nabla f(x_1)\| = \sqrt{2} < 0.0001$

คำนวณ $\beta_1 = \left\|\dfrac{\nabla f(x_1)}{\nabla f(x_0)}\right\|^2 = \left[\dfrac{\sqrt{2}}{\sqrt{2}}\right]^2 = 1.0$

ได้ทิศทางการค้นหาเป็น $\quad p_1 = -\nabla f(x_1) + \beta_1 p_0 = -\begin{bmatrix} -1.0 \\ -1.0 \end{bmatrix} + (1.0)\begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 2.0 \end{bmatrix}$

<u>รอบการคำนวณที่ 2</u>:

จาก $p_1 = \begin{bmatrix} 0.0 \\ 2.0 \end{bmatrix}$ จะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

Minimize $\quad F(\lambda) = f\left(\begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix} + \lambda \begin{bmatrix} 0.0 \\ 2.0 \end{bmatrix}\right) = f\left(\begin{bmatrix} -1 \\ 1 + 2\lambda \end{bmatrix}\right) = 4\lambda^2 - 2\lambda - 1$

แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหาคิวบิก จะได้ $\lambda_2 = ¼ = 0.25$

จะได้ $\quad x_2 = x_1 + \lambda_1 p_1 = \begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix} + (0.25)\begin{bmatrix} 0.0 \\ 2.0 \end{bmatrix} = \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}$

จะได้ว่า $\nabla f(x_2) = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} \Rightarrow \|\nabla f(x_2)\| = 0 < 0.0001$ สิ้นสุดการคำนวณ

จะได้ผลเฉลยเป็น $\quad x^* = \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}$

ใช้การคำนวณทั้งสิ้น 2 รอบ เท่ากับจำนวนมิติของปัญหา

โปรแกรม SCILAB สำหรับตัวอย่างที่ 3.6

// SCILAB source programs

```
// กำหนดฟังก์ชันระเบียบวิธีเกรเดียนต์สังยุค                        conjgrad.sci
========================
        conjgrad.sci
========================
function [x,f,k]=conjgrad(fobj,fgrad,x0,ftol,vtol,opt)
k=0;
g=fgrad(x0);
s_dir0=-g;
s_dir=s_dir0;
lambda=lsearch(fobj,x0,s_dir);
x=x0-lambda*s_dir;
f=fobj(x);
if opt==1 then
   disp([k x' f norm(g,%inf)]);
end
k=k+1;
x0=x;
x_init=x;
s_dir0=s_dir;
g0=g;
while norm(g,%inf)>ftol
  beta1=g'*g/(g0'*g0);
  s_dir=-g+beta1*s_dir0;
  lambda=lsearch(fobj,x0,s_dir);
  x=x0+lambda*s_dir;
  f=fobj(x);
  g=fgrad(x);
  if opt==1 then
      disp([k x' f norm(g,%inf)]);
  end
  k=k+1;
  x0=x;
```

```
 s_sdir0=s_dir;
 g0=g;
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('f_unc01.sci');            ประกาศฟังก์ชันวัตถุประสงค์
--> exec('g_unc01.sci');            ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('lsearch.sci');            ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('flsearch.sci');           ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('fgradnum1.sci');          ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('cubicsearch1.sci');       ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('conjgrad.sci');           ประกาศฟังก์ชันหลักของการคำนวณ
--> [xmin,fmin,k]=conjgrad(f_unc01,g_unc01,[0;0],1e-4,1e-4,1);
// Iter      X1            X2          F(x)        ||g(x)||
   1.   1.00000000  - 1.0000000   3.000000   1.000000
   2. - 1.00000000    1.0000000 - 1.000000   1.000000
   3. - 1.00000000    1.5000000 - 1.250000   8.017D-12
```

3.7 ระเบียบวิธีนิวตัน (Newton Method)

ระเบียบวิธีนิวตันใช้การวิเคราะห์จากการกระจายอนุกรมเทย์เลอร์เป็นหลัก โดยอาศัยการประมาณค่าถึงพจน์ของอนุพันธ์อันดับที่สองเท่านั้น ดังสมการต่อไปนี้

$$f(x_k+\lambda_k p_k)\approx f(x_k)+\nabla f(x_k)^T p_k+\frac{1}{2}p_k^T\nabla^2 f(x_k)\,p_k \qquad (3.12)$$

คำนวณอนุพันธ์ย่อยของสมการดังกล่าว จะได้ความสัมพันธ์ ดังนี้

$$\nabla f(x_k+p_k)=\nabla f(x_k)+\nabla^2 f(x_k)\,p_k \qquad (3.13)$$

โดยวิธีการนี้คาดหวังว่าจุดคำตอบต่อไปที่ค้นพบคือจุดที่สอดคล้องกับเงื่อนไข

$$\nabla f(x_{k+1})=0 \qquad (3.14)$$

ดังนั้นจะได้สมการดังนี้

$$\nabla f(x_k)+\nabla^2 f(x_k)\,p_k=0 \qquad (3.15)$$

$$p_k=-\left[\nabla^2 f(x_k)\right]^{-1}\nabla f(x_k) \qquad (3.16)$$

เดยปกติ ระเบียบวิธีนี้ใช้การกำหนดค่า $\lambda = 1.0$ และมีอัตราการลู่เข้าที่รวดเร็วมากเป็นการลู่เข้าแบบกำลังสอง (quadratic convergence) แต่มีข้อเสีย คือ ไม่สามารถรับประกันการลู่เข้า

ของผลเฉลยขึ้นอยู่กับการกำหนดจุดเริ่มต้นและคุณสมบัติบางประการของฟังก์ชันวัตถุประสงค์ ได้อัลกอริทึมการคำนวณดังนี้

ขั้นที่ 1: กำหนดค่าเริ่มต้น $k = 0$, เลือกจุดเริ่มต้นของการค้นหา $\boldsymbol{x}_k$
ถ้า $\|\nabla f(\boldsymbol{x}_k)\| < \varepsilon$ สิ้นสุดการคำนวณ ข้ามไปขั้นที่ 6

ขั้นที่ 2: คำนวณทิศทางการค้นหา $\boldsymbol{p}_k = -\left[\nabla^2 f(\boldsymbol{x}_k)\right]^{-1} \nabla f(\boldsymbol{x}_k)$

ขั้นที่ 3: ปรับปรุงจุดคำตอบ $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{p}_k$

ขั้นที่ 4: ถ้า $\|\nabla f(\boldsymbol{x}_{k+1})\| < \varepsilon$ สิ้นสุดการคำนวณ ข้ามไปขั้นตอนที่ 6

ขั้นที่ 5: เพิ่มตัวนับ $k = k+1$ ทำซ้ำขั้นตอนที่ 2

ขั้นที่ 6: ได้ผลเฉลย $\boldsymbol{x}^* = \boldsymbol{x}_{k+1}$

ตัวอย่างที่ 3.7 จงแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับโดยใช้ระเบียบวิธีนิวตัน

$$\text{Minimize} \qquad f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

กำหนดให้ $\boldsymbol{x}_0 = [0.0 \;\; 0.0]^T$ ใช้ความคลาดเคลื่อนสูงสุดที่ยอมรับได้ $\varepsilon = 0.0001$

วิธีทำ คำนวณเกรเดียนต์และเฮสเซียนของฟังก์ชันวัตถุประสงค์จะได้

$$\nabla f(\boldsymbol{x}_k) = \begin{bmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{bmatrix}; \quad \boldsymbol{H}(\boldsymbol{x}_k) = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}$$

รอบการคำนวณที่ 1:

$$\nabla f(\boldsymbol{x}_0) = \begin{bmatrix} 1.0 \\ -1.0 \end{bmatrix}; \qquad \boldsymbol{H}(\boldsymbol{x}_0) = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}$$

จะได้

$$\boldsymbol{p}_0 = -\left[\nabla^2 f(\boldsymbol{x}_0)\right]^{-1} \nabla f(\boldsymbol{x}_0) = -\boldsymbol{H}^{-1}(\boldsymbol{x}_0)\nabla f(\boldsymbol{x}_0)$$

$$= -\begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 1.0 \\ -1.0 \end{bmatrix} = \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}$$

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{p}_0 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix} = \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}$$

จะได้ว่า $\nabla f(\boldsymbol{x}_1) = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} \Rightarrow \|\nabla f(\boldsymbol{x}_1)\| = 0 < 0.0001$

จะได้ผลเฉลยเป็น $\boldsymbol{x}^* = \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}$

ใช้การคำนวณทั้งสิ้น 1 รอบ เท่านั้น

3.8 ระเบียบวิธีมาร์ควอดต์ (Marquardt Method) และระเบียบวิธีนิวตันที่ใช้การค้นหาตามเส้น (Line-search Newton Method)

ระเบียบวิธีนิวตันมีปัญหาเรื่องการรับประกันการลู่เข้า โดยจะลู่เข้าสู่เข้าตอบเมื่อทุกรอบการคำนวณให้เมตริกซ์เฮสเซียนเป็น *pdf* ดังนั้น เพื่อรับประกันการลู่เข้า มาร์ควอดต์ได้เสนอการปรับปรุงเมตริกซ์เฮสเซียนโดยเพิ่มพจน์ในแนวทแยงมุมดังนี้

$$\tilde{\boldsymbol{H}}_k = \boldsymbol{H}_k + \alpha_k \boldsymbol{I} \tag{3.17}$$

เมื่อ $\boldsymbol{I}$ คือ เมตริกซ์เอกลักษณ์ (identity matrix)

ด้วยหลักการนี้ ในขณะเริ่มต้นการคำนวณ พารามิเตอร์ α จะถูกกำหนดให้มีค่าสูง ประมาณ 10^4 หรือมีค่ามากพอที่จะทำให้

$$\left[\tilde{\boldsymbol{H}}_k\right]^{-1} = \left[\boldsymbol{H}_k + \alpha_k \boldsymbol{I}\right]^{-1} \approx \frac{1}{\alpha_k}\boldsymbol{I} \tag{3.18}$$

จะได้ทิศทางการค้นหาเป็น

$$\boldsymbol{p}_k = -\left[\tilde{\boldsymbol{H}}_k\right]^{-1} \nabla f\left(\boldsymbol{x}_k\right) \tag{3.19}$$

ที่ค่า α นี้ จะทำให้ในช่วงต้นของการคำนวณระเบียบวิธีนี้จะมีคุณสมบัติของระเบียบวิธีชันที่สุด โดยหลักการ เมื่อจำนวนรอบการคำนวณเพิ่มมากขึ้น α จะมีค่าลดลงเข้าสู่ศูนย์ นั่นคือ เมื่อผลเฉลยที่ถูกปรับปรุงอยู่ใกล้จุดคำตอบมากพอ ระเบียบวิธีมาร์ควอดต์นี้จะมีคุณสมบัติเป็นระเบียบวิธีนิวตันนั่นเอง

นอกจากนี้ การแก้ปัญหาการลู่เข้าของระเบียบวิธีนิวตันอาจจะทำได้โดยใช้การแก้ปัญหาย่อยการค้นหาตามเส้น นั่นคือ ให้ $\lambda > 0.0$ และ $\lambda \neq 1.0$ เรียกว่า ระเบียบวิธีนิวตันที่ใช้การค้นหาตามเส้น ดังนี้

$$\boldsymbol{p}_k = -\left[\nabla^2 f\left(\boldsymbol{x}_k\right)\right]^{-1} \nabla f\left(\boldsymbol{x}_k\right) \tag{3.20}$$

ในแต่ละรอบการคำนวณให้แก้ปัญหาย่อยการค้นหาตามเส้นดังต่อไปนี้

$$\begin{aligned} &\text{Minimize} \qquad F\left(\lambda_k\right) \equiv f\left(\boldsymbol{x}_k + \lambda_k \boldsymbol{p}_k\right) \\ &\quad \lambda_k > 0 \end{aligned} \tag{3.21}$$

3.9 ระเบียบวิธีคล้ายนิวตัน (Quasi-Newton Methods)

การแก้ปัญหาการลู่เข้าของระเบียบวิธีนิวตันที่ได้รับความนิยมมากที่สุด และถูกนำมาใช้ในการแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับ ได้แก่ ระเบียบวิธีคล้ายนิวตัน เนื่องจากคุณสมบัติการลู่เข้าที่รวดเร็วแบบเชิงเส้นยิ่งยวด (super linear convergence) และการรับประกันการลู่เข้าของจุดคำตอบ นอกจากนี้ระเบียบวิธีเหล่านี้มีข้อได้เปรียบคือ ไม่ต้องอาศัยการคำนวณเมตริกซ์เฮสเซียนของฟังก์ชันวัตถุประสงค์ ต้องการเฉพาะเกรเดียนต์เท่านั้น โดยกำหนดทิศทางการค้นหาดังนี้

$$\boldsymbol{p}_k = -\left[\boldsymbol{B}_k\right]^{-1} \nabla f\left(\boldsymbol{x}_k\right) \tag{3.22}$$

กรณีของฟังก์ชันตัวแปรเดี่ยวการประมาณค่าเฮสเซียนหรืออนุพันธ์อันดับสองนี้ ใช้เทคนิคเชิงตัวเลขดังต่อไปนี้

$$f''\left(x_{k+1}\right) \approx \frac{f'\left(x_{k+1}\right)-f'\left(x_k\right)}{x_{k+1}-x_k} \tag{3.23}$$

นิยมเรียกระเบียบวิธีคล้ายนิวตันสำหรับตัวแปรหนึ่งมิติว่า ระเบียบวิธีซีแคนต์ (secant method) อย่างไรก็ตาม ในกรณีของตัวแปรหลายมิติ รูปแบบการประมาณค่าจะอยู่ในรูปความสัมพันธ์ดังต่อไปนี้

$$\nabla^2 f\left(\boldsymbol{x}_{k+1}\right)\left(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right) \approx \nabla f\left(\boldsymbol{x}_{k+1}\right)-\nabla f\left(\boldsymbol{x}_k\right) \tag{3.24}$$

ให้ $\boldsymbol{B}_{k+1}$ แทนการประมาณค่าเฮสเซียนดังกล่าว จะได้ว่า

$$\boldsymbol{B}_{k+1}\left(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\right)=\nabla f\left(\boldsymbol{x}_{k+1}\right)-\nabla f\left(\boldsymbol{x}_k\right) \tag{3.25}$$

ถ้ากำหนดให้ $\boldsymbol{s}_k=\boldsymbol{x}_{k+1}-\boldsymbol{x}_k, \boldsymbol{y}_k=\nabla f\left(\boldsymbol{x}_{k+1}\right)-\nabla f\left(\boldsymbol{x}_k\right)$ จะได้ว่า

$$\boldsymbol{B}_{k+1}\boldsymbol{s}_k=\boldsymbol{y}_k \tag{3.26}$$

เรียกสมการนี้ว่า สมการซีแคนต์ (secant equation)

การประมาณค่านี้อาศัยหลักการเปลี่ยนแปลงเมตริกซ์เฮสเซียนให้มีคุณสมบัติสมมาตรและเป็น *pdf* การเริ่มต้นกระบวนการ อาจจะกำหนดให้ $\boldsymbol{B}_0$ เป็นเมตริกซ์เอกลักษณ์ได้ โดยใช้หลักการ

$$\boldsymbol{B}_{k+1}=\boldsymbol{B}_k+\Delta\boldsymbol{B}_k \tag{3.27}$$

ตัวอย่างหนึ่งของการประมาณค่า $\boldsymbol{B}_{k+1}$ ให้ได้เมตริกซ์สมมาตร แต่ไม่ใช่เมตริกซ์ *pdf* เรียกว่า สูตรการปรับปรุงสมมาตรลำดับขั้นที่หนึ่ง (symmetric rank-one update formula) ดังนี้

$$\boldsymbol{B}_{k+1}=\boldsymbol{B}_k+\frac{\left(\boldsymbol{y}_k-\boldsymbol{B}_k\boldsymbol{s}_k\right)\left(\boldsymbol{y}_k-\boldsymbol{B}_k\boldsymbol{s}_k\right)^T}{\left(\boldsymbol{y}_k-\boldsymbol{B}_k\boldsymbol{s}_k\right)^T\boldsymbol{s}_k} \tag{3.28}$$

ตัวอย่างที่ 3.8 จงแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับโดยใช้ระเบียบวิธีคล้ายนิวตัน ด้วยสูตรการปรับปรุงสมมาตรตำแหน่งที่หนึ่ง

Minimize $\quad f\left(x_1,x_2\right)=x_1-x_2+2x_1^2+2x_1x_2+x_2^2$

กำหนดให้ $\boldsymbol{x}_0=[0.0\ \ 0.0]^T$ ใช้ความคลาดเคลื่อนสูงสุดที่ยอมรับได้ $\varepsilon=0.0001$

วิธีทำ คำนวณเกรเดียนต์และการประมาณเฮสเซียนของฟังก์ชันวัตถุประสงค์จะได้

$$\nabla f\left(\boldsymbol{x}_k\right)=\begin{bmatrix}1+4x_1+2x_2\\-1+2x_1+2x_2\end{bmatrix};\quad \boldsymbol{B}_0=\begin{bmatrix}4&0\\0&4\end{bmatrix}$$

<u>รอบการคำนวณที่ 1</u>:

จะได้ทิศทางการค้นหา $\boldsymbol{p}_0=-\left[\boldsymbol{B}_0\right]^{-1}\nabla f\left(\boldsymbol{x}_0\right)=-\begin{bmatrix}4&0\\0&4\end{bmatrix}^{-1}\begin{bmatrix}1\\-1\end{bmatrix}=\begin{bmatrix}-0.25\\0.25\end{bmatrix}$

จะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

Minimize $\quad F(\lambda)=f\left(\begin{bmatrix}0.0\\0.0\end{bmatrix}+\lambda\begin{bmatrix}-0.25\\0.25\end{bmatrix}\right)=f\left(\begin{bmatrix}-0.25\lambda\\0.25\lambda\end{bmatrix}\right)=\frac{1}{16}\lambda^2-\frac{1}{2}\lambda$

$\lambda>0$

แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหาคิวบิก จะได้ $\lambda_0 = 4.0$

จะได้ $\quad x_1 = x_0+\lambda_0 p_0=\begin{bmatrix}0.0\\0.0\end{bmatrix}+(4.0)\begin{bmatrix}-0.25\\0.25\end{bmatrix}=\begin{bmatrix}-1.0\\1.0\end{bmatrix}$

$$\nabla f(x_1)=\begin{bmatrix}1+4(-1)+2(1)\\-1+2(-1)+2(1)\end{bmatrix}=\begin{bmatrix}-1\\-1\end{bmatrix}$$

คำนวณพารามิเตอร์เพิ่มเติม จะได้

$$s_1=x_1-x_0=\begin{bmatrix}-1\\1\end{bmatrix}-\begin{bmatrix}0\\0\end{bmatrix}=\begin{bmatrix}-1\\1\end{bmatrix}$$

$$y_1=\nabla f(x_1)-\nabla f(x_0)=\begin{bmatrix}-1\\1\end{bmatrix}-\begin{bmatrix}1\\-1\end{bmatrix}=\begin{bmatrix}-2\\2\end{bmatrix}$$

$$B_1=B_0+\frac{(y_0-B_0s_0)(y_0-B_0s_0)^T}{(y_0-B_0s_0)^T s_0}$$

$$=\begin{bmatrix}1&0\\0&1\end{bmatrix}+\frac{\left(\begin{bmatrix}-2\\2\end{bmatrix}-\begin{bmatrix}4&0\\0&4\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix}\right)^T\left(\begin{bmatrix}-2\\2\end{bmatrix}-\begin{bmatrix}4&0\\0&4\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix}\right)}{\left(\begin{bmatrix}-2\\2\end{bmatrix}-\begin{bmatrix}4&0\\0&4\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix}\right)^T\begin{bmatrix}-1\\1\end{bmatrix}}=\begin{bmatrix}10/3&4/3\\4/3&4/3\end{bmatrix}$$

<u>รอบการคำนวณที่ 2</u>:

จะได้ทิศทางการค้นหา $p_1=-[B_1]^{-1}\nabla f(x_1)=-\begin{bmatrix}10/3&4/3\\4/3&4/3\end{bmatrix}^{-1}\begin{bmatrix}-1\\-1\end{bmatrix}=\begin{bmatrix}0.00\\0.75\end{bmatrix}$

จะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

Minimize $\quad F(\lambda)=f\left(\begin{bmatrix}-1\\1\end{bmatrix}+\lambda\begin{bmatrix}0\\0.75\end{bmatrix}\right)=f\left(\begin{bmatrix}-1\\1+0.75\lambda\end{bmatrix}\right)=\frac{9}{16}\lambda^2-\frac{3}{4}\lambda-1$

$\lambda>0$

แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหาคิวบิก จะได้ $\lambda_1=\frac{2}{3}$

จะได้ $\quad x_2=x_1+\lambda_1 p_1=\begin{bmatrix}-1\\1\end{bmatrix}+\left(\frac{2}{3}\right)\begin{bmatrix}0\\3/4\end{bmatrix}=\begin{bmatrix}-1.0\\1.5\end{bmatrix}$

$$\nabla f(x_1)=\begin{bmatrix}1+4(-1)+2(1.5)\\-1+2(-1)+2(1.5)\end{bmatrix}=\begin{bmatrix}0\\0\end{bmatrix};$$

จะได้ว่า $\nabla f(x_2)=\begin{bmatrix}0.0\\0.0\end{bmatrix}\Rightarrow\|\nabla f(x_2)\|=0<0.0001$

จะได้ผลเฉลยเป็น $x^* = \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}$

ใช้การคำนวณทั้งสิ้น 2 รอบ เท่านั้น

เนื่องจากสูตรการปรับปรุงสมมาตรลำดับขั้นที่หนึ่งนี้ ไม่รับประกันความเป็น *pdf* ดังนั้น ถ้าเลือกค่าเริ่มต้นของเมตริกซ์ $\boldsymbol{B}$ ไม่เหมาะสมปัญหาจะไม่ลู่เข้า สามารถทดลองได้โดยกำหนดให้ค่าเริ่มต้นเป็น $\boldsymbol{B}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ ให้แก้ปัญหานี้ซ้ำอีกครั้ง

นอกจากสูตรการปรับปรุงสมมาตรลำดับขั้นที่หนึ่งแล้ว ยังมีสูตรการปรับปรุงขั้นอื่นๆ ซึ่งนอกจากจะทำให้เมตริกซ์ $\boldsymbol{B}$ มีความสมมาตรแล้ว ยังมีคุณสมบัติ *pdf* อีกด้วย ซึ่งจะช่วยรับประกันการลู่เข้าของปัญหาค่าต่ำสุดของฟังก์ชันไม่เชิงเส้นใดๆ ได้ ด้วยการประมาณค่ากำลังสอง สำหรับสูตรการปรับปรุงลำดับขั้นที่หนึ่งนี้ ทำให้เมตริกซ์ดังกล่าวมีความสมมาตรแต่ไม่จำเป็นต้องได้เมตริกซ์ *pdf* ดังนั้น การใช้สูตรการปรับปรุงลำดับขั้นที่สองจะช่วยให้เมตริกซ์มีความเป็น *pdf* เนื่องจากสูตรการปรับปรุงลำดับขั้นที่สองนี้ มีนักคณิตศาสตร์สร้างสูตรขึ้นหลากหลาย แต่ที่นิยมใช้กันแพร่หลายและเป็นมาตรฐานสำหรับการแก้ปัญหาค่าเหมาะที่สุด ได้แก่ สูตรการปรับปรุง BFGS (BFGS update formula) ตามชื่อของผู้พัฒนา ได้แก่ บรอยเดน (Broyden) เฟตเชอร์ (Fletcher) โกลด์ฟาร์บ (Goldfarb) และ ชานนอน (Shannon) ดังต่อไปนี้

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k - \frac{(\boldsymbol{B}_k \boldsymbol{s}_k)(\boldsymbol{B}_k \boldsymbol{s}_k)^T}{\boldsymbol{s}_k^T \boldsymbol{B}_k \boldsymbol{s}_k} + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k} = \boldsymbol{B}_k - \frac{\boldsymbol{B}_k \boldsymbol{y}_k \boldsymbol{y}_k^T \boldsymbol{B}_k}{\boldsymbol{y}_k^T \boldsymbol{B}_k \boldsymbol{y}_k} + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k} \tag{3.29}$$

นอกจากนี้ ยังมีสูตรการปรับปรุงที่สำคัญอีกสูตรหนึ่ง ได้แก่ สูตรการปรับปรุง DFP (DFP update formula) ตามชื่อของ เดวิดัน (Davidon) เฟตเชอร์ (Fletcher) และเพาเวลล์ (Powell) ดังสมการต่อไปนี้

$$\boldsymbol{B}_{k+1} = \left(\boldsymbol{I} - \gamma_k \boldsymbol{y}_k \boldsymbol{s}_k^T\right) \boldsymbol{B}_k \left(\boldsymbol{I} - \gamma_k \boldsymbol{s}_k \boldsymbol{y}_k^T\right) + \gamma_k \boldsymbol{y}_k \boldsymbol{y}_k^T \tag{3.30}$$

อย่างไรก็ตาม สูตรการปรับปรุง DFP ไม่นิยมใช้ในรูปการประมาณค่าเฮสเซียน แต่จะใช้ประมาณค่าเมตริกซ์ผกผันเฮสเซียน $\hat{\boldsymbol{B}}_k = \boldsymbol{B}_k^{-1}$ ดังนี้

$$\hat{\boldsymbol{B}}_{k+1} = \hat{\boldsymbol{B}}_{k+1} - \frac{\hat{\boldsymbol{B}}_k \boldsymbol{y}_k \boldsymbol{y}_k^T \hat{\boldsymbol{B}}_k}{\boldsymbol{y}_k^T \hat{\boldsymbol{B}}_k \boldsymbol{y}_k} + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k} \tag{3.31}$$

ในกรณีของการใช้สูตรปรับปรุง DFP นี้ จะคำนวณทิศทางการค้นหาจากสมการต่อไปนี้

$$\boldsymbol{p}_k = -\left[\boldsymbol{B}_k\right]^{-1} \nabla f(\boldsymbol{x}_k) = -\hat{\boldsymbol{B}}_k \nabla f(\boldsymbol{x}_k) \tag{3.32}$$

ตัวอย่างที่ 3.9 จงแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับโดยใช้ระเบียบวิธีคล้ายนิวตัน ด้วยสูตรการปรับปรุง BFGS

Minimize $f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1 x_2 + x_2^2$

กำหนดให้ $x_0 = [0.0 \;\; 0.0]^T$ ใช้ความคลาดเคลื่อนสูงสุดที่ยอมรับได้ $\varepsilon = 0.0001$

วิธีทำ คำนวณเกรเดียนต์และการประมาณเฮสเซียนของฟังก์ชันวัตถุประสงค์จะได้

$$\nabla f(x_k)=\begin{bmatrix}1+4x_1+2x_2\\-1+2x_1+2x_2\end{bmatrix};\quad \boldsymbol{B}_0=\begin{bmatrix}1&0\\0&1\end{bmatrix}$$

<u>รอบการคำนวณที่ 1</u>:

จะได้ทิศทางการค้นหา $\boldsymbol{p}_0=-[\boldsymbol{B}_0]^{-1}\nabla f(\boldsymbol{x}_0)=-\begin{bmatrix}1&0\\0&1\end{bmatrix}^{-1}\begin{bmatrix}1\\-1\end{bmatrix}=\begin{bmatrix}-1\\1\end{bmatrix}$

จะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

$$\text{Minimize}\qquad F(\lambda)=f\left(\begin{bmatrix}0.0\\0.0\end{bmatrix}+\lambda\begin{bmatrix}-1.0\\1.0\end{bmatrix}\right)=f\left(\begin{bmatrix}-\lambda\\\lambda\end{bmatrix}\right)=\lambda^2-2\lambda$$

$\lambda>0$

แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหาคิวบิก จะได้ $\lambda_0=1.0$

จะได้ $\boldsymbol{x}_1=\boldsymbol{x}_0+\lambda_0\boldsymbol{p}_0=\begin{bmatrix}0.0\\0.0\end{bmatrix}+(1.0)\begin{bmatrix}-1.0\\1.0\end{bmatrix}=\begin{bmatrix}-1.0\\1.0\end{bmatrix}$

$$\nabla f(\boldsymbol{x}_1)=\begin{bmatrix}1+4(-1)+2(1)\\-1+2(-1)+2(1)\end{bmatrix}=\begin{bmatrix}-1\\-1\end{bmatrix}$$

คำนวณพารามิเตอร์เพิ่มเติม จะได้

$$\boldsymbol{s}_1=\boldsymbol{x}_1-\boldsymbol{x}_0=\begin{bmatrix}-1\\1\end{bmatrix}-\begin{bmatrix}0\\0\end{bmatrix}=\begin{bmatrix}-1\\1\end{bmatrix}$$

$$\boldsymbol{y}_1=\nabla f(\boldsymbol{x}_1)-\nabla f(\boldsymbol{x}_0)=\begin{bmatrix}-1\\1\end{bmatrix}-\begin{bmatrix}1\\-1\end{bmatrix}=\begin{bmatrix}-2\\2\end{bmatrix}$$

$$\boldsymbol{B}_1=\boldsymbol{B}_0-\frac{(\boldsymbol{B}_0\boldsymbol{s}_0)(\boldsymbol{B}_0\boldsymbol{s}_0)^T}{\boldsymbol{s}_0^T\boldsymbol{B}_0\boldsymbol{s}_0}+\frac{\boldsymbol{y}_0\boldsymbol{y}_0^T}{\boldsymbol{y}_0^T\boldsymbol{s}_0}$$

$$=\begin{bmatrix}1&0\\0&1\end{bmatrix}-\frac{\left(\begin{bmatrix}1&0\\0&1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix}\right)^T\left(\begin{bmatrix}1&0\\0&1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix}\right)}{\left(-\begin{bmatrix}1&0\\0&1\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix}\right)^T\begin{bmatrix}-1\\1\end{bmatrix}}+\frac{\begin{bmatrix}-2\\2\end{bmatrix}\begin{bmatrix}-2&2\end{bmatrix}}{\begin{bmatrix}-2&2\end{bmatrix}\begin{bmatrix}-1\\1\end{bmatrix}}=\begin{bmatrix}2.5&0.5\\0.5&0.5\end{bmatrix}$$

<u>รอบการคำนวณที่ 2</u>:

จะได้ทิศทางการค้นหา $\boldsymbol{p}_1=-[\boldsymbol{B}_1]^{-1}\nabla f(\boldsymbol{x}_1)=-\begin{bmatrix}2.5&0.5\\0.5&0.5\end{bmatrix}^{-1}\begin{bmatrix}-1\\-1\end{bmatrix}=\begin{bmatrix}0\\2\end{bmatrix}$

จะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

$$\text{Minimize}\qquad F(\lambda)=f\left(\begin{bmatrix}-1\\1\end{bmatrix}+\lambda\begin{bmatrix}0\\2\end{bmatrix}\right)=f\left(\begin{bmatrix}-1\\1+2\lambda\end{bmatrix}\right)=4\lambda^2-2\lambda-1$$

$\lambda>0$

แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหาคิวบิก จะได้ $\lambda_1=0.25$

จะได้ $x_2 = x_1 + \lambda_1 p_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} + (0.25)\begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}$

$$\nabla f(x_1) = \begin{bmatrix} 1+4(-1)+2(1.5) \\ -1+2(-1)+2(1.5) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix};$$

จะได้ว่า $\nabla f(x_2) = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} \Rightarrow \|\nabla f(x_2)\| = 0 < 0.0001$

จะได้ผลเฉลยเป็น $x^* = \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}$

ใช้การคำนวณทั้งสิ้น 2 รอบ เท่านั้น

สูตรการปรับปรุง BFGS นี้รับประกันการลู่เข้าของปัญหา โดยการเลือกค่าเริ่มต้นของเมตริกซ์ $\boldsymbol{B}$ ใดๆ ที่เป็น *pdf* ที่ง่ายที่สุด คือ ให้มีค่าเป็นเมตริกซ์เอกลักษณ์ อย่างไรก็ตาม การเลือกค่าเมตริกซ์เอกลักษณ์อาจจะทำให้อัตราการลู่เข้าค่อนข้างช้าสำหรับปัญหาบางรูปแบบได้

ตัวอย่างที่ 3.11 จงแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับโดยใช้ระเบียบวิธีคล้ายนิวตันด้วยสูตรการปรับปรุง DFP

$$\text{Minimize} \qquad f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

กำหนดให้ $x_0 = [0.0 \ \ 0.0]^T$ ใช้ความคลาดเคลื่อนสูงสุดที่ยอมรับได้ $\varepsilon = 0.0001$

วิธีทำ คำนวณเกรเดียนต์และการประมาณเฮสเซียนผกผันของฟังก์ชันวัตถุประสงค์จะได้

$$\nabla f(x_k) = \begin{bmatrix} 1+4x_1+2x_2 \\ -1+2x_1+2x_2 \end{bmatrix}; \quad \hat{\boldsymbol{B}}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

<u>รอบการคำนวณที่ 1</u>:

จะได้ทิศทางการค้นหา $p_0 = -[\boldsymbol{B}_0]^{-1}\nabla f(x_0) = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

จะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

$$\text{Minimize} \qquad F(\lambda) = f\left(\begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + \lambda\begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix}\right) = f\left(\begin{bmatrix} -\lambda \\ \lambda \end{bmatrix}\right) = \lambda^2 - 2\lambda$$

$$\lambda > 0$$

แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหาคิวบิก จะได้ $\lambda_0 = 1.0$

จะได้ $x_1 = x_0 + \lambda_0 p_0 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + (1.0)\begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix} = \begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix}$

$$\nabla f(x_1) = \begin{bmatrix} 1+4(-1)+2(1) \\ -1+2(-1)+2(1) \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

คำนวณพารามิเตอร์เพิ่มเติม จะได้

$$s_1 = x_1 - x_0 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$y_1 = \nabla f(x_1) - \nabla f(x_0) = \begin{bmatrix} -1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

$$\hat{B}_1 = \hat{B}_0 - \frac{\hat{B}_0 y_0 y_0^T \hat{B}_0}{y_0^T \hat{B}_0 y_0} + \frac{s_0 s_0^T}{y_0^T s_0}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} -2 \\ 2 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}{\begin{bmatrix} -2 & 2 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} -2 \\ 2 \end{bmatrix}} + \frac{\begin{bmatrix} -1 \\ 1 \end{bmatrix}\begin{bmatrix} -1 & 1 \end{bmatrix}}{\begin{bmatrix} -2 & 2 \end{bmatrix}\begin{bmatrix} -1 \\ 1 \end{bmatrix}} = \begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 1.5 \end{bmatrix}$$

<u>**รอบ**การคำนวณที่ 2</u>:

จะได้ทิศทางการค้นหา $p_1 = -\hat{B}_1 \nabla f(x_1) = -\begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 1.5 \end{bmatrix}\begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

จะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

Minimize $\quad F(\lambda) = f\left(\begin{bmatrix} -1 \\ 1 \end{bmatrix} + \lambda \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = f\left(\begin{bmatrix} -1 \\ 1+\lambda \end{bmatrix}\right) = \lambda^2 - \lambda - 1$

$\lambda > 0$

แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหาคิวบิก จะได้ $\lambda_I = 0.5$

จะได้ $\quad x_2 = x_1 + \lambda_1 p_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} + (0.5)\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}$

$$\nabla f(x_I) = \begin{bmatrix} 1 + 4(-1) + 2(1.5) \\ -1 + 2(-1) + 2(1.5) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix};$$

จะได้ว่า $\nabla f(x_2) = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} \Rightarrow \|\nabla f(x_2)\| = 0 < 0.0001$

จะได้ผลเฉลยเป็น $\qquad x^* = \begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}$

ใช้การคำนวณทั้งสิ้น 2 รอบ เท่านั้น

จะพบว่า ข้อดีของการการใช้สูตร DFP ได้แก่ การไม่ต้องพึ่งการคำนวณเมตริกซ์ผกผันหลังจากผ่านขั้นตอนการปรับปรุงเมตริกซ์เรียบร้อยแล้ว

นอกจากสูตรปรับปรุงทั้งสามสูตรที่นำเสนอแล้ว ยังมีสูตรการปรับปรุงอีกมากมายหลายสูตร ที่น่าสนใจ ได้แก่ กลุ่มของสูตรในตระกูลบรอยเดน (Broyden family/Broyden class) ที่มีรูปทั่วไปดังนี้

$$B_{k+1} = B_k - \frac{B_k y_k y_k^T B_k}{y_k^T B_k y_k} + \frac{y_k y_k^T}{y_k^T s_k} + \phi_k \left(s_k^T B_k s_k\right) v_k v_k^T \qquad (3.33)$$

โดยที่ ϕ_k เป็นสัมประสิทธิ์ค่าคงตัว และ $v_k = \left[\frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k}\right]$

สูตร BFGS จัดเป็นสมาชิกในตระกูลบรอยเดน โดยกำหนดให้ $\phi_k = 0$ และสูตร DFP เกิดจากการปรับตั้งค่า $\phi_k = 1$

โปรแกรม SCILAB สำหรับตัวอย่างที่ 3.10 และ 3.11

// SCILAB source programs

```
// กำหนดฟังก์ชันระเบียบวิธีคล้ายนิวตันบีเอฟจีเอส                    bfgs.sci
========================
              bfgs.sci
========================
function [x,f,k]=bfgs(fobj,fgrad,x0,B0,ftol,vtol,opt)
k=0;
g0=fgrad(x0);
if opt==1 then
   disp([k x0' fobj(x0) norm(g0,%inf)]);
end
s_dir=-linsolve(B0,g0);
lambda=lsearch(fobj,x0,s_dir);
x=x0+lambda*s_dir;
f=fobj(x);
g=fgrad(x);
k=k+1;
if opt==1 then
   disp([k x' f norm(g,%inf)]);
end
s=x-x0;
y=g-g0;
k=k+1;
x0=x;
x_init=x;
g0=g;
while norm(g,%inf)>ftol
   B=B0-(B0*s)*(B0*s)'/(s'*B0*s)+(y*y')/(y'*s);
   s_dir=-linsolve(B,g);
```

```
  lambda=lsearch(fobj,x0,s_dir);
  x=x0+lambda*s_dir;
  f=fobj(x);
  g=fgrad(x);
  if opt==1 then
    disp([k x' f norm(g,%inf)]);
  end
  s=x-x0;
  y=g-g0;
  k=k+1;
  x0=x;
  g0=g;
  B0=B;
end
endfunction
```

```
// กำหนดฟังก์ชันระเบียบวิธีคล้ายนิวตันดีเอฟพี                    dfp.sci
========================
              dfp.sci
========================
function [x,f,k]=dfp(fobj,fgrad,x0,B0,ftol,vtol,opt)
k=0;
g0=fgrad(x0);
s_dir=-B0*g0;
if opt==1 then
   disp([k x0' fobj(x0) norm(g0,%inf)]);
end
lambda=lsearch(fobj,x0,s_dir);
x=x0+lambda*s_dir;
f=fobj(x);
g=fgrad(x);
k=k+1;
if opt==1 then
   disp([k x' f norm(g,%inf)]);
end
```

```
s=x-x0;
y=g-g0;
k=k+1;
x0=x;
x_init=x;
g0=g;
while norm(g,%inf)>ftol
   B=B0-(B0*y*y'*B0)/(y'*B0*y)+(s*s')/(y'*s);
   s_dir=-B*g;
   lambda=lsearch(fobj,x0,s_dir);
   x=x0+lambda*s_dir;
   f=fobj(x);
   g=fgrad(x);
   if opt==1 then
      disp([k x' f norm(g,%inf)]);
   end
   s=x-x0;
   y=g-g0;
   k=k+1;
   x0=x;
   g0=g;
   B0=B;
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('f_unc01.sci');          ประกาศฟังก์ชันวัตถุประสงค์
--> exec('g_unc01.sci');          ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('lsearch.sci');          ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('flsearch.sci');         ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('fgradnum1.sci');        ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('cubicsearch1.sci');     ประกาศฟังก์ชันย่อยเพื่อช่วยคำนวณ
--> exec('bfgs.sci');             ประกาศฟังก์ชันหลักของการคำนวณ
--> exec('dfp.sci');              ประกาศฟังก์ชันหลักของการคำนวณ
```

```
--> [x,f,k]=bfgs(f_unc01,g_unc01,[0;0],[1 0;0 1],1e-4,1e-4,1);
// Iter     X1            X2          F(x)         ||g(x)||
   0.   0.00000000    0.000000    0.0000000    1.0000000
   1. - 1.00000000    1.000000  - 1.0000000    1.0000000
   2. - 1.00000000    1.500000  - 1.2500000    2.325D-11
-->[x,f,k]=dfp(f_unc01,g_unc01,[0;0],[1 0;0 1],1e-4,1e-4,1);
// Iter     X1            X2          F(x)         ||g(x)||
   0.   0.00000000    0.000000    0.0000000    1.0000000
   1. - 1.00000000    1.000000  - 1.0000000    1.0000000
   2. - 1.00000000    1.500000  - 1.2500000    2.914D-10
```

3.10 สรุป

บทนี้ นำเสนอระเบียบวิธีการแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับ ระเบียบวิธีที่ได้รับความนิยมสูงและถูกนำมาใช้พัฒนาชุดคำสั่งโปรแกรมคอมพิวเตอร์กันอย่างแพร่หลาย ได้แก่ ระเบียบวิธีคล้ายนิวตัน อย่างไรก็ตาม ระเบียบวิธีชันที่สุด ถือได้ว่าเป็นมาตรฐานและถูกอ้างถึงควบคู่กับระเบียบวิธีอื่น ๆ เพื่อประเมินสมรรถนะของอัลกอริทึมใหม่ที่ได้รับการพัฒนาขึ้น

3.11 แบบฝึกหัดท้ายบท

จากปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับดังต่อไปนี้ จงใช้ระเบียบวิธีการเดินสุ่ม ระเบียบวิธีตัวแปรเดี่ยว ระเบียบวิธีทิศทางสังยุค ระเบียบวิธีชันที่สุด ระเบียบวิธีเกรเดียนต์สังยุค ระเบียบวิธีนิวตัน ระเบียบวิธีคล้ายนิวตัน DFP และ BFGS เพื่อแก้ปัญหา

1. $f(\boldsymbol{x}) = 6x_1^2 - 6x_1x_2 + x_2^2 - x_1 - 2x_2$, $\boldsymbol{x_0} = [1.0 \quad 1.0]^T$
2. $f(\boldsymbol{x}) = x_1^2 + 10x_2^2$, $\boldsymbol{x_0} = [1.0 \quad 1.0]^T$
3. $f(\boldsymbol{x}) = (x_1 + x_2 + 4)^2 + (x_1 - x_2 + 3)^2$, $\boldsymbol{x_0} = [1.0 \quad 1.0]^T$
4. $f(\boldsymbol{x}) = x_1^3 + x_1x_2 - x_1^2x_2^2$, $\boldsymbol{x_0} = [1.0 \quad 1.0]^T$
5. $f(\boldsymbol{x}) = 1 + x_1 + x_2 + \dfrac{4}{x_1} + \dfrac{9}{x_2}$, $\boldsymbol{x_0} = [1.0 \quad 1.0]^T$
6. $f(\boldsymbol{x}) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$, $\boldsymbol{x_0} = [-1.2 \quad 1.0]^T$
7. $f(\boldsymbol{x}) = x_1 + 2x_2^2 + e^{x_1^2 + x_2^2}$, $\boldsymbol{x_0} = [1.0 \quad 0.0]^T$
8. $f(\boldsymbol{x}) = (x_1 - 2)^4 + (x_1 - 2x_2)^2$, $\boldsymbol{x_0} = [1.0 \quad 1.0]^T$

บทที่ 4 กำหนดการเชิงเส้นเบื้องต้น
(Introduction to Linear Programming)

"Seek simplicity and distrust it"

4.1 เกริ่นนำ

ความเป็นเชิงเส้น คือ คุณสมบัติที่นักวิจัยส่วนใหญ่ต้องการ เนื่องจากฟังก์ชันเชิงเส้นเข้าใจง่าย และถือเป็นเครื่องมือพื้นฐานในการแก้ปัญหาไม่เชิงเส้นด้วย สมการของระบบไม่เชิงเส้น เมื่อพิจารณาในช่วงการเปลี่ยนแปลงค่าพารามิเตอร์ในช่วงแคบ ๆ สามารถประมาณได้ด้วยฟังก์ชันเชิงเส้น นอกจากนี้ ปัญหาบางประเภทนิยมกำหนดให้อยู่ในรูปของฟังก์ชันเชิงเส้นเพื่อให้เข้าใจง่ายและไม่ซับซ้อน เช่น ปัญหาทางด้านเศรษฐศาสตร์ การประเมินต้นทุน ผลกำไรของสินค้าที่ผลิตหรือออกแบบ เป็นต้น ดังนั้น การศึกษาการแก้ปัญหาในรูปของฟังก์ชันเชิงเส้นมีความสำคัญและจำเป็นอย่างยิ่งในงานการออกแบบทางวิศวกรรม โดยปัญหากำหนดการเชิงเส้นเป็นปัญหาค่าเหมาะที่สุดที่มีฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับในรูปฟังก์ชันเชิงเส้น ดังนี้

$$
\begin{aligned}
\text{Minimize} \quad & f(x_1, x_2) = -x_1 - 2x_2 \\
\text{Subject to} \quad & 2x_1 - x_2 \ge 4 \\
& x_1 - x_2 \le 2 \\
& x_1 \le 3 \\
& x_1, x_2 \ge 0
\end{aligned}
$$

หรือเขียนปัญหากำหนดการเชิงเส้นในรูปความสัมพันธ์แบบคาโนนิคอล (canonical form of linear programming problems) ได้ดังต่อไปนี้

$$
\begin{aligned}
\text{Minimize} \quad & f(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x} \\
\text{Subject to} \quad & \boldsymbol{A}\boldsymbol{x} \ge \boldsymbol{b} \\
& \boldsymbol{x} \ge 0
\end{aligned}
$$

ปัญหากำหนดการเชิงเส้นถูกกำหนดขึ้นประมาณปี ค.ศ. 1930 ในเบื้องต้น จะนำเสนอเทคนิคการแก้ปัญหาทางเรขาคณิตเพื่อสร้างความเข้าใจในหลักการแก้ปัญหา ดังต่อไปนี้

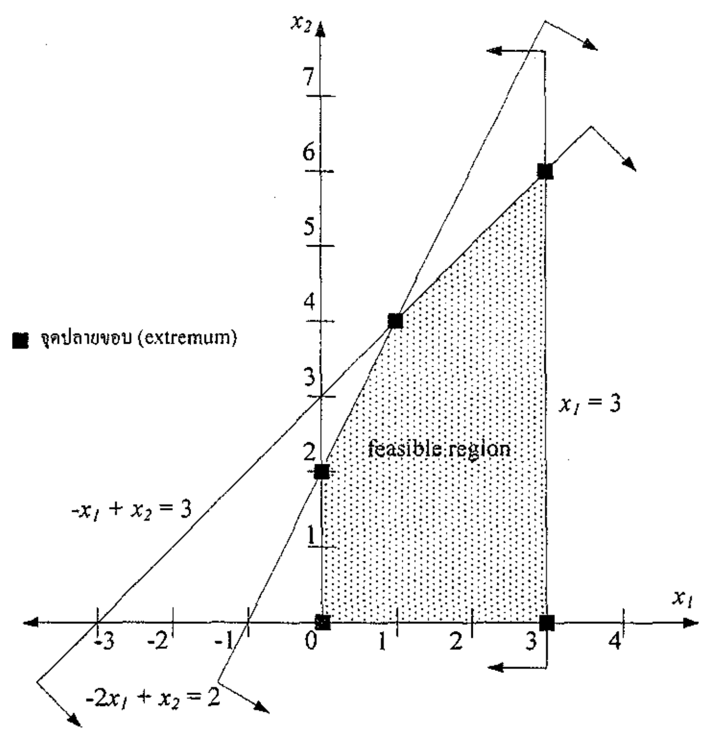
4.2 เรขาคณิตของกำหนดการเชิงเส้น (Geometry of Linear Programming)

ปัญหากำหนดการเชิงเส้นที่ใช้การแก้ปัญหาด้วยเทคนิคทางเรขาคณิตนี้ ใช้ได้กับปัญหาตัวแปรสองมิติเท่านั้น เนื่องจากต้องอาศัยการสร้างกราฟเพื่อแก้ปัญหา เมื่อได้กราฟออกมา การพิจารณาหาจุดต่ำสุดจากกราฟทำได้ไม่ยากนัก พิจารณาจากตัวอย่างต่อไปนี้

ตัวอย่างที่ 4.1 พิจารณาปัญหากำหนดการเชิงเส้นต่อไปนี้

$$\begin{aligned} &\text{Minimize} && f(x_1, x_2) = -x_1 - 2x_2 \\ &\text{Subject to} && -2x_1 + x_2 \le 2 \\ &&& -x_1 + x_2 \le 3 \\ &&& x_1 \le 3 \\ &&& x_1, x_2 \ge 0 \end{aligned}$$

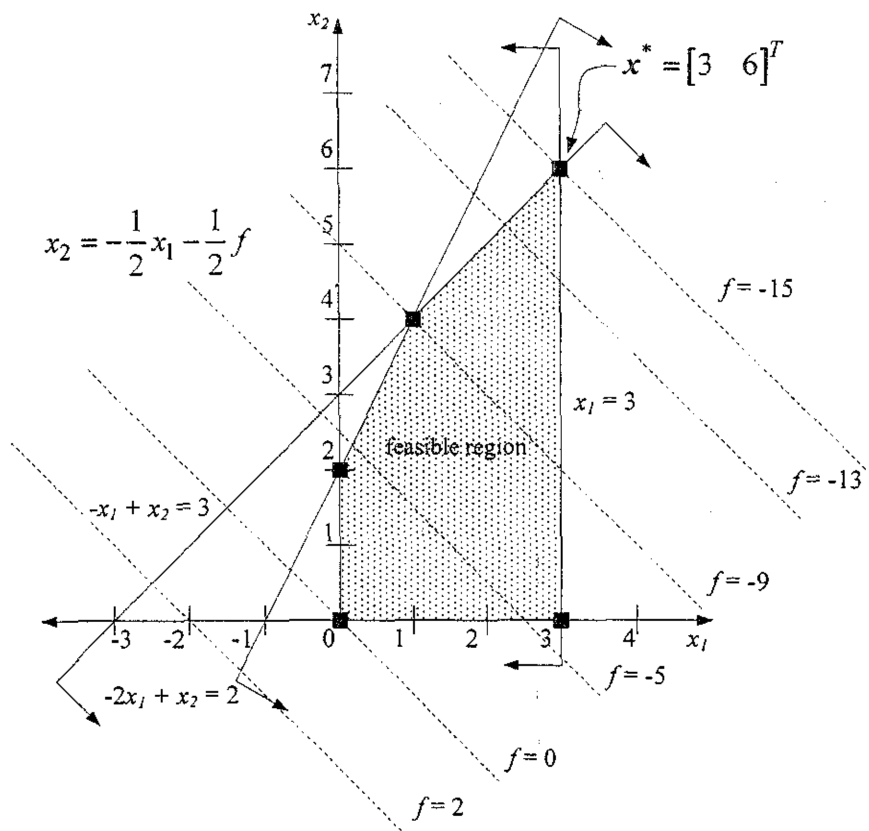
พิจารณาเงื่อนไขบังคับ จากสมการเชิงเส้นในรูป $a_1x_1 + a_2x_2 = b$ เมื่อนำมาเขียนกราฟบนระนาบสองมิติ ที่มีแกนตั้งแทนตัวแปร x_2 และแกนนอนแทนตัวแปร x_1 จะได้สมการเชิงเส้นในรูป $x_2 = -\dfrac{a_1}{a_2}x_1 + \dfrac{b}{a_2}$ สามารถนำไปวาดกราฟบนระนาบดังกล่าวได้ดังรูปต่อไปนี้



รูปที่ 4.1 เงื่อนไขบังคับของตัวอย่างที่ 4.1

จากกราฟ จะพบว่า พื้นที่ที่แรเงาเป็นบริเวณที่จุดคำตอบสอดคล้องตามเงื่อนไขบังคับทุกเงื่อนไข เรียกว่า เซตคำตอบที่เป็นไปได้ (feasible set) นั่นหมายความว่า จุดคำตอบที่เหมาะสมที่ให้ค่าต่ำสุดต้องเป็นสมาชิกของพื้นที่แรเงานี้ จุดอื่นๆ ถึงแม้จะให้ค่าฟังก์ชันวัตถุประสงค์ที่ต่ำกว่านี้ จะไม่สอดคล้องกับเงื่อนไขบังคับ ดังนั้น จะไม่ถูกนำมาพิจารณา

อย่างไรก็ตาม ในกรณีของฟังก์ชันวัตถุประสงค์ในรูป $f = c_1x_1 + c_2x_2$ ยังคงอยู่ในรูปสมการเส้นตรงเช่นกัน ในรูป $x_2 = -\dfrac{c_1}{c_2}x_1 + \dfrac{f}{c_2}$ เมื่อนำมาวาดกราฟลงบนระนาบเดียวกันกับเงื่อนไขบังคับ จะได้ดังรูปต่อไปนี้



รูปที่ 4.2 เงื่อนไขบังคับและเส้นโครงร่างของฟังก์ชันวัตถุประสงค์ในตัวอย่างที่ 4.1

จากรูป เส้นประแสดงถึงเส้นโครงร่างหรือคอนทัวร์ (contour) ของฟังก์ชันวัตถุประสงค์มีคุณสมบัติลดค่าลงจากซ้ายไปขวา เป็นจริงเฉพาะตัวอย่างนี้เท่านั้น เนื่องจากความชันของฟังก์ชันวัตถุประสงค์มีค่าติดลบ ถ้าความชันมีค่าเป็นบวกเส้นโครงร่างจะลดค่าลงจากขวาไปซ้าย ในตัวอย่างนี้ จุดคำตอบที่เหมาะที่สุดมีค่าเป็น $x^* = [3 \quad 6]^T$ ให้ค่าฟังก์ชันวัตถุประสงค์เท่ากับ -15 ซึ่งเป็นจุดปลายขอบจุดหนึ่งของเซตคำตอบที่เป็นไปได้

การแก้ปัญหากำหนดการเชิงเส้นนี้ จุดคำตอบจะเป็นจุดปลายขอบจุดใดจุดหนึ่งเสมอ ไม่ว่าฟังก์ชันวัตถุประสงค์จะมีสมการเป็นอย่างไรก็ตาม ดังนั้น การแก้ปัญหานี้ นิยมใช้การหาคำตอบโดยการตรวจสอบจุดปลายขอบ โดยจุดคำตอบที่เหมาะที่สุดจะเป็นจุดปลายขอบที่ให้ค่าฟังก์ชันวัตถุประสงค์ต่ำที่สุดนั่นเอง ปัญหานี้ดูเหมือนง่ายในกรณีสองมิติ แต่ในความเป็นจริง ถึงแม้จะมีตัวแปรเพียงสองตัว เมื่อเงื่อนไขบังคับมีความซับซ้อน จะทำให้เซตคำตอบที่เป็นไปได้มีลักษณะเป็นรูปหลายเหลี่ยม ถ้าความซับซ้อนของปัญหาทำให้จำนวนเหลี่ยมมีค่ามากมาย การตรวจสอบจุดปลายขอบทุกจุดย่อมทำได้ลำบากหรือสิ้นเปลืองเวลาในการคำนวณ ดังนั้น ประมาณปี ค.ศ. 1940 แดนต์ซิก (Dantzig) ได้พัฒนาอัลกอริทึมที่มีประสิทธิภาพเพื่อใช้แก้ปัญหากำหนดการเชิงเส้นโดยอาศัยการตรวจสอบจุดปลายขอบบางจุดเท่านั้น เรียกว่า ระเบียบวิธีซิมเพล็กซ์ (simplex method) โดยระเบียบวิธีซิมเพล็กซ์สำหรับแก้ปัญหากำหนดการเชิงเส้นมีรายละเอียดดังต่อไปนี้

4.3 ระเบียบวิธีซิมเพล็กซ์ (Simplex Method)

การแก้ปัญหาด้วยระเบียบวิธีนี้ จะเริ่มต้นจากรูปแบบของปัญหากำหนดการเชิงเส้นในรูปทั่วไป (general form of linear programming problems) ดังนี้

$$\begin{aligned} &\text{Minimize} && f(x) = c^T x \\ &\text{Subject to} && Ax = b \\ & && x \geq 0 \end{aligned} \tag{4.1}$$

ด้วยการแบ่งผลเฉลยออกเป็นสองส่วน ได้แก่ ผลเฉลยที่เป็นไปได้พื้นฐานจำนวน n ตัว (basic feasible solution: x_B) และผลเฉลยที่เป็นไปได้ที่เหลือจำนวน m ตัว (non-basic feasible solution: x_N) จะได้ว่า $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_B \\ \boldsymbol{x}_N \end{bmatrix}$

จากฟังก์ชันวัตถุประสงค์สามารถเขียนใหม่ได้เป็น $f = \boldsymbol{c}_B^T \boldsymbol{x}_B + \boldsymbol{c}_N^T \boldsymbol{x}_N$ และในกรณีของเงื่อนไขบังคับก็เช่นเดียวกันจะได้ว่า $\boldsymbol{Ax} = \boldsymbol{Bx}_B + \boldsymbol{Nx}_N = \boldsymbol{b}$ นั่นคือ

$$\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b} - \boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{x}_N \tag{4.2}$$

แทนค่าลงในฟังก์ชันวัตถุประสงค์จะได้

$$\begin{aligned} f = \boldsymbol{c}_B^T \boldsymbol{x}_B + \boldsymbol{c}_N^T \boldsymbol{x}_N &= \boldsymbol{c}_B^T \left(\boldsymbol{B}^{-1}\boldsymbol{b} - \boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{x}_N \right) + \boldsymbol{c}_N^T \boldsymbol{x}_N \\ &= \boldsymbol{c}_B^T \boldsymbol{B}^{-1}\boldsymbol{b} + \left(\boldsymbol{c}_N^T - \boldsymbol{c}_B^T \boldsymbol{B}^{-1}\boldsymbol{N} \right) \boldsymbol{x}_N \end{aligned}$$

การแก้ปัญหาเริ่มต้นด้วยการกำหนดให้ $x_N = 0$ จะได้ว่า

$$\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b} = \hat{\boldsymbol{b}} \qquad \Rightarrow \qquad \hat{f} = \boldsymbol{c}_B^T \boldsymbol{B}^{-1}\boldsymbol{b}$$

นั่นคือ

$$f = \hat{f} + \left(\boldsymbol{c}_N^T - \boldsymbol{c}_B^T \boldsymbol{B}^{-1}\boldsymbol{N} \right) \boldsymbol{x}_N = \hat{f} + \hat{\boldsymbol{c}}_N^T \boldsymbol{x}_N \tag{4.3}$$

เรียก $\hat{\boldsymbol{c}}_N^T$ ว่า สัมประสิทธิ์การลดค่าฟังก์ชันวัตถุประสงค์ (reduced cost coefficients)

เนื่องจาก ข้อกำหนดเบื้องต้น ให้ $x_N = 0$ จะพบว่า ฟังก์ชันวัตถุประสงค์ ที่จะถูกปรับปรุงต่อไป จะขึ้นอยู่กับตัวแปร x_N นั่นเอง โดยการปลดเงื่อนไขของ x_N ทีละตัว โดยกำหนดให้

$$\hat{\boldsymbol{c}}_N^T \boldsymbol{x}_N = \hat{c}_{N,1} x_{N,1} + \hat{c}_{N,2} x_{N,2} + \cdots + \hat{c}_{N,m} x_{N,m} \tag{4.4}$$

ถ้าเลือกให้ตัวแปร $x_{N,j} = \varepsilon \neq 0$ จะพบว่า $f = \hat{f} + \hat{c}_{N,j} x_{N,j} \Rightarrow \Delta f = \varepsilon \hat{c}_{N,j}$ เมื่อให้ตัวแปรควบคุมทุกตัวมีค่ามากกว่าหรือเท่ากับศูนย์เสมอ ($\varepsilon > 0$) ดังนั้น การเปลี่ยนแปลงค่าของฟังก์ชันวัตถุประสงค์ จะลดลงหรือมีทิศทางลาดลง (descent property) ก็ต่อเมื่อ สัมประสิทธิ์ $\hat{c}_{N,j} < 0$ ดังนั้น ถ้าการคำนวณวนรอบยังคงดำเนินการไปโดยที่ยังมีค่าสัมประสิทธิ์บางตัวทำให้ $\hat{c}_{N,j} < 0$ หมายความว่า ผลเฉลยที่ถูกปรับปรุงล่าสุดยังไม่ใช่จุดต่ำสุด ต้องดำเนินการวนรอบต่อไป การคำนวณด้วยระเบียบวิธีซิมเพล็กซ์จะใช้หลักการดังกล่าว โดยการกำหนด x_N ในรอบการคำนวณเริ่มต้น จากนั้น ในรอบการคำนวณถัดไปจะเลือกตัวแปรย่อยหนึ่งตัว $x_{N,j}$ จากตัวแปร x_N ที่ทำให้ค่าการเปลี่ยนแปลงของฟังก์ชันวัตถุประสงค์มีค่าลดลงมากที่สุด นั่นคือ ให้เลือกตามเงื่อนไขต่อไปนี้

$$\underset{1\le j\le m}{\text{Minimize}} \left\{\hat{c}_{N,j} \middle| \hat{c}_{N,j} < 0\right\} \tag{4.5}$$

จากดัชนี $j = J$ ดังกล่าวจะได้ตัวแปร $x_{N,J}$ ซึ่งเรียกว่า ตัวแปรเลือกเข้า (entering variable: x_{ent}) จะทำให้จำนวนตัวแปรใน x_N ลดลงหนึ่งตัว ตัวแปรเลือกเข้านี้ จะถูกนำไปแทนที่เพื่อให้เป็นตัวแปรผลเฉลยที่เป็นไปได้พื้นฐาน แต่คำถามคือ จะนำไปแทนตัวแปรตัวใดใน x_B ย่อมหมายความว่า ต้องมีตัวแปรย่อยจำนวนหนึ่งตัว $x_{B,k}$ ใน x_B ที่จะถูกเลือกออก เพื่อสลับกับ x_{ent} นั่นเอง พิจารณาจากความสัมพันธ์ต่อไปนี้

$$\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b} - \boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{x}_N = \boldsymbol{B}^{-1}\boldsymbol{b} - \hat{\boldsymbol{A}}\boldsymbol{x}_N \tag{4.6}$$

เนื่องจาก $x_{N,j} = 0$ ทุกตัวยกเว้น x_{ent} ดังนั้น จะได้

$$\boldsymbol{x}_B = \hat{\boldsymbol{b}} - \hat{\boldsymbol{A}}_J x_{ent}$$ เมื่อ $\hat{\boldsymbol{A}}_J$ แทนเวกเตอร์หลักที่ J ของ $\hat{\boldsymbol{A}}$

หรือเขียนในรูปแจกแจงสมาชิก จะได้ว่า

$$\boldsymbol{x}_B = \begin{bmatrix} x_{B,1} \\ x_{B,2} \\ \vdots \\ x_{B,n} \end{bmatrix} = \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \vdots \\ \hat{b}_n \end{bmatrix} - \begin{bmatrix} \hat{a}_{1,J} \\ \hat{a}_{2,J} \\ \vdots \\ \hat{a}_{n,J} \end{bmatrix} x_{ent} \tag{4.7}$$

เมื่อแปรค่า x_{ent} จะส่งผลให้ $x_{B,I}$ แต่ละตำแหน่งลดค่าลง จนกระทั่งมีค่าเท่ากับศูนย์หรือน้อยกว่าศูนย์ เนื่องจากตัวแปรควบคุมทุกตัวต้องมากกว่าศูนย์เสมอตามข้อตกลงเบื้องต้น ดังนั้น การทำให้ฟังก์ชันวัตถุประสงค์ลดค่าลงมากที่สุด นั่นคือ $\underset{x_{ent}\ge 0}{\text{Minimize}}\ \Delta f = \hat{c}_{N,J}x_{ent}$ จากเงื่อนไขของผลเฉลยเป็นไปได้พื้นฐานข้างต้น x_{ent} จะมีค่าได้มากที่สุดเท่ากับค่าที่ทำให้ $x_{B,I}$ ตัวแรกมีค่าเท่ากับศูนย์พอดี โดยที่ตัวอื่นๆ ยังคงมากกว่าศูนย์ นอกจากนี้ จากสมการ สัมประสิทธิ์ $\hat{a}_{i,J}$ ก็ไม่อาจมองข้าม จะพบว่า เงื่อนไขที่กล่าวมานั้นจะเป็นจริงในกรณีที่ $\hat{a}_{i,J} > 0$ ถ้า $\hat{a}_{i,J} < 0$ การเพิ่มค่า x_{ent} จะทำให้ $x_{B,I}$ มีค่าเพิ่มขึ้น จะไม่มีผลต่อการพิจารณาข้อจำกัดของผลเฉลยที่เป็นไปได้พื้นฐานนั่นเอง เพื่อให้ง่ายในการพิจารณา เงื่อนไขดังกล่าวอาจจะเขียนได้ดังนี้

$$\begin{bmatrix} x_{B,1} \\ x_{B,2} \\ \vdots \\ x_{B,n} \end{bmatrix}^{(test)} = \begin{bmatrix} \frac{\hat{b}_1}{\hat{a}_{1,J}} \\ \frac{\hat{b}_2}{\hat{a}_{2,J}} \\ \vdots \\ \frac{\hat{b}_n}{\hat{a}_{n,J}} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} x_{ent} \tag{4.8}$$

เรียกว่า สมการทดสอบอัตราส่วน (ratio test equation) จะพบว่า ที่ค่า x_{ent} คงที่ใดๆ

$$x_{B,i}^{(test)} = \frac{\hat{b}_i}{\hat{a}_{i,J}} - x_{ent} \tag{4.9}$$

$$\underset{1\le j\le m}{\text{Minimize}}\ \left\{\hat{c}_{N,j} \middle| \hat{c}_{N,j} < 0\right\} \tag{4.5}$$

จากดัชนี $j = J$ ดังกล่าวจะได้ตัวแปร $x_{N,J}$ ซึ่งเรียกว่า ตัวแปรเลือกเข้า (entering variable: x_{ent}) จะทำให้จำนวนตัวแปรใน x_N ลดลงหนึ่งตัว ตัวแปรเลือกเข้านี้ จะถูกนำไปแทนที่เพื่อให้เป็นตัวแปรผลเฉลยที่เป็นไปได้พื้นฐาน แต่คำถามคือ จะนำไปแทนตัวแปรตัวใดใน x_B ย่อมหมายความว่า ต้องมีตัวแปรย่อยจำนวนหนึ่งตัว $x_{B,k}$ ใน x_B ที่จะถูกเลือกออก เพื่อสลับกับ x_{ent} นั่นเอง พิจารณาจากความสัมพันธ์ต่อไปนี้

$$\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b} - \boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{x}_N = \boldsymbol{B}^{-1}\boldsymbol{b} - \hat{\boldsymbol{A}}\boldsymbol{x}_N \tag{4.6}$$

เนื่องจาก $x_{Nj} = 0$ ทุกตัวยกเว้น x_{ent} ดังนั้น จะได้

$$\boldsymbol{x}_B = \hat{\boldsymbol{b}} - \hat{\boldsymbol{A}}_J x_{ent} \qquad \text{เมื่อ } \hat{\boldsymbol{A}}_J \text{ แทนเวกเตอร์หลักที่ } J \text{ ของ } \hat{\boldsymbol{A}}$$

หรือเขียนในรูปแจกแจงสมาชิก จะได้ว่า

$$\boldsymbol{x}_B = \begin{bmatrix} x_{B,1} \\ x_{B,2} \\ \vdots \\ x_{B,n} \end{bmatrix} = \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \vdots \\ \hat{b}_n \end{bmatrix} - \begin{bmatrix} \hat{a}_{1,J} \\ \hat{a}_{2,J} \\ \vdots \\ \hat{a}_{n,J} \end{bmatrix} x_{ent} \tag{4.7}$$

เมื่อแปรค่า x_{ent} จะส่งผลให้ $x_{B,I}$ แต่ละตำแหน่งลดค่าลง จนกระทั่งมีค่าเท่ากับศูนย์หรือน้อยกว่าศูนย์ เนื่องจากตัวแปรควบคุมทุกตัวต้องมากกว่าศูนย์เสมอตามข้อตกลงเบื้องต้น ดังนั้น การทำให้ฟังก์ชันวัตถุประสงค์ลดค่าลงมากที่สุด นั่นคือ $\underset{x_{ent}\ge 0}{\text{Minimize}}\ \Delta f = \hat{c}_{N,J}x_{ent}$ จากเงื่อนไขของผลเฉลยเป็นไปได้พื้นฐานข้างต้น x_{ent} จะมีค่าได้มากที่สุดเท่ากับค่าที่ทำให้ $x_{B,I}$ ตัวแรกมีค่าเท่ากับศูนย์พอดี โดยที่ตัวอื่นๆ ยังคงมากกว่าศูนย์ นอกจากนี้ จากสมการ สัมประสิทธิ์ $\hat{a}_{i,J}$ ก็ไม่อาจมองข้าม จะพบว่า เงื่อนไขที่กล่าวมานั้นจะเป็นจริงในกรณีที่ $\hat{a}_{i,J} > 0$ ถ้า $\hat{a}_{i,J} < 0$ การเพิ่มค่า x_{ent} จะทำให้ $x_{B,I}$ มีค่าเพิ่มขึ้น จะไม่มีผลต่อการพิจารณาข้อจำกัดของผลเฉลยที่เป็นไปได้พื้นฐานนั่นเอง เพื่อให้ง่ายในการพิจารณา เงื่อนไขดังกล่าวอาจจะเขียนได้ดังนี้

$$\begin{bmatrix} x_{B,1} \\ x_{B,2} \\ \vdots \\ x_{B,n} \end{bmatrix}^{(test)} = \begin{bmatrix} \dfrac{\hat{b}_1}{\hat{a}_{1,J}} \\ \dfrac{\hat{b}_2}{\hat{a}_{2,J}} \\ \vdots \\ \dfrac{\hat{b}_n}{\hat{a}_{n,J}} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} x_{ent} \tag{4.8}$$

เรียกว่า สมการทดสอบอัตราส่วน (ratio test equation) จะพบว่า ที่ค่า x_{ent} คงที่ใดๆ

$$x_{B,i}^{(test)} = \frac{\hat{b}_i}{\hat{a}_{i,J}} - x_{ent} \tag{4.9}$$

นั่นคือ $x_{B,i}^{(test)}$ ตัวที่มีค่าน้อยที่สุด จะเป็นตัวที่มีค่า $\frac{\hat{b}_i}{\hat{a}_{i,J}}$ มากที่สุดนั่นเอง ดังนั้น ตัวแปรที่ถูกเลือกออกจะต้องมีคุณสมบัติดังนี้

$$\underset{1\le i\le n}{\text{Maximize}} \left\{ \frac{\hat{b}_i}{\hat{a}_{i,J}} \middle| \hat{a}_{i,J} > 0 \right\} \tag{4.10}$$

จะได้ดัชนี $i = I$ ทำการแลกเปลี่ยนตัวแปรเลือกเข้า x_{ent} และตัวแปรเลือกออก (leaving variable: x_{lvg}) ทำให้ได้การแบ่งกลุ่มตัวแปรใหม่เพื่อใช้คำนวณรอบถัดไป อย่างไรก็ตาม ในกรณีที่ค่าสัมประสิทธิ์ $\forall \hat{a}_{i,J} \le 0$ ทำให้ไม่สามารถเลือกตัวแปรเลือกออกได้ ในกรณีนี้ หมายความว่าจำนวนผลเฉลยมีจำนวนไม่จำกัด เรียกว่า ปัญหาที่ไม่มีขอบเขตจำกัด (unbounded problem) พิจารณาตัวอย่างต่อไปนี้

ตัวอย่างที่ 4.2 พิจารณาปัญหากำหนดการเชิงเส้นต่อไปนี้

$$\begin{aligned}
\text{Minimize} \quad & f(x_1, x_2) = -x_1 - 2x_2 \\
\text{Subject to} \quad & -2x_1 + x_2 \le 2 \\
& -x_1 + 2x_2 \le 7 \\
& x_1 \le 3 \\
& x_1, x_2 \ge 0
\end{aligned}$$

ปัญหานี้ยังไม่อยู่ในรูปทั่วไป เนื่องจากเงื่อนไขบังคับเป็นแบบอสมการ (inequality constraints) ดังนั้น ต้องดัดแปลงปัญหาให้อยู่ในรูปทั่วไป โดยใช้ตัวแปรสแล็ก (slack variables) ดังนี้

$$\begin{aligned}
\text{Minimize} \quad & f(x_1, x_2) = -x_1 - 2x_2 \\
\text{Subject to} \quad & -2x_1 + x_2 + x_3 = 2 \\
& -x_1 + 2x_2 + x_4 = 7 \\
& x_1 + x_5 = 3 \\
& x_1, x_2, x_3, x_4, x_5 \ge 0
\end{aligned}$$

หลักการของตัวแปรสแลกจะถูกนำมาใช้เพื่อแปลงอสมการในรูป $\le$ ให้เป็นสมการเงื่อนไข ดังเช่น $-2x_1 + x_2 \le 2$ เนื่องจาก ค่าทางด้านซ้ายมือน้อยกว่าหรือเท่ากับค่าทางด้านขวามือ ดังนั้น ถ้าบวกด้านซ้ายของอสมการนี้ด้วยตัวแปรสแล็ก s_1 ที่มีค่ามากกว่าหรือเท่าศูนย์ จะได้ว่า

$$-2x_1 + x_2 + s_1 = 2$$

เพื่อให้สอดคล้องกับตัวแปรและให้ง่ายในการแก้ปัญหา เนื่องจากตัวแปรสแล็กจัดเป็นตัวแปรควบคุมตัวหนึ่งเช่นกัน ดังนั้น กำหนดให้ $x_3 = s_1$ ในกรณีของอสมการที่เหลือก็เช่นเดียวกัน.

เทียบกับรูปทั่วไปจะได้ว่า

$$c = \begin{bmatrix} -1 \\ -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}; A = \begin{bmatrix} -2 & 1 & 1 & 0 & 0 \\ -1 & 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}; b = \begin{bmatrix} 2 \\ 7 \\ 3 \end{bmatrix}$$

แก้ปัญหาโดยเลือกตัวแปร $x_B = [x_1 \quad x_2 \quad x_3]^T ; x_N = [x_4 \quad x_5]^T$ จะได้ว่า

$$c = \begin{bmatrix} c_B \\ c_N \end{bmatrix}; \qquad c_B = \begin{bmatrix} -1 \\ -2 \\ 0 \end{bmatrix}; c_N = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A = [B \quad N]; \quad B = \begin{bmatrix} -2 & 1 & 1 \\ -1 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix}; N = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

<u>รอบการคำนวณที่ 1</u>:

$$x_B = B^{-1}b = \hat{b} = \begin{bmatrix} 3 \\ 5 \\ 3 \end{bmatrix} \quad \Rightarrow \quad \hat{f} = c_B^T B^{-1} b = -13$$

$$\hat{c}_N^T = \left(c_N^T - c_B^T B^{-1} N\right) = [1 \quad 2]$$

จะพบว่า สัมประสิทธิ์เป็นบวกทั้งหมด นั่นคือ ไม่สามารถปรับปรุงฟังก์ชันวัตถุประสงค์ได้อีกแล้ว ดังนั้น ได้ผลเฉลย $x^* = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^* = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$ ให้ค่าฟังก์ชันวัตถุประสงค์ -13

การแก้ปัญหานี้ ขึ้นอยู่กับเลือกตัวแปร x_B กับ x_N ถ้าเลือกได้ดี จะใช้รอบการคำนวณน้อย พิจารณาชุดตัวแปรดังต่อไปนี้ เลือกตัวแปร $x_B = [x_2 \quad x_3 \quad x_5]^T ; x_N = [x_1 \quad x_4]^T$ จะได้ว่า

$$c = \begin{bmatrix} c_B \\ c_N \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \qquad c_B = \begin{bmatrix} -2 \\ 0 \\ 0 \end{bmatrix} \begin{array}{l} \leftrightarrow x_2 \\ \leftrightarrow x_3 \\ \rightarrow x_5 \end{array}; c_N = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$A = [B \quad N] = \begin{bmatrix} -2 & 1 & 1 & 0 & 0 \\ -1 & 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}; \quad B = \begin{array}{ccc} \begin{bmatrix} 1 & 1 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \updownarrow \quad \updownarrow \quad \updownarrow \\ x_2 \quad x_3 \quad x_5 \end{array}; N = \begin{bmatrix} -2 & 0 \\ -1 & 1 \\ 1 & 0 \end{bmatrix}$$

<u>รอบการคำนวณที่ 1</u>:

$$x_B = B^{-1}b = \hat{b} = \begin{bmatrix} 3.5 \\ -1.5 \\ 3.0 \end{bmatrix} \quad \Rightarrow \quad \hat{f} = c_B^T B^{-1} b = -7$$

$$\hat{c}_N^T = \left(c_N^T - c_B^T B^{-1} N\right) = \begin{bmatrix} -2 & 1 \end{bmatrix}$$

$$\hat{A} = B^{-1}N = \begin{bmatrix} -0.5 & 0.5 \\ -1.5 & -0.5 \\ 1.0 & 0.0 \end{bmatrix}$$

ในกรณีนี้ สามารถปรับปรุงจุดคำตอบได้ นั่นคือ พจน์ที่สัมประสิทธิ์ที่ติดลบมากที่สุดสอดคล้องกับ $x_{ent} = x_1$ นั่นคือ $J = 1$ ดำเนินการทดสอบอัตราส่วน

$$x_B = \begin{bmatrix} x_2 \\ x_3 \\ x_5 \end{bmatrix} = \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix} - \begin{bmatrix} \hat{a}_{1,1} \\ \hat{a}_{2,1} \\ \hat{a}_{3,1} \end{bmatrix} x_{ent} = \begin{bmatrix} 3.5 \\ -1.5 \\ 3.0 \end{bmatrix} - \begin{bmatrix} -0.5 \\ -1.5 \\ 1.0 \end{bmatrix} x_1$$

จะได้อัตราส่วนดังนี้

$$ratio\ test = \begin{bmatrix} \frac{\hat{b}_1}{\hat{a}_{1,1}} \\ \frac{\hat{b}_2}{\hat{a}_{2,1}} \\ \frac{\hat{b}_3}{\hat{a}_{3,1}} \end{bmatrix} = \begin{bmatrix} \frac{3.5}{-0.5} \\ \frac{-1.5}{-1.5} \\ \frac{3.0}{1.0} \end{bmatrix} = \begin{bmatrix} -7.0 \\ 1.0 \\ 3.0 \end{bmatrix} \rightarrow x_{lvg} = x_5$$

นั่นคือ สลับตัวแปรระหว่าง $x_1 \leftrightarrow x_5$ จะได้ $x_B = \begin{bmatrix} x_2 & x_3 & x_1 \end{bmatrix}^T ; x_N = \begin{bmatrix} x_5 & x_4 \end{bmatrix}^T$

$$c = \begin{bmatrix} c_B \\ c_N \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \qquad c_B = \begin{bmatrix} -2 \\ 0 \\ -1 \end{bmatrix} \begin{matrix} \leftrightarrow x_2 \\ \leftrightarrow x_3 \\ \rightarrow x_1 \end{matrix} ; c_N = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A = \begin{bmatrix} B & N \end{bmatrix} = \begin{bmatrix} -2 & 1 & 1 & 0 & 0 \\ -1 & 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}; \quad B = \begin{bmatrix} 1 & 1 & -2 \\ 2 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix}_{\begin{matrix} \updownarrow & \updownarrow & \updownarrow \\ x_2 & x_3 & x_1 \end{matrix}} ; N = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

<u>รอบการคำนวณที่ 2</u>:

$$x_B = B^{-1}b = \hat{b} = \begin{bmatrix} 5 \\ 3 \\ 3 \end{bmatrix} \quad \Rightarrow \qquad \hat{f} = c_B^T B^{-1} b = -13$$

$$\hat{c}_N^T = \left(c_N^T - c_B^T B^{-1} N\right) = \begin{bmatrix} 2 & 1 \end{bmatrix}$$

ในกรณีนี้ จะพบว่า สัมประสิทธิ์เป็นบวกทั้งหมด นั่นคือ ไม่สามารถปรับปรุงฟังก์ชันวัตถุประสงค์ได้อีกแล้ว ดังนั้น ได้ผลเฉลย $x^* = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^* = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$ ให้ค่าฟังก์ชันวัตถุประสงค์ -13

4.4 การเลือกตัวแปรที่เป็นไปได้พื้นฐาน (Selection of Basic Feasible Variables)

จากตัวอย่างที่ผ่านมา การเลือกตัวแปรที่เป็นไปได้พื้นฐานมีความสำคัญมากต่อการค้นพบจุดคำตอบ ในกรณีที่ปัญหามีตัวแปรสแล็ก ตัวแปรที่เป็นไปได้พื้นฐานชุดหนึ่งที่เหมาะสม ได้แก่ การเลือกชุดของตัวแปรสแล็กนั่นเอง อย่างไรก็ตาม ตัวแปรสแล็กไม่สามารถปรากฏอยู่ในปัญหากำหนดการเชิงเส้นได้ทุกปัญหา ทำให้ต้องมีวิธีเลือกตัวแปรที่เป็นไปได้พื้นฐานที่เหมาะสม ดังนั้น เพื่อให้รับประกันว่า การแก้ปัญหากำหนดการเชิงเส้นสามารถหาตัวแปรที่เป็นไปได้พื้นฐาน จำเป็นต้องมีข้อกำหนดเพิ่มเติมเพื่อใช้ในการแก้ปัญหาดังรายละเอียดต่อไปนี้

$$\begin{aligned}
\text{Minimize} \quad & f(x_1, x_2) = 2x_1 + 3x_2 \\
\text{Subject to} \quad & 3x_1 + 2x_2 = 14 \\
& 2x_1 - 4x_2 \geq 2 \\
& 4x_1 + 3x_2 \leq 19 \\
& x_1, x_2 \geq 0
\end{aligned}$$

ปัญหานี้เขียนในรูปทั่วไปได้เป็น

$$\begin{aligned}
\text{Minimize} \quad & f(x_1, x_2) = 2x_1 + 3x_2 \\
\text{Subject to} \quad & 3x_1 + 2x_2 \qquad\qquad = 14 \\
& 2x_1 - 4x_2 - x_3 \qquad = 2 \\
& 4x_1 + 3x_2 \qquad + x_4 = 19 \\
& x_1, x_2, x_3, x_4 \geq 0
\end{aligned}$$

เมื่อ x_3 เป็นตัวแปรสแล็ก และ x_4 เป็นตัวแปรเซอร์พลัส (surplus variable) ที่ทำหน้าที่แปลงอสมการในรูป $\geq$ ให้เป็นสมการเงื่อนไข ดังเช่น $2x_1 - 4x_2 \geq 2$ เนื่องจาก ค่าทางด้านซ้ายมือมากกว่าหรือเท่ากับค่าทางด้านขวามือ ดังนั้น ถ้าลบด้านซ้ายของอสมการนี้ด้วยตัวแปรเซอร์พลัส s_1 ที่มีค่ามากกว่าหรือเท่าศูนย์ จะได้ว่า $2x_1 - 4x_2 - s_1 = 2$

จะพบว่า ปัญหานี้ไม่สามารถเลือกผลเฉลยที่เป็นไปได้พื้นฐาน เนื่องจาก ไม่ว่าจะกำหนดให้ตัวแปรสองตัวคู่ใดเป็นศูนย์ จะส่งผลให้ตัวแปรที่เหลือมีค่าไม่สอดคล้องกับเงื่อนไข เช่น

- ถ้ากำหนดให้ $x_1 = 0, x_2 = 0$ จะได้ $x_3 = -2, x_4 = 19$ (infeasible)
- ถ้ากำหนดให้ $x_1 = 0, x_3 = 0$ จะได้ $x_2 = 7$ หรือ $-\frac{1}{2}$, $x_4 = -2$ (infeasible)

จะพบว่า ไม่ว่าจะเลือกจับคู่ตัวแปรอย่างไร จะไม่สามารถหา x_B ที่สอดคล้องกับเงื่อนไขทั้งหมดได้ ดังนั้น เพื่อแก้ปัญหานี้ สิ่งแรกที่ต้องทำ ได้แก่ การค้นหาผลเฉลยที่เป็นไปได้พื้นฐานที่สอดคล้องกับเงื่อนไขบังคับทุกกรณี จะใช้การนิยามตัวแปรประดิษฐ์ (artificial variables) เพิ่มเติม โดยการบวกตัวแปรประดิษฐ์เข้ากับชุดเงื่อนไขสมการที่ไม่มีตัวแปรสแล็กอยู่ ถึงแม้สมการนั้นจะมีตัวแปรเซอร์พลัสก็ต้องบวกตัวแปรประดิษฐ์ด้วยเช่นกัน จะได้ว่า

$$\begin{array}{lllllll}
\text{Minimize} & f(x_1,x_2)=2x_1+3x_2 \\
\text{Subject to} & 3x_1+2x_2 & & +a_1 & & =14 \\
 & 2x_1-4x_2-x_3 & & & +a_2 & =2 \\
 & 4x_1+3x_2 & +x_4 & & & =19 \\
 & x_1,x_2,x_3,x_4,a_1,a_2\geq 0
\end{array}$$

ระเบียบวิธีการแก้ปัญหานี้ มีอ้างถึงสองระเบียบวิธีด้วยกัน ได้แก่ ระเบียบวิธีสองเฟส (two-phase method) และระเบียบวิธีบิกเอ็ม (big M method) โดยมีรายละเอียดดังนี้

4.4.1 ระเบียบวิธีสองเฟส

ระเบียบวิธีนี้จะแบ่งเป็นสองส่วน หรือเรียกว่าสองเฟส ในเฟสแรก จะค้นหาผลเฉลยที่เป็นไปได้พื้นฐาน โดยการกำหนดปัญหากำหนดการเชิงเส้นขึ้นมาใหม่สำหรับเฟสนี้โดยเฉพาะ เพื่อความสะดวก กำหนดให้ตัวแปรประดิษฐ์กำหนดสัญลักษณ์เป็น x_5 และ x_6 ดังนี้

$$\begin{array}{lllllll}
\text{Minimize} & x_5+x_6 \\
\text{Subject to} & 3x_1+2x_2 & & +x_5 & & =14 \\
 & 2x_1-4x_2-x_3 & & & +x_6 & =2 \\
 & 4x_1+3x_2 & +x_4 & & & =19 \\
 & x_1,x_2,x_3,x_4,x_5,x_6\geq 0
\end{array}$$

ในขั้นตอนนี้ จะค้นหาจุดต่ำสุดที่ทำให้ผลรวมเชิงเส้นของตัวแปรประดิษฐ์ทุกตัวมีค่าน้อยที่สุด ซึ่งคาดหวังไว้ว่า จะมีค่าเป็นศูนย์ ที่จุดนี้ จะได้ผลเฉลยที่เป็นไปได้พื้นฐาน และสิ้นสุดการคำนวณในเฟสเรียก เรียกว่า Phase-I จากนั้น จะนำผลเฉลยที่เป็นไปได้พื้นฐานดังกล่าว มาใช้แก้ปัญหาด้วยระเบียบวิธีซิมเพล็กซ์ตามปกติต่อไป ซึ่งก็คือ การแก้ปัญหาในเฟสที่สอง (Phase-II) นั่นเอง พิจารณาจากตัวอย่างต่อไปนี้

ตัวอย่างที่ 4.3 จงแก้ปัญหากำหนดการเชิงเส้นต่อไปนี้ ด้วยระเบียบวิธีสองเฟส

$$\begin{array}{ll}
\text{Minimize} & f(x_1,x_2)=2x_1+3x_2 \\
\text{Subject to} & 3x_1+2x_2=14 \\
 & 2x_1-4x_2\geq 2 \\
 & 4x_1+3x_2\leq 19 \\
 & x_1,x_2\geq 0
\end{array}$$

วิธีทำ เขียนในรูปทั่วไปได้เป็น

$$\begin{array}{lllllll}
\text{Minimize} & f(x_1,x_2)=2x_1+3x_2 \\
\text{Subject to} & 3x_1+2x_2 & & +a_1 & & =14 \\
 & 2x_1-4x_2-x_3 & & & +a_2 & =2 \\
 & 4x_1+3x_2 & +x_4 & & & =19 \\
 & x_1,x_2,x_3,x_4,a_1,a_2\geq 0
\end{array}$$

ดำเนินการแก้ปัญหาในเฟสแรก จะได้

Phase-I:

$$\begin{array}{ll} \text{Minimize} & x_5 + x_6 \\ \text{Subject to} & 3x_1 + 2x_2 \qquad\qquad + x_5 \qquad = 14 \\ & 2x_1 - 4x_2 - x_3 \qquad\qquad + x_6 = 2 \\ & 4x_1 + 3x_2 \qquad + x_4 \qquad\qquad = 19 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{array}$$

เทียบกับรูปทั่วไป

$$\boldsymbol{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}; \boldsymbol{A} = \begin{bmatrix} 3 & 2 & 0 & 0 & 1 & 0 \\ 2 & -4 & -1 & 0 & 0 & 1 \\ 4 & 3 & 0 & 1 & 0 & 0 \end{bmatrix}; \boldsymbol{b} = \begin{bmatrix} 14 \\ 2 \\ 19 \end{bmatrix}$$

แก้ปัญหาโดยเลือกตัวแปร $\boldsymbol{x}_B = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T ; \boldsymbol{x}_N = \begin{bmatrix} x_4 & x_5 & x_6 \end{bmatrix}^T$ จะได้ว่า

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}_B \\ \boldsymbol{c}_N \end{bmatrix}; \qquad \boldsymbol{c}_B = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \boldsymbol{c}_N = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{N} \end{bmatrix}; \quad \boldsymbol{B} = \begin{bmatrix} 3 & 2 & 0 \\ 2 & -4 & -1 \\ 4 & 3 & 0 \end{bmatrix}; \boldsymbol{N} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

<u>รอบการคำนวณที่ 1</u>:

$$\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b} = \hat{\boldsymbol{b}} = \begin{bmatrix} 4 \\ 1 \\ 2 \end{bmatrix} \quad \Rightarrow \qquad \hat{f} = \boldsymbol{c}_B^T \boldsymbol{B}^{-1} \boldsymbol{b} = 0$$

ผลการคำนวณได้ฟังก์ชันวัตถุประสงค์ของ Phase-I เป็นศูนย์ นั่นคือ ได้ผลเฉลยที่เป็นไปได้พื้นฐาน ในกรณีนี้ ให้ตัดตัวแปรประดิษฐ์ทิ้งไป จะได้ว่า $\boldsymbol{x}_B = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T ; \boldsymbol{x}_N = \begin{bmatrix} x_4 \end{bmatrix}^T$

Phase-II:

$$\begin{array}{ll} \text{Minimize} & 2x_1 + 3x_2 \\ \text{Subject to} & 3x_1 + 2x_2 \qquad\qquad = 14 \\ & 2x_1 - 4x_2 - x_3 \qquad = 2 \\ & 4x_1 + 3x_2 \qquad + x_4 = 19 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{array}$$

เทียบกับรูปทั่วไป

$$\boldsymbol{c}=\begin{bmatrix}2\\3\\0\\0\end{bmatrix};\boldsymbol{A}=\begin{bmatrix}3&2&0&0\\2&-4&-1&0\\4&3&0&1\end{bmatrix};\boldsymbol{b}=\begin{bmatrix}14\\2\\19\end{bmatrix}$$

แก้ปัญหาโดยเลือกตัวแปร $\boldsymbol{x}_B=\begin{bmatrix}x_1 & x_2 & x_3\end{bmatrix}^T;\boldsymbol{x}_N=\begin{bmatrix}x_4\end{bmatrix}^T$ จะได้ว่า

$$\boldsymbol{c}=\begin{bmatrix}\boldsymbol{c}_B\\\boldsymbol{c}_N\end{bmatrix};\qquad \boldsymbol{c}_B=\begin{bmatrix}2\\3\\0\end{bmatrix};\boldsymbol{c}_N=\begin{bmatrix}0\end{bmatrix}$$

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{B} & \boldsymbol{N}\end{bmatrix};\ \boldsymbol{B}=\begin{bmatrix}3&2&0\\2&-4&-1\\4&3&0\end{bmatrix};\boldsymbol{N}=\begin{bmatrix}0\\0\\1\end{bmatrix}$$

<u>รอบการคำนวณที่ 1</u>:

$$\boldsymbol{x}_B=\boldsymbol{B}^{-1}\boldsymbol{b}=\hat{\boldsymbol{b}}=\begin{bmatrix}4\\1\\2\end{bmatrix}\Rightarrow\qquad \hat{f}=\boldsymbol{c}_B^T\boldsymbol{B}^{-1}\boldsymbol{b}=11$$

$$\hat{\boldsymbol{c}}_N^T=\left(\boldsymbol{c}_N^T-\boldsymbol{c}_B^T\boldsymbol{B}^{-1}\boldsymbol{N}\right)=\begin{bmatrix}-5\end{bmatrix}$$

$$\hat{\boldsymbol{A}}=\boldsymbol{B}^{-1}\boldsymbol{N}=\begin{bmatrix}-2\\3\\-16\end{bmatrix}$$

ในกรณีนี้ สามารถปรับปรุงจุดคำตอบได้ นั่นคือ พจน์ที่สัมประสิทธิ์ $\hat{c}_{N,i}$ ติดลบมากที่สุ สอดคล้องกับ $x_{ent}=x_4$ นั่นคือ $J=1$ ดำเนินการทดสอบอัตราส่วน

$$\boldsymbol{x}_B=\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}\hat{b}_1\\\hat{b}_2\\\hat{b}_3\end{bmatrix}-\begin{bmatrix}\hat{a}_{1,1}\\\hat{a}_{2,1}\\\hat{a}_{3,1}\end{bmatrix}x_{ent}=\begin{bmatrix}4\\1\\2\end{bmatrix}-\begin{bmatrix}-2\\3\\-16\end{bmatrix}x_4$$

จะได้อัตราส่วนดังนี้

$$ratio\ test=\begin{bmatrix}\frac{\hat{b}_1}{\hat{a}_{1,1}}\\\frac{\hat{b}_2}{\hat{a}_{2,1}}\\\frac{\hat{b}_3}{\hat{a}_{3,1}}\end{bmatrix}=\begin{bmatrix}\frac{4}{-2}\\\frac{1}{3}\\\frac{2}{-16}\end{bmatrix}=\begin{bmatrix}-2.0\\1/3\\-1/8\end{bmatrix}\rightarrow x_{lvg}=x_2$$

นั่นคือ สลับตัวแปรระหว่าง $x_4\leftrightarrow x_2$ จะได้ $\boldsymbol{x}_B=\begin{bmatrix}x_1 & x_4 & x_3\end{bmatrix}^T;\boldsymbol{x}_N=\begin{bmatrix}x_2\end{bmatrix}^T$

$$\boldsymbol{c}=\begin{bmatrix}\boldsymbol{c}_B\\\boldsymbol{c}_N\end{bmatrix}=\begin{bmatrix}2\\3\\0\\0\end{bmatrix};\qquad \boldsymbol{c}_B=\begin{bmatrix}2\\0\\0\end{bmatrix}\begin{matrix}\leftrightarrow x_1\\\leftrightarrow x_4\\\leftrightarrow x_3\end{matrix};\boldsymbol{c}_N=\begin{bmatrix}3\end{bmatrix}$$

$$A=\begin{bmatrix}B & N\end{bmatrix}=\begin{bmatrix}3 & 2 & 0 & 0\\ 2 & -4 & -1 & 0\\ 4 & 3 & 0 & 1\end{bmatrix};\qquad B=\begin{array}{c}\begin{bmatrix}3 & 0 & 0\\ 2 & 0 & -1\\ 4 & 1 & 0\end{bmatrix}\\ \begin{array}{ccc}\updownarrow & \updownarrow & \updownarrow\\ x_1 & x_4 & x_3\end{array}\end{array};N=\begin{bmatrix}2\\ -4\\ 3\end{bmatrix}$$

<u>รอบการคำนวณที่ 2</u>:

$$x_B=B^{-1}b=\hat{b}=\begin{bmatrix}4.67\\ 0.33\\ 7.33\end{bmatrix}\qquad\Rightarrow\qquad \hat{f}=c_B^TB^{-1}b=9.33$$

$$\hat{c}_N^T=\left(c_N^T-c_B^TB^{-1}N\right)=\begin{bmatrix}1.67\end{bmatrix}$$

พบว่าได้สัมประสิทธิ์เป็นบวกทั้งหมด นั่นคือ ไม่สามารถปรับปรุงฟังก์ชันวัตถุประสงค์ได้อีกแล้ว ดังนั้น ได้ผลเฉลย $x^*=\begin{bmatrix}x_1\\ x_2\end{bmatrix}^*=\begin{bmatrix}4.67\\ 0.0\end{bmatrix}$ ให้ค่าฟังก์ชันวัตถุประสงค์ 9.33

4.4.2 ระเบียบวิธีบิกเอ็ม

ระเบียบวิธีนี้ใช้การสร้างฟังก์ชันการปรับโทษ (penalty function) โดยใช้ตัวประกอบการปรับโทษ M คูณกับตัวแปรประดิษฐ์ เมื่อ M มีค่ามากๆ จะส่งผลให้ตัวแปรประดิษฐ์ต้องมีค่าเข้าสู่ศูนย์ มิฉะนั้นแล้ว ฟังก์ชันวัตถุประสงค์จะเพิ่มค่าขึ้นไปตามค่าตัวแปรประดิษฐ์นั่นเอง กำหนดฟังก์ชันการปรับโทษดังต่อไปนี้

$$p=f+M\sum_{i=1}^{N_A}a_i$$

จากนั้น ให้ใช้ระเบียบวิธีซิมเพล็กซ์แก้ปัญหาตามปกติ ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 4.4 จงแก้ปัญหากำหนดการเชิงเส้นต่อไปนี้ ด้วยระเบียบวิธีบิกเอ็ม

$$\begin{aligned}\text{Minimize}\quad & f(x_1,x_2)=2x_1+3x_2\\ \text{Subject to}\quad & 3x_1+2x_2=14\\ & 2x_1-4x_2\ge 2\\ & 4x_1+3x_2\le 19\\ & x_1,x_2\ge 0\end{aligned}$$

วิธีทำ เขียนในรูปทั่วไปได้เป็น

$$\begin{aligned}\text{Minimize}\quad & f(x_1,x_2)=2x_1+3x_2\\ \text{Subject to}\quad & 3x_1+2x_2+x_5=14\\ & 2x_1-4x_2-x_3+x_6=2\\ & 4x_1+3x_2+x_4=19\\ & x_1,x_2,x_3,x_4,x_5,x_6\ge 0\end{aligned}$$

แปลงฟังก์ชันวัตถุประสงค์เป็นฟังก์ชันการปรับโทษ และให้ค่า $M=1000$ จะได้ว่า

$$\begin{array}{ll} \text{Minimize} & p = 2x_1 + 3x_2 + 1000(x_5 + x_6) \\ \text{Subject to} & 3x_1 + 2x_2 \qquad\qquad + x_5 \qquad = 14 \\ & 2x_1 - 4x_2 - x_3 \qquad\qquad + x_6 = 2 \\ & 4x_1 + 3x_2 \qquad + x_4 \qquad\qquad = 19 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \ge 0 \end{array}$$

เทียบกับรูปทั่วไป

$$\boldsymbol{c} = \begin{bmatrix} 2 \\ 3 \\ 0 \\ 0 \\ 1000 \\ 1000 \end{bmatrix}; \boldsymbol{A} = \begin{bmatrix} 3 & 2 & 0 & 0 & 1 & 0 \\ 2 & -4 & -1 & 0 & 0 & 1 \\ 4 & 3 & 0 & 1 & 0 & 0 \end{bmatrix}; \boldsymbol{b} = \begin{bmatrix} 14 \\ 2 \\ 19 \end{bmatrix}$$

แก้ปัญหาโดยเลือกตัวแปร $\boldsymbol{x}_B = [x_1 \quad x_2 \quad x_3]^T$; $\boldsymbol{x}_N = [x_4 \quad x_5 \quad x_6]^T$ จะได้ว่า

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}_B \\ \boldsymbol{c}_N \end{bmatrix}; \qquad \boldsymbol{c}_B = \begin{bmatrix} 2 \\ 3 \\ 0 \end{bmatrix}; \boldsymbol{c}_N = \begin{bmatrix} 0 \\ 1000 \\ 1000 \end{bmatrix}$$

$$\boldsymbol{A} = [\boldsymbol{B} \quad \boldsymbol{N}]; \quad \boldsymbol{B} = \begin{bmatrix} 3 & 2 & 0 \\ 2 & -4 & -1 \\ 4 & 3 & 0 \end{bmatrix}; \boldsymbol{N} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

<u>รอบการคำนวณที่ 1</u>:

$$\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b} = \hat{\boldsymbol{b}} = \begin{bmatrix} 4 \\ 1 \\ 2 \end{bmatrix} \Rightarrow \qquad \hat{f} = \boldsymbol{c}_B^T \boldsymbol{B}^{-1} \boldsymbol{b} = 11$$

$$\hat{\boldsymbol{c}}_N^T = \left(\boldsymbol{c}_N^T - \boldsymbol{c}_B^T \boldsymbol{B}^{-1} \boldsymbol{N}\right) = [-5 \quad 1006 \quad 1000]$$

$$\hat{\boldsymbol{A}} = \boldsymbol{B}^{-1}\boldsymbol{N} = \begin{bmatrix} -2 & 3 & 0 \\ 3 & -4 & 0 \\ -16 & 22 & -1 \end{bmatrix}$$

ในกรณีนี้ สามารถปรับปรุงจุดคำตอบได้ นั่นคือ พจน์ที่สัมประสิทธิ์ $\hat{c}_{N,i}$ ติดลบมากที่สุด สอดคล้องกับ $x_{ent} = x_4$ นั่นคือ $J = 1$ ดำเนินการทดสอบอัตราส่วน

$$\boldsymbol{x}_B = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \end{bmatrix} - \begin{bmatrix} \hat{a}_{1,1} \\ \hat{a}_{2,1} \\ \hat{a}_{3,1} \end{bmatrix} x_{ent} = \begin{bmatrix} 4 \\ 1 \\ 2 \end{bmatrix} - \begin{bmatrix} -2 \\ 3 \\ -16 \end{bmatrix} x_4$$

จะได้อัตราส่วนดังนี้

$$ratio\ test = \begin{bmatrix} \frac{\hat{b}_1}{\hat{a}_{1,1}} \\ \frac{\hat{b}_2}{\hat{a}_{2,1}} \\ \frac{\hat{b}_3}{\hat{a}_{3,1}} \end{bmatrix} = \begin{bmatrix} \frac{4}{-2} \\ \frac{1}{3} \\ \frac{2}{-16} \end{bmatrix} = \begin{bmatrix} -2.0 \\ 1/3 \\ -1/8 \end{bmatrix} \rightarrow x_{lvg} = x_2$$

นั่นคือ สลับตัวแปรระหว่าง $x_4 \leftrightarrow x_2$ จะได้ $\boldsymbol{x}_B = \begin{bmatrix} x_1 & x_4 & x_3 \end{bmatrix}^T ; \boldsymbol{x}_N = \begin{bmatrix} x_2 \end{bmatrix}^T$

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}_B \\ \boldsymbol{c}_N \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 0 \\ 0 \\ 1000 \\ 1000 \end{bmatrix}; \quad \boldsymbol{c}_B = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} \leftrightarrow x_1 \\ \leftrightarrow x_4 \\ \leftrightarrow x_3 \end{matrix} ; \boldsymbol{c}_N = \begin{bmatrix} 3 \\ 1000 \\ 1000 \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{N} \end{bmatrix} = \begin{bmatrix} 3 & 2 & 0 & 0 & 1 & 0 \\ 2 & -4 & -1 & 0 & 0 & 1 \\ 4 & 3 & 0 & 1 & 0 & 0 \end{bmatrix}; \quad \boldsymbol{B} = \begin{array}{c} \begin{bmatrix} 3 & 0 & 0 \\ 2 & 0 & -1 \\ 4 & 1 & 0 \end{bmatrix} \\ \begin{matrix} \updownarrow & \updownarrow & \updownarrow \\ x_1 & x_4 & x_3 \end{matrix} \end{array} ; \boldsymbol{N} = \begin{bmatrix} 2 & 1 & 0 \\ -4 & 0 & 1 \\ 3 & 0 & 0 \end{bmatrix}$$

<u>รอบการคำนวณที่ 2</u>:

$$\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b} = \hat{\boldsymbol{b}} = \begin{bmatrix} 4.67 \\ 0.33 \\ 7.33 \end{bmatrix} \quad \Rightarrow \quad \hat{f} = \boldsymbol{c}_B^T \boldsymbol{B}^{-1}\boldsymbol{b} = 9.33$$

$$\hat{\boldsymbol{c}}_N^T = \left(\boldsymbol{c}_N^T - \boldsymbol{c}_B^T \boldsymbol{B}^{-1} \boldsymbol{N} \right) = \begin{bmatrix} 1.67 & 999.3 & 1000 \end{bmatrix}$$

พบว่าได้สัมประสิทธิ์เป็นบวกทั้งหมด นั่นคือ ไม่สามารถปรับปรุงฟังก์ชันวัตถุประสงค์ได้อีกแล้ว ดังนั้น ได้ผลเฉลย $\boldsymbol{x}^* = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^* = \begin{bmatrix} 4.67 \\ 0.0 \end{bmatrix}$ ให้ค่าฟังก์ชันวัตถุประสงค์ 9.33

ปัญหาที่สำคัญของวิธีบิกเอ็ม คือ การเลือกค่า M ถ้าเลือกไม่เหมาะสม เช่น น้อยเกินไปจะส่งผลให้ผลการคำนวณไม่ถูกต้องเนื่องจากตัวแปรประดิษฐ์มีค่าเข้าใกล้ศูนย์ไม่มากพอ ถ้าเลือกค่ามากเกินไปจะส่งผลต่อข้อจำกัดหรือความละเอียดในการคำนวณของคอมพิวเตอร์เอง การเลือกใช้ระเบียบวิธีที่เหมาะสมและตัวปรับคูณที่ดีจะทำให้การแก้ปัญหาทำได้ง่ายและรวดเร็วขึ้น

4.5 ผลเฉลยการแก้ปัญหากำหนดการเชิงเส้นด้วย SCILAB

โปรแกรม SCILAB มีกล่องเครื่องมือ (Toolbox) ที่ช่วยแก้ปัญหากำหนดการเชิงเส้นได้และใช้งานไม่ยาก ถ้าทราบการกำหนดปัญหาในรูปแบบคาโนนิคอลและรูปทั่วไป กล่องเครื่องมือเพื่อช่วยคำนวณผลเฉลยของปัญหากำหนดการเชิงเส้นพัฒนาขึ้นโดย เอดูอาโด คาซาส เรนทีเรีย

(Eduardo Casas Renteria) จากมหาวิทยาลัยแห่งแคนทาเบรีย (Universidad de Cantabria) ประเทศสเปน โดยมีรูปแบบของปัญหาดังนี้

$$\begin{aligned} \text{Minimize} \quad & f(x) = p^T x \\ \text{Subject to} \quad & C_1 x = b_1 \\ & C_2 x \le b_2 \\ & x_L \le x \le x_U \\ & x \ge 0 \end{aligned}$$

จะพบว่า ปัญหานี้ต้องการตัวแปร p, C_1, C_2, b_1, b_2, x_L และ x_U การแก้ปัญหาในรูปดังกล่าว ทำได้โดยใช้คำสั่ง linpro ย่อมาจาก linear programming solver ดังนี้

[x,lagr,f] = linpro(p,C,b,ci,cs,me)

p คือ สัมประสิทธิ์ของฟังก์ชันวัตถุประสงค์สำหรับกำหนดการเชิงเส้น

me คือ จำนวนเงื่อนไขบังคับสมการสำหรับกำหนดการเชิงเส้น

C คือ สัมประสิทธิ์ด้านซ้ายของเงื่อนไขบังคับสำหรับกำหนดการเชิงเส้น C = $[C_1;C_2]$

b คือ สัมประสิทธิ์ด้านขวาของเงื่อนไขบังคับสำหรับกำหนดการเชิงเส้น b = $[b_1;b_2]$

ci,cs คือ x_L และ x_U สำหรับกำหนดการเชิงเส้น

ตัวอย่างที่ 4.5 จงแก้ปัญหากำหนดการเชิงเส้นต่อไปนี้ โดยใช้คำสั่ง linpro ของ SCILAB

$$\begin{aligned} \text{Minimize} \quad & f(x_1, x_2) = 2x_1 + 3x_2 \\ \text{Subject to} \quad & 3x_1 + 2x_2 = 14 \\ & 2x_1 - 4x_2 \ge 2 \\ & 4x_1 + 3x_2 \le 19 \\ & x_1, x_2 \ge 0 \end{aligned}$$

วิธีทำ เขียนในรูปทั่วไปได้เป็น

$$\begin{aligned} \text{Minimize} \quad & f(x_1, x_2) = 2x_1 + 3x_2 \\ \text{Subject to} \quad & 3x_1 + 2x_2 = 14 \\ & -2x_1 + 4x_2 \le 2 \\ & 4x_1 + 3x_2 \le 19 \\ & x_1, x_2 \ge 0 \end{aligned}$$

นั่นคือ

$$p = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 3 & 2 \end{bmatrix}, b_1 = \begin{bmatrix} 14 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} -2 & 4 \\ 4 & 3 \end{bmatrix}, b_2 = \begin{bmatrix} 2 \\ 19 \end{bmatrix}$$

จะได้ว่า

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 4.5</u>

// SCILAB source program

```
// การแก้กำหนดการเชิงเส้นด้วย SCILAB
--> p = [2;3];
--> C1 = [3 2]; b1 = [14];
--> C2 = [-2 4;4 3]; b2 = [2;19];
--> me = 1;
--> C = [C1;C2];
--> b = [b1;b2];
--> ci = [0;0]; cs = [10;10];          กำหนดขอบเขตของตัวแปร x
--> [x,lagr,f]=linpro(p,C,b,[0;0],[10;10],me);
--> f
 f =
   9.3333333
--> x
 x =
   4.6666667
   0.0000000
```

4.6 สรุป

บทนี้ ได้นำเสนอหลักการแก้ปัญหากำหนดการเชิงเส้นขั้นพื้นฐาน โดยเน้นไปที่ระเบียบวิธีซิมเพล็กซ์เป็นสำคัญ ซึ่งเพียงพอต่อการนำไปใช้เพื่อแก้ปัญหาในงานวิศวกรรมต่อไป สำหรับหัวข้ออื่นๆ ที่น่าสนใจ เช่น duality and sensitivity, degeneracy หรือ interior point LP method เป็นต้น สามารถศึกษาเพิ่มเติมได้จากหนังสือ/ตำราทางด้านกำหนดการเชิงเส้นหรือเทคนิคการหาค่าเหมาะที่สุดเล่มอื่น ๆ ได้

4.7 แบบฝึกหัดท้ายบท

1. จงแก้ปัญหากำหนดการเชิงเส้นต่อไปนี้

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) = 3x_1 + x_2 + x_3 \\ \text{Subject to} \quad & x_1 - 2x_2 + x_3 \le 11 \\ & -x_1 + x_2 + 2x_3 \ge 3 \\ & 2x_1 - x_3 = -1 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

2. จงแก้ปัญหากำหนดการเชิงเส้นต่อไปนี้

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) = 6x_1 + 2x_2 \\ \text{Subject to} \quad & 10x_1 + x_2 \ge 10 \\ & 5x_1 + 4x_2 \ge 20 \\ & 3x_1 + 7x_2 \ge 21 \\ & x_1 + 12x_2 \ge 12 \end{aligned}$$

3. จงแก้ปัญหากำหนดการเชิงเส้นต่อไปนี้

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) = -3x_1 + 2x_2 \\ \text{Subject to} \quad & 0 \le x_1 \le 4 \\ & 1 \le x_2 \le 6 \\ & x_1 + x_2 \le 5 \end{aligned}$$

4. จงแก้ปัญหากำหนดการเชิงเส้นต่อไปนี้

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) = x_1 - 3x_2 + 2x_3 \\ \text{Subject to} \quad & -2 \le x_1 \le 4, \quad 0 \le x_2 \le 4, \quad 2 \le x_3 \le 5 \end{aligned}$$

5. จงแก้ปัญหากำหนดการเชิงเส้นต่อไปนี้

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) = 3x_1 + x_2 + x_3 + x_4 \\ \text{Subject to} \quad & -2x_1 + 2x_2 + x_3 = 4 \\ & 3x_1 + x_2 + x_4 = 6 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{aligned}$$

บทที่ 5 ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับ (Constrained Optimization Problems)

"In the beginner's mind there are many possible solutions, in the expert's mind there are few"

5.1 เกริ่นนำ

การแก้ปัญหาค่าเหมาะที่สุดที่ผ่านมาเน้นไปที่ปัญหาแบบไม่มีเงื่อนไขบังคับ แบบจำลองด้วยฟังก์ชันกำลังสองสามารถนำมาใช้ประมาณค่าฟังก์ชันไม่เชิงเส้นได้ ถ้าจุดที่พิจารณาอยู่ใกล้จุดต่ำสุดมากพอ เมื่อปัญหามีความซับซ้อนและสะท้อนการออกแบบทางวิศวกรรมมากขึ้น เงื่อนไขบังคับเป็นสิ่งที่ไม่อาจหลีกเลี่ยงได้ ส่วนใหญ่อยู่ในรูปของข้อจำกัดต่าง ๆ เช่น ปริมาณวัสดุหรือวัตถุดิบที่มีอยู่ ความแข็งแรงหรือความทนทานของชิ้นงาน หรือข้อจำกัดด้านการออกแบบอื่น ๆ เป็นต้น การจัดการเงื่อนไขบังคับได้กล่าวถึงมาบ้างแล้วในปัญหาการโปรแกรมเชิงเส้น อย่างไรก็ตาม ปัญหาดังกล่าวถูกจำกัดอยู่ที่เงื่อนไขบังคับในรูปฟังก์ชันเชิงเส้นเท่านั้น ปัญหาในงานวิศวกรรมโดยทั่วไปไม่จำเป็นต้องมีคุณสมบัติความเป็นเชิงเส้น ดังนั้น การแก้ปัญหาภายใต้เงื่อนไขบังคับในรูปทั่วไปเป็นสิ่งที่สำคัญและเป็นหัวข้อหลักในบทนี้ ก่อนที่จะกล่าวถึงรายละเอียด ต้องกำหนดปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับในรูปทั่วไปดังนี้

$$\begin{array}{lllll} \text{Minimize} & f(\boldsymbol{x}) & & & \\ \text{Subject to} & g_i(\boldsymbol{x}) = 0, & i \in E & & \text{(equality constraints)} \\ & h_j(\boldsymbol{x}) \geq 0, & j \in I & & \text{(inequality constraints)} \end{array}$$

เทคนิคการแก้ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับมีรายละเอียดดังต่อไปนี้

5.2 ปัญหาที่มีเงื่อนไขบังคับสมการเชิงเส้น (linear equality constraint problems)

กำหนดปัญหาค่าเหมาะที่สุดดังต่อไปนี้

$$\begin{array}{lll} \text{Minimize} & f(\boldsymbol{x}) & \\ \text{Subject to} & \boldsymbol{Ax} = \boldsymbol{b}, & \text{(linear equality constraints)} \end{array}$$

ถ้าให้ $\bar{\boldsymbol{x}}$ เป็นจุดคำตอบใด ๆ ที่ทำให้สมการ $\boldsymbol{Ax} = \boldsymbol{b}$ เป็นจริง จะเรียกจุดคำตอบนี้ว่าคำตอบที่เป็นไปได้ของปัญหา (feasible solution) ซึ่งจุดนี้ ไม่จำเป็นต้องเป็นผลเฉลยของปัญหา ดังนั้น การค้นหาผลเฉลยของปัญหานี้ต้องทำการปรับปรุงจุดคำตอบที่เป็นไปได้ดังกล่าว โดยให้จุดคำตอบที่ถูกปรับปรุง มีค่าดังนี้ $\boldsymbol{x} = \bar{\boldsymbol{x}}+\boldsymbol{p}$ เมื่อ $\boldsymbol{p}$ คือ ทิศทางการค้นหา หรืออาจจะเรียกว่า ทิศทางที่เป็นไปได้ (feasible direction) สำหรับปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับ หมายถึงทิศทางการค้นหาที่ทำให้จุดคำตอบที่ถูกปรับปรุงยังคงเป็นคำตอบที่สอดคล้องกับเงื่อนไขบังคับของปัญหาและในขณะเดียวกัน จะทำให้ค่าฟังก์ชันวัตถุประสงค์ที่จุดคำตอบที่ถูกปรับปรุงลดค่าลงด้วย นั่นคือ

$$\boldsymbol{Ax} = \boldsymbol{A}(\bar{\boldsymbol{x}}+\boldsymbol{p}) = \boldsymbol{A}\bar{\boldsymbol{x}}+\boldsymbol{Ap} = \boldsymbol{b}+\boldsymbol{Ap} \;\Rightarrow\; \boldsymbol{Ap} = 0 \tag{5.1}$$

นั่นคือ ถ้ากำหนดเมตริกซ์การแปลงเชิงเส้นแทนด้วย A ใด ๆ การแปลงเวกเตอร์ p ใด ๆ ในปริภูมิให้เป็นเวกเตอร์ศูนย์นั้น จะเรียกปริภูมิที่ประกอบด้วยสมาชิกเวกเตอร์ p ที่มีคุณสมบัติดังกล่าวว่า null space หรือ kernel ของการแปลงเชิงเส้น A ทำให้ทราบทิศทางที่เป็นไปได้ที่ใช้ปรับปรุงจุดคำตอบของปัญหา อย่างไรก็ตาม ถ้าการแปลงเชิงเส้นดังกล่าวแทนได้ด้วยเมตริกซ์การแปลง $m \times n$ โดยที่ $n \geq m$ แล้ว เวกเตอร์ p สามารถเขียนในอยู่ในรูปของผลรวมเชิงเส้นของเวกเตอร์จำนวน m ชุด เมื่อ $m = rank(A)$ นั่นเอง

$$p = \sum_{j=1}^{n} \sum_{i=1}^{m} z_{i,j} v_i = Zv \tag{5.2}$$

โดย $Z = n \times (n\text{-}m)$ เมตริกซ์, v = เวกเตอร์หลักมีสมาชิกเท่ากับ $n\text{-}m$ ตำแหน่ง

นั่นคือ เมื่อแทนค่า $p = Zv$ ลงในสมการปรับปรุงจุดคำตอบที่เป็นไปได้ จะส่งผลให้ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับสมการลดรูปเป็นปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับ ดังนี้

$$\begin{array}{ll} \text{Minimize} & \phi(v) = f(x) = f(\bar{x} + Zv) \\ v \in \Re^{n-m} & \end{array} \tag{5.3}$$

เรียก ฟังก์ชัน $\phi(v)$ ว่า ฟังก์ชันลดทอน (reduced function) และเรียกเทคนิคการลดรูปแบบนี้ว่า ระเบียบวิธีการกำจัดตัวแปร (variable-elimination method)

ถ้ากำหนดให้ p ซึ่งสอดคล้องกับปัญหา $Ap = 0$ แบ่งออกเป็นตัวแปรพื้นฐาน (basic variables: p_B) และตัวแปรที่เหลือ (non-basic variables: p_N) จะได้

$$Ap = \begin{bmatrix} B & N \end{bmatrix} \begin{bmatrix} p_B \\ p_N \end{bmatrix} = Bp_B + Np_N = 0 \qquad \Rightarrow \qquad p_B = -B^{-1} N p_N$$

นั่นคือ

$$p = \begin{bmatrix} p_B \\ p_N \end{bmatrix} = \begin{bmatrix} -B^{-1}N \\ I \end{bmatrix} p_N = Zv \qquad \Rightarrow \qquad v \equiv p_N, Z = \begin{bmatrix} -B^{-1}N \\ I \end{bmatrix}$$

ถ้าเลือกจุดคำตอบที่เป็นไปได้เริ่มต้น โดยกำหนดให้ $x_N = 0$

$$\bar{x} = \begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix} \text{ จะได้ } x = \bar{x} + Zp_N$$

สำหรับปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไข พิจารณาที่ค่าผลเฉลยที่เหมาะสม จะได้

- <u>เงื่อนไขจำเป็น (Necessary conditions)</u>

$$\nabla \phi(v^*) = Z^T \nabla f(x^*) = 0$$

$$\nabla^2 \phi(v^*) = Z^T \nabla^2 f(x^*) Z \text{ is positive semi-definite}(psdf)$$

- <u>เงื่อนไขที่เพียงพอ (Sufficient conditions)</u>

$$Ax^* = b$$
$$\nabla\phi\left(v^*\right) = Z^T \nabla f\left(x^*\right) = 0$$
$$\nabla^2\phi\left(v^*\right) = Z^T \nabla^2 f\left(x^*\right) Z \textit{ is positive definite}(pdf)$$
$\nabla\phi(x)$: *reduced gradient / projected gradient*

$\nabla^2\phi(x)$: *reduced Hessian matrix / projected Hessian matrix*

พิจารณาจากตัวอย่างต่อไปนี้

ตัวอย่างที่ 5.1 จงหาค่าเหมาะที่สุดของปัญหาต่อไปนี้

Minimize $\quad f(x) = x_1^2 - 2x_1 + x_2^2 - x_3^2 + 4x_3$

subject to $\quad x_1 - x_2 + 2x_3 = 2$

จากปัญหาที่กำหนดจะได้

$$A = [1 \quad -1 \quad 2]; b = [2]; B = [1]; N = [-1 \quad 2]$$
$$Z = \begin{bmatrix} -B^{-1}N \\ I \end{bmatrix} = \begin{bmatrix} 1 & -2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}; \bar{x} = \begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$$
$$\therefore x = \bar{x} + Zv = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & -2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_1 - 2v_2 + 2 \\ v_1 \\ v_2 \end{bmatrix}$$

จะได้ฟังก์ชันลดทอนเป็น

$$\phi(v) = (v_1 - 2v_2 + 2)^2 + v_1^2 - v_2^2 - 2(v_1 - 2v_2 + 2) + 4v_2$$
$$\nabla\phi\left(v^*\right) = \begin{bmatrix} 4v_1 - 4v_2 + 2 \\ -4v_1 + 6v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \Rightarrow \qquad v^* = \begin{bmatrix} -1 \\ -1.5 \end{bmatrix}$$
$$\nabla^2\phi\left(v^*\right) = \begin{bmatrix} 4 & -4 \\ -4 & 6 \end{bmatrix} \textit{ is pdf} \rightarrow \qquad v^* \text{ is a local minimum}$$
$$x^* = \bar{x} + Zv^* = \begin{bmatrix} 2.5 \\ -1.5 \\ -1 \end{bmatrix}$$

เมื่อค้นพบผลเฉลยที่เหมาะที่สุดแล้ว ย่อมหมายความว่า การคำนวณโดยกำหนดให้ $\bar{x} = x^*$ นี้ย่อมส่งผลให้ $Zv^* = 0$ นั่นเอง

5.3 ตัวคูณลากรองจ์และฟังก์ชันลากรองจ์ (Lagrange multiplier and Lagrangian function)

กำหนดการแปลงเชิงเส้น A: $\Re^m \rightarrow \Re^n$ ปริภูมิศูนย์ (null space/kernel)

$$\aleph(A) = \{p | p \in \Re^n, Ap = 0\} \qquad (5.4)$$

จากคุณสมบัติการแผ่ทั่วปริภูมิ (vector spanned), จะพบว่า สำหรับเวกเตอร์ $\boldsymbol{q} \in \Re^n$ ใด ๆ ที่เกิดจากการแปลง A ซึ่งเป็นเมตริกซ์ขนาด $m \times n$ และมี $rank(A) = m$ โดยที่ $m \leq n$ ดังกล่าว สามารถเขียนให้อยู่ในรูปของความสัมพันธ์ $\boldsymbol{q} = A^T\lambda$ โดยที่ $\lambda \in \Re^m$ จะเรียกปริภูมินี้ว่า ปริภูมิเรนจ์ (range space)

$$\mathcal{R}(A) = \{\boldsymbol{q} \mid \boldsymbol{q} \in \Re^n, A^T\lambda = \boldsymbol{q} \text{ สำหรับ } \exists\lambda \in \Re^m \} \tag{5.5}$$

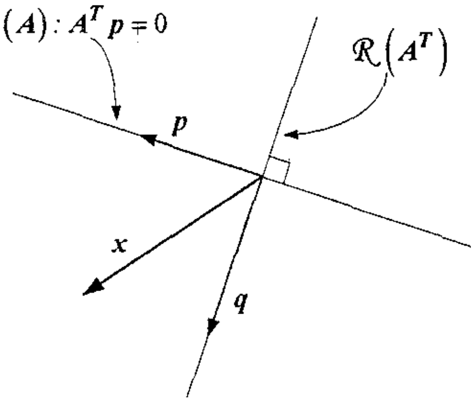
สำหรับปริภูมิย่อยใด ๆ สองปริภูมิที่มีคุณสมบัติตั้งฉาก (orthogonal property) แล้ว ผลคูณภายใน (inner product) ระหว่างเวกเตอร์ใด ๆ จากปริภูมิย่อยทั้งสองจะมีค่าเท่ากับศูนย์ หรือเพื่อให้เฉพาะเจาะจงกับปริภูมิศูนย์และปริภูมิเรนจ์ จะได้ว่า

$$\boldsymbol{Q}^T\boldsymbol{p} = (A^T\lambda)^T\boldsymbol{p} = \lambda^T A\boldsymbol{p} = \lambda^T(A\boldsymbol{p}) = 0 \tag{5.6}$$

และเนื่องจากปริภูมิย่อยทั้งสองตั้งฉากกันและมีมิติเท่ากับ n จะพบว่าเวกเตอร์ $\boldsymbol{x}$ ใด ๆ ในปริภูมิ $\Re^n$ ย่อมสามารถเขียนแยกออกเป็นองค์ประกอบตั้งฉากให้อยู่ในปริภูมิย่อยทั้งสองได้เป็น

$$\boldsymbol{X} = \boldsymbol{p} + \boldsymbol{q} \tag{5.7}$$

พิจารณากราฟนำเสนอผลดังกล่าวได้ดังรูปต่อไปนี้



รูปที่ 5.1 องค์ประกอบของเวกเตอร์ในปริภูมิศูนย์และปริภูมิเรนจ์

จากปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับสมการที่เขียนในรูปฟังก์ชันลดทอน $\phi(\boldsymbol{v})$ จะพบว่า

$\nabla\phi(\boldsymbol{v}^*) = 0$ แต่เกรเดียนต์ของ $f(\boldsymbol{x})$ ไม่จำเป็นต้องเท่ากับ 0 ดังนั้น เกรเดียนต์ของฟังก์ชันวัตถุประสงค์อาจจะเขียนให้อยู่ในรูปขององค์ประกอบตั้งฉากได้ดังนี้

$\nabla f(\boldsymbol{x}^*) = \boldsymbol{p} + \boldsymbol{q} = \boldsymbol{Z}\,\boldsymbol{v}^* + \boldsymbol{A}^T\lambda$ และจากความสัมพันธ์ $\boldsymbol{Z}\,\boldsymbol{v}^* = 0$ นั่นคือ

$\nabla f(\boldsymbol{v}^*) = \boldsymbol{A}^T\lambda^*$ เมื่อ $\boldsymbol{x}^*$ เป็นจุดต่ำสุดเฉพาะถิ่น (local minimizer)

เรียก λ ว่า ตัวคูณลากรองจ์ (Lagrange multiplier) จะได้ว่า

$$\nabla L(\boldsymbol{x}^*, \lambda^*) = \nabla f(\boldsymbol{x}^*) - \nabla[\lambda^{*T}(A\,\boldsymbol{x}^* - \boldsymbol{b})] = 0 \tag{5.8}$$

เมื่อจุดที่พิจารณาไม่ใช่จุดต่ำสุดเฉพาะถิ่น

$$\nabla L(\boldsymbol{x}, \lambda) = \nabla f(\boldsymbol{x}) - \nabla[\lambda^T(A\boldsymbol{x} - \boldsymbol{b})] \tag{5.9}$$

จะได้ว่า

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) - \lambda^T\boldsymbol{g}(\boldsymbol{x}) \tag{5.10}$$

เรียก $L(x, \lambda)$ ว่า ฟังก์ชันลากรองจ์ (Lagrangian function)

ตัวอย่างที่ 5.2 จงหาค่าเหมาะที่สุดของปัญหาต่อไปนี้

$$\text{Minimize} \quad f(x) = 2x_1 + x_2 + 10$$

$$\text{subject to} \quad 2x_1^2 - x_2 + 3 = 0$$

จากปัญหาที่กำหนดจะได้ฟังก์ชันลากรองจ์ดังนี้

$$\mathcal{L}(x,\lambda) = 2x_1 + x_2 + 10 - \lambda\left(2x_1^2 - x_2 + 3\right)$$

จากเงื่อนไขที่จำเป็น $\nabla\mathcal{L}(x^*,\lambda^*) = 0$ จะได้

$$\frac{\partial\mathcal{L}}{\partial x_1} = 2 - 4\lambda x_1 = 0$$

$$\frac{\partial\mathcal{L}}{\partial x_2} = 1 + \lambda = 0$$

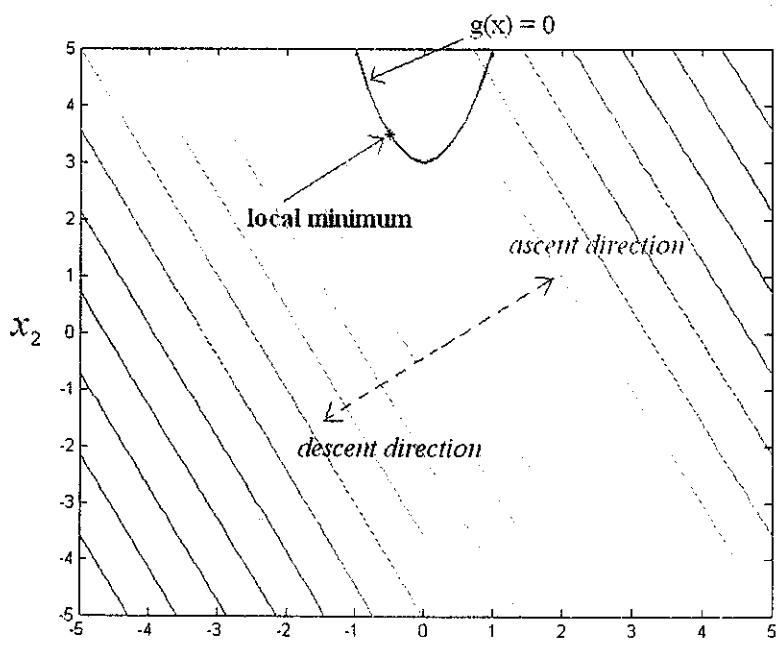
$$\frac{\partial\mathcal{L}}{\partial\lambda} = 2x_1^2 - x_2 + 3 = 0$$

นั่นคือ

$$\lambda^* = -1,\ \boldsymbol{x}^* = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^* = \begin{bmatrix} -\frac{1}{2} \\ \frac{7}{2} \end{bmatrix}$$

จุดนี้เป็นจุดต่ำสุดของปัญหาที่พิจารณา (*ให้ตรวจสอบโดยคำนวณเมตริกซ์เฮสเซียนของฟังก์ชันลากรองจ์ที่จุดนี้ ต้องเป็น pdf*)

เพื่อให้เห็นภาพได้ชัดเจน พิจารณาจากกราฟต่อไปนี้ประกอบ



รูปที่ 5.2 คอนทัวร์ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับสำหรับตัวอย่างที่ 5.2

5.4 ปัญหาที่มีเงื่อนไขบังคับอสมการเชิงเส้น (linear inequality constraint problems)

มีรูปสมการทั่วไปเป็นดังนี้

Minimize $f(x)$
Subject to $\boldsymbol{Ax} \geq \boldsymbol{b}$, (linear inequality constraints)

การแก้ปัญหาในลักษณะนี้ทำได้หลายแบบ ในที่นี้ จะนำเสนอเพียง 2 รูปแบบเท่านั้น

- การใช้ตัวแปรสแล็ก (slack variables) เพื่อแปลงเงื่อนไขบังคับแบบอสมการให้เป็นเงื่อนไขบังคับแบบสมการ

จาก $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{Ax} - \boldsymbol{b} \geq 0$ เมื่อใช้ตัวแปรสแล็กจะได้

$\boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{s}^2 = 0$ เมื่อ $\boldsymbol{s}^2 \geq 0$

เมื่อแปลงเงื่อนไขบังคับเรียบร้อยแล้ว ขั้นตอนต่อไปก็คือ การสร้างฟังก์ชันลากรองจ์เพื่อแก้ปัญหา โดยจะมีตัวแปรสแล็กเพิ่มเติมเข้ามา ทำให้มิติของปัญหาเพิ่มขึ้น ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 5.3 จงหาค่าเหมาะที่สุดของปัญหาต่อไปนี้

Minimize $f(x) = x_1^2 + 2x_2^2$
subject to $2x_1 - x_2 \geq 4$

จากปัญหาที่กำหนดจะได้ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับสมการดังนี้

Minimize $f(x) = x_1^2 + 2x_2^2$
subject to $h(x) = 2x_1 - x_2 - 4 - s^2 = 0$

จะได้

$$\mathcal{L}(x, s, \lambda) = x_1^2 + 2x_2^2 - \lambda\left(2x_1 - x_2 - 4 - s^2\right)$$

จากเงื่อนไขที่จำเป็น $\nabla\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{s}^*, \lambda^*) = 0$ จะได้

$$\frac{\partial \mathcal{L}}{\partial x_1} = x_1 - \lambda = 0$$
$$\frac{\partial \mathcal{L}}{\partial x_2} = 4x_2 + \lambda = 0$$
$$\frac{\partial \mathcal{L}}{\partial s} = 2\lambda s = 0$$
$$\frac{\partial \mathcal{L}}{\partial \lambda} = 2x_1 - x_2 - 4 - s^2 = 0$$

สมการที่ 3 นำเสนอทางเลือก 2 ทาง ได้แก่ $\lambda = 0$ หรือ $s = 0$

กรณีที่ให้ $\lambda = 0$

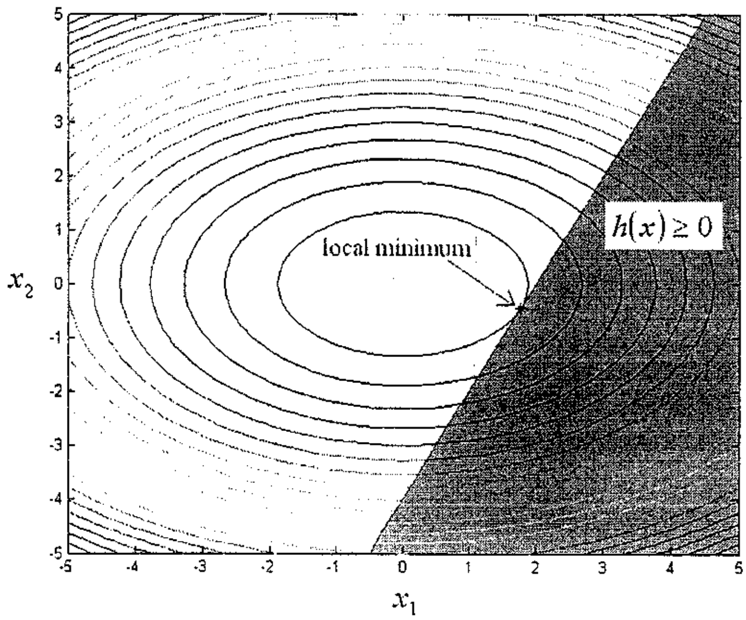
จะได้ $x_1 = 0$ และ $x_2 = 0$ แต่จะไม่สามารถหาค่า s ที่เป็นจำนวนจริงได้ ดังนั้น คำตอบนี้เป็นไปไม่ได้ (infeasible solution) จุดนี้ เป็นจุดต่ำสุดของปัญหานี้แบบไม่มีเงื่อนไขนั่นเอง

กรณีที่ให้ $s = 0$

สมการจะเหลือเพียง $\boldsymbol{h}(\boldsymbol{x}) \geq 0$ นั่นคือ

$$x_1 + 4x_2 = 0$$
$$2x_1 - x_2 = 4$$
$$\therefore x_* = \begin{bmatrix} \frac{16}{9} \\ -\frac{4}{9} \end{bmatrix} \quad \Rightarrow \quad \lambda_* = \frac{16}{9}$$

จะพบว่า จุดคำตอบนี้เป็นต่ำสุดเฉพาะถิ่น (ให้ตรวจสอบโดยคำนวณเมตริกซ์เฮสเซียนของฟังก์ชันลากรองจ์ที่จุดนี้ ต้องเป็น psdf)



รูปที่ 5.3 คอนทัวร์ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับสำหรับตัวอย่างที่ 5.3

- เงื่อนไขเสริมสำหรับความเป็นสแล็ก (complementary slackness condition) จะใช้การตรวจสอบเงื่อนไขบังคับแบบอสมการว่าเป็นเงื่อนไขแบบแอกทีฟหรือไม่แอกทีฟ (active constraint/inactive constraint) จะได้ว่า

Minimize $f(x)$
Subject to $h(x) = Ax - b \geq 0$, (linear inequality constraints)

เขียนฟังก์ชันลากรองจ์ได้ดังนี้

$$L(x,\lambda) = f(x) - \lambda^T[Ax - b] \tag{5.11}$$

การแก้ปัญหาจะใช้เงื่อนไขของการกำหนดให้เงื่อนไขเป็นแบบแอกทีฟหรือแบบไม่แอกทีฟ หมายความว่า เงื่อนไขบังคับแบบอสมการ $h(x) \geq 0$ ใด ๆ จะแอกทีฟก็ต่อเมื่อ $h(x) = 0$ และจะไม่แอกทีฟก็ต่อเมื่อ $h(x) > 0$ ผลดังกล่าวมีความสำคัญต่อการแก้ปัญหาดังนี้คือ

ถ้า $h(x)$ แอกทีฟ หมายความว่า $h(x)$ มีค่าเป็นศูนย์ ดังนั้น จากเงื่อนไขดังกล่าว ค่าตัวคูณลากรองจ์ไม่จำเป็นต้องเป็นศูนย์ และด้วยหลักการเดียวกัน กรณีเงื่อนไขไม่แอกทีฟ $h(x) > 0$ หมายความว่า ตัวคูณลากรองจะต้องมีค่าเป็นศูนย์นั่นเอง

ตัวอย่างที่ 5.4 จงหาค่าเหมาะที่สุดของปัญหาต่อไปนี้

$$\begin{aligned} \text{Minimize} \quad & f(x) = x_1^2 + x_2^2 \\ \text{subject to} \quad & h_1(x) = 2x_1 + x_2 \geq 4 \\ & h_2(x) = x_1 - x_2 \geq 0 \end{aligned}$$

จะได้

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = x_1^2 + x_2^2 - \lambda_1(2x_1 + x_2 - 4) - \lambda_2(x_1 - x_2)$$

จากเงื่อนไขที่จำเป็น $\nabla\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{s}^*, \boldsymbol{\lambda}^*) = 0$ จะได้

$$\frac{\partial\mathcal{L}}{\partial x_1} = 2x_1 - 2\lambda_1 - \lambda_2 = 0$$

$$\frac{\partial\mathcal{L}}{\partial x_2} = 2x_2 - \lambda_1 + \lambda_2 = 0$$

$$\frac{\partial\mathcal{L}}{\partial \lambda_1} = 2x_1 + x_2 - 4 = 0$$

$$\frac{\partial\mathcal{L}}{\partial \lambda_2} = x_1 - x_2 = 0$$

ให้ $\boldsymbol{h}_1(\boldsymbol{x})$ แอกทีฟ, $\boldsymbol{h}_2(\boldsymbol{x})$ ไม่แอกทีฟ นั่นคือ $\lambda_2 = 0$ และ $\boldsymbol{h}_1(\boldsymbol{x}) = 0$ จะได้

$$\frac{\partial\mathcal{L}}{\partial x_1} = 2x_1 - 2\lambda_1 = 0 \qquad\qquad x_1 = \frac{8}{5}$$

$$\frac{\partial\mathcal{L}}{\partial x_2} = 2x_2 - \lambda_1 = 0 \quad \text{จะได้} \quad x_2 = \frac{4}{5} \quad \text{กำหนดให้เป็น } u_1$$

$$\frac{\partial\mathcal{L}}{\partial \lambda_1} = 2x_1 + x_2 - 4 = 0 \qquad\qquad \lambda_1 = \frac{8}{5}, \quad \lambda_2 = 0$$

ให้ $\boldsymbol{h}_1(\boldsymbol{x})$ ไม่แอกทีฟ, $\boldsymbol{h}_2(\boldsymbol{x})$ แอกทีฟ นั่นคือ $\lambda_1 = 0$ และ $\boldsymbol{h}_2(\boldsymbol{x}) = 0$ จะได้

$$\frac{\partial\mathcal{L}}{\partial x_1} = 2x_1 - \lambda_2 = 0 \qquad\qquad x_1 = 0$$

$$\frac{\partial\mathcal{L}}{\partial x_2} = 2x_2 + \lambda_2 = 0 \quad \text{จะได้} \quad x_2 = 0 \quad \text{กำหนดให้เป็น } u_2, \quad \lambda_1 = 0$$

$$\frac{\partial\mathcal{L}}{\partial \lambda_2} = x_1 - x_2 = 0 \qquad\qquad \lambda_2 = 0$$

ให้ $\boldsymbol{h}_1(\boldsymbol{x})$ แอกทีฟ, $\boldsymbol{h}_2(\boldsymbol{x})$ แอกทีฟ นั่นคือ $\boldsymbol{h}_2(\boldsymbol{x}) = 0$ และ $\boldsymbol{h}_2(\boldsymbol{x}) = 0$ จะได้

$$\frac{\partial\mathcal{L}}{\partial x_1} = 2x_1 - 2\lambda_1 - \lambda_2 = 0 \qquad\qquad x_1 = \frac{4}{3}$$

$$\frac{\partial\mathcal{L}}{\partial x_2} = 2x_2 - \lambda_1 + \lambda_2 = 0 \quad \text{จะได้} \quad x_2 = \frac{4}{3} \quad \text{กำหนดให้เป็น } u_3$$

$$\frac{\partial\mathcal{L}}{\partial \lambda_1} = 2x_1 + x_2 - 4 = 0 \qquad\qquad \lambda_1 = \frac{16}{9}$$

$$\frac{\partial\mathcal{L}}{\partial \lambda_2} = x_1 - x_2 = 0 \qquad\qquad \lambda_2 = -\frac{8}{9}$$

ให้ $\boldsymbol{h}_1(x)$ ไม่แอกทีฟ, $\boldsymbol{h}_2(x)$ ไม่แอกทีฟ นั่นคือ $\lambda_1 = 0$ และ $\lambda_2 = 0$ จะได้

$$\frac{\partial \mathcal{L}}{\partial x_1} = 2x_1 = 0 \qquad \frac{\partial \mathcal{L}}{\partial x_2} = 2x_2 = 0$$

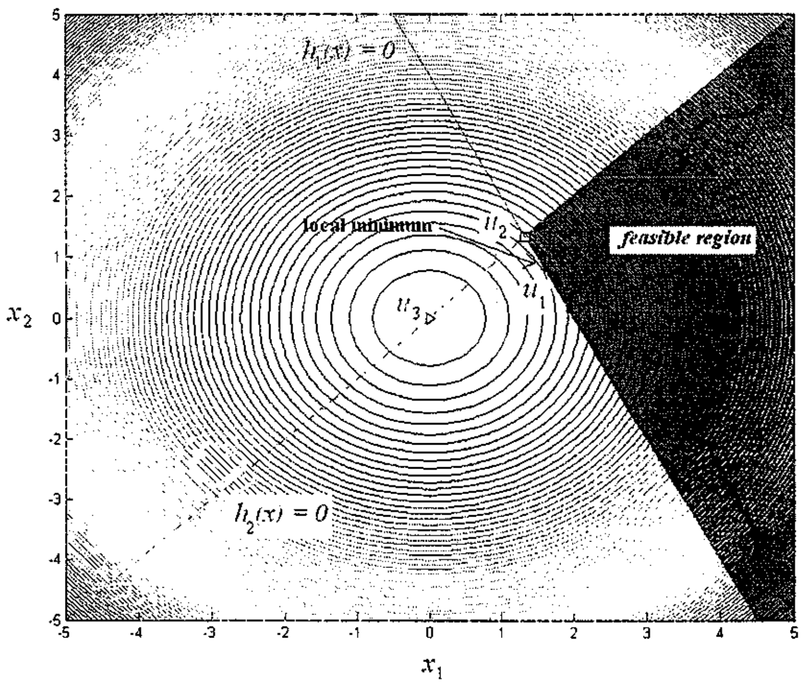
จะได้

$$x_1 = 0,\quad x_2 = 0,\quad \lambda_1 = 0,\quad \lambda_2 = 0$$

มีค่าเท่ากับ u_2

จะได้จุดหยุดนิ่ง (stationary point) ทั้งสิ้น 3 จุด โดย u_3 จะเป็นจุดต่ำสุด (<u>*ให้หาจุดต่ำสุดโดยการตรวจสอบเมตริกซ์เฮสเซียนของฟังก์ชันลากรองจ์ที่จุดนี้ ต้องเป็น psdf*</u>)

พิจารณาได้จากกราฟในรูปต่อไปนี้



รูปที่ 5.4 คอนทัวร์ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับสำหรับตัวอย่างที่ 5.4

5.5 ปัญหาเงื่อนไขบังคับไม่เชิงเส้น (non-linear constrained problems)

มีรูปสมการทั่วไปเป็นดังนี้

Minimize	$f(\boldsymbol{x})$	
Subject to	$\boldsymbol{g}(\boldsymbol{x}) = 0$	(non-linear equality constraints)
	$\boldsymbol{h}(\boldsymbol{x}) \geq 0$	(non-linear inequality constraints)

โดยที่

f: $\Re^n \rightarrow \Re$

g: $\Re^n \rightarrow \Re^m$, $m \leq n$

h: $\Re^n \rightarrow \Re^w$,

เมื่อ n คือ มิติของปัญหา, m และ w คือ จำนวนเงื่อนไขบังคับสมการและอสมการตามลำดับ

สำหรับเงื่อนไขบังคับแบบอสมการ $h_j(x)$ ลำดับที่ j ใด ๆ ถ้า $h_j(x) = 0$ จะเรียกว่า เป็นเงื่อนไขบังคับแอกทีฟ (active constraint) ถ้า $h_j(x) > 0$ เรียกว่า เป็นเงื่อนไขบังคับไม่แอกทีฟ (inactive constraint)

- <u>ทฤษฎีบทของ คารัช-คูนห์-ทักเกอร์</u> (Karush-Kuhn-Tucker's theorem)

ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขใด ๆ ดังแสดงในรูปแบบข้างต้น สำหรับปัญหาในรูปค่าต่ำสุด ถ้ากำหนดให้ x^* เป็นจุดต่ำสุดวงแคบเฉพาะถิ่น จะมีเวกเตอร์ค่าจริง $\lambda \in \Re^m$ และ $\mu \in \Re^w$ ที่ทำให้

- $\mu_j \geq 0$ สำหรับ $j = 1, 2, \ldots, w$
- $\nabla f(x^*) - \lambda^T \nabla g(x^*) - \mu^T \nabla h(x^*) = 0$
- $\mu^T h(x^*) = 0$

λ คือ ตัวคูณลากรองจ์นั่นเอง ส่วน μ จะเรียกว่า ตัวคูณ<u>คารัช-คูนห์-ทักเกอร์</u> หรือตัวคูณ KKT (Karush-Kuhn-Tucker multiplier)

เงื่อนไขนี้ คือ เงื่อนไขที่จำเป็นอันดับหนึ่ง (first-order necessary condition) นั่นเอง แต่โดยทั่วไปสำหรับกรณีของปัญหาแบบมีเงื่อนไขบังคับนี้ นิยมเรียกว่า เงื่อนไขคารัช-คูนห์-ทักเกอร์ (Karush-Kuhn-Tucker condition หรือ KKT condition) นอกจากเงื่อนไขที่จำเป็นแล้ว ยังมีเงื่อนไขที่จำเป็นอันดับสอง (second-order necessary condition) และ เงื่อนไขที่เพียงพออันดับสอง (second-order sufficient condition) ดังนี้

เมื่อไม่สนใจว่าจุดต่ำสุดที่ได้จะเป็น strict local minimizer หรือ non-strict local minimizer จะได้เงื่อนไขที่จำเป็นอันดับ 2 ดังนี้

ถ้าให้ $F(x)$ เป็นเมตริกซ์เฮสเซียนของ $f(x)$, $G(x)$ เป็นเมตริกซ์เฮสเซียนของ $g(x)$, เป็นเมตริกซ์เฮสเซียนของ $h(x)$ และ $L(x,\lambda,\mu) = F(x) - \lambda^T G(x) - \mu^T H(x)$

ถ้ากำหนดให้ x^* เป็นจุดต่ำสุดเฉพาะถิ่น ซึ่งสอดคล้องกับ เงื่อนไขบังคับ KKT อันดับหนึ่ง ,ล้ว $L(x^*,\lambda^*,\mu^*)$ ต้องเป็น psdf

โดยปกติแล้ว การแก้ปัญหาโดยใช้การตรวจสอบเงื่อนไข KKT ทำได้ยากในทางปฏิบัติ ทำให้การเขียนปัญหาแบบมีเงื่อนไขให้อยู่ในรูปปัญหาแบบไม่มีเงื่อนไข โดยอาศัยตัวคูณลากรองจ์และตัวคูณ KKT ได้รับความนิยมมากกว่า ดังนั้น ฟังก์ชันลากรองจ์ซึ่งแทนปัญหาแบบไม่มีเงื่อนไขบังคับและจะใช้การแก้ปัญหาแบบไม่มีเงื่อนไขเพื่อค้นหาคำตอบ มีค่าดังนี้

$$L(x,\lambda,\mu) = f(x) - \lambda^T g(x) - \mu^T h(x) \tag{5.12}$$

ตัวอย่างที่ 5.5 จงใช้เงื่อนไข KKT เพื่อแก้ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับต่อไปนี้

$$\begin{aligned} \text{Minimize} \quad & f(x) = (x_1 - 1)^2 + x_2 - 2 \\ \text{subject to} \quad & x_2 - x_1 - 1 = 0 \\ & x_1 + x_2 - 2 \leq 0 \end{aligned}$$

เขียนในรูปฟังก์ชันลากรองจ์ได้ดังนี้

$$\mathcal{L}(x,\lambda)=(x_1-1)^2+x_2-\lambda(x_2-x_1-1)-\mu(2-x_1-x_2)$$

ด้วยเงื่อนไข 1st-order KKT

$$2(x_1-1)-\lambda+\mu=0$$
$$1+\lambda+\mu=0$$
$$\mu(x_1+x_2-2)=0$$
$$\mu\geq 0$$
$$x_2-x_1-1=0$$
$$x_1+x_2-2\leq 0$$

เริ่มต้นแก้ปัญหาโดยกำหนดให้ $\mu>0$ ดังนั้น $x_1+x_2-2=0$ จะได้สมการดังนี้

$x_1+x_2-2=0$ และ $x_2-x_1-1=0$ จะได้ $\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}1/2\\3/2\end{bmatrix}$

แทนค่าในสมการที่เหลือจะได้

$$2\mu+2x_1-1=2\mu+2\times\frac{1}{2}-1=2\mu=0 \quad\Rightarrow\quad \mu=0$$

จะพบว่า จากเงื่อนไขที่กำหนดไว้ว่า $\mu>0$ ดังนั้น ในกรณีนี้จะขัดแย้งกับสมมติฐานข้างต้น นั่นคือ เมื่อ $\mu=0$ จะได้ $\lambda=-1$ แก้ชุดสมการที่เหลือ จะได้

$$2(x_1-1)-\lambda+\mu=2(x_1-1)-(-1)+0=2x_1-1=0 \quad\Rightarrow\quad x_1=\frac{1}{2}$$

$$x_2-x_1-1=x_2-\frac{1}{2}-1=x_2-\frac{3}{2} \quad\Rightarrow\quad x_2=\frac{3}{2}$$

ดังนั้น จะได้จุดคำตอบดังนี้

$$\begin{bmatrix}x_1\\x_2\\\lambda\\\mu\end{bmatrix}^*=\begin{bmatrix}1/2\\3/2\\-1\\0\end{bmatrix}$$

ตรวจสอบการเป็นจุดต่ำสุดด้วย 2nd-order KKT condition

$$\boldsymbol{L}(\boldsymbol{x},\lambda,\boldsymbol{\mu})=\boldsymbol{F}(\boldsymbol{x})+\lambda\boldsymbol{G}(\boldsymbol{x})+\mu\boldsymbol{H}(\boldsymbol{x})=\begin{bmatrix}2&0\\0&0\end{bmatrix}-\lambda\begin{bmatrix}0&0\\0&0\end{bmatrix}-\mu\begin{bmatrix}0&0\\0&0\end{bmatrix}=\begin{bmatrix}2&0\\0&0\end{bmatrix}$$

จะพบว่าเมตริกซ์ดังกล่าวปืน *psdf* ดังนั้น จุดคำตอบที่ได้เป็นจุดต่ำสุดวงแคบเฉพาะถิ่น

5.6 ระเบียบวิธีจุดเป็นไปได้ (Feasible Point Methods)

สำหรับการแก้ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับนี้ จะอาศัยหลักการรักษาการปรับปรุงจุดคำตอบในแต่ละรอบให้เป็นจุดคำตอบที่เป็นไปได้ (feasible solution) โดยจุดคำตอบใด ๆ จะเรียกว่า จุดคำตอบที่เป็นไปได้ ก็ต่อเมื่อ จุดคำตอบนั้น ๆ ทำให้เงื่อนไขบังคับทุกเงื่อนไขเป็นจริง เพื่อให้ง่ายในการศึกษา จะเริ่มต้นด้วยปัญหาที่มีเฉพาะเงื่อนไขบังคับสมการเท่านั้น ดังนี้

Minimize $\quad f(\boldsymbol{x})$
Subject to $\quad g(\boldsymbol{x})=\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}=0$

เริ่มต้นพิจารณาปัญหาด้วยจุดคำตอบที่เป็นไปได้ $\bar{x}$ ใด ๆ จุดคำตอบที่เป็นไปได้อื่น ๆ สามารถคำนวณได้จากการใช้ทิศทางการค้นหาที่เป็นไปได้ (feasible direction: $\boldsymbol{p}$) ดังที่ได้กล่าวไว้ในหัวข้อที่ผ่านมา นั่นคือ คำตอบที่เหมาะที่สุดจะอยู่ในรูปของความสัมพันธ์

$$x^* = \bar{x} + p \tag{5.13}$$

ปัญหานี้สามารถแปลงให้อยู่ในรูปของปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขได้ในรูปของฟังก์ชันลดทอน นั่นคือ

$$\begin{aligned} &\text{Minimize} \quad \phi(v) = f(x) = f(\bar{x} + Zv) \\ &v \in \Re^{n-m} \end{aligned} \tag{5.14}$$

โดยทั่วไปแล้ว การแก้ปัญหานี้สามารถใช้ระเบียบวิธีการแก้ปัญหาต่าง ๆ สำหรับปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขได้ เช่น ระเบียบวิธีเกรเดียน (gradient methods) ระเบียบวิธีนิวตัน (Newton's methods) หรือระเบียบวิธีคล้ายนิวตัน (Quasi-Newton methods) เป็นต้น การแก้ปัญหาจะใช้การประมาณค่าฟังก์ชันลดทอนด้วยอนุกรมเทย์เลอร์ นั่นคือ

$$\phi(v) = f(\bar{x} + Zv) = f(\bar{x}) + v^T Z^T \nabla f(\bar{x}) + \tfrac{1}{2} v^T Z^T \nabla f(\bar{x}) Zv + \cdots$$

ถ้าใช้การแก้ปัญหาด้วยระเบียบวิธีของนิวตัน ทิศทางการค้นหาของนิวตันจะถูกนำมาใช้โดยการประมาณฟังก์ชันลดทอนด้วยแบบจำลองกำลังสอง จะได้ทิศทางการค้นหาเป็น

$$p = -Z\left(Z^T \nabla^2 f(\bar{x}) Z\right)^{-1} Z^T \nabla f(\bar{x}) \tag{5.15}$$

ทิศทางการค้นหานี้เรียกว่า ทิศทางการค้นหานิวตันลดทอน (reduced Newton direction) ที่จุด $\bar{x}$ และเรียกระเบียบวิธีนี้ว่า ระเบียบวิธีนิวตันลดทอน (reduced Newton's method) พิจารณาจากตัวอย่างต่อไปนี้

ตัวอย่างที่ 5.6 จงหาค่าเหมาะที่สุดโดยใช้ระเบียบวิธีนิวตันลดทอน

$$\begin{aligned} \text{Minimize} \quad & f(x) = x_1^2 + x_2^2 + x_3^2 \\ \text{subject to} \quad & x_1 - x_2 - x_3 = 0 \\ & x_1 + x_3 = 4 \end{aligned}$$

จะพบว่า จากมิติของปัญหาและมิติของเงื่อนไขบังคับแบบสมการ จะได้

$$x = \begin{bmatrix} x_B \\ x_N \end{bmatrix} \quad \Rightarrow \quad x_B = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, x_N = [x_3]$$

$$\therefore \bar{x} = \begin{bmatrix} -B^{-1}b \\ 0 \end{bmatrix}; \quad A = \begin{bmatrix} 1 & -1 & -1 \\ 1 & 0 & 1 \end{bmatrix}; b = \begin{bmatrix} 0 \\ 4 \end{bmatrix} \quad \Rightarrow B = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}, N = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

เลือกจุดคำตอบที่เป็นไปได้เริ่มต้น

$$x^{(0)} = \bar{x} = \begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix} = \begin{bmatrix} -\begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 4 \end{bmatrix} \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 0 \end{bmatrix}$$

$$Z = \begin{bmatrix} -B^{-1}N \\ I \end{bmatrix} = \begin{bmatrix} -\begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ 1 \end{bmatrix}$$

คำนวณเกรเดียนต์และเมตริกซ์เฮสเซียนได้ดังนี้

$$\nabla f(x) = \begin{bmatrix} 2x_1 & 2x_2 & 2x_3 \end{bmatrix}^T$$

$$\nabla^2 f(x) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$$x^{(0)} = \begin{bmatrix} 4 \\ 4 \\ 0 \end{bmatrix} \qquad \Rightarrow \nabla\phi\left(v^{(0)}\right) = Z^T \nabla f\left(x^{(0)}\right) = \begin{bmatrix} -1 & -2 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 8 \\ 0 \end{bmatrix} = -24$$

เกรเดียนต์ของฟังก์ชันลดทอนมีค่าไม่เท่ากับศูนย์ ดังนั้น เริ่มการคำนวณรอบที่ 1 จากจุดเริ่มต้นที่กำหนด

<u>รอบการคำนวณที่ 1</u>:

คำนวณทิศทางนิวตันลดทอน

$$p^{(0)} = -Z\left(Z^T \nabla^2 f\left(x^{(0)}\right) Z\right)^{-1} Z^T \nabla f\left(x^{(0)}\right) = \begin{bmatrix} -2 \\ -4 \\ 2 \end{bmatrix}$$

$$\therefore x^{(1)} = x^{(0)} + p^{(0)} = \begin{bmatrix} 4 \\ 4 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 \\ -4 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix}$$

ตรวจสอบเกรเดียนต์ของฟังก์ชันลดทอนในรอบการคำนวณที่ 1 จะได้

$$x^{(1)} = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix} \qquad \Rightarrow \nabla\phi\left(v^{(1)}\right) = \begin{bmatrix} -1 & -2 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 0 \\ 4 \end{bmatrix}$$

$$= (-1)(4) + (-2)(0) + (1)(4) = 0$$

เกรเดียนต์ของฟังก์ชันลดทอนมีค่าเท่ากับศูนย์ ดังนั้น ได้จุดคำตอบของปัญหา

ตรวจสอบ reduced Hessian ว่าเป็น *psdf* หรือไม่

$$\nabla^2\phi\left(v^{(1)}\right) = Z^T \nabla^2 f\left(x^{(1)}\right) Z = \begin{bmatrix} -1 & -2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ -2 \\ 1 \end{bmatrix} = 12 > 0 \quad (pdf)$$

ดังนั้น

$x^{(1)} = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix}$ เป็นจุดต่ำสุดเฉพาะถิ่น ที่ให้ค่าต่ำสุดเท่ากับ 8

จากตัวอย่างนี้ ปัญหาที่ได้อยู่ในรูปของฟังก์ชันกำลังสอง การใช้ระเบียบวิธีของนิวตันจะช่วยให้ค้นหาคำตอบได้ภายใน 1 รอบการค้นหาเท่านั้น ถ้าเมตริกซ์เฮสเซียนเป็น *pdf* แต่ไม่สามารถรับประกันได้ว่าเฮสเซียนของฟังก์ชันใด ๆ หรือแม้แต่ฟังก์ชันที่เขียนในรูปกำลังสองได้จะเป็น *pdf* เสมอไป ส่งผลให้ระเบียบวิธีนิวตันลดทอนไม่ได้รับความนิยมในการแก้ปัญหา นอกจากจะพิสูจน์ได้ว่า ปัญหาที่พิจารณาให้ค่าเฮสเซียนเป็น *pdf* ในทุก ๆ รอบการค้นหา ดังนั้น เพื่อให้การแก้ปัญหาสามารถรับประกันจุดคำตอบได้ว่าเมื่อสิ้นสุดการค้นหาแล้ว จุดคำตอบที่ได้จะเป็นจุดต่ำสุดเฉพาะถิ่นจุดหนึ่งของปัญหา วิธีการดังกล่าวนี้จะใช้การคำนวณเกรเดียนต์ของฟังก์ชันลดทอนเป็นทิศทางการค้นหา เรียกว่า ทิศทางเกรเดียนต์ลดทอน (reduced gradient direction) หรือทิศทางชันที่สุดที่ถูกลดทอน (reduced steepest descent direction) นั่นเอง ดังต่อไปนี้

5.7 ระเบียบวิธีเกรเดียนต์ลดทอน (Reduced Gradient Methods)

โดยปราศจากการนำเสนอบทพิสูจน์ใด ๆ ในที่นี้ ระเบียบวิธีเกรเดียนต์ลดทอนจะใช้การคำนวณเกรเดียนต์ของฟังก์ชันลดทอนในการกำหนดทิศทางการค้นหา ซึ่งจุดคำตอบทุกจุดที่เกิดขึ้นจากระเบียบวิธีนี้เป็นจุดคำตอบที่สอดคล้องกับเงื่อนไขบังคับเสมอ ถ้าพิจารณาปัญหาที่มีเงื่อนไขบังคับสมการเท่านั้น จะได้ทิศทางการค้นหาและจุดคำตอบที่ถูกปรับปรุงดังนี้

$$\boldsymbol{p}_k = -\boldsymbol{Z}\boldsymbol{Z}^T\nabla f\left(\boldsymbol{x}_k\right) \quad \Rightarrow \quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_{k+1}\boldsymbol{p}_{k+1} \tag{5.16}$$

ทิศทางการค้นหานี้เรียกว่า ทิศทางการค้นหาเกรเดียนต์ลดทอน (reduced gradient direction) และเรียกระเบียบวิธีนี้ว่า ระเบียบวิธีเกรเดียนต์ลดทอน (reduced gradient method) พิจารณาจากตัวอย่างต่อไปนี้

ตัวอย่างที่ 5.7 จงหาค่าเหมาะที่สุดโดยใช้ระเบียบวิธีเกรเดียนต์ลดทอน

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) = x_1^2 + x_2^2 + x_3^2 \\ \text{subject to} \quad & x_1 - x_2 - x_3 = 0 \\ & x_1 + x_3 = 4 \end{aligned}$$

การคำนวณเมตริกซ์ประกอบต่าง ๆ ที่จำเป็นให้ดูในตัวอย่างที่ 5.6 ในตัวอย่างนี้จะนำเสนอรายละเอียดเฉพาะส่วนที่เป็นการคำนวณที่ไม่ปรากฏในระเบียบวิธีนิวตันลดทอน จะได้

$$\boldsymbol{x}^{(0)} = \begin{bmatrix} 4 \\ 4 \\ 0 \end{bmatrix}, \qquad \nabla\phi\left(\boldsymbol{x}^{(0)}\right) = -24 \neq 0$$

<u>คำนวณรอบที่ 1</u>

$$p_1 = \begin{bmatrix} -24 \\ -48 \\ 24 \end{bmatrix}$$

แก้ปัญหาการค้นหาตามเส้นโดยใช้ระเบียบวิธีคิวบิก จะได้ $\lambda_1 = 0.0833$ นั่นคือ

$$x^{(1)} = \begin{bmatrix} 2.0012 \\ 0.0024 \\ 1.9988 \end{bmatrix}, \qquad \nabla\phi\left(x^{(1)}\right) = -0.0144 \neq 0$$

<u>คำนวณรอบที่ 2</u>

$$p_2 = \begin{bmatrix} -0.0144 \\ -0.0288 \\ 0.0144 \end{bmatrix}$$

แก้ปัญหาการค้นหาตามเส้นโดยใช้ระเบียบวิธีคิวบิก จะได้ $\lambda_2 = 0.0833$ นั่นคือ

$$x^{(2)} = \begin{bmatrix} 2.0000 \\ 0.0000 \\ 2.0000 \end{bmatrix}, \qquad \nabla\phi\left(x^{(2)}\right) = -8.6261\times10^{-6} \approx 0$$

เงื่อนไขการหยุดจะขึ้นกับค่าเกรเดียนต์ลดทอนว่ามีค่าเข้าใกล้ศูนย์เพียงพอหรือยัง จากการตรวจสอบความคลาดเคลื่อน ดังนี้

$$\left\|x_{k+1} - x_k\right\| < \varepsilon_1 = 10^{-8}, \qquad \left\|Z^T \nabla f\left(x_k\right)\right\| < \varepsilon_2 = 10^{-8}$$

เขียนโปรแกรมเพื่อแก้ปัญหาจะได้ผลการคำนวณเป็น

k	$x_1(k)$	$x_2(k)$	$x_3(k)$	$f(x_1,x_2)$
0	4.0000	4.0000	0.0000	32.0000
1	2.0012	0.0024	1.9988	8.0000
2	2.0000	0.0000	2.0000	8.0000

ตัวอย่างที่ 5.8 จงหาค่าเหมาะที่สุดโดยใช้ระเบียบวิธีเกรเดียนต์ลดทอน

$$\begin{aligned} \text{Minimize} \quad & f(x) = x_1^2 + x_2^2 + x_3^2 \\ \text{subject to} \quad & x_1 - x_2 - x_3 = 0 \\ & x_1 + x_3 \geq 4 \end{aligned}$$

เมื่อแก้ปัญหาที่มีเงื่อนไขบังคับอสมการเชิงเส้นรวมกับเงื่อนไขบังคับแบบสมการเชิงเส้นรูปแบบการใช้ตัวแปรสแล็ก ทำให้การแก้ปัญหาไม่ยุ่งยากมากนัก เมื่อกำหนดให้ตัวแปร x_4 เป็นตัวแปรสแล็ก จะได้ว่า

$$\begin{aligned} \text{Minimize} \quad & f(x) = x_1^2 + x_2^2 + x_3^2 \\ \text{subject to} \quad & x_1 - x_2 - x_3 = 0 \\ & x_1 + x_3 + x_4 = 4 \end{aligned}$$

จากการใช้โปรแกรมคอมพิวเตอร์ช่วยแก้ปัญหาจะได้ผลการคำนวณในแต่ละรอบเป็นดังนี้ โดยให้เงื่อนไขการหยุดเกิดขึ้นเมื่อความคลาดเคลื่อนมีค่าน้อยกว่า 10^{-8}

Iteration	x_1	x_2	x_3	x_4	$f(x)$
0	4.00000000	4.00000000	0.00000000	0.00000000	32.0000000
1	1.34890000	-0.2417000	1.59060000	1.06040000	4.40820000
2	0.50097000	0.51929000	-0.0183130	3.51730000	0.52097000
3	0.15973000	-0.0281620	0.1878900	3.65240000	0.06160800
4	0.05270100	0.05687400	-0.0041733	3.95150000	0.00602950
5	0.01566600	-0.0027124	0.01837900	3.96600000	0.00059056
6	0.00444430	0.00502370	-0.00057947	3.99610000	0.00004532
7	0.00120540	-0.0002043	0.00140970	3.99740000	0.00000348
8	0.00027866	0.00033284	-0.00005418	3.99980000	0.00000019
9	0.00006645	-0.00001098	0.00007743	3.99990000	0.00000001
10	0.00001129	0.00001445	-0.00000316	4.00000000	0.00000000

ระเบียบวิธีนี้มีข้อดี คือ รับประกันการลู่เข้าของจุดคำตอบ ถ้าสามารถปรับปรุงจุดคำตอบได้ในทุกรอบการคำนวณ แต่ข้อเสีย คือ การปรับปรุงจุดคำตอบเพื่อให้ได้จุดคำตอบที่เป็นไปได้นั้นส่วนใหญ่ใช้การคำนวณวนรอบซ้อนเข้าไปอีกชั้นหนึ่ง ทำให้เสียเวลาคำนวณยาวนาน การแก้ปัญหาที่กำหนดไว้นี้ อยู่ภายใต้ข้อสมมติที่ให้เงื่อนไขบังคับอยู่ในรูปฟังก์ชันเชิงเส้น ดังนั้นเพื่อให้ได้ระเบียบวิธีที่สามารถใช้งานได้ทั่วไปกับเงื่อนไขบังคับไม่เชิงเส้น ทำให้เกิดการพัฒนาระเบียบวิธีเกรเดียนต์ลดทอนทั่วไป (generalized reduced gradient method: GRG method) ดังต่อไปนี้

$$\begin{aligned} &\text{Minimize} && f(\boldsymbol{x}) \\ &\text{Subject to} && \boldsymbol{g}(\boldsymbol{x}) = 0 \end{aligned}$$

ถ้ากำหนดให้ฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับสมการเป็นแบบไม่เชิงเส้น ให้ m เป็นสมการเงื่อนไขบังคับสมการ และ n เป็นจำนวนตัวแปรทั้งหมด โดยที่ $m \le n$ ตัวแปรจำนวน n ตัวถูกแบ่งเป็น 2 กลุ่ม โดยที่ $\boldsymbol{x} = [\boldsymbol{y} \;\; \boldsymbol{z}]^T$ จะได้

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_{k+1}\boldsymbol{p}_{k+1} \;\Rightarrow\; \begin{bmatrix} \boldsymbol{y}_{k+1} \\ \boldsymbol{z}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}_k \\ \boldsymbol{z}_k \end{bmatrix} + \lambda_{k+1}\begin{bmatrix} \boldsymbol{q}_{k+1} \\ \boldsymbol{r}_{k+1} \end{bmatrix} \tag{5.17}$$

นั่นคือ

$$f(\boldsymbol{x}_{k+1}) \approx f(\boldsymbol{x}_k) + \nabla^T f(\boldsymbol{x}_k)\,\boldsymbol{p}_{k+1} = f(\boldsymbol{y}_k, \boldsymbol{z}_k) + \nabla^T f_y(\boldsymbol{y}_k, \boldsymbol{z}_k)\boldsymbol{q}_{k+1} + \nabla^T f_z(\boldsymbol{y}_k, \boldsymbol{z}_k)\boldsymbol{r}_{k+1}$$

$$\boldsymbol{g}(\boldsymbol{x}_{k+1}) \approx \boldsymbol{g}(\boldsymbol{x}_k) + \nabla^T \boldsymbol{g}(\boldsymbol{x}_k)\,\boldsymbol{p}_{k+1} = \boldsymbol{g}(\boldsymbol{y}_k, \boldsymbol{z}_k) + \nabla^T \boldsymbol{g}_y(\boldsymbol{y}_k, \boldsymbol{z}_k)\boldsymbol{q}_{k+1} + \nabla^T \boldsymbol{g}_z(\boldsymbol{y}_k, \boldsymbol{z}_k)\boldsymbol{r}_{k+1} = 0$$

จะได้

$$f(\boldsymbol{y}_{k+1}, \boldsymbol{z}_{k+1}) = f(\boldsymbol{y}_k, \boldsymbol{z}_k) + \nabla^T f_y(\boldsymbol{y}_k, \boldsymbol{z}_k)\boldsymbol{q}_{k+1} + \nabla^T f_z(\boldsymbol{y}_k, \boldsymbol{z}_k)\boldsymbol{r}_{k+1}$$

$$\boldsymbol{g}(\boldsymbol{y}_{k+1}, \boldsymbol{z}_{k+1}) = \boldsymbol{g}(\boldsymbol{y}_k, \boldsymbol{z}_k) + \nabla^T \boldsymbol{g}_y(\boldsymbol{y}_k, \boldsymbol{z}_k)\boldsymbol{q}_{k+1} + \nabla^T \boldsymbol{g}_z(\boldsymbol{y}_k, \boldsymbol{z}_k)\boldsymbol{r}_{k+1} = 0$$

ถ้าเริ่มต้นจากจุดคำตอบที่เป็นไปได้ นั่นคือ $\boldsymbol{g}(\boldsymbol{x}_k) = 0$ จะได้ว่า

$$q_{k+1} = -\left[\nabla^T g_y(x_k)\right]^{-1} \nabla^T g_z(x_k) r_{k+1} \tag{5.18}$$

แทนค่ากลับในฟังก์ชันวัตถุประสงค์ที่ถูกประมาณค่าด้วยอนุกรมเทย์เลอร์

$$\therefore f(y_{k+1}, z_{k+1}) = f(y_k, z_k) - \nabla^T f_y(x_k)\left[\nabla^T g_y(x_k)\right]^{-1} \nabla^T g_z(x_k) r_{k+1} + \nabla^T f_z(x_k) r_{k+1}$$

$$= f(y_k, z_k) + \left\{\nabla^T f_z(x_k) - \nabla^T f_y(x_k)\left[\nabla^T g_y(x_k)\right]^{-1} \nabla^T g_z(x_k)\right\} r_{k+1}$$

นั่นคือ ถ้าให้

$$s_{k+1} = \nabla^T f_z(x_k) - \nabla^T f_y(x_k)\left[\nabla^T g_y(x_k)\right]^{-1} \nabla^T g_z(x_k) \tag{5.19}$$

เป็นทิศทางการค้นหาเกรเดียนต์ลดทอนทั่วไปของฟังก์ชันวัตถุประสงค์ที่ถูกลดทอน และเพื่อรับประกันทิศทางลาดลง จะให้ $r_{k+1} = -\lambda_{k+1} s_{k+1}$ อย่างไรก็ตาม ต้องทำการหาค่าช่วงก้าวที่เหมาะสม λ_{k+1} โดยการแก้ปัญหาย่อยในรูปแบบของการค้นหาตามเส้นจะได้

Minimize $\quad F(\lambda) = f(x_k + \lambda_{k+1} p_{k+1})$

Subject to $\quad g(x_k + \lambda_{k+1} p_{k+1}) = 0$

ถ้าเริ่มต้นด้วย

i) $\lambda_{k+1} = 0$:

$$r'_{k+1} = -\lambda_{k+1} s_{k+1} = 0$$

$$q'_{k+1} = -\left[\nabla^T g_y(x_k)\right]^{-1} \nabla^T g_z(x_k) r_{k+1} = 0$$

$$F(0) = f(x_k)$$

ii) $\lambda_{k+1} = a$: ให้เลือกค่า a ตามความเหมาะสม

$$r''_{k+1} = -\lambda_{k+1} s_{k+1} = -a\, s_{k+1}$$

$$q''_{k+1} = -\left[\nabla^T g_y(x_k)\right]^{-1} \nabla^T g_z(x_k) r''_{k+1} = 0$$

$$x''_{k+1} = x_k + \begin{bmatrix} q''_{k+1} \\ r''_{k+1} \end{bmatrix}$$

ก. ตรวจสอบเงื่อนไขบังคับสมการ

- ถ้า $\quad g(x''_{k+1}) = 0 \rightarrow \|g(x''_{k+1})\| \le \varepsilon$, หยุดการคำนวณ ไปข้อ ค.
- ถ้า $\quad g(x''_{k+1}) \ne 0 \rightarrow \|g(x''_{k+1})\| > \varepsilon$, ให้ปรับปรุง q''_{k+1}, x''_{k+1} ดังนี้

$$q''_{k+1} = q''_{k+1} - \left[\nabla^T g_y(x_k)\right]^{-1} g(x''_{k+1})$$

$$x''_{k+1} = x_k + \begin{bmatrix} q''_{k+1} \\ r''_{k+1} \end{bmatrix}$$

; ทำซ้ำข้อ ก.

ค. ให้คำนวณ

$$F(a) = f(x''_{k+1})$$

iii) $\lambda_{k+1} = b$: ให้เลือกค่า b ตามความเหมาะสม โดยที่ $a \neq b$ ทำซ้ำข้อ ii) โดยใช้ค่า b แทน a เมื่อการคำนวณลู่เข้า จะได้ $F(b) = f(x''_{k+1})$

ในที่นี้จะใช้คำนวณหาช่วงก้าวที่เหมาะสมโดยวิธีการประมาณค่ากำลังสอง (quadratic interpolation) นั่นคือ

$$F(\lambda) = \alpha + \beta\lambda + \gamma\lambda^2$$

จะได้

$$\left.\begin{aligned} F(0) &= \alpha \\ F(a) &= \alpha + \beta a + \gamma a^2 \\ F(b) &= \alpha + \beta b + \gamma b^2 \end{aligned}\right\rangle \qquad \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & a & a^2 \\ 1 & b & b^2 \end{bmatrix}^{-1} \begin{bmatrix} F(0) \\ F(a) \\ F(b) \end{bmatrix}$$

หาจุดต่ำสุดของฟังก์ชันกำลังสองได้

$$\frac{d}{d\lambda} F(\lambda^*_{k+1}) = 0 \qquad \Rightarrow \qquad \lambda^*_{k+1} = -\frac{\beta}{2\gamma} \tag{5.20}$$

ที่ค่าช่วงก้าวที่เหมาะสมต้องตรวจสอบเงื่อนไขบังคับสมการเช่นกัน และต้องปรับปรุง q''_{k+1}, x''_{k+1} จนกระทั่งเงื่อนไขบังคับสมการเป็นจริง ซึ่งจะเป็นการคำนวณครั้งสุดท้ายของรอบการคำนวณนี้ การแก้ปัญหาการค้นหาตามเส้นนี้เป็นแบบไม่แม่นตรง (inexact line search)

ตัวอย่างที่ 5.9 จงใช้ระเบียบวิธีเกรเดียนต์ลดทอนทั่วไปเพื่อแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้

Minimize $\quad f(\boldsymbol{x}) = x_1^4 - 2x_1^2 x_2 + x_1^2 + x_1 x_2^2 - 2x_1 + 4$

Subject to $\quad g(\boldsymbol{x}) = x_1^2 + x_2^2 - 2 = 0$

$\qquad\qquad\quad h(\boldsymbol{x}) = 0.25x_1^2 + 0.75x_2^2 \geq 0$

ใช้ตัวแปรสแลก x_3 แปลงให้เป็นปัญหาที่มีเฉพาะเงื่อนไขบังคับสมการได้ดังนี้

Minimize $\quad f(\boldsymbol{x}) = x_1^4 - 2x_1^2 x_2 + x_1^2 + x_1 x_2^2 - 2x_1 + 4$

Subject to $\quad g_1(\boldsymbol{x}) = x_1^2 + x_2^2 - 2 = 0$

$\qquad\qquad\quad g_2(\boldsymbol{x}) = 0.25x_1^2 + 0.75x_2^2 - x_3^2 = 0$

โดยการเลือกจุดเริ่มต้นที่เป็นไปได้ นั่นคือ

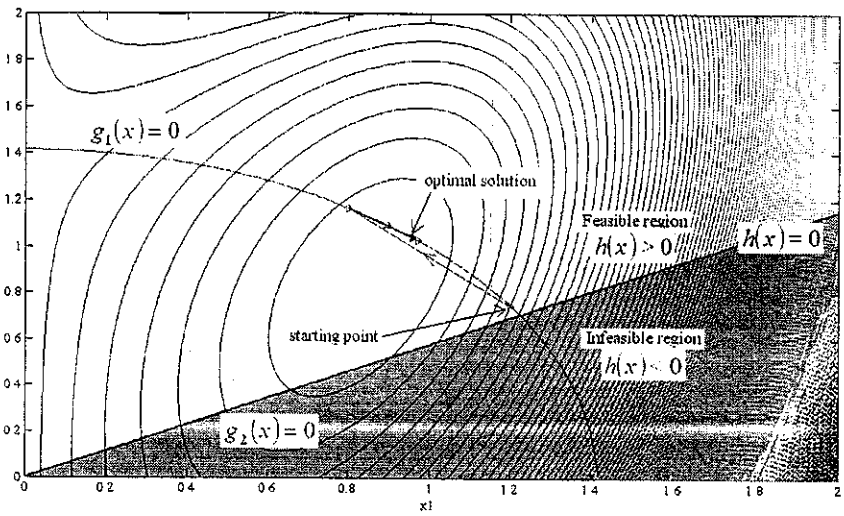
จุดที่ทำให้เงื่อนไขบังคับทุกชุดแอกทีฟ $\boldsymbol{x}_0 = [1.2000 \quad 0.7438 \quad 0.8803]^{\mathrm{T}}$ ปัญหาการลู่เข้าของระเบียบวิธีนี้อยู่ที่การเลือกค่าพารามิเตอร์ a และ b ถ้าเลือกน้อยเกินไป จะทำให้การลู่เข้าช้า ถ้าเลือกมากไป จะทำให้ GRG ไม่ลู่เข้า การเลือกค่าที่พอเหมาะไม่มีหลักเกณฑ์ที่ตายตัว ขึ้นอยู่กับปัญหาที่กำลังพิจารณา วิธีการสุ่มเลือก หรือลองผิดลองถูก (trial & error) สำหรับตัวอย่างนี้ จะใช้ $a = 0.02$ และ $b = 0.04$ จะได้ผลการคำนวณดังนี้

เมื่อใช้ความคลาดเคลื่อนที่ยอมรับได้สูงสุดไม่เกิน 1×10^{-6}

Iteration	x_1	x_2	x_3	$f(x)$
0	1.2000	0.7438	0.8803	3.6353
1	0.8018	1.1649	1.0856	3.0428
2	0.9723	1.0269	1.0136	2.9782
3	0.9595	1.0389	1.0197	2.9719

จะได้เงื่อนไขบังคับที่ค่า x^*: $g_1 = 0.9050\times10^{-4}$, $g_2 = 0.9865\times10^{-4}$

เมื่อตัดตัวแปรสแลกออก นำผลการคำนวณมาสร้างกราฟ ดูพฤติกรรมการลู่เข้าของจุดคำตอบ จะได้ดังกราฟต่อไปนี้



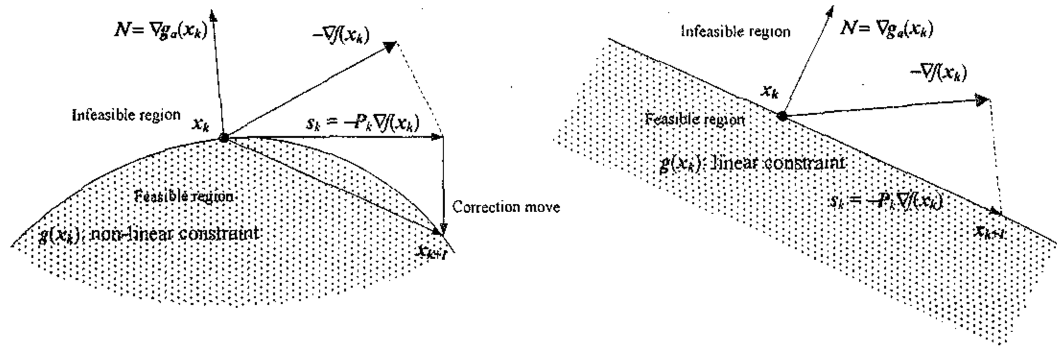
รูปที่ 5.5 คอนทัวร์ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับสำหรับตัวอย่างที่ 5.9

การแก้ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับที่ใช้ระเบียบวิธีเกรเดียนต์ลดทอนนี้อาศัยการสร้างจุดคำตอบที่เป็นไปได้ในทุก ๆ รอบการคำนวณ และในขณะเดียวกัน จุดคำตอบที่สร้างขึ้นต้องมีทิศทางที่ทำให้ฟังก์ชันวัตถุประสงค์ลดค่าลง ในหลาย ๆ กรณี การรักษาความสอดคล้องกับเงื่อนไขบังคับของจุดคำตอบที่ถูกปรับปรุง (feasibility) ในทิศทางลาดลง (descent direction) ทำได้ยาก และก่อให้เกิดปัญหาในการลู่เข้าอันเนื่องมาจากความไม่เป็นเชิงเส้นของเงื่อนไขบังคับ ระเบียบวิธีนี้จึงมีข้อจำกัด

5.8 ระเบียบวิธีเกรเดียนต์โปรเจกชัน (Gradient Projection Method)

ระเบียบวิธีนี้ ใช้หลักการคำนวณทิศทางการค้นหาด้วยทิศทางลาดชันที่สุด (steepest descent direction) อย่างไรก็ตาม โดยทั่วไปแล้ว ถ้าจุดคำตอบในรอบการคำนวณที่ผ่านมาเป็นจุดขอบ (boundary point) ของเงื่อนไขบังคับที่ทำให้เงื่อนไขดังกล่าวแอกทีฟ จะทำให้ทิศทางการค้นหาลบเกรเดียนต์มีทิศทางพุ่งออกจากเซตคำตอบที่เป็นไปได้ การแก้ปัญหานี้สามารถใช้การสร้าง

ภาพฉายหรือโปรเจกชันของเกรเดียนต์ลงมาบนเซตคำตอบที่เป็นไปได้นั่นเอง จะช่วยให้ทิศทางการค้นหายังคงชี้นำไปสู่จุดคำตอบที่เป็นไปได้ ดังแสดงในรูปต่อไปนี้



รูปที่ 5.6 การฉายภาพเกรเดียนต์ลงบนเซตคำตอบที่เป็นไปได้

เทคนิคนี้อาศัยการฉายภาพลบเกรเดียนต์ลงบนเซตของเงื่อนไขบังคับ ผ่านการเมตริกซ์แปลงเชิงเส้นที่เรียกว่า เมตริกซ์โปรเจกชัน (projection matrix) เมื่อต้องการฉายภาพเวกเตอร์ $\boldsymbol{x}$ ใด ๆ จะได้ว่า $\boldsymbol{x} = \boldsymbol{p} + \boldsymbol{q}$ โดยแยกเป็นองค์ประกอบในปริภูมิศูนย์ $\boldsymbol{p}$ และปริภูมิเรนจ์ $\boldsymbol{q}$ จากคุณสมบัติ $\boldsymbol{Mp} = 0$ เมื่อ $\boldsymbol{M}$ แทนเมตริกซ์การแปลงเวกเตอร์ไปยังปริภูมิศูนย์ นั่นคือ

$$\boldsymbol{p} = \boldsymbol{x} - \boldsymbol{q} = \boldsymbol{x} - \boldsymbol{M}^T\lambda$$
$$\boldsymbol{Mp} = \boldsymbol{Mx} - \boldsymbol{MM}^T\lambda = 0$$
$$\lambda = \left(\boldsymbol{MM}^T\right)^{-1}\boldsymbol{Mx}$$

จะได้ว่า

$$\begin{aligned}\boldsymbol{p} &= \boldsymbol{x} - \boldsymbol{q} = \boldsymbol{x} - \boldsymbol{M}^T\lambda = \boldsymbol{x} - \boldsymbol{M}^T\left(\boldsymbol{MM}^T\right)^{-1}\boldsymbol{Mx} \\ \boldsymbol{p} &= \left[\boldsymbol{I} - \boldsymbol{M}^T\left(\boldsymbol{MM}^T\right)^{-1}\boldsymbol{M}\right]\boldsymbol{x} = \boldsymbol{Px}\end{aligned} \tag{5.21}$$

ในกรณีนี้ $\boldsymbol{p}$ เป็นองค์ประกอบของเวกเตอร์ $\boldsymbol{x}$ ที่ถูกฉายมายังปริภูมิศูนย์ เรียก $\boldsymbol{P}$ เป็นเมตริกซ์โปรเจกชัน ซึ่งเมตริกซ์โปรเจกชันนี้มีได้หลายรูปแบบ รูปแบบที่กำหนดนี้ มีคุณสมบัติที่สำคัญ คือ $\boldsymbol{PP}^T = \boldsymbol{P}^2 = \boldsymbol{P}$ และมีคุณสมบัติความสมมาตรรอบแนวทะแยงมุม เรียกเมตริกซ์ $\boldsymbol{P}$ ว่าเป็น orthogonal projection matrix

การแก้ปัญหาในกรณีของเงื่อนไขบังคับไม่เชิงเส้นมีความซับซ้อน เนื่องจากต้องคำนวณ correction move ในที่นี้ จะพิจารณาเฉพาะเงื่อนไขบังคับเชิงเส้นเท่านั้น ถึงแม้จะพิจารณาเฉพาะเงื่อนไขบังคับเชิงเส้น แต่ปัญหาที่สำคัญอีกประการหนึ่ง ได้แก่ เมื่อได้ทิศทางการค้นหาที่เป็นไปได้แล้ว การค้นหาตามเส้นต้องถูกจำกัดด้วยเงื่อนไขบังคับนั่นเอง ดังนั้น ก่อนที่จะแก้ปัญหาการค้นหาตามเส้น จำเป็นต้องทราบขอบเขตที่เป็นไปได้ของช่วงก้าว λ ก่อน จากปัญหาต่อไปนี้

$$\begin{aligned} \text{Minimize} \quad & f(x) \\ \text{Subject to} \quad & Ax \le b \\ & Ex = e \end{aligned} \tag{5.22}$$

นั่นคือ

$$\begin{aligned} \text{Minimize} \quad & F(\lambda_k) \equiv f(x_k + \lambda_k p_k) \\ \text{Subject to} \quad & A(x_k + \lambda_k p_k) \le b \\ & E(x_k + \lambda_k p_k) = e \end{aligned} \tag{5.23}$$

เนื่องจาก ระเบียบวิธีที่พิจารณานี้อยู่ภายใต้ข้อสมมติการสร้างจุดคำตอบที่สอดคล้องกับเงื่อนไขบังคับ ดังนั้น เงื่อนไขบังคับสมการถือเป็นส่วนเกิน ไม่จำเป็นต้องพิจารณา ในกรณีของเงื่อนไขบังคับอสมการ สามารถแบ่งเงื่อนไขได้เป็นสองส่วน ได้แก่ เงื่อนไขที่แอกทีฟ และเงื่อนไขไม่แอกทีฟ เงื่อนไขแอกทีฟนั้น จะเหมือนกับกรณีของเงื่อนไขบังคับสมการนั่นเอง ดังนั้น พิจารณาเฉพาะเงื่อนไขบังคับอสมการที่ไม่แอกทีฟ ถ้าให้ $A = [A_1^T \quad A_2^T]^T$ โดยที่ A_1 แทนเมตริกซ์สัมประสิทธิ์ของเงื่อนไขบังคับอสมการที่แอกทีฟ และ A_2 แทนเมตริกซ์สัมประสิทธิ์ของเงื่อนไขบังคับอสมการที่ไม่แอกทีฟ จะได้ว่า

$$A_2(x_k + \lambda_k p_k) \le b_2 \quad \Rightarrow \quad \lambda_k A_2 p_k \le b_2 - A_2 x_k \tag{5.24}$$

พิจารณาเงื่อนไขดังกล่าวที่ละแถว จะได้ว่า

$$\lambda_k^{(max)} = min \quad \left\{ \frac{\tilde{b}_i}{\tilde{p}_i} \middle| \tilde{p}_i > 0 \right\}; \tilde{b} = b_2 - A_2 x_k, \tilde{p} = A_2 p_k \tag{5.25}$$

จะได้การค้นหาตามเส้นในช่วง $0 \le \lambda \le \lambda^{max}$ อย่างไรก็ตาม การแก้ปัญหาดังกล่าวตามทิศทางเกรเดียนต์โปรเจกชันจะกรณีที่ทำให้ $-P\nabla f(x_k) = 0$ ในกรณีนี้ หมายความว่า ไม่สามารถจะปรับปรุงผลเฉลยได้อีกแล้วในทิศทางดังกล่าว ต้องทำการตรวจสอบว่า จุดที่ได้เป็นจุดคำตอบที่ต้องการหรือไม่ หรือยังมีทิศทางอื่นที่สามารถปรับปรุงผลเฉลยได้อีก ถ้ายังสามารถปรับปรุงผลเฉลยได้ ต้องดำเนินการคำนวณเมตริกซ์โปรเจกชันใหม่ ดังนี้

$$\begin{aligned} 0 = P\nabla f(x) &= \left[I - M^T \left(MM^T \right)^{-1} M \right] \nabla f(x) \\ &= \nabla f(x) - M^T \left(MM^T \right)^{-1} M \nabla f(x) = \nabla f(x) + M^T w \\ &= \nabla f(x) + A_1^T u + E^T v \end{aligned}$$

จะพบว่า เมตริกซ์ M สร้างขึ้นจากเงื่อนไขบังคับที่แอกทีฟทุกชนิดไม่ว่าจะเป็นเงื่อนไขบังคับสมการหรืออสมการ ดังนั้น แยก M ออกเป็นสององค์ประกอบ ได้ A_1 และ E นั่นเอง จากสมการจะได้

$$w = -\left(MM^T \right)^{-1} M \nabla f(x) = \begin{bmatrix} u \\ v \end{bmatrix} \tag{5.26}$$

องค์ประกอบ $\boldsymbol{u}$ เป็นตัวบ่งชี้ถึงเงื่อนไขบังคับอสมการ ถ้า $u_i \in \boldsymbol{u}$ ตำแหน่งที่ I ใด ๆ เนื่องจาก $\boldsymbol{A}_I\boldsymbol{u} = 0$ ตามข้อกำหนดของเงื่อนไขบังคับแอกทีฟ ถ้าผลการคำนวณ ได้ค่า $\forall u_i \geq 0$ นั้น หมายความว่า ทิศทางของเกรเดียนต์โปรเจกชันจะทำไปสู่ผลเฉลยที่ออกนอกเซตคำตอบที่เป็นไปได้ ดังนั้น ไม่มีผลเฉลยที่ดีกว่าจุดคำตอบปัจจุบันอีกแล้ว ให้หยุดการคำนวณ ถ้า $\exists u_i < 0$ หมายความว่า มีบางเงื่อนไข (เงื่อนไขลำดับที่ I เทียบจากตำแหน่งที่ถูกจัดเรียงในเมตริกซ์ $\boldsymbol{A}_I$) ที่ไม่แอกทีฟสามารถเลื่อนจุดคำตอบไปได้อีก ดังนั้น ให้ปลดเงื่อนไขบังคับลำดับที่ I ดังกล่าวออก โดยการลบแถวที่สอดคล้องกับเงื่อนไขบังคับลำดับที่ I นั้น ออกจากเมตริกซ์ $\boldsymbol{M}$ จะได้ เมตริกซ์ $\hat{\boldsymbol{M}}$ ดังนั้น ทิศทางการค้นหาเกรเดียนต์โปรเจกชันที่ถูกแก้ไขมีค่าเป็น

$$\hat{\boldsymbol{p}}_k = -\hat{\boldsymbol{P}}\nabla f(\boldsymbol{x}) \qquad \Rightarrow \qquad \hat{\boldsymbol{P}} = \boldsymbol{I} - \hat{\boldsymbol{M}}^T\left(\hat{\boldsymbol{M}}\hat{\boldsymbol{M}}^T\right)^{-1}\hat{\boldsymbol{M}} \tag{5.27}$$

ดำเนินการแก้ปัญหาการค้นหาตามเส้น เพื่อคำนวณ λ ในช่วง $0 \leq \lambda \leq \lambda^{max}$ ตามปกติ พิจารณาได้จากตัวอย่างต่อไปนี้

ตัวอย่างที่ 5.10 จงใช้ระเบียบวิธีเกรเดียนต์โปรเจกชันเพื่อแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้

$$\begin{array}{ll} \text{Minimize} & f(\boldsymbol{x}) = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 - x_1 - 2x_2 \\ \text{Subject to} & 2x_1 + 3x_2 \leq 6 \\ & x_1 + 4x_2 \leq 5 \\ & x_1, x_2 \geq 0 \end{array}$$

โดยเริ่มต้นการคำนวณที่จุด $\boldsymbol{x}_0 = [0.0 \quad 1.0]^T$

วิธีทำ จะได้ปัญหาในรูปทั่วไปสำหรับระเบียบวิธีเกรเดียนต์โปรเจกชันเป็น

$$\begin{array}{ll} \text{Minimize} & f(\boldsymbol{x}) = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 - x_1 - 2x_2 \\ \text{Subject to} & g_1(\boldsymbol{x}) = 2x_1 + 3x_2 \leq 6 \\ & g_2(\boldsymbol{x}) = x_1 + 4x_2 \leq 5 \\ & g_3(\boldsymbol{x}) = -x_1 \leq 0 \\ & g_4(\boldsymbol{x}) = -x_2 \leq 0 \end{array}$$

คำนวณเกรเดียนต์ของฟังก์ชันวัตถุประสงค์ได้เป็น $\nabla f(\boldsymbol{x}) = \begin{bmatrix} x_1 - 1 \\ x_2 - 2 \end{bmatrix}$

<u>รอบการคำนวณที่ 1</u>:

ที่จุดเริ่มต้นนี้ $\boldsymbol{x}_0 = [0.0 \quad 0.0]^T$ จะได้

$$\begin{array}{ll} f(\boldsymbol{x}_0) = 0.0 & \rightarrow \textit{initial obj. func.} \\ g_1(\boldsymbol{x}_0) = 0 \leq 6 & \rightarrow \textit{inactive constraint} \\ g_2(\boldsymbol{x}_0) = 0 \leq 5 & \rightarrow \textit{inactive constraint} \\ g_3(\boldsymbol{x}_0) = 0 \leq 0 & \rightarrow \textit{active constraint} \\ g_4(\boldsymbol{x}_0) = 0 \leq 0 & \rightarrow \textit{active constraint} \end{array}$$

ดังนั้น (g_a แทนเงื่อนไขบังคับแอกทีฟ) M ถูกสร้างขึ้นจาก g_3 และ g_4 นั่นคือ

$$g_a(x) = Mx \le \phi \qquad \Rightarrow \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{matrix} \leftrightarrow g_3 \\ \leftrightarrow g_4 \end{matrix}$$

$$\therefore \qquad M = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

ดังนั้น

$$P = I - M^T \left(MM^T \right)^{-1} M = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

เข้าเงื่อนไข $P = 0$ จะได้

$$w = -\left(MM^T \right)^{-1} M \nabla f(x) = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$$

เลือกปลดเงื่อนไขที่ให้ค่าติดลบ ในกรณีนี้ เลือกปลด g_4 ออก จะได้

$$\hat{M} = \begin{bmatrix} -1 & 0 \end{bmatrix}$$

นั่นคือ

$$\hat{P} = I - \hat{M}^T \left(\hat{M}\hat{M}^T \right)^{-1} \hat{M} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

จะได้ทิศทางเกรเดียนต์โปรเจกชันเป็น

$$p_0 = -\hat{P} \nabla f(x_0) = -\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ -2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

แก้ปัญหาการค้นหาตามเส้นในทิศทางเกรเดียนต์โปรเจกชัน

$$\underset{\lambda_0 > 0}{\text{Minimize}} \qquad f(x_0 + \lambda_0 p_0) = f\left(\begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + \lambda_0 \begin{bmatrix} 0.0 \\ 2.0 \end{bmatrix} \right)$$

พิจารณาขอบเขตของช่วงก้าวการค้นหา λ_0^{max} จะได้

$$\tilde{b} = b_2 - A_2 x_0 = \begin{bmatrix} 6 \\ 5 \end{bmatrix} - \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}$$

$$\tilde{p} = A_2 p_0 = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \end{bmatrix}$$

$$\Rightarrow \qquad \lambda_0^{(max)} = min \left\{ \frac{\tilde{b}_i}{\tilde{p}_i} \middle| \tilde{p}_i > 0 \right\} = min \left\{ \frac{6}{6}, \frac{5}{8} \right\} = 0.625$$

จะได้ว่า

$$\underset{0 < \lambda_0 \le 0.625}{\text{Minimize}} \qquad f(x_0 + \lambda_0 p_0) = f\left(\begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + \lambda_0 \begin{bmatrix} 0.0 \\ 2.0 \end{bmatrix} \right)$$

แก้ปัญหาการค้นหาตามเส้นจะได้ $\lambda_0 = 1.0 > \lambda_0^{max}$ $\therefore \lambda_0 = \lambda_0^{max} = 0.625$

นั่นคือ
$$x_1 = x_0 + \lambda_0 p_0 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix} + (0.625) \begin{bmatrix} 0.0 \\ 2.0 \end{bmatrix} = \begin{bmatrix} 0.00 \\ 1.25 \end{bmatrix}$$

<u>รอบการคำนวณที่ 2</u>:

ที่จุด $x_1 = [0.0 \quad 1.25]^T$ จะได้

$$\begin{aligned} f(x_1) &= -1.7188 && \rightarrow \textit{decreased} \\ g_1(x_1) &= 3.75 \le 6 && \rightarrow \textit{inactive} \\ g_2(x_1) &= 5 \le 5 && \rightarrow \textit{active} \\ g_3(x_1) &= 0 \le 0 && \rightarrow \textit{active} \\ g_4(x_1) &= -1.25 \le 0 && \rightarrow \textit{inactive} \end{aligned}$$

ดังนั้น (g_a แทนเงื่อนไขบังคับแอกทีฟ) $\boldsymbol{M}$ ถูกสร้างขึ้นจาก g_2 และ g_3 นั่นคือ

$$\boldsymbol{g}_a(\boldsymbol{x}) = \boldsymbol{M}\boldsymbol{x} \le \phi \qquad \Rightarrow \begin{bmatrix} 1 & 4 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 5 \\ 0 \end{bmatrix} \begin{matrix} \leftrightarrow g_2 \\ \leftrightarrow g_3 \end{matrix}$$

$$\therefore \qquad \boldsymbol{M} = \begin{bmatrix} 1 & 4 \\ -1 & 0 \end{bmatrix}$$

ดังนั้น

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{M}^T \left(\boldsymbol{M}\boldsymbol{M}^T\right)^{-1} \boldsymbol{M} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

เข้าเงื่อนไข $\boldsymbol{P} = 0$ จะได้

$$\boldsymbol{w} = -\left(\boldsymbol{M}\boldsymbol{M}^T\right)^{-1} \boldsymbol{M} \nabla f(\boldsymbol{x}) = \begin{bmatrix} 0.1875 \\ -0.8125 \end{bmatrix}$$

เลือกปลดเงื่อนไขที่ให้ค่าลบ ดังนั้น ต้องปลด g_3 ออก จะได้

$$\hat{\boldsymbol{M}} = [1 \quad 4]$$

นั่นคือ

$$\hat{\boldsymbol{P}} = \boldsymbol{I} - \hat{\boldsymbol{M}}^T \left(\hat{\boldsymbol{M}}\hat{\boldsymbol{M}}^T\right)^{-1} \hat{\boldsymbol{M}} = \begin{bmatrix} 0.9412 & -0.2353 \\ -0.2353 & 0.0588 \end{bmatrix}$$

จะได้ทิศทางเกรเดียนต์โปรเจกชันเป็น

$$\boldsymbol{p}_1 = -\hat{\boldsymbol{P}} \nabla f(\boldsymbol{x}_1) = -\begin{bmatrix} 0.9412 & -0.2353 \\ -0.2353 & 0.0588 \end{bmatrix} \begin{bmatrix} -1.00 \\ -0.75 \end{bmatrix} = \begin{bmatrix} 0.7647 \\ -0.1912 \end{bmatrix}$$

แก้ปัญหาการค้นหาตามเส้นในทิศทางเกรเดียนต์โปรเจกชัน

$$\underset{\lambda_1 > 0}{\text{Minimize}} \qquad f(\boldsymbol{x}_1 + \lambda_1 \boldsymbol{p}_1) = f\left(\begin{bmatrix} 0.00 \\ 1.25 \end{bmatrix} + \lambda_1 \begin{bmatrix} 0.7647 \\ -0.1912 \end{bmatrix}\right)$$

พิจารณาขอบเขตของช่วงก้าวการค้นหา λ_1^{max} จะได้

$$\tilde{\boldsymbol{b}} = \boldsymbol{b}_2 - \boldsymbol{A}_2 \boldsymbol{x}_1 = \begin{bmatrix} 6 \\ 0 \end{bmatrix} - \begin{bmatrix} 2 & 3 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0.00 \\ 1.25 \end{bmatrix} = \begin{bmatrix} 2.25 \\ 1.25 \end{bmatrix}$$

$$\tilde{\boldsymbol{p}} = \boldsymbol{A}_2 \boldsymbol{p}_1 = \begin{bmatrix} 2 & 3 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0.7647 \\ -0.1912 \end{bmatrix} = \begin{bmatrix} 0.9558 \\ 0.1912 \end{bmatrix}$$

$$\Rightarrow \qquad \lambda_1^{(max)} = min \quad \left\{ \frac{\tilde{b}_i}{\tilde{p}_i} \middle| \tilde{p}_i > 0 \right\} = min \ \left\{ \frac{2.25}{0.9558}, \frac{1.25}{0.1912} \right\} = 2.354$$

จะได้ว่า

$$\underset{0<\lambda_1\le 2.354}{\text{Minimize}} \qquad f\left(x_1 + \lambda_1 p_1\right) = f\left(\begin{bmatrix} 0.00 \\ 1.25 \end{bmatrix} + \lambda_1 \begin{bmatrix} 0.7647 \\ -0.1912 \end{bmatrix}\right)$$

แก้ปัญหาการค้นหาตามเส้นจะได้ $\lambda_l = 0.944 < \lambda_1^{max} \ \therefore \ \lambda_l = 0.944$

นั่นคือ $$x_2 = x_1 + \lambda_1 p_1 = \begin{bmatrix} 0.00 \\ 1.25 \end{bmatrix} + (0.944)\begin{bmatrix} 0.7647 \\ -0.1912 \end{bmatrix} = \begin{bmatrix} 0.7219 \\ 1.0695 \end{bmatrix}$$

<u>รอบการคำนวณที่ 3</u>:

ที่จุด $x_2 = [0.7219 \quad 1.0695]^T$ จะได้

$$f(x_2) = -2.0284 \qquad \rightarrow decreased$$
$$g_1(x_2) = 4.6523 \le 6 \qquad \rightarrow inactive$$
$$g_2(x_2) = 4.9999 \approx 5 \le 5 \qquad \rightarrow active$$
$$g_3(x_2) = -0.7219 \le 0 \qquad \rightarrow inactive$$
$$g_4(x_2) = -1.0695 \le 0 \qquad \rightarrow inactive$$

ดังนั้น (g_a แทนเงื่อนไขบังคับแอกทีฟ) M ถูกสร้างขึ้นจาก g_2 นั่นคือ

$$g_a(x) = Mx \le \phi \qquad \Rightarrow [1 \quad 4]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le [5] \leftrightarrow g_2$$

$$\therefore \qquad M = [1 \quad 4]$$

ดังนั้น

$$P = I - M^T\left(MM^T\right)^{-1} M = \begin{bmatrix} 0.9412 & -0.2353 \\ -0.2353 & 0.0588 \end{bmatrix}$$

จะได้ทิศทางเกรเดียนต์โปรเจกชันเป็น

$$p_2 = -\hat{P}\nabla f(x_2) = -\begin{bmatrix} 0.9412 & -0.2353 \\ -0.2353 & 0.0588 \end{bmatrix}\begin{bmatrix} -0.2781 \\ -0.9305 \end{bmatrix} = \begin{bmatrix} 0.0428 \\ -0.0107 \end{bmatrix}$$

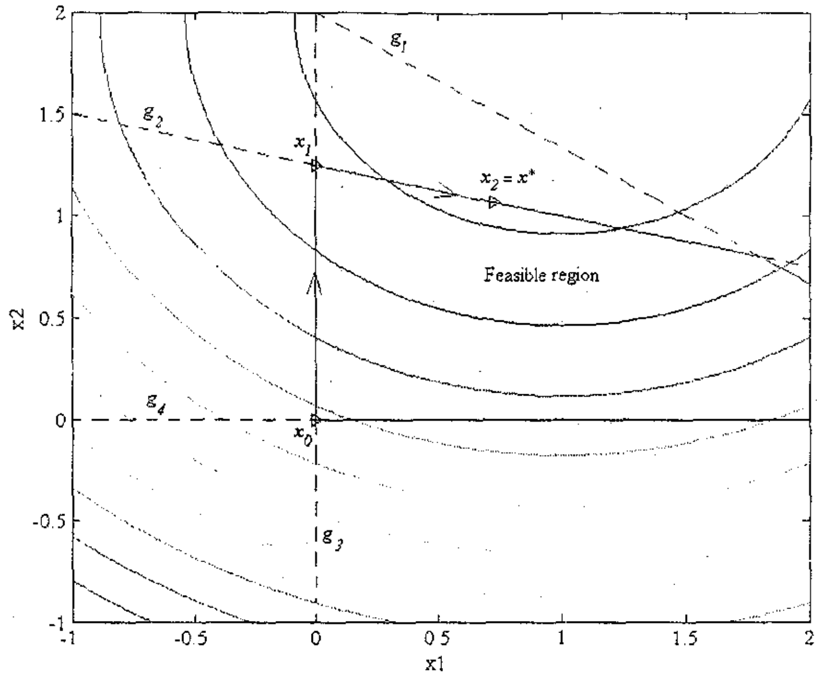
ถือว่า p_2 มีค่าน้อย จุดคำตอบนี้ อาจจะเป็น KKT point ตรวจสอบจากค่า w จะได้

$$w = -\left(MM^T\right)^{-1} M\nabla f(x) = 0.2353 \ge 0 \qquad \rightarrow KKT \ \ point$$

หยุดการคำนวณ จะได้คำตอบเป็น

$$x^* = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^* = \begin{bmatrix} 0.7219 \\ 1.0695 \end{bmatrix} \qquad \rightarrow f\left(x^*\right) = -2.0284$$

ดังกราฟต่อไปนี้



รูปที่ 5.7 คอนทัวร์ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับสำหรับตัวอย่างที่ 5.10

5.9 กำหนดการลำดับกำลังสอง (Sequential Quadratic Programming)

กำหนดการลำดับกำลังสอง (sequential quadratic programming: SQP) เป็นวิธีการแก้ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับที่ถูกพัฒนาขึ้นมาเป็นลำดับหลังสุด แต่อาจจะเรียกได้ว่าเป็นระเบียบวิธีที่มีประสิทธิภาพมากที่สุดเท่าที่นักคณิตศาสตร์สามารถคิดค้นขึ้นมาได้จนถึงปัจจุบัน ถึงแม้จะมีเทคนิคอื่น ๆ เช่น ระเบียบวิธีการปรับโทษ (จะกล่าวถึงในหัวข้อถัดไป) ทำให้แก้ปัญหาได้ง่ายกว่า แต่ผลเสียก็คือการเลือกพารามิเตอร์การปรับโทษที่ไม่เหมาะสม จะส่งผลให้การลู่เข้าช้า และในบางกรณีไม่ลู่เข้าเลย เนื่องจากการสร้างกำแพงกั้นระหว่างเซตที่เป็นไปได้กับเซตที่เป็นไปไม่ได้ของคำตอบนั้น ทำให้การคำนวณค่าเกรเดียนต์บริเวณขอบเขตของเซตดังกล่าวมีปัญหานั่นเอง การพัฒนาระเบียบวิธี SQP นี้มีหลายรูปแบบ แต่เพื่อให้ง่ายในการพิจารณา จะนำเสนอรูปแบบที่ง่ายที่สุด ถึงแม้จะไม่มีประสิทธิภาพสูงสุดเท่ากับ SQP บางวิธีที่ถูกพัฒนาขึ้นก็ตาม แต่ระเบียบวิธีที่จะนำเสนอเหมาะสำหรับการศึกษาขั้นต้น อัลกอริทึมไม่มีความซับซ้อนมากนัก สามารถนำไปเขียนเป็นชุดคำสั่งสำหรับโปรแกรมคอมพิวเตอร์ได้ไม่ยากนัก อย่างไรก็ตาม นอกจากชื่อดังกล่าวแล้ว SQP อาจจะเรียกได้อีกหลายชื่อ เช่น Successive Quadratic Programming (SQP), Iterative Quadratic Programming (IQP), Constrained Variable Metric (CVM), Projected Lagrangian method เป็นต้น พิจารณาปัญหาต่อไปนี้

$$\begin{aligned} &\text{Minimize} && f(x) \\ &\text{subject to} && g(x)=0 \end{aligned}$$

จากฟังก์ชันลากรองจ์

$$L(x,\lambda)=f(x)-\lambda^{\mathrm{T}}g(x)$$

โดยใช้การแก้ปัญหาด้วยระเบียบวิธีนิวตัน ถ้าเริ่มต้นที่จุดคำตอบ $[x_k, \lambda_k]^T$ ใด ๆ สามารถปรับปรุงจุดคำตอบได้ดังนี้

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} p_k \\ v_k \end{bmatrix} \tag{5.28}$$

ทิศทางการค้นหาคำนวณได้จากการแก้สมการเมตริกซ์ต่อไปนี้

$$\begin{bmatrix} \nabla_{xx}^2 \mathcal{L}(x_k, \lambda_k) & -\nabla g(x_k, \lambda_k)^T \\ -\nabla g(x_k, \lambda_k) & 0 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} = \begin{bmatrix} -\nabla_x \mathcal{L}(x_k, \lambda_k) \\ g(x_k, \lambda_k) \end{bmatrix} \tag{5.29}$$

กระบวนการวนรอบจะดำเนินไปเรื่อย ๆ จนกว่า $max \parallel p_k \parallel < \varepsilon$ ซึ่งปัญหาในรูปแบบดังกล่าว สอดคล้องกับการประมาณค่าฟังก์ชันวัตถุประสงค์ด้วยฟังก์ชันกำลังสอง และการประมาณค่าเงื่อนไขบังคับในรูปสมการเชิงเส้น ต่อไปนี้

$$\begin{array}{lll} \text{Minimize} & f(x) & \\ \text{subject to} & g(x) = 0 & \end{array} \Rightarrow \begin{array}{ll} \text{Minimize} & \nabla_x^T \mathcal{L}(x_k, \lambda_k) p_k + \frac{1}{2} p_k^T \left[\nabla_{xx}^2 \mathcal{L}(x_k, \lambda_k) \right] p_k \\ \text{subject to} & \nabla g(x_k, \lambda_k)^T p_k = 0 \end{array}$$

อย่างไรก็ตาม ปัญหาการปรับปรุงจุดคำตอบด้วยเมตริกซ์เฮสเซียนของลากรองจ์ฟังก์ชันทำให้ไม่สามารถรับประกันการลู่เข้าของคำตอบได้ ดังนั้น ต้องแทนเมตริกซ์เฮสเซียนด้วยสูตรการคำนวณเมตริกซ์ $\boldsymbol{B}$ ด้วยสมการคล้ายนิวตัน เช่น สมการ BFGS ดังนี้

$$B_{k+1} = B_k - \frac{(B_k s_k)(B_k s_k)^T}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \tag{5.30}$$

โดยที่

$$s_k = x_{k+1} - x_k$$
$$y_k = \nabla_x \mathcal{L}(x_{k+1}, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_k)$$

จะได้สมการคำนวณเป็นดังนี้

$$\begin{bmatrix} B_k & -\nabla g(x_k, \lambda_k)^T \\ -\nabla g(x_k, \lambda_k) & 0 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} = \begin{bmatrix} -\nabla_x \mathcal{L}(x_k, \lambda_k) \\ g(x_k, \lambda_k) \end{bmatrix} \tag{5.31}$$

ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 5.11 จงแก้ปัญหาค่าเหมาะที่สุดโดยใช้ระเบียบวิธีกำหนดการลำดับกำลังสอง

$$\begin{array}{ll} \text{Minimize} & f(x) = (x_1 - 4)^2 + (x_2 - 4)^2 \\ \text{subject to} & g(x) = 5 - x_1^2 - x_2 = 0 \end{array}$$

กำหนดให้ใช้ค่าความคลาดเคลื่อนสูงสุดไม่เกิน 1×10^{-5} และใช้จุดเริ่มต้น $x_0 = [1 \ \ 0]^T$ จะได้ว่า

$$\mathcal{L}(x, \lambda) = (x_1 - 4)^2 + (x_2 - 4)^2 - \lambda(5 - x_1^2 - x_2)$$

$$\nabla_x \mathcal{L}(x, \lambda) = \begin{bmatrix} 2(x_1 - 4) + 2\lambda x_1 \\ 2(x_2 - 4) + \lambda \end{bmatrix} \qquad \nabla_{xx}^2 \mathcal{L}(x, \lambda) = \begin{bmatrix} 2 + 2\lambda & 0 \\ 0 & 2 \end{bmatrix}$$

$$\nabla f = \begin{bmatrix} 2(x_1-4) \\ 2(x_2-4) \end{bmatrix} \qquad \nabla \boldsymbol{g} = \begin{bmatrix} -2x_1 & -1 \end{bmatrix}$$

เริ่มต้นด้วย $\boldsymbol{x_0} = [1\ \ 0]^T,\ \lambda_0 = 1$

<u>รอบการคำนวณที่ 1</u>:

$$\boldsymbol{g} = 5 - x_1^2 - x_2 = 4$$

$$\nabla_x \mathcal{L} = \begin{bmatrix} 2(x_1-4)+2\lambda x_1 \\ 2(x_2-4)+\lambda \end{bmatrix} = \begin{bmatrix} -4 \\ -7 \end{bmatrix} \qquad \nabla_x^2 \mathcal{L} = \begin{bmatrix} 2+2\lambda & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\nabla \boldsymbol{g} = [-2x_1\ \ -1] = [-2\ \ -1]$$

จะได้

$$\begin{bmatrix} \nabla^2 \mathcal{L} & -\nabla \boldsymbol{g}^T \\ -\nabla \boldsymbol{g} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} -\nabla \mathcal{L} \\ \boldsymbol{g} \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 0 & 2 \\ 0 & 2 & 1 \\ 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \\ 4 \end{bmatrix} \rightarrow \begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix} = \begin{bmatrix} 0.5 \\ 3 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ v \end{bmatrix}^{(1)} = \begin{bmatrix} x_1 \\ x_2 \\ v \end{bmatrix}^{(0)} + \begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix}^{(0)} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 3 \\ 2 \end{bmatrix}$$

<u>รอบการคำนวณที่ 2</u>:

$$\boldsymbol{g} = 5 - x_1^2 - x_2 = -0.25$$

$$\nabla_x \mathcal{L} = \begin{bmatrix} 2(x_1-4)+2\lambda x_1 \\ 2(x_2-4)+\lambda \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\nabla_x^2 \mathcal{L} = \begin{bmatrix} 2+2\lambda & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 6 & 0 \\ 0 & 2 \end{bmatrix}$$

$$\nabla \boldsymbol{g} = [-2x_1\ \ -1] = [-3\ \ -1]$$

จะได้

$$\begin{bmatrix} \nabla^2 \mathcal{L} & -\nabla \boldsymbol{g}^T \\ -\nabla \boldsymbol{g} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} -\nabla \mathcal{L} \\ \boldsymbol{g} \end{bmatrix} \rightarrow \begin{bmatrix} 6 & 0 & 3 \\ 0 & 2 & 1 \\ 3 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ -0.25 \end{bmatrix} \rightarrow \begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix} = \begin{bmatrix} -0.1042 \\ 0.0625 \\ -0.1250 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ v \end{bmatrix}^{(2)} = \begin{bmatrix} x_1 \\ x_2 \\ v \end{bmatrix}^{(1)} + \begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix}^{(1)} = \begin{bmatrix} 1.5 \\ 3 \\ 2 \end{bmatrix} + \begin{bmatrix} -0.1042 \\ 0.0625 \\ -0.1250 \end{bmatrix} = \begin{bmatrix} 1.3958 \\ 3.0625 \\ 1.8750 \end{bmatrix}$$

<u>รอบการคำนวณที่ 3</u>:

$$\boldsymbol{g} = 5 - x_1^2 - x_2 = -0.0109$$

$$\nabla_x \mathcal{L} = \begin{bmatrix} 2(x_1-4)+2\lambda x_1 \\ 2(x_2-4)+\lambda \end{bmatrix} = \begin{bmatrix} 0.026 \\ 0 \end{bmatrix}$$

$$\nabla_x^2 \mathcal{L} = \begin{bmatrix} 2+2\lambda & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 5.75 & 0 \\ 0 & 2 \end{bmatrix}$$

$\nabla \boldsymbol{g} = [-2x_1 \quad -1] = [-2.7917 \quad -1]$

จะได้

$$\begin{bmatrix} \nabla^2\mathcal{L} & -\nabla \boldsymbol{g}^T \\ -\nabla \boldsymbol{g} & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{p} \\ v \end{bmatrix} = \begin{bmatrix} -\nabla\mathcal{L} \\ \boldsymbol{g} \end{bmatrix} \rightarrow \begin{bmatrix} 5.75 & 0 & 2.7917 \\ 0 & 2 & 1 \\ 2.7917 & 1 & 0 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix} = \begin{bmatrix} -0.026 \\ 0 \\ -0.0109 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix} = \begin{bmatrix} -0.0041 \\ 0.0005 \\ -0.001 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ v \end{bmatrix}^{(3)} = \begin{bmatrix} x_1 \\ x_2 \\ v \end{bmatrix}^{(2)} + \begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix}^{(2)} = \begin{bmatrix} 1.3958 \\ 3.0625 \\ 1.8750 \end{bmatrix} + \begin{bmatrix} -0.0041 \\ 0.0005 \\ -0.001 \end{bmatrix} = \begin{bmatrix} 1.3918 \\ 3.0630 \\ 1.8740 \end{bmatrix}$$

<u>รอบการคำนวณที่ 4</u>:

$$\boldsymbol{g} = 5 - x_1^2 - x_2 = -1.65\times10^{-5}$$

$$\nabla_x\mathcal{L} = \begin{bmatrix} 2(x_1-4)+2\lambda x_1 \\ 2(x_2-4)+\lambda \end{bmatrix} = \begin{bmatrix} 7.85\times10^{-6} \\ 0 \end{bmatrix}$$

$$\nabla_x^2\mathcal{L} = \begin{bmatrix} 2+2\lambda & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 5.7481 & 0 \\ 0 & 2 \end{bmatrix}$$

$\nabla \boldsymbol{g} = [-2x_1 \quad -1] = [-2.7835 \quad -1.00]$

จะได้

$$\begin{bmatrix} \nabla^2\mathcal{L} & -\nabla \boldsymbol{g}^T \\ -\nabla \boldsymbol{g} & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{p} \\ v \end{bmatrix} = \begin{bmatrix} -\nabla\mathcal{L} \\ \boldsymbol{g} \end{bmatrix} \rightarrow \begin{bmatrix} 5.7481 & 0 & 2.7835 \\ 0 & 2 & 1 \\ 2.7835 & 1 & 0 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix} = \begin{bmatrix} -7.85\times10^{-6} \\ 0 \\ -1.65\times10^{-5} \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix} = \begin{bmatrix} -4.69\times10^{-6} \\ -3.43\times10^{-6} \\ 6.86\times10^{-6} \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ v \end{bmatrix}^{(4)} = \begin{bmatrix} x_1 \\ x_2 \\ v \end{bmatrix}^{(3)} + \begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix}^{(3)} = \begin{bmatrix} 1.3918 \\ 3.0630 \\ 1.8740 \end{bmatrix} + \begin{bmatrix} -4.69\times10^{-6} \\ -3.43\times10^{-6} \\ 6.68\times10^{-6} \end{bmatrix} = \begin{bmatrix} 1.3918 \\ 3.0630 \\ 1.8740 \end{bmatrix}$$

เนื่องจากความคลาดเคลื่อนสูงสุดของตัวแปรมีค่าน้อยกว่า 1×10^{-5} จะได้

$\boldsymbol{x}^* = [x_1 \quad x_2]^* = [1.3918 \quad 3.0630]^T,\ f^* = 7.6809$

โปรแกรม SCILAB สำหรับตัวอย่างที่ 5.11

// SCILAB source program

```
// ฟังก์ชันสำหรับคำนวณการกำหนดลำดับกำลังสอง        ex5_11a.sci
========================
        ex5_11a.sci                    // SQP using Newton's Update formula
========================
function [x1,lambda,f,k] = ex5_11a(x0,lambda,err_max)
k = 0; xerr = %inf;
f = (x0(1)-4)^2 + (x0(2)-4)^2;
g = 5-x0(1)^2 - x0(2);
L = f - lambda*g;
df = [2*(x0(1)-4); 2*(x0(2)-4)];
dg = [-2*x0(1);-1];
d2f = [2 0;0 2];
d2g = [-2 0;0 0];
Grad = df - lambda*dg;
Hessian = d2f - lambda*d2g;
disp([k x0' f g xerr]);
while xerr>err_max
   M = [Hessian -dg;-dg' 0];
   r = [-Grad;g];
   v = linsolve(M,-r);
   x1 = x0 + v(1:2,1);
   xerr = norm(x1-x0,%inf);
   lambda1 = lambda+v(3,1);
   x0 = x1; lambda = lambda1; k = k+1;
   f = (x0(1)-4)^2 + (x0(2)-4)^2;
   g = 5 - x0(1)^2 - x0(2);
   L = f - lambda*g;
   df = [2*(x0(1)-4);2*(x0(2)-4)];
   dg = [-2*x0(1);-1];
   d2f = [2 0;0 2];
   d2g = [-2 0;0 0];
   Grad = df - lambda*dg;
```

```
    Hessian = d2f - lambda*d2g;
    disp([k x0' f g xerr])
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('ex5_11a.sci');                         ประกาศฟังก์ชันเพื่อการคำนวณ
--> [x1,lambda,f,k]=ex5_11a([1;0],1,1e-4);
 Iter    X1         X2          f(x)         g(x)         Xerr
  0.   1.0000000  0.000000   25.0000000   4.0000000     Inf
  1.   1.5000000  3.0000000   7.2500000  -0.2500000   3.0000000
  2.   1.3958333  3.0625000   7.6605903  -0.0108507   0.1041667
  3.   1.3917735  3.0629831   7.6808463  -0.0000165   0.0040599
  4.   1.3917688  3.0629797   7.6808772  -2.198D-11   0.0000047
```

```
// ฟังก์ชันสำหรับคำนวณการกำหนดลำดับกำลังสอง        ex5_11b.sci
=========================
        ex5_11b.sci                               // SQP using BFGS Update formula
=========================
function [x1,lambda,f,k] = ex5_11b(x0,lambda,err_max)
k = 0; xerr = %inf;
B0 = eye(length(x0),length(x0));
f = (x0(1)-4)^2 + (x0(2)-4)^2;
g = 5 - x0(1)^2 - x0(2);
L = f - lambda*g;
df = [2*(x0(1)-4);2*(x0(2)-4)];
dg = [-2*x0(1);-1];
d2f = [2 0;0 2];
d2g = [-2 0;0 0];
Grad = df - lambda*dg;
disp([k x0' f g xerr]);
while xerr>err_max
   M = [B0 -dg;-dg' 0];
   r = [-Grad;g];
   v = linsolve(M,-r);
```

```
    x1 = x0 + v(1:2,1);
    xerr = norm(x1-x0,%inf);
    lambda1 = lambda + v(3,1);
    sk = x1-x0;
    x0 = x1; lambda = lambda1; k = k+1;
    f = (x0(1)-4)^2 + (x0(2)-4)^2;
    g = 5 - x0(1)^2 - x0(2);
    L = f - lambda*g;
    df = [2*(x0(1)-4);2*(x0(2)-4)];
    dg = [-2*x0(1);-1];
    d2f = [2 0;0 2];
    d2g = [-2 0;0 0];
    Grad1 = df - lambda*dg;
    yk = Grad1 - Grad;
    Grad = Grad1;
    B0 = B0 + ((yk-B0*sk)*(yk-B0*sk)')/((yk-B0*sk)'*sk);
    disp([k x0' f g xerr])
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('ex5_11b.sci');                    ประกาศฟังก์ชันเพื่อการคำนวณ
--> [x1,lambda,f,k] = ex5_11b([1;0],1,1e-4);
```

Iter	X1	X2	f(x)	||g(x)||	Xerr
0.	1.0000000	0.0000000	25.000000	4.0000000	Inf
1.	0.6000000	4.8000000	12.200000	- 0.1600000	4.8000000
2.	2.7224084	2.0931099	5.2684701	- 4.5046174	2.7068901
3.	1.7202136	3.0452594	6.1089554	- 1.0043944	1.0021948
4.	1.4600119	2.9360702	7.5834862	- 0.0677049	0.2602017
5.	1.4018212	3.0382834	7.6754318	- 0.0033862	0.1022132
6.	1.3931934	3.0590866	7.6807587	- 0.0000744	0.0208032
7.	1.3919032	3.0626072	7.6808743	- 0.0000017	0.0035206
8.	1.3917947	3.0629069	7.6808781	0.0000005	0.0002997
9.	1.3917770	3.0629602	7.6808708	- 0.0000034	0.0000533

ตัวอย่างที่ 5.12 จงแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้

$$\begin{aligned}
\text{Minimize} \quad & f(\boldsymbol{x}) = \tfrac{1}{3}(x_1+3)^3 + x_2^2 \\
\text{subject to} \quad & g(\boldsymbol{x}) = 2x_1^2 - x_2 - 1 = 0 \\
& -1 \le x_1, x_2 \le 2
\end{aligned}$$

โดยใช้ระเบียบวิธีลำดับกำลังสอง ด้วยการใช้ค่าความคลาดเคลื่อนสูงสุดไม่เกิน 1×10^{-4} และใช้จุดเริ่มต้น $x_0 = [1.0 \;\; 1.0]^T$

วิธีทำ เนื่องจากปัญหานี้มีขอบเขตของตัวแปรค้นหา จะได้ว่า

$$\begin{aligned}
\text{Minimize} \quad & f(\boldsymbol{x}) = \tfrac{1}{3}(x_1+3)^3 + x_2^2 \\
\text{subject to} \quad & g_1(\boldsymbol{x}) = 2x_1^2 - x_2 - 1 = 0 \\
& g_2(\boldsymbol{x}) = x_1 + 1 - x_3^2 = 0 \qquad \textit{lower limit for } x_1 \\
& g_3(\boldsymbol{x}) = x_1 - 2 + x_4^2 = 0 \qquad \textit{upper limit for } x_1 \\
& g_4(\boldsymbol{x}) = x_2 + 1 - x_5^2 = 0 \qquad \textit{lower limit for } x_2 \\
& g_5(\boldsymbol{x}) = x_2 - 2 + x_6^2 = 0 \qquad \textit{upper limit for } x_2
\end{aligned}$$

โดยที่ x_3, x_4, x_5, x_6 เป็นตัวแปรสแล็กที่นิยามขึ้น และมีค่าได้อย่างอิสระโดยไม่ถูกจำกัดขอบเขต ตัวอย่างการแปลงเงื่อนไขขอบเขตตัวแปรเป็นเงื่อนไขบังคับสมการแสดงไว้ดังต่อไปนี้ ในที่นี้จะนำเสนอเฉพาะตัวแปร x_1 เท่านั้น

$$-1 \le x_1 \le 2 \quad \rightarrow \begin{cases} -1 \le x_1 \Rightarrow x_1 + 1 \ge 0 \Rightarrow x_1 + 1 - x_3^2 = 0 \leftrightarrow g_2(\boldsymbol{x}) \\ \\ x_1 \le 2 \Rightarrow x_1 - 2 \le 0 \Rightarrow x_1 - 2 + x_4^2 = 0 \leftrightarrow g_3(\boldsymbol{x}) \end{cases}$$

อย่างไรก็ตาม การแปลงนี้ขึ้นอยู่กับเป้าหมายว่า ต้องการเงื่อนไขบังคับหลักการแปลงในรูปสมการหรืออสมการ หรือต้องการเงื่อนไขบังคับในรูปสมการเชิงเส้นหรือสมการไม่เชิงเส้น

นั่นคือ

$$\begin{aligned}
\mathcal{L}(\boldsymbol{x},\lambda) = {} & \tfrac{1}{3}(x_1+3)^3 + {x_2}^2 - \lambda_1\left(2x_1^2 - x_2 - 1\right) - \lambda_2\left(x_1 + 1 - x_3^2\right) \\
& - \lambda_3\left(x_1 - 2 + x_4^2\right) - \lambda_4\left(x_2 + 1 - x_5^2\right) - \lambda_5\left(x_2 - 2 + x_6^2\right)
\end{aligned}$$

$$\nabla_x \mathcal{L}(\boldsymbol{x},\lambda) = \begin{bmatrix} (x_1+3)^2 - 4x_1\lambda_1 - \lambda_2 - \lambda_3 \\ 2x_2 + \lambda_1 - \lambda_4 - \lambda_5 \\ 2x_3\lambda_2 \\ -2x_4\lambda_3 \\ 2x_5\lambda_4 \\ -2x_6\lambda_5 \end{bmatrix}$$

$$\nabla^2_{xx}\mathcal{L}(\boldsymbol{x},\lambda)=\begin{bmatrix} 2x_1-4\lambda_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\lambda_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2\lambda_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\lambda_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2\lambda_5 \end{bmatrix}$$

$$\nabla f=\begin{bmatrix} (x_1+3)^2 \\ 2x_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \nabla \boldsymbol{g}=\begin{bmatrix} 4x_1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2x_3 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2x_4 & 0 & 0 \\ 0 & 1 & 0 & 0 & -2x_5 & 0 \\ 0 & 1 & 0 & 0 & 0 & 2x_6 \end{bmatrix}.$$

เริ่มต้นด้วย $\boldsymbol{x_0}=[1.00\ \ 1.00]^T,\ \lambda_0=1.00$

เนื่องด้วยระเบียบวิธีนี้อยู่ภายใต้การกำหนดจุดเริ่มต้นที่ต้องเป็นจุดคำตอบที่เป็นไปได้ ดังนั้น เลือก

$$x_3^{(0)}=\sqrt{2};x_4^{(0)}=1;x_5^{(0)}=\sqrt{2};x_6^{(0)}=1;$$

$$\lambda_2^{(0)}=\lambda_3^{(0)}=\lambda_4^{(0)}=\lambda_5^{(0)}=1$$

ปัญหานี้มีจำนวนตัวแปรปกติสองตัว ตัวแปรสแล็กสี่ตัว และตัวคูณลากรองจ์อีกห้าตัว รวมทั้งสิ้น 11 ตัว นั่นคือ การแก้ปัญหาย่อยการโปรแกรมกำลังสองจะต้องใช้การแก้สมการเชิงเส้นขนาด 11×11 ไม่สะดวกที่จะนำเสนอในที่นี้ เมื่อแก้ปัญหาด้วยโปรแกรมคอมพิวเตอร์ จะได้ผลเฉลยดังต่อไปนี้

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 5.12</u>

// SCILAB source program

```
// ฟังก์ชันสำหรับคำนวณการกำหนดลำดับกำลังสอง        ex5_12.sci
=========================
        ex5_12.sci                    // SQP with Newton's update for example 5.12
=========================
function [x,f] = ex5_12()
x1=1;x2=1;
x3=sqrt(2);x4=1;x5=sqrt(2);x6=1;
L1=1;L2=1;L3=1;L4=1;L5=1;
x0=[x1;x2;x3;x4;x5;x6];
L0=[L1;L2;L3;L4;L5];
B0=10*eye(6,6);
```

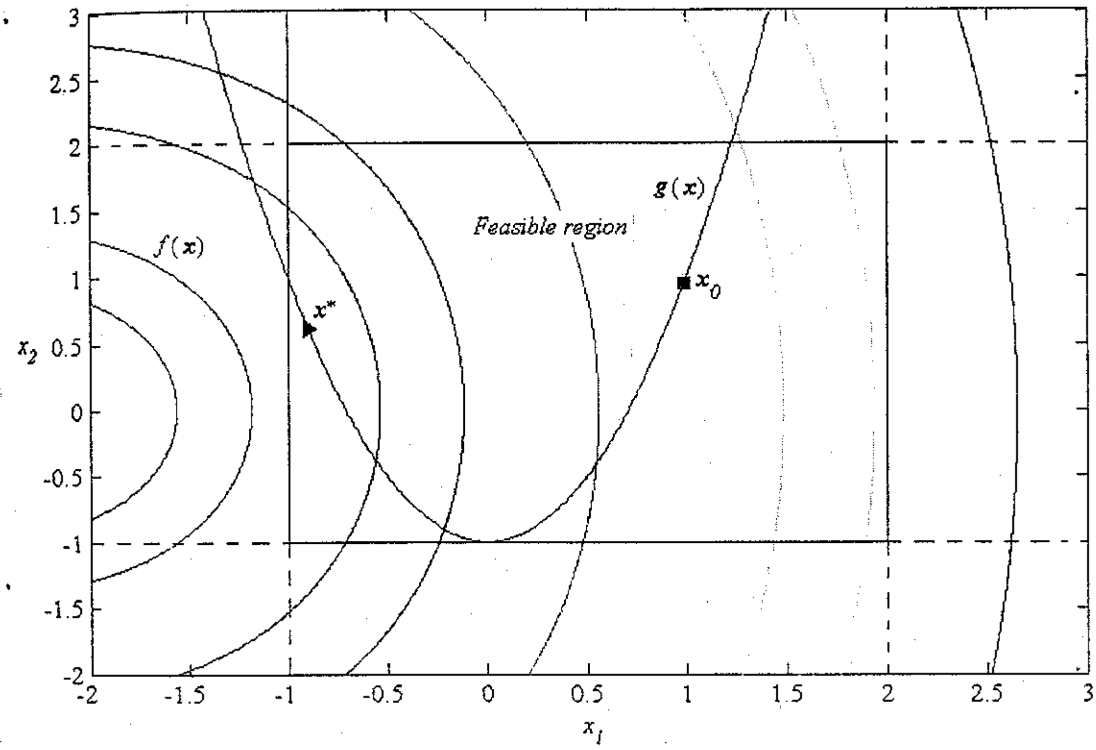
```
LL0=zeros(5,5);
f=1/3*(x1+3)^2+x2^2;
g=[2*x1^2-x2-1;x1+1-x3^3;x1-2+x4^2;x2+1-x5^2;x2-2+x6^2];
Lgrad=[(x1+3)^2-4*x1*L1-L2-L3;
   2*x2+L1-L4-L5;
   2*x3*L2;
   -2*x4*L3;
   2*x5*L4;
   -2*x6*L5];
L2grad=[2*x1-4*L1 0 0 0 0 0;0 2 0 0 0 0;0 0 2*L2 0 0 0;0 0 0 -2*L3 0 0;
         0 0 0 0 2*L4 0;0 0 0 0 0 -2*L5];
Frad=[(x1+3)^2;2*x2;0;0;0;0];
Ggrad=[4*x1 -1 0 0 0 0;
   1 0 -2*x3 0 0 0;
   1 0 0 2*x4 0 0;
   0 1 0 0 -2*x5 0;
   0 1 0 0 0 2*x6];
disp([0 x1 x2 f norm(g,%inf)])
A=[L2grad -Ggrad';-Ggrad LL0];b=[-Lgrad;g];
s=linsolve(A,-b);p=s(1:6,:);v=s(7:11,:);
for k=1:1000
   x=x0+p;
   L=L0+v;
   x0=x;
   L0=L;
   x1=x(1);x2=x(2);x3=x(3);x4=x(4);x5=x(5);x6=x(6);
   L1=L(1);L2=L(2);L3=L(3);L4=L(4);L5=L(5);
   f=1/3*(x1+3)^2+x2^2;
   g=[2*x1^2-x2-1;x1+1-x3^3;x1-2+x4^2;x2+1-x5^2;x2-2+x6^2];
   Lgrad=[(x1+3)^2-4*x1*L1-L2-L3;
      2*x2+L1-L4-L5;
      2*x3*L2;
      -2*x4*L3;
      2*x5*L4;
```

```
    -2*x6*L5];
  L2grad=[2*x1-4*L1 0 0 0 0 0;0 2 0 0 0 0;0 0 2*L2 0 0 0;0 0 0 -2*L3 0 0;
          0 0 0 0 2*L4 0;0 0 0 0 0 -2*L5];
  Frad=[(x1+3)^2;2*x2;0;0;0;0];
  Ggrad=[4*x1 -1 0 0 0 0;
    1 0 -2*x3 0 0 0;
    1 0 0 2*x4 0 0;
    0 1 0 0 -2*x5 0;
    0 1 0 0 0 2*x6];
  disp([k x1 x2 f norm(g,%inf)])
  if (norm(p,%inf)<1e-4)&(norm(g,%inf)<1e-4) then
    break;
  end
  A=[L2grad -Ggrad';-Ggrad LL0];b=[-Lgrad;g];
  s=linsolve(A,-b);p=s(1:6,:);v=s(7:11,:);
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('ex5_12.sci');                    ประกาศฟังก์ชันเพื่อการคำนวณ
--> [x,f] = ex5_12();
 Iter      X1            X2          f(x)        ||g(x)||
  0.    1.00000000    1.0000000    6.3333333    0.8284271
  1.    0.07600410  - 2.6959834   10.422260     3.4150733
  2.    0.76845200  - 0.7779311    5.3389202    2.2637790
  3.  - 0.15139820  - 2.6464058    9.7083077    1.6922486
  4.    0.26888920  - 1.2086802    5.0227865    2.0330278
  5.    1.54515810    0.5173023    7.1537558    3.2577249
  6.    0.73992470  - 1.2018243    6.1067273    1.2968016
  7.  - 0.80892300  - 4.4891458   21.752703     4.7978586
  8.  - 0.63993590  - 0.2380774    1.9133151    0.8371762
  9.  - 1.29491920    1.4956252    3.2059949    0.8580063
 10.  - 0.9542248     0.5889446    1.7419211    0.2321454
 11.  - 0.8890159     0.5721942    1.8128241    0.0250026
 12.  - 0.9006301     0.6219994    1.8560012    0.0042165
```

13.	- 0.8980171	0.6128556	1.8483694	0.0020756
14.	- 0.8985496	0.6147820	1.8499883	0.0004652
15.	- 0.8984388	0.6143845	1.8496549	0.0001731
16.	- 0.8984617	0.6144670	1.8497240	0.0000451
17.	- 0.8984570	0.6144499	1.8497097	0.0000150

ใช้จำนวนรอบการคำนวณทั้งสิ้น 17 รอบ จะได้กราฟดังต่อไปนี้



รูปที่ 5.8 คอนทัวร์ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับสำหรับตัวอย่างที่ 5.12

5.10 ระเบียบวิธีการแปลง (Transformation Methods)

การแก้ปัญหาด้วยระเบียบวิธีจุดเป็นไปได้ ไม่ว่าจะเป็นการใช้ระเบียบวิธีเกรเดียนต์ลดทอน ระเบียบวิธีเกรเดียนต์โปรเจกชันหรือระเบียบวิธีกำหนดการลำดับกำลังสอง อาศัยหลักการสร้างจุดคำตอบภายใต้เงื่อนไขของความสอดคล้องกับเงื่อนไขบังคับ ในหลาย ๆ กรณี การหาจุดที่เป็นไปได้ในเวลาเดียวกันกับการทำให้ค่าฟังก์ชันวัตถุประสงค์ มีค่าต่ำที่สุดในทิศทางการค้นหาที่กำหนดเป็นไปได้ยาก หรืออาจจะเป็นไปไม่ได้เลย กระบวนการวนรอบเพื่อหาช่วงก้าวที่เหมาะสมอาจจะเสี่ยงต่อการไม่ลู่เข้า หรือใช้เวลาการคำนวณช่วงก้าวที่เหมาะสมนานเกินไป ทำให้ข้อด้อยของระเบียบวิธีที่ใช้หลักการดังกล่าวไม่คงทน (lack of robustness) เพื่อให้การแก้ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับสามารถทำได้อย่างมีประสิทธิภาพและมีความคงทนเพื่อรับประกันว่ากระบวนการค้นหาต้องรับประกันคำตอบวงแคบเฉพาะถิ่นอย่างน้อยที่สุด 1 คำตอบ

การแก้ปัญหาโดยไม่ใช้การหาคำตอบที่สอดคล้องกับเงื่อนไขบังคับจะทำให้ลดความยุ่งยาก และการแก้ปัญหามีความคงทนขึ้น แต่อย่างไรก็ตาม เมื่อกระบวนการค้นหาที่ใช้สร้างจุดคำตอบที่ไม่สอดคล้องกับเงื่อนไขบังคับ ถึงแม้จุดคำตอบนั้นจะให้ค่าฟังก์ชันวัตถุประสงค์เป็นที่น่าพอใจ ก็ต้องไม่นำจุดคำตอบดังกล่าวมาใช้งาน การจะทำเช่นนี้ได้นั้น จะต้องมีการกำหนดค่าฟังก์ชันวัตถุประสงค์เพิ่มเติมในกรณีที่คำตอบไม่สอดคล้องกันสมการ อาจจะอยู่ในรูปของการบวกค่าปรับโทษ (penalty term) ที่กำหนดขึ้นมาเป็นการเฉพาะกิจ ทำให้เมื่อจุดคำตอบที่ได้ไม่สอดคล้องกับเงื่อนไขบังคับจะทำให้ได้ค่าฟังก์ชันวัตถุประสงค์สูง และจุดนี้จะโดนกำจัดออกไปในกระบวนการค้นหาช่วงก้าวที่เหมาะสม ดังนั้น ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขบังคับจะถูกลดรูปไปเป็นปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับทำให้วิธีการแก้ปัญหาอาจจะใช้ ระเบียบวิธีชันที่สุดหรือระเบียบวิธีคล้ายนิวตัน

○ **ระเบียบวิธีการปรับโทษ (penalty method)**

พิจารณาปัญหาค่าเหมาะที่สุดดังต่อไปนี้

$$\begin{aligned} &\text{Minimize} && f(\boldsymbol{x}) \\ &\text{subject to} && \boldsymbol{x} \in S \end{aligned} \tag{5.32}$$

ปัญหานี้จะแบ่งจุดคำตอบออกเป็น 2 กลุ่ม ได้แก่ กลุ่มที่สอดคล้องกับเงื่อนไขบังคับ และกลุ่มที่ไม่สอดคล้องกับเงื่อนไขบังคับ โดยกำหนดพจน์การปรับโทษ (penalty term) $\Omega(\boldsymbol{x})$ ดังนี้

$$\Omega(\boldsymbol{x}) = \begin{cases} 0 & ; \boldsymbol{x} \in S \\ \\ +\infty & ; \boldsymbol{x} \notin S \end{cases} \tag{5.33}$$

ทำให้ได้ปัญหาในรูปฟังก์ชันการปรับโทษ (penalty function) $P(\boldsymbol{x})$ ดังนี้

$$\text{Minimize} \quad P(\boldsymbol{x}) = f(\boldsymbol{x}) + \Omega(\boldsymbol{x}) \tag{5.34}$$

อย่างไรก็ตาม ในทางปฏิบัติแล้ว เมื่อค่าฟังก์ชันวัตถุประสงค์ใหม่ที่ได้มีค่าเป็นอนันต์ในกรณีที่คำตอบไม่สอดคล้องกับเงื่อนไขบังคับ จะทำให้การคำนวณค่าเกรเดียนต์หรือข้อมูลอื่น ๆ ทำได้ยาก เนื่องจากไม่มีนิยามการคำนวณค่า ∞ ในการคำนวณเชิงตัวเลข นอกจากนี้ ฟังก์ชันที่ได้ยังไม่มีความต่อเนื่องอีกด้วย ดังนั้น การใช้ฟังก์ชันต่อเนื่องที่ให้ค่าเพิ่มขึ้นอย่างต่อเนื่องเมื่อคำตอบเคลื่อนตัวเข้าหาจุดขอบ (boundary) ของเซตคำตอบที่เป็นไปได้ โดยจะใช้พจน์การปรับโทษที่เขียนในรูปของฟังก์ชันของเงื่อนไขบังคับ ดังนี้

$$\Omega(\boldsymbol{x},\rho) = \rho \cdot \gamma\{\boldsymbol{g}(\boldsymbol{x}), \boldsymbol{h}(\boldsymbol{x})\} \tag{5.35}$$

พจน์การปรับโทษอาจจะใช้ฟังก์ชันดังต่อไปนี้

<u>ฟังก์ชันพาราโบลาหรือกำลังสอง</u>:

$$\Omega(\boldsymbol{x},\rho) = \rho \cdot \left[\boldsymbol{g}(\boldsymbol{x})\right]^2, \Omega(\boldsymbol{x},\rho) = \rho \cdot \left[\boldsymbol{h}(\boldsymbol{x})\right]^2 \tag{5.36}$$

<u>ฟังก์ชันล็อกการิทึม</u>:

$$\Omega(\boldsymbol{x},\rho) = -\rho \cdot ln\,\{\boldsymbol{g}(\boldsymbol{x})\}, \qquad \Omega(\boldsymbol{x},\rho) = -\rho \cdot ln\,\{\boldsymbol{h}(\boldsymbol{x})\} \tag{5.37}$$

ฟังก์ชันส่วนกลับ:

$$\Omega(x,\rho)=\rho\cdot\left[\frac{1}{g(x)}\right], \qquad \Omega(x,\rho)=\rho\cdot\left[\frac{1}{h(x)}\right] \tag{5.38}$$

ปัญหาสำคัญที่ทำให้เงื่อนไขบังคับอสมการมีความยุ่งยากก็คือ ถ้า $h(x) < 0$ ค่าของพจน์การปรับโทษต้องมีค่าสูง คล้ายการสร้างกำแพงขึ้นกันไม่ให้จุดคำตอบข้ามไปในฝั่งของเซตคำตอบที่ไม่สอดคล้องกับเงื่อนไขบังคับ ดังนั้น การกำหนดค่าพจน์การปรับโทษในกรณีนี้ต้องพิจารณาสถานการณ์ดังกล่าวด้วยนั่นเอง ซึ่งอาจจะทำได้ดังนี้

$$\Omega(x,\rho)=\rho\cdot\left[\min\left\{h(x),0\right\}\right]^2 \tag{5.39}$$

พิจารณาตัวอย่างการเขียนฟังก์ชันการปรับโทษดังต่อไปนี้

ตัวอย่างที่ 5.13 พิจารณาปัญหาค่าเหมาะที่สุดต่อไปนี้

Minimize $\quad f(x)=(x_1-4)^2+(x_2-4)^2$

subject to $\quad g(x)=x_1+x_2-5=0$

เขียนฟังก์ชันการปรับโทษ โดยใช้สมการกำลังสอง ได้ดังนี้

$$P(x)=(x_1-4)^2+(x_2-4)^2+\rho(x_1+x_2-5)^2$$

เมื่อแปรค่าพารามิเตอร์การปรับโทษ (penalty parameter: ρ) จะได้ผลดังกราฟ ซึ่งชี้ให้เห็นว่า ในกรณีของปัญหาที่มีเฉพาะเงื่อนไขสมการนั้น ผลของค่าพารามิเตอร์การปรับโทษจะทำให้เส้นโครงร่างของปัญหาเกิดการเปลี่ยนรูปไป และค่าเหมาะที่สุดของปัญหาดั้งเดิมจะกลายเป็นจุดต่ำสุดเฉพาะถิ่นของฟังก์ชันการปรับโทษนั่นเอง การแก้ปัญหาเพื่อหาจุดคำตอบดังกล่าวสามารถใช้ระเบียบวิธีชันที่สุดหรือระเบียบวิธีคล้ายนิวตันได้ตามความเหมาะสม

ทดลองแก้ปัญหานี้โดยใช้ระเบียบวิธีชันที่สุด

เมื่อเริ่มด้วย $\rho = 0.1$ จะได้

Iteration	x_1	x_2	$P(x_1,x_2)$	$\nabla P(x_1,x_2)$
0	0.0000	0.0000	34.500	9.0000
1	3.7496	3.7496	0.7500	1.0789×10^{-3}
2	3.7500	3.7500	0.7500	1.2943×10^{-7}

เมื่อให้ $\rho = 1.0$ จะได้

Iteration	x_1	x_2	$P(x_1,x_2)$	$\nabla P(x_1,x_2)$
0	0.0000	0.0000	57.000	18.000
1	2.9991	2.9991	3.0000	5.4000×10^{-3}
2	3.0000	3.0000	3.0000	1.6176×10^{-6}
3	3.0000	3.0000	3.0000	7.1548×10^{-8}

เมื่อให้ ρ = 10.0 จะได้

Iteration	x_1	x_2	$P(x_1,x_2)$	$\nabla P(x_1,x_2)$
0	0.0000	0.0000	282.00	108.00
1	2.5660	2.5660	4.2869	2.2680×10^{-1}
2	2.5714	2.5714	4.2857	4.7545×10^{-4}
3	2.5714	2.5714	4.2857	1.0012×10^{-6}

เมื่อให้ ρ = 100 จะได้

Iteration	x_1	x_2	$P(x_1,x_2)$	$\nabla P(x_1,x_2)$
0	0.0000	0.0000	2532	1008
1	2.4571	2.4571	5.4988	20.2610
2	2.5064	2.5064	4.4780	4.0723×10^{-1}
3	2.5074	2.5074	4.4776	8.1723×10^{-3}
4	2.5075	2.5075	4.4776	1.6400×10^{-4}

เมื่อให้ ρ = 1000 จะได้

Iteration	x_1	x_2	$P(x_1,x_2)$	$\nabla P(x_1,x_2)$
0	0.0000	0.0000	25032	10008
1	2.0003	2.0003	1006.6	2002.6
2	2.4006	2.4006	44.622	400.72
3	2.4807	2.4807	6.1043	80.184
4	2.4967	2.4967	4.5621	16.045
5	2.4999	2.4999	4.5003	3.2097
6	2.5006	2.5006	4.4979	6.4180×10^{-1}
7	2.5007	2.5007	4.4978	1.2833×10^{-1}
8	2.5007	2.5007	4.4978	2.5661×10^{-2}
9	2.5007	2.5007	4.4978	5.1310×10^{-3}

จะพบว่า ผลเฉลยจะขึ้นอยู่กับพารามิเตอร์การปรับโทษนั่นเอง ดังนั้น การแก้ปัญหานี้ต้องดำเนินการโดยการกำหนดค่าพารามิเตอร์การปรับโทษ จากนั้นใช้ระเบียบวิธีการแก้ปัญหาแบบไม่มีเงื่อนไขเพื่อหาคำตอบ ดำเนินการปรับเปลี่ยนค่าพารามิเตอร์การปรับโทษ แล้วทำซ้ำการหาคำตอบไปเรื่อย ๆ จนกระทั่งจุดคำตอบที่ได้ลู่เข้าสู่ค่า ๆ หนึ่ง ซึ่งก็คือค่าเหมาะที่สุดของปัญหานั่นเอง พิจารณาตัวอย่างโปรแกรมคำนวณโดยใช้ SCILAB ดังต่อไปนี้

โปรแกรม SCILAB สำหรับตัวอย่างที่ 5.13

// SCILAB source program

```
// ฟังก์ชันสำหรับคำนวณด้วยระเบียบวิธีการปรับโทษ      penalty.sci
========================
        penalty.sci                    // Penalty Method with Steepest Decent Algorithm
========================
function [x,f,RHO]=penalty(fobj,fgrad,x0,ferr_max)
```

```
global rho
count=0;
rho=0;
ferr=%inf;
f0=%inf;
ferr_max=1e-6;
disp('The constrained optimization problem was solved by the Penalty method')
disp('==> Steepest descent method was employed to solve iterative unconstrained
problems');
while ferr>ferr_max
   [x1,f1,k] = steepest(fobj,fgrad,x0,1e-6,1e-6,0);
   ferr=abs(f1-f0);
   disp([count rho x1' f1 ferr k]);
   f0=f1;
   rho=rho+10;
   x(count+1,1)=x1(1,1);
   x(count+1,2)=x1(2,1);
   f(count+1)=f1;
   RHO(count+1)=rho;
   if count>1000 then
      break;
   end
   count=count+1;
end
endfunction
```

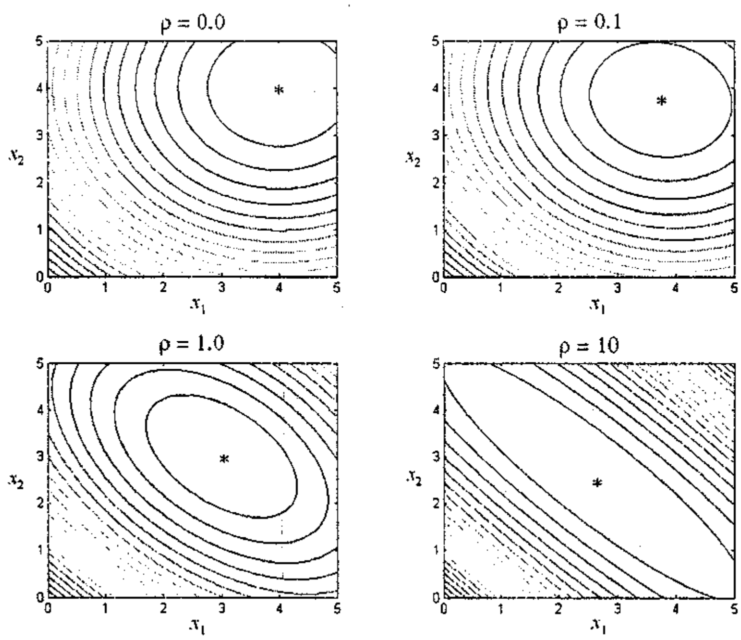
```
// ฟังก์ชันการปรับโทษสำหรับตัวอย่างที่ 5.13                fpen5_13.sci
========================
      fpen5_13.sci
========================
function f=fpen5_13(x)
global rho
P=x(1)+x(2)-5;
f=(x(1)-4)^2+(x(2)-4)^2+rho*P^2;
endfunction
```

```
// เกรเดียนต์ของฟังก์ชันการปรับโทษสำหรับตัวอย่างที่ 5.13          gpen5_13.sci
=======================
     gpen5_13.sci
=======================
function g=gpen5_13(x)
// f=(x(1)-4)^2+(x(2)-4)^2+rho*(x(1)+x(2)-5)^2;
global rho
g = [2*(x(1)-4) + 2*rho*(x(1) + x(2) - 5);2*(x(2)-4) + 2*rho*(x(1) + x(2) - 5)];
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('cubicsearch.sci');          ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('fgradnum1.sci');            ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('lsearch.sci');              ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('flsearch.sci');             ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('steepest.sci');             ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('fpen5_13.sci');             ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('gpen5_13.sci');             ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('penalty.sci');              ประกาศฟังก์ชันเพื่อการคำนวณ
--> [x,f,RHO]=penalty(fpen5_13,gpen5_13,[0;0],1e-4);
 The constrained optimization problem was solved by the Penalty method
 ==> Steepest descent method was employed to solve iterative unconstrained problems
// Iter  ρ     X1          X2          f(x)        Ferr        k
   0.   0.0   4.0000000   4.0000000   1.578D-30   Inf         1.
   1.   10.   2.5714286   2.5714286   4.2857143   4.2857143   1.
   2.   20.   2.5365854   2.5365854   4.3902439   0.1045296   1.
   3.   30.   2.5245902   2.5245902   4.4262295   0.0359856   1.
   4.   40.   2.5185185   2.5185185   4.4444444   0.0182149   1.
   5.   50.   2.5148515   2.5148515   4.4554455   0.0110011   1.
   6.   60.   2.5123967   2.5123967   4.4628099   0.0073644   1.
   7.   70.   2.5106383   2.5106383   4.4680851   0.0052752   1.
   8.   80.   2.5093168   2.5093168   4.4720497   0.0039646   1.
   9.   90.   2.5082873   2.5082873   4.4751381   0.0030884   1.
   10. 100.   2.5074627   2.5074627   4.4776119   0.0024738   1.
   11. 110.   2.5067873   2.5067873   4.4796380   0.0020261   1.
```

12.	120.	2.5062241	2.5062241	4.4813278	0.0016898	1.
13.	130.	2.5057471	2.5057471	4.4827586	0.0014308	1.
14.	140.	2.5053381	2.5053381	4.4839858	0.0012271	1.
15.	150.	2.5049834	2.5049834	4.4850498	0.0010641	1.
16.	160.	2.5046729	2.5046729	4.4859813	0.0009315	1.
17.	170.	2.5043988	2.5043988	4.4868035	0.0008222	1.
18.	180.	2.5041551	2.5041551	4.4875346	0.0007311	1.
19.	190.	2.5039370	2.5039370	4.4881890	0.0006544	1.
20.	200.	2.5037406	2.5037406	4.4887781	0.0005891	1.
21.	210.	2.5035629	2.5035629	4.4893112	0.0005331	1.
22.	220.	2.5034014	2.5034014	4.4897959	0.0004848	1.
23.	230.	2.5032538	2.5032538	4.4902386	0.0004427	1.
24.	240.	2.5031185	2.5031185	4.4906445	0.0004059	1.
25.	250.	2.5029940	2.5029940	4.4910180	0.0003735	1.
26.	260.	2.5028791	2.5028791	4.4913628	0.0003448	1.
27.	270.	2.5027726	2.5027726	4.4916821	0.0003193	1.
28.	280.	2.5026738	2.5026738	4.4919786	0.0002965	1.
29.	290.	2.5025818	2.5025818	4.4922547	0.0002761	1.
30.	300.	2.5024958	2.5024958	4.4925125	0.0002577	1.
31.	310.	2.5024155	2.5024155	4.4927536	0.0002411	1.
32.	320.	2.5023401	2.5023401	4.4929797	0.0002261	1.
33.	330.	2.5022693	2.5022693	4.4931921	0.0002124	1.
34.	340.	2.5022026	2.5022026	4.4933921	0.0001999	1.
35.	350.	2.5021398	2.5021398	4.4935806	0.0001885	1.
36.	360.	2.5020804	2.5020804	4.4937587	0.0001781	1.
37.	370.	2.5020243	2.5020243	4.4939271	0.0001685	1.
38.	380.	2.5019711	2.5019711	4.4940867	0.0001596	1.
39.	390.	2.5019206	2.5019206	4.4942382	0.0001514	1.
40.	400.	2.5018727	2.5018727	4.4943820	0.0001439	1.
41.	410.	2.5018270	2.5018270	4.4945189	0.0001369	1.
42.	420.	2.5017836	2.5017836	4.4946492	0.0001303	1.
43.	430.	2.5017422	2.5017422	4.4947735	0.0001243	1.
44.	440.	2.5017026	2.5017026	4.4948922	0.0001186	1.
45.	450.	2.5016648	2.5016648	4.4950055	0.0001134	1.

46.	460.	2.5016287	2.5016287	4.4951140	0.0001085	1.
47.	470.	2.5015940	2.5015940	4.4952179	0.0001038	1.
48.	480.	2.5015609	2.5015609	4.4953174	0.0000995	1.



รูปที่ 5.9 ผลของสัมประสิทธิ์การปรับโทษต่อเส้นโครงร่างหรือคอนทัวร์ของปัญหา

จากตัวอย่างนี้ เมื่อปรับพารามิเตอร์การปรับโทษให้สูงขึ้นเรื่อย ๆ จะพบว่า คำตอบจะลู่เข้าสู่จุด

$$x^* = [2.50 \quad 2.50]^T, f(x^*) = 4.50 \qquad \text{เมื่อ} \quad \rho \to +\infty$$

อย่างไรก็ตาม การกำหนดค่าพารามิเตอร์การปรับโทษให้มีค่าสูงมาก ๆ จะเกิดผลเสีย คือ การลู่เข้าจะช้าลงตามค่าพารามิเตอร์ที่เพิ่มขึ้น และจากความคลาดเคลื่อนสะสมเนื่องการคำนวณเชิงตัวเลขที่ค่าพารามิเตอร์การปรับโทษมีค่าสูง อาจจะส่งผลให้ปัญหาไม่ลู่เข้าได้

กราฟในรูปที่ 5.9 แสดงถึงการเปลี่ยนรูปร่างของเส้นโครงร่างของปัญหาตามค่าพารามิเตอร์การปรับโทษ

ตัวอย่างที่ 5.14 พิจารณาปัญหาค่าเหมาะที่สุดต่อไปนี้

Minimize $\quad f(\boldsymbol{x}) = (x_1 - 4)^2 + (x_2 - 4)^2$

subject to $\quad \boldsymbol{g}(\boldsymbol{x}) = x_1 + x_2 - 5 \le 0$

เขียนฟังก์ชันการปรับโทษได้ดังนี้

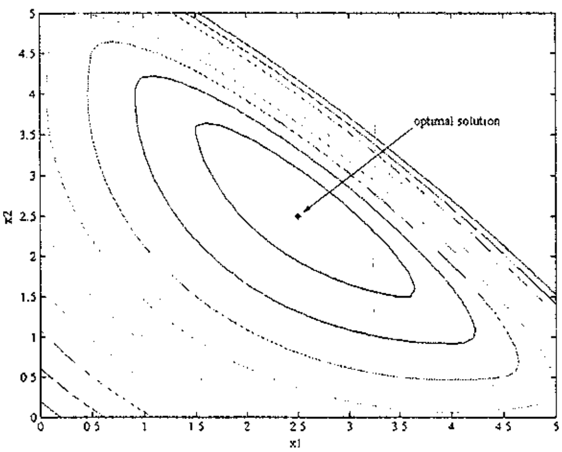
$$P(\boldsymbol{x}) = (x_1 - 4)^2 + (x_2 - 4)^2 + \rho \cdot \left[min\ \{(5 - x_1 - x_2), 0\} \right]^2$$

ตัวอย่างนี้ มีจุดคำตอบเดียวกันกับตัวอย่างที่ 5.13

ถ้าเลือก $\rho = 1000$ จะได้ว่า

Iteration	x_1	x_2	$P(x_1,x_2)$	$\nabla P(x_1,x_2)$
0	0.0000	0.0000	32.000	8.0000
1	2.5003	2.5000	4.4984	1.6005
2	2.5007	2.5007	4.4978	3.2004×10^{-1}
3	2.5007	2.5007	4.4978	6.3994×10^{-2}
4	2.5007	2.5007	4.4978	1.2796×10^{-2}

วาดเส้นโครงร่างได้ดังนี้



รูปที่ 5.10 การใช้ฟังก์ชันกำลังสองเป็นฟังก์ชันปรับโทษสำหรับตัวอย่างที่ 5.14

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 5.14</u>

// SCILAB source program

```
// ฟังก์ชันสำหรับคำนวณด้วยระเบียบวิธีการปรับโทษ      penalty1.sci
========================
        penalty1.sci
========================
function [x,f,k]=penalty1(fobj,fgrad,x0,ferr_max)
// Penalty Method with Steepest Decent Algorithm
// Use a pre-defined penalty factor

disp('The constrained optimization problem was solved by the Penalty method')
disp('==> Steepest descent method was employed to solve iterative unconstrained
problems');
[x,f,k] = steepest(fobj,fgrad,x0,ferr_max,ferr_max,1);
endfunction
```

```
// ฟังก์ชันการปรับโทษสำหรับตัวอย่างที่ 5.14                    fpen5_14.sci
========================
        fpen5_14.sci
========================
function f = fpen5_14(x)
rho = 1000;
P = x(1) + x(2) - 5;
if P<=0 then
   P = 0;
end
f = (x(1)-4)^2 + (x(2)-4)^2 + rho*P^2;
endfunction
```

```
// เกรเดียนต์ของฟังก์ชันการปรับโทษสำหรับตัวอย่างที่ 5.14        gpen5_14.sci
========================
        gpen5_14.sci
========================
function g = gpen5_14(x)
rho = 1000;
P = x(1) + x(2) - 5;
if P<=0 then
   g = [2*(x(1)-4);2*(x(2)-4)];
else
   g = [2*(x(1)-4) + 2*rho*(x(1) + x(2) - 5);2*(x(2)-4) + 2*rho*(x(1) + x(2) - 5)];
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('cubicsearch.sci');          ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('fgradnum1.sci');            ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('lsearch.sci');              ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('flsearch.sci');             ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('steepest.sci');             ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('fpen5_14.sci');             ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('gpen5_14.sci');             ประกาศฟังก์ชันเพื่อการคำนวณ
```

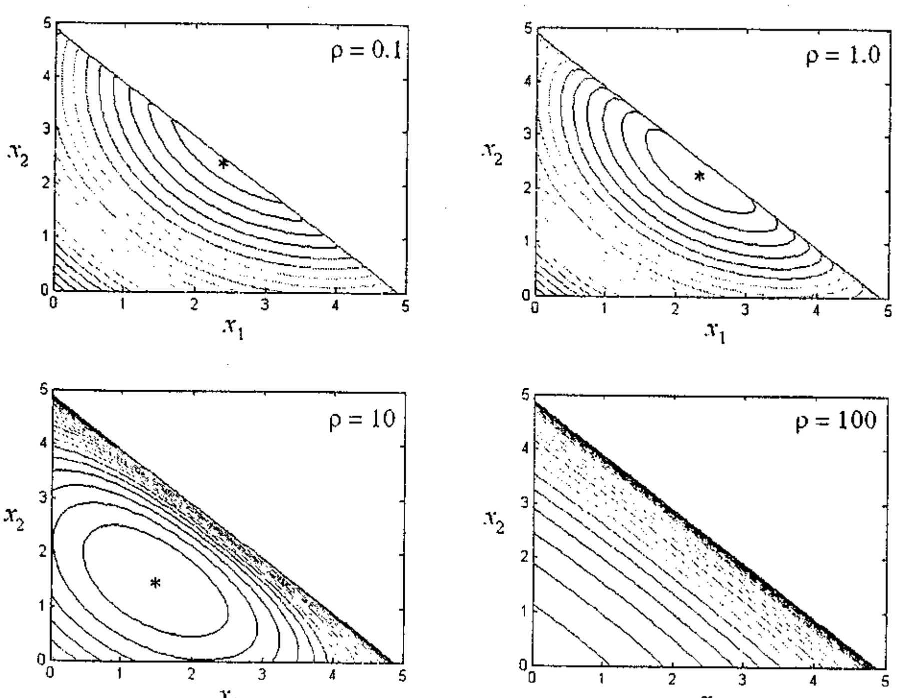
```
> exec('penalty1.sci');                    ประกาศฟังก์ชันเพื่อการคำนวณ
> [x,f,k]=penalty1(fpen5_14,gpen5_14,[0;0],1e-4);
The constrained optimization problem was solved by the Penalty method
==> Steepest descent method was employed to solve iterative unconstrained problems
// Iter    X1          X2          f(x)         ||g(x)||
   0.    0.0000000   0.0000000   32.000000    11.313708
   1.    2.5007496   2.5007496   4.4977511    1.324D-11
```

ทดลองเปลี่ยนพจน์การปรับโทษเป็นฟังก์ชันล็อกการิทึม ได้ดังนี้

$$P(x)=(x_1-4)^2+(x_2-4)^2-\rho\cdot ln\,(5-x_1-x_2)$$

พิจารณาการแปรค่าพารามิเตอร์การปรับโทษ จากน้อยไปมาก จะพบว่า ให้ผลตรงกันข้ามกับกรณีการใช้ฟังก์ชันพาราโบลา นั่นคือ ยิ่งค่า ρ มีค่าน้อยหรือเข้าใกล้ 0 จะทำให้คำตอบที่ได้ลู่เข้าสู่ค่าที่แท้จริงของปัญหา ดังแสดงในรูปที่ 5.11

การเลือกพารามิเตอร์การปรับโทษ ρ ไม่ใช่เรื่องที่ซับซ้อน จะพบว่า ในกรณีที่ใช้ฟังก์ชันพาราโบลา ค่า ρ ต้องมีค่าสูง ดังนั้น ในการแก้ปัญหาโดยใช้ฟังก์ชันพาราโบลานี้ จะเริ่มต้นคำนวณด้วย $\rho=0$ จากนั้น ปัญหาจะถูกแปลงให้อยู่ในรูปของการหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับ เมื่อได้คำตอบที่เหมาะสม จะต้องทำการคำนวณซ้ำใหม่ โดยการปรับค่าพารามิเตอร์การปรับโทษให้เพิ่มขึ้นเรื่อย ๆ มีข้อแนะนำให้เลือก $\Delta\rho=10^n$



รูปที่ 5.11 การใช้ฟังก์ชันล็อกการิทึมเป็นฟังก์ชันปรับโทษ

ตัวอย่างที่ 5.15 จงแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้ โดยใช้ระเบียบวิธีการปรับโทษ

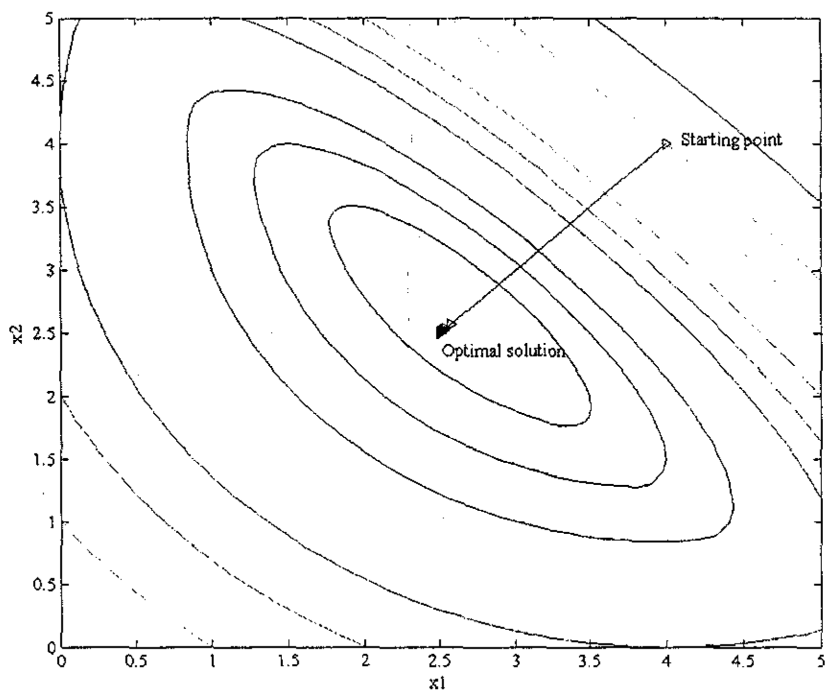
Minimize $f(x) = (x_1 - 4)^2 + (x_2 - 4)^2$

subject to $g(x) = 5 - x_1 - x_2 \geq 0$

สร้างฟังก์ชันการปรับโทษได้เป็น

$$P(x, \rho) = (x_1 - 4)^2 + (x_2 - 4)^2 + \rho\left[min\{(5 - x_1 - x_2), 0\}\right]^2$$

แก้ปัญหาโดยเริ่มต้นที่ $\rho = 0$ ปรับปรุงค่าพารามิเตอร์การปรับโทษโดยใช้ $\Delta\rho = 10$ ในแต่ละรอบ จะได้ดังกราฟ (ใช้จำนวนรอบการค้นหาทั้งสิ้น 48 รอบ)



รูปที่ 5.12 คอนทัวร์ของปัญหาในตัวอย่างที่ 5.15 กรณีที่ให้ $\rho = 50$

จะได้ผลเฉลยดังตารางต่อไปนี้

k	ρ	x_1	x_2	$f(x_1, x_2)$	Change of $f(x_1, x_2)$
0	0	4.0000	4.0000	0.0000	Inf
1	10	2.5714	2.5714	4.2857	4.29E+00
2	20	2.5366	2.5366	4.3902	1.05E-01
3	30	2.5246	2.5246	4.4262	3.60E-02
4	40	2.5185	2.5185	4.4444	1.82E-02
5	50	2.5149	2.5149	4.4554	1.10E-02
6	60	2.5124	2.5124	4.4628	7.40E-03
7	70	2.5106	2.5106	4.4681	5.30E-03
8	80	2.5093	2.5093	4.4720	4.00E-03
9	90	2.5083	2.5083	4.4751	3.10E-03
10	100	2.5075	2.5075	4.4776	2.50E-03
11	110	2.5068	2.5068	4.4796	2.00E-03

12	120	2.5062	2.5062	4.4813	1.70E-03
13	130	2.5057	2.5057	4.4828	1.40E-03
14	140	2.5053	2.5053	4.4840	1.20E-03
15	150	2.5050	2.5050	4.4850	1.10E-03
16	160	2.5047	2.5047	4.4860	9.00E-04
17	170	2.5044	2.5044	4.4868	8.00E-04
18	180	2.5042	2.5042	4.4875	7.00E-04
19	190	2.5039	2.5039	4.4882	7.00E-04
20	200	2.5037	2.5037	4.4888	6.00E-04
21	210	2.5036	2.5036	4.4893	5.00E-04
22	220	2.5034	2.5034	4.4898	5.00E-04
23	230	2.5033	2.5033	4.4902	4.00E-04
24	240	2.5031	2.5031	4.4906	4.00E-04
25	250	2.5030	2.5030	4.4910	4.00E-04
26	260	2.5029	2.5029	4.4914	3.00E-04
27	270	2.5028	2.5028	4.4917	3.00E-04
28	280	2.5027	2.5027	4.4920	3.00E-04
29	290	2.5026	2.5026	4.4923	3.00E-04
30	300	2.5025	2.5025	4.4925	3.00E-04
31	310	2.5024	2.5024	4.4928	2.00E-04
32	320	2.5023	2.5023	4.4930	2.00E-04
33	330	2.5023	2.5023	4.4932	2.00E-04
34	340	2.5022	2.5022	4.4934	2.00E-04
35	350	2.5021	2.5021	4.4936	2.00E-04
36	360	2.5021	2.5021	4.4938	2.00E-04
37	370	2.5020	2.5020	4.4939	2.00E-04
38	380	2.5020	2.5020	4.4941	2.00E-04
39	390	2.5019	2.5019	4.4942	2.00E-04
40	400	2.5019	2.5019	4.4944	1.00E-04
41	410	2.5018	2.5018	4.4945	1.00E-04
42	420	2.5018	2.5018	4.4946	1.00E-04
43	430	2.5017	2.5017	4.4948	1.00E-04
44	440	2.5017	2.5017	4.4949	1.00E-04
45	450	2.5017	2.5017	4.4950	1.00E-04
46	460	2.5016	2.5016	4.4951	1.00E-04
47	470	2.5016	2.5016	4.4952	1.00E-04
48	480	2.5016	2.5016	4.4953	1.00E-04

5.11 สรุป

การแก้ปัญหาค่าเหมาะที่สุดแบบมีเงื่อนไขที่นำเสนอไปนั้น ในกรณีที่ฟังก์ชันวัตถุประสงค์เป็นฟังก์ชันกำลังสองและเงื่อนไขบังคับเป็นฟังก์ชันเชิงเส้น ระเบียบวิธีเกรเดียนต์ลดทอนหรือระเบียบวิธีเกรเดียนต์โปรเจกชันสามารถนำมาใช้แก้ปัญหาได้ อย่างไรก็ตาม ถึงแม้ระเบียบวิธีทั้งสองจะมีเวอร์ชันสำหรับแก้ปัญหาไม่เชิงเส้น กระบวนการแก้ปัญหาค่อยข้างยุ่งยากซับซ้อน เมื่อปัญหามีฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับไม่เชิงเส้น กระบวนการแก้ปัญหาด้วยระเบียบวิธีกำหนดการลำดับกำลังสองสามารถนำมาใช้ได้อย่างมีประสิทธิภาพ แต่ด้วยการยึดหลักการปรับปรุง

จุดคำตอบด้วยหลักการของนิวตันส่งผลให้เมตริกซ์เฮสเซียนต้องมีความเป็น *pdf* ดังนั้น อาจจะใช้สูตรการปรับปรุง DFP หรือ BFGS เข้ามาช่วย ในส่วนของระเบียบวิธีการปรับโทษเป็นอีกทางเลือกหนึ่งที่น่าสนใจ อย่างไรก็ตาม ปัญหาของระเบียบวิธีนี้ ได้แก่ การเลือกค่าสัมประสิทธิ์การปรับโทษและการเลือกฟังก์ชันการปรับโทษให้เหมาะสม จะเห็นได้ว่า แต่ละวิธีมีข้อดี และข้อด้อยที่แตกต่างกัน การเลือกใช้วิธีการแก้ปัญหาจะขึ้นอยู่กับปัจจัยหลายประการ การเลือกใช้เทคนิคที่ดีและเหมาะสมช่วยให้สามารถแก้ปัญหาได้ และใช้เวลาในการคำนวณรวดเร็ว

5.12 แบบฝึกหัดท้ายบท

1. จงพิสูจน์ความสัมพันธ์สำหรับฟังก์ชันลดทอน $\phi(v)$ ต่อไปนี้

$$\nabla\phi(v) = \boldsymbol{Z}^T\nabla f(\boldsymbol{x})$$
$$\nabla^2\phi(v) = \boldsymbol{Z}^T\nabla^2 f(\boldsymbol{x})\boldsymbol{Z}$$

2. จงหาค่าเหมาะที่สุดของปัญหาต่อไปนี้ โดยใช้วิธี variable-elimination method

Minimize $f(\boldsymbol{x}) = 5x_1x_2x_3$

subject to $x_1 + x_2 + 2x_3 - 1 = 0$

3. จงหาค่าเหมาะที่สุดของปัญหาต่อไปนี้ โดยใช้วิธี variable-elimination method

Minimize $f(\boldsymbol{x}) = x_1 + x_2$

subject to $x_1^2 + x_2^2 - 1 = 0$

ให้กำจัดตัวแปร x_2 แก้ปัญหาโดยแสดงให้เห็นว่าการเลือกเครื่องหมายของการถอดรากผิดจะนำไปสู่คำตอบของปัญหาที่ผิด

4. จงหาค่าเหมาะที่สุดของปัญหาต่อไปนี้ โดยใช้วิธี variable-elimination method

Minimize $f(\boldsymbol{x}) = x_1^2 + x_2^2$

subject to $x_1x_2 - 9 = 0$

5. จงหาค่าเหมาะที่สุดของปัญหาต่อไปนี้ พร้อมทั้งวาดกราฟเพื่อยืนยันจุดคำตอบที่คำนวณได้

Minimize $f(\boldsymbol{x}) = \frac{1}{2}x_1^2 + x_2^2$

subject to $2x_1 + x_2 \geq 2$

$x_1 - x_2 \leq 1$

$x_1, x_2 \geq 0$

6. จงหาค่าเหมาะที่สุดของปัญหาต่อไปนี้ พร้อมทั้งวาดกราฟเพื่อยืนยันจุดคำตอบที่คำนวณได้

Minimize $f(\boldsymbol{x}) = -x_1^2 + x_2^2 - x_1x_2$

subject to $2x_1 - x_2 \geq 2$

$x_1 + x_2 \leq 4$

$x_1, x_2 \geq 0$

จงหาค่าเหมาะที่สุดของปัญหาต่อไปนี้ โดยใช้เงื่อนไข KKT พร้อมทั้งวาดกราฟเพื่อยืนยันจุดคำตอบที่คำนวณได้

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{subject to} \quad & x_1^2 + 2x_1x_2 + x_1^2 = 1 \\ & x_1^2 - x_2 \le 0 \end{aligned}$$

8. จงหาค่าเหมาะที่สุดของปัญหาต่อไปนี้ โดยใช้เงื่อนไข KKT พร้อมทั้งวาดกราฟเพื่อยืนยันจุดคำตอบที่คำนวณได้

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{subject to} \quad & 4 - x_1 - x_2^2 \le 0 \\ & x_1 - 3x_2 \ge 0 \\ & x_1 + 3x_2 \ge 0 \end{aligned}$$

9. จงหาค่าเหมาะที่สุดโดยใช้ระเบียบวิธีเกรเดียนต์ลดทอน และระเบียบวิธีเกรเดียนต์โปรเจกชัน โดยให้ใช้ความคลาดเคลื่อนน้อยกว่า 1×10^{-4}

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) = x_1^2 + (x_2 - 2)^2 \\ \text{subject to} \quad & x_1 + 4x_2 = 1 \end{aligned}$$

10. จงหาค่าเหมาะที่สุดโดยใช้ระเบียบวิธีเกรเดียนต์ลดทอน และระเบียบวิธีเกรเดียนต์โปรเจกชัน โดยให้ใช้ความคลาดเคลื่อนน้อยกว่า 1×10^{-4}

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) = -\frac{1}{\left(x_1^2 + x_2^2 + 10\right)} \\ \text{subject to} \quad & x_1 - x_2 = 3 \end{aligned}$$

11. จงหาค่าเหมาะที่สุดโดยใช้ระเบียบวิธีเกรเดียนต์ลดทอน และระเบียบวิธีเกรเดียนต์โปรเจกชัน โดยให้ใช้ความคลาดเคลื่อนน้อยกว่า 1×10^{-4}

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) = sin(x_1 + x_2) + x_3^2 + \frac{1}{3}\left(x_4 + x_5^4 + \frac{x_6}{2}\right) \\ \text{subject to} \quad & 8x_1 - 6x_2 + x_3 + 9x_4 + 4x_5 = 6 \\ & 3x_1 + 2x_2 - x_4 + 6x_5 + 4x_6 = -4 \end{aligned}$$

จงหาค่าเหมาะที่สุดโดยใช้ระเบียบวิธีเกรเดียนต์ลดทอน และระเบียบวิธีเกรเดียนต์โปรเจกชัน โดยให้ใช้ความคลาดเคลื่อนน้อยกว่า 1×10^{-4}

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) = (x_1 - 1)^2 + x_2^2 - 3x_1 + x_2 - 2 \\ \text{subject to} \quad & -x_1 + x_2 = 1 \\ & x_1 + x_2 + 2 \le 0 \end{aligned}$$

13. จงหาค่าเหมาะที่สุดโดยใช้ระเบียบวิธีกำหนดการลำดับกำลังสองและระเบียบวิธีการปรับโทษ ให้มีความคลาดเคลื่อนไม่เกิน 1×10^{-4}

Minimize $f(x) = x_1^2 + (x_2 - 2)^2$

subject to $x_1^2 + 4x_2^2 = 1$

$x_1^2 - x_2 \geq 0$

14. จงหาค่าเหมาะที่สุดโดยใช้ระเบียบวิธีกำหนดการลำดับกำลังสองและระเบียบวิธีการปรับโทษ ให้มีความคลาดเคลื่อนไม่เกิน 1×10^{-4}

Minimize $f(x) = x_1^{1/4} + \left(\dfrac{x_2}{x_1}\right)^{1/4} + \left(\dfrac{64}{x_2}\right)^{1/4}$

subject to $x_1 \geq 1,\ \ x_2 \geq x_1,\qquad x_2 \leq 64$

15. จงหาค่าเหมาะที่สุดโดยใช้ระเบียบวิธีกำหนดการลำดับกำลังสองและระเบียบวิธีการปรับโทษ ให้มีความคลาดเคลื่อนไม่เกิน 1×10^{-4}

Minimize $f(x) = x_1 + x_2^2$

subject to $25 - x_1^2 - x_2^2 \leq 0$

$9 - x_1^2 - x_2 \geq 0$

$0 \leq x_1 \leq 5$

$0 \leq x_2 \leq 10$

16. จงหาค่าเหมาะที่สุดโดยใช้ระเบียบวิธีกำหนดการลำดับกำลังสองและระเบียบวิธีการปรับโทษ ให้มีความคลาดเคลื่อนไม่เกิน 1×10^{-4}

Minimize $f(x) = x_1^2 + x_2^2 + x_3^2$

subject to $x_1x_2x_3 = 9$

$x_1 + x_2^2 \leq 6$

$x_1^2 + x_2 \leq 4$

$x_1 \geq 0, x_2 \geq 0$

บทที่ 6 การสร้างปัญหาค่าเหมาะที่สุด (Problem Formulation)

"Nothing in life is to be feared, it is to be understood"

6.1 เกริ่นนำ

รายละเอียดที่ได้นำเสนอไปนั้น เป็นเทคนิคการแก้ปัญหาค่าเหมาะที่สุดภายใต้ข้อกำหนดของฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับที่จัดเตรียมไว้ ในรูปของความสัมพันธ์ทางคณิตศาสตร์ที่ชัดเจน การนำเทคนิคการหาค่าเหมาะที่สุดไปใช้ในงานวิศวกรรมจำเป็นต้องกำหนดรูปแบบของปัญหาโดยผู้ออกแบบเป็นผู้กำหนด ทั้งฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับ บางกรณี ไม่มีรูปแบบที่ชัดเจน หรือข้อมูลจากระบบที่ต้องการอยู่ในรูปชุดข้อมูลเชิงตัวเลขที่ได้จากการทดสอบ ถึงแม้ระบบจะมีสมการหรือแบบจำลองที่แน่ชัด ปัญหาที่สำคัญที่สุด ได้แก่ การได้มาซึ่งค่าพารามิเตอร์หรือค่าสัมประสิทธิ์ของสมการหรือความสัมพันธ์เหล่านั้น ต้องพึ่งพาข้อมูลจากการทดสอบนั่นเอง จะพบว่า การเลือกฟังก์ชันทางคณิตศาสตร์เพื่อแทนชุดข้อมูลดังกล่าวมีความสำคัญเป็นอย่างมาก เนื่องจากรูปแบบของฟังก์ชันที่ใช้จะสะท้อนการมีอยู่ของคำตอบหรือจุดต่ำสุดนั่นเอง การเลือกฟังก์ชันที่เหมาะสมไม่มีกฎเกณฑ์ที่ตายตัว ขึ้นอยู่กับผู้ออกแบบระบบเป็นสำคัญและอยู่นอกเหนือขอบเขตการศึกษาในชั้นนี้ ในบทนี้จะนำเสนอการแปลงหรือการสร้างแบบจำลองทางคณิตศาสตร์ของปัญหาอย่างง่ายโดยใช้ฟังก์ชันพื้นฐานเป็นตัวนำเสนอ เช่น ฟังก์ชันเชิงเส้น หรือฟังก์ชันกำลังสอง เป็นต้น เพื่อเป็นแนวทางในการศึกษาและทำงานวิจัยต่อไป

6.2 การสร้างแบบจำลองด้วยเทคนิคกำลังสองน้อยที่สุด (Least-Square Modeling)

ภายใต้ชุดข้อมูลที่ได้จากการทดสอบ $(\hat{x}_i, \hat{y}_i)$ ใด ๆ สามารถประมาณฟังก์ชัน $y_i = f(x_i)$ ได้ โดยการกำหนดรูปแบบของฟังก์ชัน เช่น ฟังก์ชันเชิงเส้น ฟังก์ชันกำลังสอง ฟังก์ชันกำลังสาม ฟังก์ชันเอ็กซ์โพเนนเชียล หรืออื่น ๆ เป็นต้น ตัวอย่างเช่น การประมาณค่าฟังก์ชันที่พิจารณาด้วยสมการกำลังสอง โดยกำหนดให้ $y_i = f(x_i) = a + bx_i + cx_i^2$ เมื่อ a, b, c เป็นสัมประสิทธิ์ของฟังก์ชันที่ใช้ในการประมาณค่า และถือว่าเป็นเป้าหมายสำหรับการสร้างแบบจำลองทางคณิตศาสตร์นั่นเอง ค่าสัมประสิทธิ์ที่เหมาะสมจะต้องทำให้ค่าฟังก์ชัน y_i จากสมการประมาณค่าใกล้เคียงกับชุดข้อมูลจากการทดสอบ $\hat{y}_i$ มากที่สุด อย่างไรก็ตาม การพิจารณาค่าผลต่างอาจจะให้ค่าบวกหรือลบก็ได้ การพิจารณาเฉพาะขนาดของผลต่างถึงแม้สามารถทำได้ แต่มีปัญหาในเรื่องของการคำนวณค่าอนุพันธ์ ณ จุดบางจุด ข้อจำกัดดังกล่าวถูกกำจัดไปได้ โดยใช้หลักการกำลังสองน้อยที่สุด หมายความว่าให้นำความคลาดเคลื่อนยกกำลังสองนั่นเอง จากนั้น ปรับแต่งชุดพารามิเตอร์ให้ได้ผลต่างกำลังสองน้อยที่สุด นิยมเรียกว่า เทคนิคกำลังสองน้อยที่สุด (least-square technique) หรือความคลาดเคลื่อนกำลังสองน้อยที่สุด (least square error: LSE) ดังสมการปัญหาค่าเหมาะที่สุดต่อไปนี้

$$\text{Minimize} \qquad LSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}\left(a + bx_i + cx_i^2 - \hat{y}_i\right)^2 \qquad (6.1)$$

เมื่อ n คือจำนวนชุดข้อมูลทั้งหมดที่ใช้ในการพิจารณา

จะพบว่า ปัญหา LSE นี้เป็นปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขบังคับนั่นเอง อาจจะใช้ระเบียบวิธีชันที่สุด ระเบียบวิธีนิวตัน หรือระเบียบวิธีคล้ายนิวตันในการแก้ปัญหาก็ได้ พิจารณาจากตัวอย่างต่อไปนี้

ตัวอย่างที่ 6.1 จากชุดข้อมูลต่อไปนี้ จงประมาณค่าฟังก์ชัน $y = f(x)$ ด้วยสมการกำลังสอง

i	1	2	3	4	5
x_i	1	2	4	5	8
y_i	3	4	6	11	20

เขียนฟังก์ชันวัตถุประสงค์ของปัญหาความคลาดเคลื่อนกำลังสองน้อยที่สุด

$$\begin{aligned} LSE &= \sum_{i=1}^{5}\left(a + bx_i + cx_i^2 - \hat{y}_i\right)^2 \\ &= (a+b+c-3)^2 + (a+2b+4c-4)^2 + (a+4b+16c-6)^2 \\ &\quad + (a+5b+25c-11)^2 + (a+8b+64c-20)^2 \end{aligned}$$

แก้ปัญหาด้วยระเบียบวิธีคล้ายนิวตันโดยใช้สูตรการปรับปรุง BFGS จะได้ว่า

$$\begin{aligned} \text{Minimize} \qquad f(a,b,c) &= (a+b+c-3)^2 + (a+2b+4c-4)^2 + (a+4b+16c-6)^2 \\ &\quad + (a+5b+25c-11)^2 + (a+8b+64c-20)^2 \end{aligned}$$

ให้ $z = [a\ b\ c]^T$ นั่นคือ

$$\text{Minimize} \qquad f(z)$$

เริ่มต้นด้วย $z_0 = [0 \quad 0 \quad 0]^T$ และ $\boldsymbol{B}_0 = \boldsymbol{I}$

<u>รอบการคำนวณที่ 1</u>:

จะได้ทิศทางการค้นหา $\boldsymbol{p}_0 = -[\boldsymbol{B}_0]^{-1}\nabla f(z_0) = -\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1}\begin{bmatrix} -88 \\ -500 \\ -3340 \end{bmatrix} = \begin{bmatrix} 88 \\ 500 \\ 3340 \end{bmatrix}$

จะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

$$\text{Minimize} \qquad F(\lambda) = f(z_0 + \lambda \boldsymbol{p}_0)\, f\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \lambda\begin{bmatrix} 88 \\ 500 \\ 3340 \end{bmatrix}\right) = f\left(\begin{bmatrix} 88\lambda \\ 500\lambda \\ 3340\lambda \end{bmatrix}\right)$$

$$\lambda > 0$$

แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหากำลังสาม จะได้ $\lambda_0 = 9.81 \times 10^{-5}$

จะได้ $\quad z_1 = z_0 + \lambda_0 \boldsymbol{p}_0 = \begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} + \left(9.81 \times 10^{-5}\right)\begin{bmatrix} 88 \\ 500 \\ 3340 \end{bmatrix} = \begin{bmatrix} 0.0086 \\ 0.0490 \\ 0.3276 \end{bmatrix}$

$$f(z_1)=22.2182,\qquad \nabla f(z_1)=\begin{bmatrix}-13.8734\\-23.6314\\3.9032\end{bmatrix}$$

ะได้ว่า $\nabla f(z_1)=\begin{bmatrix}-13.8734\\-23.6314\\3.9032\end{bmatrix}\Rightarrow\|\nabla f(z_1)\|=27.6794>1\times10^{-4}$

คำนวณพารามิเตอร์เพิ่มเติม จะได้

$$s_0=z_1-z_0=\begin{bmatrix}0.0086\\0.0490\\0.3276\end{bmatrix}\quad y_0=\nabla f(z_1)-\nabla f(z_0)=\begin{bmatrix}74.1\\476.4\\3343.9\end{bmatrix}$$

$$B_1=B_0-\frac{(B_0s_0)(B_0s_0)^T}{s_0^TB_0s_0}+\frac{y_0y_0^T}{y_0^Ts_0}=\begin{bmatrix}5.9&31.5&221.4\\31.5&203.7&1422.7\\221.4&1422.7&9987.6\end{bmatrix}$$

<u>รอบการคำนวณที่ 2</u>:

จะได้ทิศทางการค้นหา $p_1=-[B_1]^{-1}\nabla f(z_1)=\begin{bmatrix}13.8794\\23.6649\\-3.6790\end{bmatrix}$

ะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

$$\text{Minimize}\qquad F(\lambda)=f\left(\begin{bmatrix}0.0086\\0.0490\\0.3276\end{bmatrix}+\lambda\begin{bmatrix}13.8794\\23.6649\\-3.6790\end{bmatrix}\right)$$

$$\lambda>0$$

แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหากำลังสาม จะได้ $\lambda_1=0.0454$

จะได้ $z_2=z_1+\lambda_1p_1=\begin{bmatrix}0.0086\\0.0490\\0.3276\end{bmatrix}+(0.0454)\begin{bmatrix}13.8794\\23.6649\\-3.6790\end{bmatrix}=\begin{bmatrix}0.6390\\1.1238\\0.1605\end{bmatrix}$

$$f(z_2)=4.8209,\qquad \nabla f(z_2)=\begin{bmatrix}-1.3380\\0.7722\\-0.0803\end{bmatrix}$$

จะได้ว่า $\nabla f(z_2)=\begin{bmatrix}-1.3380\\0.7722\\-0.0803\end{bmatrix}\Rightarrow\|\nabla f(z_2)\|=1.5469>1\times10^{-4}$

คำนวณพารามิเตอร์เพิ่มเติม จะได้

$$s_1=z_2-z_1=\begin{bmatrix}0.6303\\1.0747\\-0.1671\end{bmatrix}\quad y_1=\nabla f(z_2)-\nabla f(z_1)=\begin{bmatrix}12.5355\\24.4036\\-3.9835\end{bmatrix}$$

$$B_2 = B_1 - \frac{(B_1 s_1)(B_1 s_1)^T}{s_1^T B_1 s_1} + \frac{y_1 y_1^T}{y_1^T s_1} = \begin{bmatrix} 10.2 & 39.9 & 220.0 \\ 39.9 & 220.1 & 1420.0 \\ 220.0 & 1420.0 & 9988.0 \end{bmatrix}$$

<u>รอบการคำนวณที่ 3</u>:

จะได้ทิศทางการค้นหา $p_2 = -[B_2]^{-1} \nabla f(z_2) = \begin{bmatrix} 1.3813 \\ -0.6983 \\ 0.0689 \end{bmatrix}$

จะได้ปัญหาย่อยการค้นหาตามเส้นเป็น

$$\text{Minimize} \qquad F(\lambda) = f\left(\begin{bmatrix} 0.6390 \\ 1.1238 \\ 0.1605 \end{bmatrix} + \lambda \begin{bmatrix} 1.3813 \\ -0.6983 \\ 0.0689 \end{bmatrix}\right)$$

$$\lambda > 0$$

แก้ปัญหาโดยใช้ระเบียบวิธีการค้นหากำลังสาม จะได้ $\lambda_2 = 1.299$

จะได้ $z_3 = z_2 + \lambda_2 p_2 = \begin{bmatrix} 0.6390 \\ 1.1238 \\ 0.1605 \end{bmatrix} + (1.299)\begin{bmatrix} 1.3813 \\ -0.6983 \\ 0.0689 \end{bmatrix} = \begin{bmatrix} 2.4333 \\ 0.2167 \\ 0.2500 \end{bmatrix}$

$$f(z_3) = 3.2667, \qquad \nabla f(z_3) = \begin{bmatrix} 1.39\times10^{-10} \\ 5.96\times10^{-11} \\ 7.36\times10^{-10} \end{bmatrix}$$

จะได้ว่า $\nabla f(z_3) = \begin{bmatrix} 1.39\times10^{-10} \\ 5.96\times10^{-11} \\ 7.36\times10^{-10} \end{bmatrix} \Rightarrow \|\nabla f(z_3)\| = 7.5138\times10^{-10} < 1\times10^{-4}$

จะได้ผลเฉลยเป็น $z^* = \begin{bmatrix} a \\ b \\ c \end{bmatrix}^* = \begin{bmatrix} 2.4333 \\ 0.2167 \\ 0.2500 \end{bmatrix}$

ใช้การคำนวณทั้งสิ้น 3 รอบ จะได้สมการประมาณค่าเป็น

$$y = f(x) = 2.4333 + 0.2167x + 0.25x^2$$

นำมาเขียนกราฟจะได้ดังรูปต่อไปนี้

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 6.1</u>

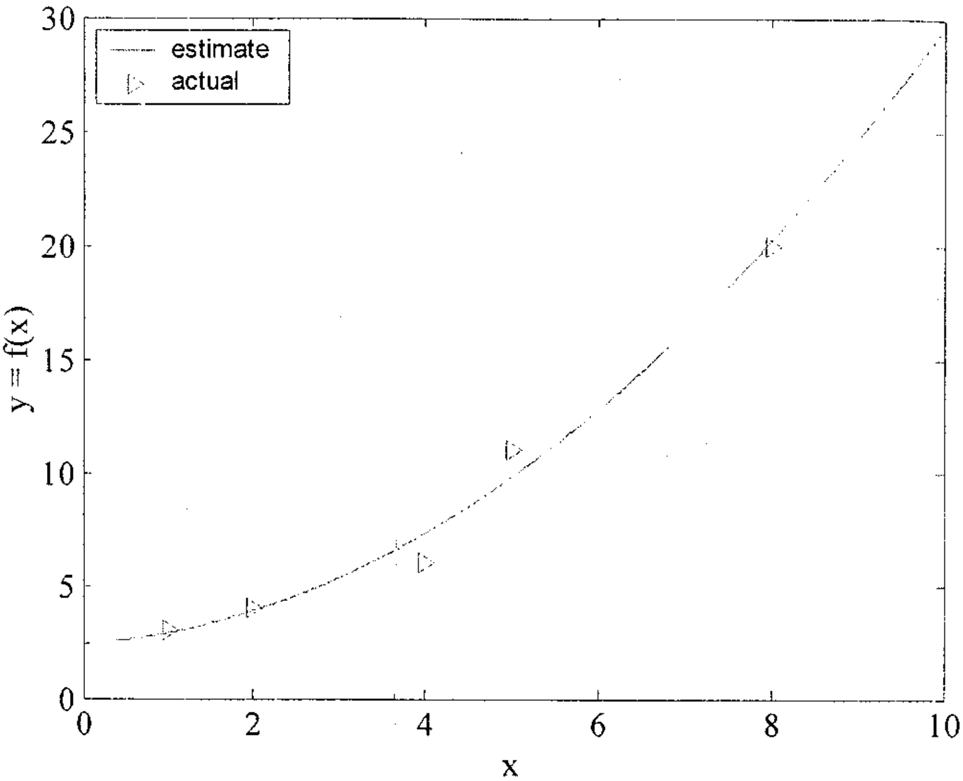
// SCILAB source program

```
// กำหนดฟังก์ชันวัตถุประสงค์                    f_lse01.sci
==========================
        f_lse01.sci
==========================
```

```
function f = f_lse01(x)
a=x(1);b=x(2);c=x(3);
f  =  (a+b+c-3)^2+(a+2*b+4*c-4)^2+(a+4*b+16*c-6)^2+(a+5*b+25*c-11)^2+(a+8*b+64*c-
20)^2;
endfunction
```

```
// กำหนดเกรเดียนต์ของฟังก์ชันวัตถุประสงค์                    g_lse01.sci
=========================
        g_lse01.sci
=========================
function g = g_lse01(x)
a=x(1);b=x(2);c=x(3);
g = [2*(a+b+c-3)+2*(a+2*b+4*c-4)+2*(a+4*b+16*c-6)+2*(a+5*b+25*c-11)+2*(a+8*b+64*c-
20);
  2*(a+b+c-3)+4*(a+2*b+4*c-4)+8*(a+4*b+16*c-6)+10*(a+5*b+25*c-
11)+16*(a+8*b+64*c-20);
  2*(a+b+c-3)+8*(a+2*b+4*c-4)+32*(a+4*b+16*c-6)+50*(a+5*b+25*c-
11)+128*(a+8*b+64*c-20)];
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('f_lse01.sci');                 ประกาศฟังก์ชันวัตถุประสงค์
--> exec('g_lse01.sci');                 ประกาศเกรเดียนต์ของฟังก์ชันวัตถุประสงค์
--> exec('lsearch.sci');                 ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('lsearch.sci');                 ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('flsearch.sci');                ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('cubicsearch.sci');             ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('fgradnum1.sci');               ประกาศฟังก์ชันเพื่อการคำนวณ
--> exec('bfgs.sci');                    ประกาศฟังก์ชันเพื่อการคำนวณ
--> [x,f,k]=bfgs(f_lse01,g_lse01,[0;0;0],eye(3,3),1e-4,1e-4,1);
 Iter     a           b           c           LSE         Max Grad
  0.    0.0000000   0.0000000   0.0000000   582.00000   3340.0000
  1.    0.0086321   0.0490463   0.3276290   22.218200   23.631365
  2.    0.6389622   1.1237875   0.1605496   4.8209037   1.3379589
  3.    2.4333333   0.2166667   0.2500000   3.2666667   7.365D-10
```



รูปที่ 6.1 การสร้างแบบจำลองด้วยเทคนิคความคลาดเคลื่อนกำลังสองน้อยที่สุด

อย่างไรก็ตาม แบบจำลองที่ได้ขึ้นอยู่กับการเลือกใช้สมการประมาณค่า ซึ่งไม่จำเป็นต้องเป็นฟังก์ชันกำลังสองเสมอไป การเลือกฟังก์ชันประมาณค่าที่ดีจะทำให้ LSE มีค่าน้อยที่สุด พิจารณาจากตัวอย่างที่ 6.2

ตัวอย่างที่ 6.2 จากชุดข้อมูลในตัวอย่างที่ 6.1 จงประมาณค่าฟังก์ชันด้วยสมการต่อไปนี้

$$y = f(x) = ae^{bx}$$

เขียนฟังก์ชันวัตถุประสงค์ของปัญหาความคลาดเคลื่อนกำลังสองน้อยที่สุด

$$LSE = \sum_{i=1}^{5}\left(ae^{bx_i} - \hat{y}_i\right)^2$$

$$= \left(ae^{b} - 3\right)^2 + \left(ae^{2b} - 4\right)^2 + \left(ae^{4b} - 6\right)^2 + \left(ae^{5b} - 11\right)^2 + \left(ae^{8b} - 20\right)^2$$

แก้ปัญหาด้วยระเบียบวิธีคล้ายนิวตันโดยใช้สูตรการปรับปรุง BFGS จะได้ว่า

$$\text{Minimize} \qquad f(a,b) = \left(ae^{b} - 3\right)^2 + \left(ae^{2b} - 4\right)^2 + \left(ae^{4b} - 6\right)^2 + \left(ae^{5b} - 11\right)^2 + \left(ae^{8b} - 20\right)^2$$

ให้ $z = [a\ b]^T$ นั่นคือ

$$\text{Minimize} \qquad f(z)$$

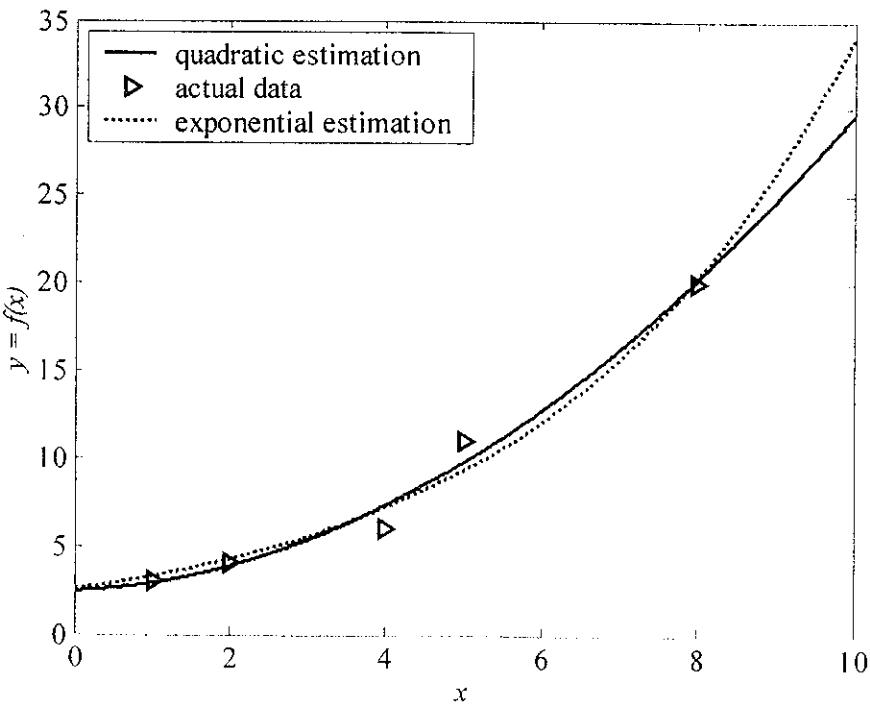
เริ่มต้นด้วย $z_0 = [0 \quad 0]^T$ และ $\boldsymbol{B}_0 = \boldsymbol{I}$ จะใช้การคำนวณทั้งสิ้น 18 รอบ สรุปได้ดังนี้

k	a	b	f	max. grad.
0.0000	0.0000	0	0.1948	1.3024
1.0000	9.4796	0.0459	158.5722	377.6354
2.0000	3.1925	0.2196	8.9918	450.5844
3.0000	3.1738	0.2294	6.8358	14.7266
4.0000	2.6191	0.2534	4.6543	117.3063
5.0000	2.5698	0.2581	4.5002	8.4377
6.0000	2.5418	0.2594	4.4943	1.4704
7.0000	2.5411	0.2595	4.4943	0.0047
8.0000	2.5410	0.2595	4.4943	0.0005
9.0000	2.5410	0.2595	4.4943	0.0045
10.0000	2.5410	0.2595	4.4943	0.0022
11.0000	2.5411	0.2595	4.4943	0.0034
12.0000	2.5410	0.2595	4.4943	0.0035
13.0000	2.5410	0.2595	4.4943	0.0017
14.0000	2.5411	0.2595	4.4943	0.0020
15.0000	2.5410	0.2595	4.4943	0.0001
16.0000	2.5410	0.2595	4.4943	0.0002
17.0000	2.5410	0.2595	4.4943	0.0001
18.0000	2.5410	0.2595	4.4943	0.0000

ใช้การคำนวณทั้งสิ้น 18 รอบ จะได้สมการประมาณค่าเป็น

$$y = f(x) = 2.541e^{0.2595x}$$

นำมาเขียนกราฟเทียบกับชุดข้อมูลและสมการประมาค่าในตัวอย่างที่ 6.1 จะได้ดังรูปต่อไปนี้



รูปที่ 6.2 การสร้างแบบจำลองด้วยเทคนิคความคลาดเคลื่อนกำลังสองน้อยที่สุด

นอกจากการใช้เทคนิคการหาค่าเหมาะที่สุดดังที่ได้กล่าวมาแล้วนั้น การแก้ปัญหาความคลาดเคลื่อนกำลังสองน้อยที่สุด สามารถดำเนินการได้ด้วยระเบียบวิธีเชิงตัวเลขที่พัฒนาขึ้นเพื่อให้สอดคล้องกับการแก้ปัญหาในลักษณะนี้มากที่สุด หลักการที่ใช้จะอาศัยการสร้างทิศทางการค้นหาเหมือนกับระเบียบวิธีนิวตัน ต่างกันที่การคำนวณเฮสเซียนของฟังก์ชันวัตถุประสงค์ จะใช้การประมาณค่าด้วยสูตรอย่างง่าย เรียกระเบียบวิธีนี้ว่า ระเบียบวิธีเกาส์-นิวตัน (Gauss-Newton method) พิจารณาปัญหาในรูปทั่วไปดังต่อไปนี้

$$\underset{x}{\text{Minimize}} \qquad LSE = \frac{1}{2}\sum_{i=1}^{n} f_i(x)^2 = \frac{1}{2}F(x)^T F(x) \tag{6.2}$$

เมื่อ $F(x) = [f_1(x)\, f_2(x) \ldots f_n(x)]^T$ โดยที่ $\nabla F(x)$ เป็นเมตริกซ์จัตุรัส

ประมาณทิศทางการค้นหาของนิวตันจากสมการต่อไปนี้ $\nabla^2 f(x)\, p = -\nabla f(x)$ โดยใช้สมการ $\nabla F(x) \nabla F(x)^T\, p = -\nabla F(x) \nabla F(x)$ อย่างไรก็ตาม ระเบียบวิธีเกาส์-นิวตันไม่อยู่ในหัวข้อที่จะนำเสนอในรายวิชานี้ ผู้สนใจสามารถศึกษาเพิ่มเติมจากหนังสือหรือตำรารายวิชาการคำนวณเชิงตัวเลข (numerical computation) ได้ในรายละเอียด

6.3 การสร้างฟังก์ชันวัตถุประสงค์ทางเศรษฐศาสตร์ (Formulation of Economical Objective Functions)

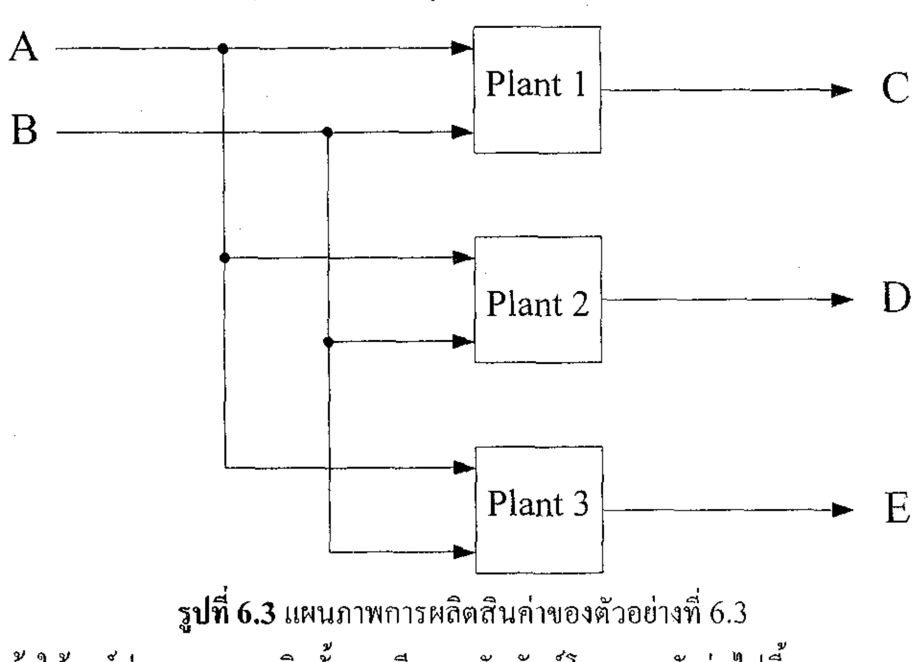
งานทางด้านวิศวกรรมมีความแตกต่างกันไปตามแขนงวิชา ความสนใจหรือเป้าหมายของผู้ออกแบบระบบ ดังนั้น ผู้ออกแบบต้องกำหนดเป้าหมายที่ชัดเจนในการดำเนินงาน เป้าหมายดังกล่าวจะเรียกว่า ฟังก์ชันวัตถุประสงค์ที่ผู้ออกแบบต้องกำหนด ส่วนใหญ่แล้วจะมีเงื่อนไขบังคับปรากฏอยู่ด้วยเสมอ การกำหนดเป้าหมายนั้นมีหลากหลาย เช่น การออกแบบผลิตภัณฑ์เพื่อให้ใช้<u>พลังงานน้อยที่สุด</u> การออกแบบผลิตภัณฑ์เพื่อให้มี<u>ต้นทุนการผลิตต่ำที่สุด</u> การออกแบบกระบวนการผลิตเพื่อให้การผลิตชิ้นงานใช้<u>เวลาน้อยที่สุด</u> หรือการออกแบบจุดทำงานเพื่อให้สิ้นเปลือง<u>ค่าใช้จ่ายในการดำเนินงานน้อยที่สุด</u> เป็นต้น การออกแบบระบบด้วยเป้าหมายด้านสมรรถนะเป็นลักษณะเฉพาะของระบบ ไม่สามารถระบุให้อยู่ในรูปทั่วไปได้ เป็นรายละเอียดในการศึกษาแบบจำลองระบบ (system modeling) ในบทนี้จะยกตัวอย่างระบบเพื่อนำมาใช้สร้างฟังก์ชันวัตถุประสงค์ทางด้านสมรรถนะของระบบเพื่อให้เข้าใจหลักการเบื้องต้นเท่านั้น เมื่อไม่พิจารณาสมรรถนะที่เฉพาะเจาะจง การสร้างฟังก์ชันวัตถุประสงค์จากต้นทุนหรือผลกำไรเป็นสิ่งที่น่าสนใจและถือได้ว่าเป็นปลายทางของงานออกแบบทางวิศวกรรมส่วนใหญ่ ในส่วนนี้จะนำเสนอการสร้างฟังก์ชันวัตถุประสงค์จากการพิจารณาต้นทุนและผลกำไร ซึ่งมีความซับซ้อนเกี่ยวพันถึงสิ่งที่เรียกว่า การไหลของเงิน (cash flows) ประกอบไปด้วยรายรับ (income) และรายจ่าย (expense) ของโครงการ นอกจากนี้ ในแต่ละส่วนยังแยกย่อยเป็นองค์ประกอบต่าง ๆ ที่อาจจะมีค่าคงที่ เช่น เงินลงทุนทางด้านโครงสร้างหลักของโครงการที่เรียกว่า (capital cost) หรือแปรไปตามช่วงของเวลาได้ เช่น เงินรายจ่ายค่าดำเนินการ (operating cost) การศึกษาพลวัตของต้นทุน

รือผลกำไรในช่วงเวลาที่เฉพาะเจาะจง จำเป็นต้องทราบหลักการของการลดค่าลงเนื่องจากการหลของเงิน (discounted cash flows: DCF) อันเนื่องมาจากหลายสาเหตุ ดังรายละเอียดต่อไปนี้

6.3.1 เงินการลงทุน ค่าดำเนินการ และผลกำไร (investment cost, operating cost and profit)

เงินลงทุน ค่าดำเนินการและผลกำไร เป็นองค์ประกอบในขั้นพื้นฐานในการสร้างฟังก์ชันวัตถุประสงค์ทางเศรษฐศาสตร์ เงินลงทุนอาจจะมีค่าคงที่หรือไม่ก็ได้ เช่น ค่าใช้จ่ายการสร้างตัวอาคาร ระบบต้นกำลัง เครื่องจักรหรือวัสดุอุปกรณ์ในการผลิต เป็นต้น ถ้ามีการวางแผนที่แน่นอนเงินลงทุนในส่วนนี้มีค่าคงที่ และจะใช้จ่ายในช่วงเริ่มต้นโครงการ ทำให้ในฟังก์ชันวัตถุประสงค์อาจจะไม่ต้องพิจารณาในส่วนนี้ได้ ในส่วนของค่าดำเนินการนั้น ประกอบไปด้วยค่าแรง ค่าบำรุงรักษา เงินเดือนผู้ปฏิบัติการ สวัสดิการต่าง ๆ ซึ่งแปรเปลี่ยนไปตามเงื่อนไขการทำงาน กำไรเป็นอีกปัจจัยที่สำคัญ เป้าหมายของการประกอบกิจการทุกประเภทเพื่อแสวงหาผลกำไรมากที่สุดจากการขายสินค้านั่นเอง พิจารณาจากตัวอย่างต่อไปนี้

ตัวอย่างที่ 6.3 กระบวนการผลิตสินค้าชนิดหนึ่งประกอบไปด้วยสายการผลิต 3 ส่วน ได้ผลผลิตทั้งสิ้น 3 ผลิตภัณฑ์ โดยใช้วัตถุดิบ 2 ชนิด ดังรูป



รูปที่ 6.3 แผนภาพการผลิตสินค้าของตัวอย่างที่ 6.3

ถ้าให้องค์ประกอบการผลิตทั้งสามมีความสัมพันธ์โดยมวล ดังต่อไปนี้

ผลิตภัณฑ์ C: $A + 3B \rightarrow C$

ผลิตภัณฑ์ D: $2A + 3B \rightarrow D$

ผลิตภัณฑ์ E: $A + B \rightarrow E$

วัตถุดิบทั้ง 2 ชนิดมีข้อกำหนดดังต่อไปนี้

A:	1.2 บาท/กิโลกรัม	จัดซื้อได้สูงสุด	4000 กิโลกรัม/วัน
B:	2.0 บาท/กิโลกรัม	จัดซื้อได้สูงสุด	2500 กิโลกรัม/วัน

ผลิตภัณฑ์ทั้ง 3 ชนิดมีข้อกำหนดดังต่อไปนี้

Process	Product	Processing cost (บาท/กิโลกรัม-ผลิตภัณฑ์)	Selling price (บาท/กิโลกรัม-ผลิตภัณฑ์)
1	C	1.5	4.0
2	D	0.5	3.8
3	E	1.0	3.3

โดยมีอัตราการขายสูงสุดได้

ผลิตภัณฑ์ C ไม่เกิน 1200 กิโลกรัม/วัน

ผลิตภัณฑ์ D ไม่เกิน 1800 กิโลกรัม/วัน

ผลิตภัณฑ์ E ไม่เกิน 1400 กิโลกรัม/วัน

กำหนดตัวแปรควบคุมดังต่อไปนี้

x_1 แทนมวลของวัตถุดิบ A ที่ใช้ต่อวัน

x_2 แทนมวลของวัตถุดิบ B ที่ใช้ต่อวัน

x_3 แทนมวลของผลิตภัณฑ์ C ที่ผลิตได้ต่อวัน

x_4 แทนมวลของผลิตภัณฑ์ D ที่ผลิตได้ต่อวัน

x_5 แทนมวลของผลิตภัณฑ์ E ที่ผลิตได้ต่อวัน

เริ่มต้นพิจารณาสมดุลของมวลในกระบวนการผลิต

ผลิตภัณฑ์ C: ใช้ $A = 0.25x_3$ และใช้ $B = 0.75x_3$ ได้ $C = x_3$

ผลิตภัณฑ์ D: ใช้ $A = 0.4x_4$ และใช้ $B = 0.6x_4$ ได้ $D = x_4$

ผลิตภัณฑ์ E: ใช้ $A = 0.5x_5$ และใช้ $B = 0.5x_5$ ได้ $E = x_5$

นั่นคือ

$$x_1 = 0.25x_3 + 0.4x_4 + 0.5x_5 \quad \text{กิโลกรัมต่อวัน}$$

$$x_2 = 0.75x_3 + 0.6x_4 + 0.5x_5 \quad \text{กิโลกรัมต่อวัน}$$

อย่างไรก็ตามเงื่อนไขดังกล่าวไม่จำเป็นต้องอยู่ในรูปสมการเนื่องจากโดยปกติจำนวนวัตถุดิบต้องมากกว่าที่จะใช้ในการผลิต จะได้เงื่อนไขบังคับอสมการในรูป

$$x_1 - 0.25x_3 - 0.4x_4 - 0.5x_5 \geq 0 \quad \text{กิโลกรัมต่อวัน}$$

$$x_2 - 0.75x_3 - 0.6x_4 - 0.5x_5 \geq 0 \quad \text{กิโลกรัมต่อวัน}$$

พิจารณาต้นทุนที่ใช้

$$MC \text{ (material cost)} = 1.2x_1 + 2.0x_2 \quad \text{บาทต่อวัน}$$

$$PC \text{ (processing cost)} = 1.5x_3 + 0.5x_4 + x_5 \quad \text{บาทต่อวัน}$$

พิจารณารายได้จากการขายผลิตภัณฑ์

$$SI \text{ (selling income)} = 4.0x_3 + 3.8x_4 + 3.3x_5$$

ดังนั้น กำไร (profit) จากการขายสินค้ามีค่าเป็น

$$profit = SI - MC - PC = -1.2x_1 - 2x_2 + 2.5x_3 + 3.3x_4 + 2.3x_5$$

อย่างไรก็ตาม เนื่องจากปัญหาค่าเหมาะที่สุดที่นำเสนออยู่ในรูปปัญหาค่าต่ำที่สุด ดังนั้นการกำหนดฟังก์ชันวัตถุประสงค์ไม่สามารถเขียนในรูปการหาค่ากำไรสูงสุดได้ ต้องเปลี่ยนรูปแบบเป็นการค่าลบของกำไรต่ำที่สุดแทน (maximize profit $\triangleq$ minimize –profit) จะได้ปัญหาค่าเหมาะที่สุดเป็น

$$\begin{aligned}
\text{Minimize} \quad & f(x) = 1.2x_1 + 2x_2 - 2.5x_3 - 3.3x_4 - 2.3x_5 \\
\text{Subject to} \quad & x_1 - 0.25x_3 - 0.4x_4 + 0.5x_5 \ge 0 \\
& x_2 - 0.75x_3 - 0.6x_4 - 0.5x_5 \ge 0 \\
& 0 \le x_1 \le 4000 \\
& 0 \le x_2 \le 2500 \\
& 0 \le x_3 \le 1200 \\
& 0 \le x_4 \le 1800 \\
& 0 \le x_5 \le 1400
\end{aligned}$$

การแก้ปัญหานี้ทำได้โดยใช้ระเบียบวิธีซิมเพล็กซ์ดังที่ได้นำเสนอในบทที่ 4 อย่างไรก็ตาม โดยใช้ SCILAB's Optimization TOOLBOX ในการแก้ปัญหาการกำหนดการเชิงเส้นผ่านฟังก์ชัน "linpro.sci" จะได้ผลเฉลยเป็น

$$\boldsymbol{x}^* = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}^* = \begin{bmatrix} 1660 \\ 2500 \\ 960 \\ 1800 \\ 1400 \end{bmatrix} \Rightarrow f\left(\boldsymbol{x}^*\right) = -4568$$

นั่นคือ กำไรสูงสุดมีค่าเท่ากับ 4,568 บาทต่อวัน

6.3.2 ค่าของเงินที่แปรผันตามเวลา (time value of money)

ในโครงการขนาดใหญ่ เงินลงทุนส่วนใหญ่จะถูกใช้ไปในช่วงเริ่มต้นของการดำเนินการ ค่าใช้จ่ายดำเนินการเกิดสามารถเกิดขึ้นได้ตลอดระยะเวลาโครงการ ผลกำไรก็เช่นเดียวกัน ดังนั้นการประเมินจุดคุ้มทุนต้องพิจารณาถึงเงินลงทุน (ส่วนใหญ่จะถูกใช้เมื่อเริ่มโครงการ ถือว่าเป็นมูลค่าในปัจจุบัน: present value) ในขณะที่ค่าดำเนินการ และผลกำไรเกิดขึ้นหลังจากที่ได้ลงทุนไปแล้ว และแตกต่างกันไปในแต่ละปี (ถือว่าเป็นมูลค่าในอนาคต: future value) มูลค่าของเงิน 5,000 บาท ในปัจจุบันกับ 5,000 บาทในอีก 5 ปีข้างหน้า ไม่จำเป็นต้องมีมูลค่าเท่ากัน มีสาเหตุมาจากการลดค่าหรือเพิ่มค่าของเงิน ดังนั้น การลงทุน 5,000 บาท โดยคาดหวังให้ได้ 5,000 บาท กลับคืนมาในอนาคตนั้น ไม่สมเหตุผลในทางเศรษฐศาสตร์ การหาจุดคุ้มทุนเป็นการประเมินรายได้หรือกำไรในอนาคตเทียบกับปัจจุบันที่กำลังพิจารณานั่นเอง มูลค่าดังกล่าวอาจจะเพิ่มขึ้น (positive cash flow) หรือลดลง (negative cash flow) ก็ได้ กำหนดให้เงินลงทุนในปัจจุบันมีมูลค่า P

(present worth) ด้วยอัตราดอกเบี้ย i ต่อปีคงที่ (อาจจะเป็นดอกเบี้ยเงินฝากหรือเงินกู้ก็ได้) จะได้มูลค่าเงินในอนาคต F_k (future worth) ปีที่ k เป็น

$$F_k = P\left(1+i\right)^k \tag{6.3}$$

ถ้าให้เงินลงทุน P ถูกแบ่งจ่ายคืนแบบรายปี ภายในระยะเวลา n ปี จะได้เงินคืนรวมทั้งหมดเป็น

$$P_T = P_1 + P_2 + \cdots + P_n \tag{6.4}$$

นั่นคือ

$$P_T = \frac{F_1}{\left(1+i\right)} + \frac{F_2}{\left(1+i\right)^2} + \cdots + \frac{F_n}{\left(1+i\right)^n} \tag{6.5}$$

จะได้ F_k แทนเงินที่ต้องจ่ายคืน ซึ่งเป็นมูลค่าในอนาคต ถ้าให้อัตราการจ่ายเงินคืนมีค่าคงที่ (constant cash flow) เท่ากับ $F = F_1 = F_2 = \ldots = F_n$ ดังนั้น

$$P_T = \frac{F}{\left(1+i\right)} + \frac{F}{\left(1+i\right)^2} + \cdots + \frac{F}{\left(1+i\right)^n} \tag{6.6}$$

ย้ายข้างสมการจะได้ความสัมพันธ์ดังต่อไปนี้

$$F = \left[\frac{i\left(1+i\right)^n}{\left(1+i\right)^n - 1}\right] P_T = r \times P_T \tag{6.7}$$

เรียก r ว่าเป็น ตัวปรับคูณการคืนเงิน (repayment multiplier) จะพบว่า F เป็นยอดเงินที่ต้องจ่ายคืนต่อปีนั่นเอง พิจารณาจากตัวอย่างต่อไปนี้

ตัวอย่างที่ 6.4 นักลงทุนกู้เงินจากธนาคารเพื่อประกอบธุรกิจจำนวน 5 ล้านบาท โดยต้องเสียดอกเบี้ยในอัตรา 10% และมีการคืนเงินยืมในอัตราคงที่ทุกปีตลอด 20 ปี จงคำนวณเงินคืนต่อปี และถ้าไม่มีการคืนเงินยืมเลยตลอด 20 ปี จะต้องจ่ายเงินคืนทั้งหมดเท่าไหร่ในปีที่ 20

วิธีทำ จากโจทย์ $P_T = 5{,}000{,}000$ บาท $i = 0.1$ (10%), $n = 20$

$$F = \left[\frac{i\left(1+i\right)^n}{\left(1+i\right)^n - 1}\right] P_T = \left[\frac{0.1\times\left(1+0.1\right)^{20}}{\left(1+0.1\right)^{20} - 1}\right] \times 5{,}000{,}000 = 587{,}298 \text{ บาท}$$

นั่นคือ ในเวลา 20 ปี ต้องคืนเงินทั้งสิ้น $20\times587{,}298$ บาท = 11.75 ล้านบาท เปรียบเทียบกรณีที่คืนเงินครั้งเดียวในปีที่ 20 จะได้

$$F_{20} = P(1 + i)^{20} = 5\times(1 + 0.1)^{20} = 33.64 \text{ ล้านบาท}$$

จะเห็นได้ว่า ค่าของเงินแปรเปลี่ยนไปได้ตามเวลา การสร้างฟังก์ชันวัตถุประสงค์ทางด้านเศรษฐศาสตร์ที่สมบูรณ์ต้องคิดปัจจัยดังกล่าวด้วย การประเมินผลประกอบการทำได้หลากหลายวิธี วิธีหนึ่งที่ได้รับความนิยมอย่างแพร่หลายและคิดผลของมูลค่าเงินตามเวลาที่ผ่านไป ได้แก่ การประเมินมูลค่าปัจจุบันสุทธิ (net present value: NPV) กำหนดโดย

$$NPV = \sum_{j=1}^{n} \frac{F_j}{(1+i)^j} - I_0 \tag{6.8}$$

สมการนี้ใช้ได้เฉพาะโครงการที่มีการลงทุนเริ่มต้นเพียงครั้งเดียวเท่านั้น และอยู่ภายใต้การพิจารณาการไหลของเงินไม่คงที่ (variable cash flow) สำหรับโครงการที่มีการลงทุนเพิ่มเติมในปีต่อไป ต้องพิจารณามูลค่าของเงินลงทุนที่เพิ่มเข้ามาด้วยดังนี้

$$NPV = \sum_{j=1}^{n} \frac{F_j}{(1+i)^j} - \sum_{j=0}^{n-1} \frac{I_j}{(1+i)^j} \tag{6.9}$$

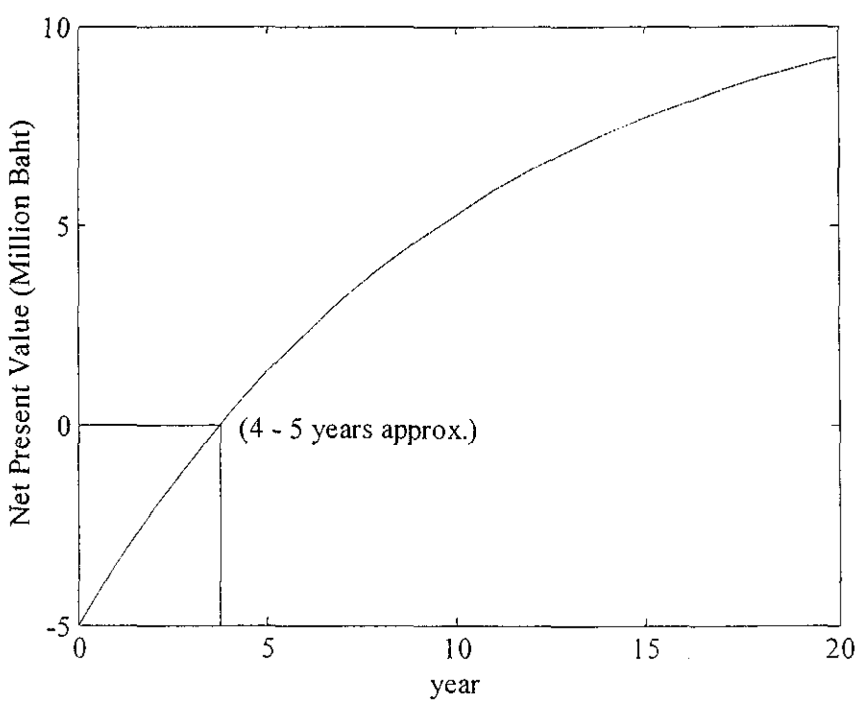
ในกรณีที่รายได้หรือผลประกอบการสุทธิที่หักรายจ่ายและภาษีเรียบร้อยแล้วมีค่าคงที่ทุกปี (constant cash flow) สามารถคำนวณได้ง่ายขึ้น ดังนี้

$$NPV = \left[\frac{(1+i)^n - 1}{i(1+i)^n} \right] F - I_0 \tag{6.10}$$

ในตัวอย่างที่ 6.3 ถ้าการประกอบกิจการดังกล่าว กู้เงินมาทั้งสิ้น 5 ล้านบาท ด้วยอัตราดอกเบี้ย 10% ดังตัวอย่างที่ 6.4 ผลการคำนวณจากตัวอย่างที่ 6.3 ได้กำไร 4,568 บาทต่อวัน คิดเป็น 4568×365 = 1,667,320 บาทต่อปี หรือ 1.67 ล้านบาทต่อปี และกำหนดให้คงที่ตลอดช่วงเวลาที่พิจารณา นั่นคือ

$$NPV_n = \left[\frac{(1+0.1)^n - 1}{0.1 \times (1+0.1)^n} \right] 1.67 - 5.0$$

เมื่อ n คือ จำนวนปีที่ใช้ นำไปสร้างกราฟโดยการแปรค่า n จะได้กราฟดังต่อไปนี้



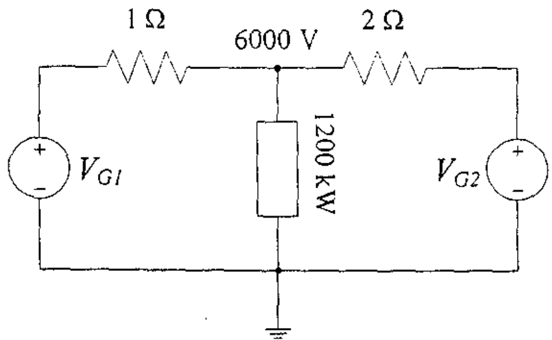
รูปที่ 6.4 จุดคุ้มทุนจากการประเมิน NPV

ในตัวอย่างที่กำหนด อยู่ภายใต้ผลประกอบการคงที่ในแต่ละปี แต่ในความเป็นจริงแล้ว บางปีอาจจะมีกำไรสูง บางปีอาจจะมีกำไรต่ำ หรือบางปีขาดทุน เป็นต้น การทำนายยอดขายด้วยทฤษฎีความน่าจะเป็นร่วมกับการแก้ปัญหาค่าเหมาะที่สุดนี้ จะเรียกว่า กำหนดการเฟ้นสุ่ม (stochastic programming) หรือ กำหนดการความน่าจะเป็น (probabilistic programming) ซึ่งอยู่นอกเหนือขอบเขตของการศึกษาในชั้นนี้

6.4 การประยุกต์ปัญหาค่าเหมาะที่สุดที่ถูกคัดสรร (Selected Applications in Engineering Optimization)

การเขียนฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับ เป็นหัวใจสำคัญในการเริ่มต้นงานวิจัย ผลเฉลยที่ได้ไม่สำคัญมากนักเนื่องจากถ้าการกำหนดปัญหามีความรัดกุมและชัดเจน ผลเฉลยสามารถหาได้ไม่ยากด้วยเทคนิคทางคณิตศาสตร์ที่มีประสิทธิภาพ ดังนั้น ในส่วนนี้จะนำเสนอตัวอย่างการสร้างปัญหาค่าเหมาะที่สุด ดังต่อไปนี้

ตัวอย่างที่ 6.5 เกาะแห่งหนึ่ง ไม่มีไฟฟ้าจากระบบส่งจ่ายของการไฟฟ้าใช้งาน ดังนั้น อ.บ.ต. ประจำเกาะดังกล่าวได้ติดตั้งเครื่องกำเนิดไฟฟ้ากระแสตรง 2 ชุด ดังรูปเพื่อจ่ายโหลดรวมทั้งสิ้น 1200 kW ที่ระดับแรงดัน 6000 V คร่อมโหลด จงหาการจ่ายกำลังงานไฟฟ้าในหน่วยกิโลวัตต์ที่เหมาะสมสำหรับเครื่องกำเนิดไฟฟ้าทั้ง 2 เพื่อให้เกิดกำลังงานสูญเสียในสายส่งน้อยที่สุด โดยให้มีแรงดันที่เครื่องกำเนิดไฟฟ้าในช่วง 5400 – 6600 V



รูปที่ 6.5 ระบบไฟฟ้าของเกาะในตัวอย่างที่ 6.5

กำหนดให้ V_{G1}, V_{G2} เป็นแรงดันไฟฟ้าที่ผลิตโดยเครื่องกำเนิดไฟฟ้าแต่ละเครื่อง คำนวณกระแสโหลดได้เป็น 200 A

จาก KVL จะได้

$$\frac{V_{G1}-6000}{1}+\frac{V_{G2}-6000}{2}=200$$

$$V_{G1}+0.5V_{G2}=9200$$

นั่นคือ จุดคำตอบที่เป็นไปได้จะต้องสอดคล้องกับสมการนี้เสมอ

นอกจากนี้ จากเงื่อนไขแรงดัน

$$5400 \le V_{G1} \le 6600$$
$$5400 \le V_{G2} \le 6600$$

คำนวณกำลังงานสูญเสียได้เป็น

$$P_{losses} = (V_{G1} - 6000)^2 + \frac{(V_{G2} - 6000)^2}{2}$$

จะได้รูปแบบปัญหาดังนี้

$$\begin{aligned} \text{Minimize} \quad & P_{losses} = (V_{G1} - 6000)^2 + \frac{(V_{G2} - 6000)^2}{2} \\ \text{Subject to} \quad & V_{G1} + 0.5V_{G2} = 9200 \\ & 5400 \le V_{G1} \le 6600 \\ & 5400 \le V_{G2} \le 6600 \end{aligned}$$

ตัวอย่างที่ 6.6 ระบบไฟฟ้ากำลังแห่งหนึ่งทำการจ่ายโหลดด้วยโรงไฟฟ้าพลังน้ำจากเขื่อนแห่งหนึ่ง และโรงไฟฟ้าพลังความร้อนที่ใช้ถ่านหินเป็นเชื้อเพลิง กำหนดให้โหลดใน 1 วัน มีค่าดังตาราง ต่อไปนี้

hours	0.00 – 8.00	8.00 – 16.00	16.00 – 24.00
load	32 MW	90 MW	125 MW

- โรงไฟฟ้าพลังน้ำมีความสัมพันธ์ระหว่างกำลังผลิตกับอัตราการไหลของน้ำผ่านกังหันน้ำดังนี้

$$q_h = 300 + 15P_h$$

เมื่อ q_h = อัตราการไหลของน้ำ (acre-ft/h)

P_h = กำลังผลิตของโรงไฟฟ้าพลังน้ำ (MW), $0 \le P_h \le 100$ MW

- โรงไฟฟ้าพลังความร้อนมีความสัมพันธ์ระหว่างกำลังผลิตกับค่าเชื้อเพลิงดังนี้

$$f_s = 54 + 12P_s + 0.02P_s^2$$

เมื่อ f_s = ค่าเชื้อเพลิงของโรงไฟฟ้าพลังความร้อน (บาท/MWh)

P_s = กำลังผลิตของโรงไฟฟ้าพลังความร้อน (MW), $10 \le P_s \le 50$ MW

กำหนดให้ปริมาตรน้ำเหนือเขื่อนที่ได้รับการจัดสรรสำหรับเขื่อนดังกล่าวเพื่อใช้ในการผลิตกระแสไฟฟ้าในวันดังกล่าวมีปริมาตร 36,000 acre-ft และจากการจัดสรรนี้วิศวกรโรงไฟฟ้าต้องใช้น้ำให้หมดพอดีใน 1 วัน

ให้ทำการจัดสรรกำลังผลิตของโรงไฟฟ้าพลังน้ำและโรงไฟฟ้าพลังความร้อนทั้ง 2 เพื่อให้ได้ต้นทุนการผลิตไฟฟ้าต่ำที่สุด (ปัญหานี้เป็นปัญหาที่เรียกว่า hydro-thermal coordination) กำหนดตัวแปรกำลังไฟฟ้าที่โรงไฟฟ้าแต่ละโรงผลิตในแต่ละช่วงเวลาเป็น

P_{h1}, P_{h2}, P_{h3} สำหรับโรงไฟฟ้าพลังน้ำในช่วงเวลาการจ่ายโหลดที่ 1, 2 และ 3 ตามลำดับ

P_{s1}, P_{s2}, P_{s3} สำหรับโรงไฟฟ้าพลังความร้อนในช่วงเวลาการจ่ายโหลดที่ 1, 2 และ 3 ตามลำดับ

การแก้ปัญหาดังกล่าวถ้าพิจารณาแบบเต็มรูปแบบ จะได้ทั้งสิ้น 6 ตัวแปร อย่างไรก็ตาม ปัญหาอาจจะสามารถลดรูปโดยการกำจัดตัวแปรได้ ในตัวอย่างนี้ ถ้าใช้หลักการสมดุลของพลังงานในแต่ละช่วงเวลานั่นคือ generation = demand จะได้เงื่อนไขบังคับสมการ 3 ชุดดังนี้

ช่วงที่ 1: $P_{h1} + P_{s1} = 32 \quad \rightarrow \quad P_{h1} = 32 - P_{s1}$

ช่วงที่ 2: $P_{h2} + P_{s2} = 90 \quad \rightarrow \quad P_{h2} = 90 - P_{s2}$

ช่วงที่ 3: $P_{h3} + P_{s3} = 125 \quad \rightarrow \quad P_{h3} = 125 - P_{s3}$

การลดรูปแบบนี้จะเหลือตัวแปรเพียง 3 ตัว ได้แก่ P_{s1}, P_{s2}, P_{s3} _อย่างไรก็ตาม การลดรูปตัวแปรอาจจะไม่สามารถใช้ได้กับทุกปัญหา ขอให้อย่ายึดถือหลักการนี้ว่าจะสามารถทำได้ในทุก ๆ กรณี_ เพื่อให้ง่ายในการสร้างปัญหาค่าเหมาะที่สุดเพื่อใช้เป็นตัวอย่างในการศึกษา ในที่นี้จะใช้การลดรูปดังกล่าว จะได้ว่า

<u>ฟังก์ชันวัตถุประสงค์</u>: ค่าเชื้อเพลิงในการผลิตกำลังงานไฟฟ้าใน 1 วัน

<u>เงื่อนไขบังคับ</u>: ขอบเขตของกำลังผลิตของโรงไฟฟ้าแต่ละโรง

ปริมาตรน้ำสูงสุดที่ได้รับการจัดสรรใน 1 วัน

เนื่องจากต้นทุนการผลิตเกิดขึ้นที่โรงไฟฟ้าพลังความร้อนเท่านั้น โรงไฟฟ้าพลังน้ำไม่มีต้นทุนในการผลิต แต่ปริมาตรน้ำที่ได้รับการจัดสรรมีจำกัด จะได้ฟังก์ชันวัตถุประสงค์ดังนี้

$$F_T = 8\sum_{i=1}^{3}\left(54 + 12P_{s,i} + 0.02P_{s,i}^2\right)$$

สำหรับเงื่อนไขบังคับที่เป็นขีดจำกัดของกำลังผลิต จะได้ว่า

$$0 \le P_{h,1} \le 100 \;\rightarrow\; 0 \le 32 - P_{s,1} \le 100 \;\rightarrow\; -68 \le P_{s,1} \le 32$$
$$0 \le P_{h,2} \le 100 \;\rightarrow\; 0 \le 90 - P_{s,2} \le 100 \;\rightarrow\; -10 \le P_{s,2} \le 90$$
$$0 \le P_{h,3} \le 100 \;\rightarrow\; 0 \le 125 - P_{s,2} \le 100 \;\rightarrow\; 25 \le P_{s,2} \le 125$$
$$10 \le P_{s,1} \le 50$$
$$10 \le P_{s,2} \le 50$$
$$10 \le P_{s,3} \le 50$$

เนื่องจากจำนวนตัวแปรถูกลดรูปลง ดังนั้นจำนวนของเงื่อนไขขีดจำกัดจะถูกลดจำนวนลงด้วยเช่นกัน จากเงื่อนไขทั้ง 6 ชุด จะได้ว่า

$$10 \le P_{s,1} \le 32$$
$$10 \le P_{s,2} \le 50$$
$$25 \le P_{s,3} \le 50$$

ซึ่งเหลือเพียง 3 ชุดเท่านั้น

เงื่อนไขบังคับชุดสุดท้าย ได้แก่ ปริมาตรน้ำที่ได้รับการจัดสรรตลอดทั้งวัน จะได้ว่า

$$8q_{h1} + 8q_{h2} + 8q_{h3} = 36000 \quad \rightarrow \quad q_{h1} + q_{h2} + q_{h3} = 4500$$
$$(300 + 15P_{h1}) + (300 + 15P_{h2}) + (300 + 15P_{h3}) = 4500$$
$$[300 + 15(32 - P_{s1})] + [300 + 15(90 - P_{s2})] + [300 + 15(125 - P_{s3})] = 4500$$
$$P_{s1} + P_{s2} + P_{s3} = 105$$

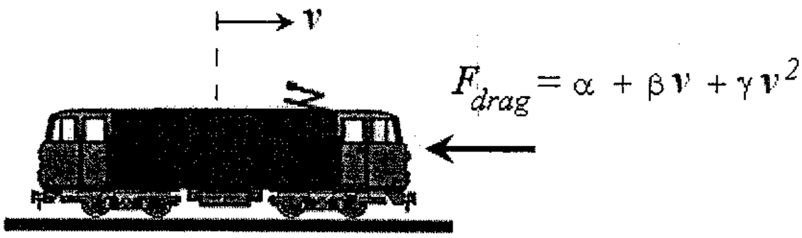
จะได้ปัญหาค่าเหมาะที่สุดดังนี้

$$\text{Minimize} \quad F_T = 8\sum_{i=1}^{3}\left(54+12P_{s,i}+0.02P_{s,i}^2\right)$$

$$\begin{aligned}\text{Subject to} \quad & P_{s1}+P_{s2}+P_{s3}=105\\ & 10\le P_{s1}\le 32\\ & 10\le P_{s2}\le 50\\ & 25\le P_{s3}\le 50\end{aligned}$$

ตัวอย่างที่ 6.7 จากข้อมูลการทดสอบเพื่อหาแรงต้านของอากาศต่อหัวรถจักรทดสอบคันหนึ่งบนรางทดสอบ จากการใช้สมการของเดวี (Davie's equation) สามารถประมาณแรงต้านของอากาศ (dynamic drag force) ในรูปของสมการกำลังสองหรือกำลังสองฟังก์ชัน โดยแรงต้านของอากาศดังกล่าวแปรตามค่าความเร็วของหัวรถจักรที่กำลังเคลื่อนที่นั่นเอง

$$F_{drag}^{model} = \alpha+\beta v+\gamma v^2$$



รูปที่ 6.6 แผนภาพการเคลื่อนที่ของหัวรถจักรทดสอบเพื่อประมาณแรงต้านของอากาศ

ผลการทดสอบได้ดังตารางต่อไปนี้

v (kph)	0	20	40	60	80	100	120	160
$F_{drag,measured}$ (kN)	0.2	2.5	6	12	30	55	90	180

kph = kilometer-per-hour

การหาสัมประสิทธิ์ดังกล่าว จะอยู่ในรูปของปัญหาที่เรียกว่าการประมาณเส้นโค้งด้วยวิธีกำลังสองน้อยที่สุดแบบไม่เชิงเส้น (nonlinear least-square estimation) ด้วยจำนวนจุดทั้งสิ้น 8 จุด จะได้ปัญหาดังนี้

$$\text{Minimize} \quad \sum_{i=1}^{8}\left[\left(\alpha+\beta v_i+\gamma v_i^2\right)-F_{drag,i}^{measured}\right]^2$$

$$\alpha,\beta,\gamma\in\Re$$

6.5 สรุป

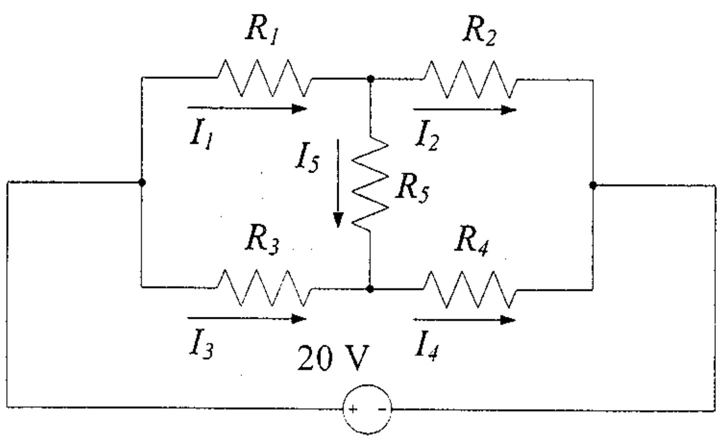
การสร้างปัญหาค่าเหมาะที่สุดมีความสำคัญเป็นอย่างยิ่ง ถือเป็นจุดเริ่มต้นการออกแบบในทางวิศวกรรม น่าเสียดายที่ไม่มีสูตรตายตัว ประสบการณ์ ความชำนาญและความใฝ่รู้ของผู้ออกแบบเท่านั้นจะช่วยให้การสร้างปัญหาค่าเหมาะที่สุดมีความรัดกุม ชัดเจน และมีประสิทธิภาพ

6.6 แบบฝึกหัดท้ายบท

1. จงแก้ปัญหาค่าเหมาะที่สุดในตัวอย่างที่ 6.5 ถึง ตัวอย่างที่ 6.7 โดยให้เลือกวิธีการแก้ปัญหาตามความเหมาะสม
2. โรงงานแห่งหนึ่งขายสินค้าในอัตรา x กิโลกรัมต่อวัน โดยมีต้นทุนการผลิตดังนี้

   $C = 50 + 0.1x + 9000/x$ บาท

   ถ้ากำหนดให้ราคาขายสินค้ามีค่าเป็น 300 บาทต่อกิโลกรัม จงคำนวณยอดขายที่ทำให้ต้นทุนการผลิตต่ำที่สุด ยอดขายที่ทำให้ได้กำไรมากที่สุดต่อวัน และยอดขายที่เท่าทุน
3. จากวงจรบริดจ์ดังรูป จงหาค่าความต้านทาน $R_1 - R_5$ ที่ทำให้เกิดกำลังไฟฟ้าสูญเสียน้อยที่สุดในวงจร เมื่อกำหนดให้กระแสที่ไหลผ่านตัวต้านทานแต่ละตัวต้องมีขนาดไม่เกิน 10 A และค่าความต้านทานมีค่าในช่วง $1 \le R_i \le 5\ \Omega$



4. กระบวนการผลิตสินค้าชนิดหนึ่งใช้น้ำบริสุทธิ์เป็นส่วนประกอบที่สำคัญ ซึ่งอาจจะทำน้ำให้บริสุทธิ์โดยใช้การกรอง (distillation) หรือการแลกเปลี่ยนประจุ (ion exchange) โดยมีเงื่อนไขดังต่อไปนี้

   <u>การกรอง (distillation)</u>:

ต้นทุนการติดตั้งชุดตัวกรอง	\$800
ค่าปฏิบัติการ (operating cost) ต่อน้ำที่ผลิตได้ 1000 แกลลอน	\$1.00

   <u>การแลกเปลี่ยนประจุ (ion exchange)</u>:

ต้นทุนการติดตั้งชุดแลกเปลี่ยนประจุต่อแกลลอนต่อชั่วโมง	\$2.00
ค่าเรซินที่ใช้ในกระบวนการต่อน้ำหนัก 1 ปอนด์	\$1.00
ต้องใช้เรซิน 1 ปอนด์ต่อน้ำ 10000 แกลลอน	
ค่าปฏิบัติการ (operating cost) ต่อน้ำที่ผลิตได้ 1000 แกลลอน	\$0.40

   โดยใช้หลักการมูลค่าในปัจจุบัน (present value) จงวินิจฉัยว่ากระบวนการใดเหมาะสมที่สุดทางด้านเศรษฐศาสตร์เพื่อใช้ผลิตน้ำบริสุทธิ์ 1000 แกลลอนต่อชั่วโมง โดยให้อายุการใช้งานของทั้งสองกระบวนการเท่ากับ 10 ปี เท่ากัน กำหนดให้คิดอัตราดอกเบี้ย 12%

จงประมาณค่าฟังก์ชัน $y = f(x)$ จากข้อมูลต่อไปนี้

x	0.5	1.0	2.1	3.4
y	0.6	1.4	2.0	3.6

ก. โดยใช้ฟังก์ชันเชิงเส้น

ข. โดยใช้ฟังก์ชันกำลังสอง

ค. โดยใช้ฟังก์ชันกำลังสาม

จากข้อมูลต่อไปนี้ การประมาณค่าฟังก์ชันในรูปใดที่กำหนดให้มีความเหมาะสมมากที่สุด

x	10	20	30	40	50
y	1.00	1.26	1.86	3.31	7.08

ก. $y = e^{a+bx}$

ข. $y = e^{a+bx+cx^2}$

ค. $y = ax^b$

7. ในกระบวนการผลิตวัสดุในงานก่อสร้างชนิดหนึ่ง ถ้าความสามารถในการทนแรงดึงสูงสุดของวัสดุขึ้นอยู่กับปัจจัยสองประการ ได้แก่ อุณหภูมิในการอบวัสดุและเวลาที่ใช้ในการอบวัสดุ จงใช้การประมาณค่าด้วยแบบจำลองอันดับที่หนึ่ง $F = a + bT + ct$

t (hour)	1	5	1	5	1	5	1	5
T (°C)	240	240	280	280	240	240	280	280
F (kN)	24	42	3	19	24	46	5	21

บทที่ 7 การหาค่าเหมาะที่สุดโดยใช้เทคนิคชาญฉลาด (Intelligent Optimization Techniques)

"Make everything as simple as possible, but not more so"

7.1 เกริ่นนำ

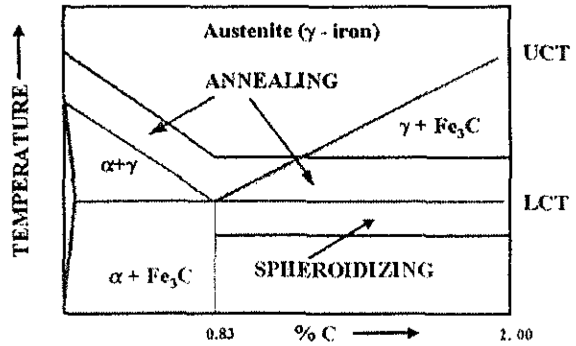
เมื่อปัญหาค่าเหมาะที่สุดที่มีความซับซ้อนและเป็นปัญหาแบบมัลติโมดอล (multimodal problem) ซึ่งมีจุดต่ำสุดหลายจุดในปริภูมิค้นหาและมีความไม่เชิงเส้นสูง ระเบียบวิธีกำหนดการทางคณิตศาสตร์ที่ได้นำเสนอมานั้นอาจจะไม่สามารถค้นหาจุดต่ำสุดโดยรวมของปัญหาได้ ทำให้ปัญญาประดิษฐ์หรือเทคนิคชาญฉลาดถูกนำมาใช้งานกันอย่างแพร่หลาย ถึงแม้เทคนิคเหล่านี้จะใช้เวลาค้นหาที่ยาวนานมาก แต่ด้วยสมรรถนะของคอมพิวเตอร์ที่เพิ่มสูงขึ้นมากในปัจจุบัน ทำให้การแก้ปัญหาด้วยเทคนิคชาญฉลาด อยู่ในเกณฑ์ที่ยอมรับได้ในสำหรับการประยุกต์ใช้งานบางประเภท เทคนิคชาญฉลาดมีหลากหลายชนิด ในที่นี้จะนำเสนอเฉพาะที่สำคัญและที่นิยมใช้ในงานวิจัยทางวิศวกรรมไฟฟ้า ได้แก่

- การจำลองการอบอ่อน (simulated annealing: SA)
- จีนเนติกอัลกอริทึม (genetic algorithms: GA)
- กำหนดการวิวัฒนาการ (evolutionary programming: EP)
- การหาค่าเหมาะที่สุดของฝูงอนุภาค (particle swarm optimization: PSO)
- การค้นหาตาบู (tabu search: TS)

ดังรายละเอียดต่อไปนี้

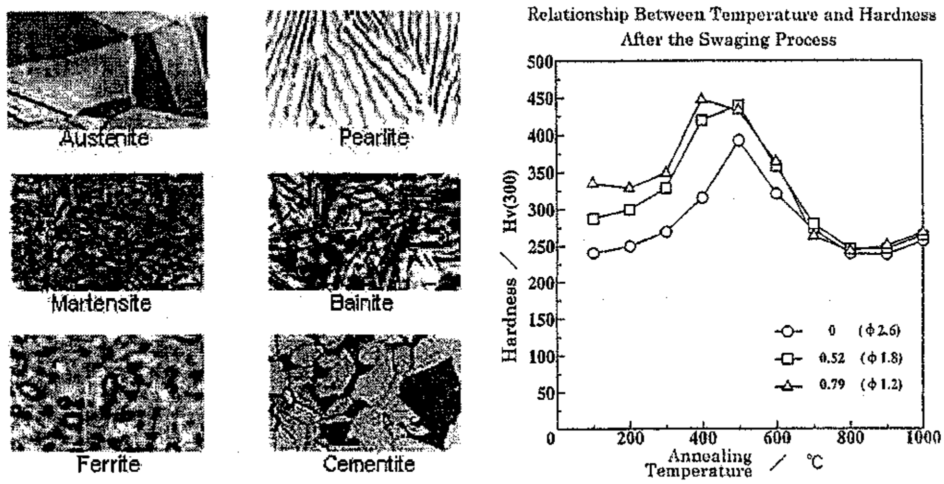
7.2 การจำลองการอบอ่อน (Simulated Annealing: SA)

เทคนิคการจำลองการอบอ่อน (annealing) นี้ เรียกอีกอย่างหนึ่งว่า การจำลองการตกผลึก เป็นการเลียนแบบปรากฏการณ์ที่ใช้ในการขึ้นรูปโลหะ ถ้าให้โลหะร้อนหรือสารละลายเย็นตัวลงอย่างช้าๆ จะเกิดการตกผลึกหรือได้โครงสร้างที่มีความแข็งแกร่ง มากกว่าการบังคับให้เย็นตัวลงอย่างรวดเร็ว (quenching) ในอากาศ ในน้ำหรือในสารละลายบางชนิด พิจารณาแผนภูมิสถานะของโลหะดังรูป



รูปที่ 7.1 แผนภูมิสถานะ (phase diagram) ของเหล็กกล้า

การอบอ่อนอาศัยหลักการให้ความร้อนแก่เหล็กกล้า (steel) ให้มีอุณหภูมิสูงกว่าประมาณ 100 องศา จากนั้นทำให้เย็นเล็กอย่างช้าๆ ในอัตราที่เหมาะสมในเตาหลอดจนถึงอุณหภูมิห้อง โครงสร้างออสเทไนต์ (austenite) ของเหล็กจะถูกทำลายกลายเป็นโครงสร้างอื่น เช่น เฟอร์ไรต์ (ferrite) ซีเมนไทต์ (cementite) หรือเพอร์ไลต์ (pearlite) เป็นต้น ดังรูปที่ 7.2 ช่วยทำให้เหล็กกล้าที่ผ่านการอบอ่อนคุณลักษณะสมบัติของเหล็กกล้าดีขึ้น เช่น ความเหนียว ความแข็งแกร่ง ทำให้ขึ้นรูปได้ง่าย



รูปที่ 7.2 โครงสร้างของโลหะและความแข็งแกร่งตามอุณหภูมิการอบอ่อน

ในปี ค.ศ. 1983 เคอร์กแพทริก และคณะ (Kirkpatric et al) นำเอาหลักการดังกล่าวมาใช้ในการหาค่าเหมาะที่สุดสำหรับออกแบบแผงวงจรขนาดใหญ่ (VLSI design) พิจารณาจากปัญหาค่าเหมาะที่สุดต่อไปนี้

$$\text{Minimize} \quad f(x)$$
$$x \in S$$

การแก้ปัญหาด้วยเทคนิค SA มีกระบวนการดังนี้

<u>ขั้นที่ 1</u>: สุ่มเลือกคำตอบเริ่มต้น x_0 กำหนดพารามิเตอร์ T_0 (cooling temperature) ค่าสูงๆ

ให้ $f_0 = f(x_0)$ และ $j = 1, c = 1$

<u>ขั้นที่ 2</u>: สร้างผลเฉลยแบบสุ่มจากเซตข้างเคียง $\aleph(x_0)$ ของ x_0 โดยที่ $x \in \aleph(x_0)$

คำนวณ $\Delta f = f(x) - f(x_0)$

<u>ขั้นที่ 3</u>: ถ้า $\Delta f < 0$ ให้ $x_0 = x$, ข้ามไปขั้นที่ 6

ถ้าไม่ใช่ ให้สุ่มค่า δ อย่างสม่ำเสมอในช่วง [0,1]

<u>ขั้นที่ 4</u>: ถ้า $\delta < e^{-\frac{\Delta f}{kT_0}}$ ให้ $x_0 = x$, ข้ามไปขั้นที่ 6

ขั้นตอนนี้ เรียกว่า การทดสอบการยอมรับจุดคำตอบด้วยการกระจายแบบโบลต์ซ์มานน์ (Boltzmann distribution)

โดยที่ k = Boltzmann's constant

T_0 = cooling temperature

ถ้าไม่ใช่ ให้ $c = c + 1$

<u>ขั้นที่ 5</u>: ถ้า $c < c_{max}$ ข้ามไปขั้นที่ 6

ถ้าไม่ใช่ ให้ $c = 0$; $T_0 = \rho T_0$, โดยที่ $\rho < 1$

<u>ขั้นที่ 6</u>: ตรวจสอบเงื่อนไขการหยุด ถ้าไม่สอดคล้อง ทำซ้ำขั้นที่ 2

<u>ขั้นที่ 7</u>: ได้ผลเฉลย

พิจารณาอัลกอริทึมการค้นหาโดยรวมได้จากแผนภาพในรูปที่ 7.3

ตัวอย่างที่ 7.1 ดำเนินการแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้โดยใช้เทคนิคการจำลองการอบอ่อน

$$\text{Minimize} \qquad f(x) = x_1^2 + x_2^2$$

วิธีทำ ผลการคำนวณที่แสดงต่อไปนี้ได้จากการสร้างตัวเลขสุ่มด้วยโปรแกรมคอมพิวเตอร์

เริ่มต้นการคำนวณด้วยผลเฉลยแบบสุ่ม $x_0 = [0.5619 \;\; -0.1258]^T$ $\quad f_0 = 0.3316$

ทดลองกำหนดพารามิเตอร์ $T_0 = 1000, k = 0.0001, \rho = 0.8$

<u>รอบการคำนวณที่ 1</u>: สุ่ม $\boldsymbol{x}_1 \in \aleph(\boldsymbol{x}_0)$

$$\boldsymbol{x}_1 = [0.5214 \;\; -0.1306]^T, \qquad f_1 = 0.2889, \qquad \Delta f = -0.0427 < 0$$

เข้าเงื่อนไขปรับปรุงจุดคำตอบ

<u>รอบการคำนวณที่ 2</u>: สุ่ม $\boldsymbol{x}_2 \in \aleph(\boldsymbol{x}_1)$

$$\boldsymbol{x}_2 = [0.5280 \;\; -0.1277]^T, \qquad f_2 = 0.2951, \qquad \Delta f = 0.0062 > 0$$

ดำเนินการทดสอบการยอมรับจุดคำตอบด้วยการกระจายแบบโบลต์ซ์มานน์ (Boltzmann distribution) คำนวณ $e^{-\frac{\Delta f}{kT_0}} = e^{\left(-\frac{0.0062}{0.0001\times1000}\right)} = 0.9399$ โดยการสุ่มค่า δ จะได้ว่า

$\delta = 0.229 < 0.9399$ ดังนั้น ยอมรับ $\boldsymbol{x}_2$ เป็นจุดคำตอบในรอบถัดไป

<u>รอบการคำนวณที่ 3</u>: สุ่ม $\boldsymbol{x}_3 \in \aleph(\boldsymbol{x}_2)$

$$\boldsymbol{x}_3 = [0.5634 \;\; -0.1487]^T, \qquad f_3 = 0.3266, \qquad \Delta f = 0.0315 > 0$$

ดำเนินการทดสอบการยอมรับจุดคำตอบด้วยการกระจายแบบโบลต์ซ์มานน์ (Boltzmann distribution) คำนวณ $e^{-\frac{\Delta f}{kT_0}} = e^{\left(-\frac{0.0315}{0.0001\times1000}\right)} = 0.7298$ โดยการสุ่มค่า δ จะได้ว่า

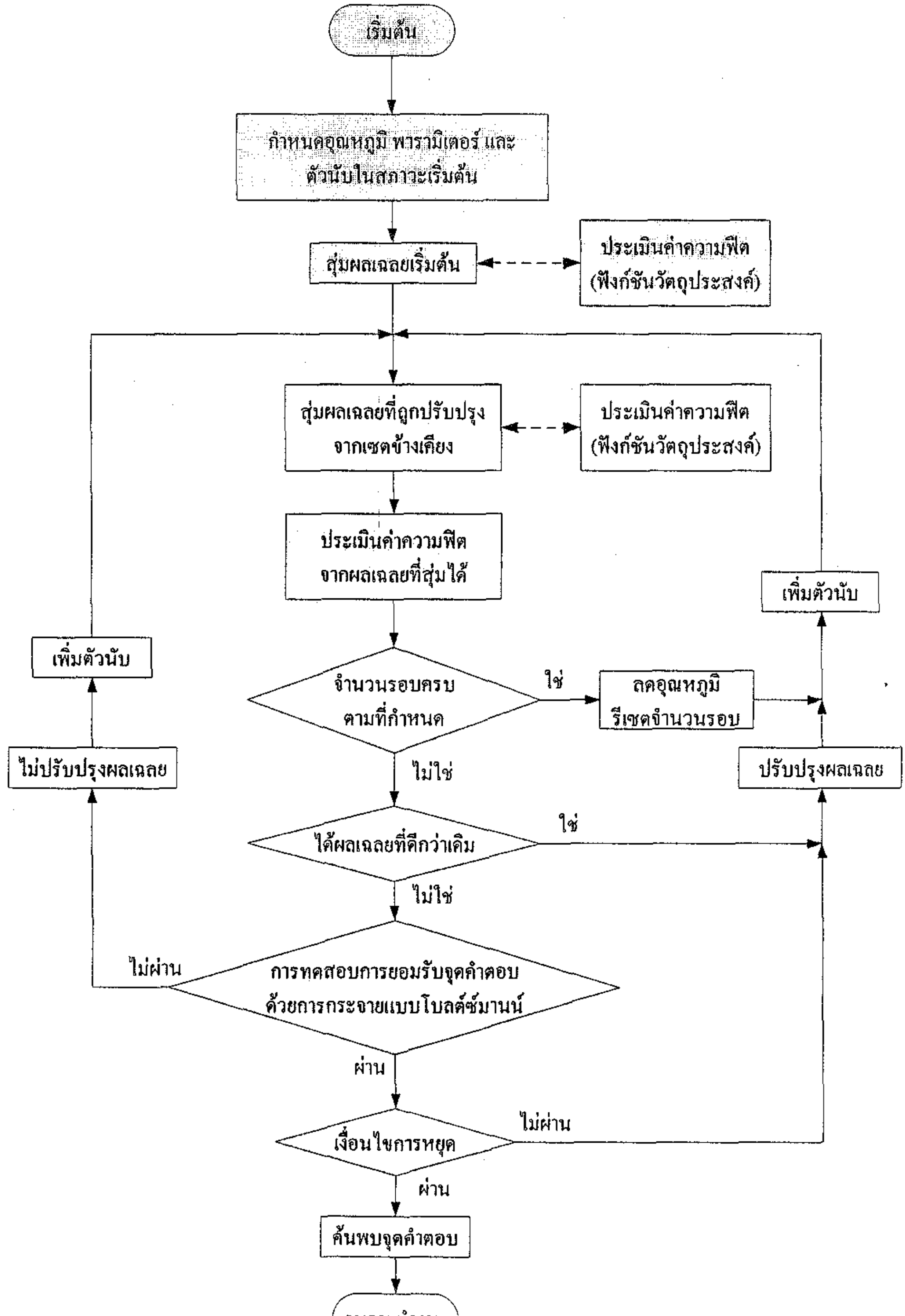
$\delta = 0.8450 > 0.7298$ ดังนั้น ละทิ้งจุดคำตอบ $\boldsymbol{x}_3$ ใช้ $\boldsymbol{x}_2$ สร้างจุดคำตอบในรอบถัดไป

<u>รอบการคำนวณที่ 4</u>: สุ่ม $\boldsymbol{x}_4 \in \aleph(\boldsymbol{x}_2)$

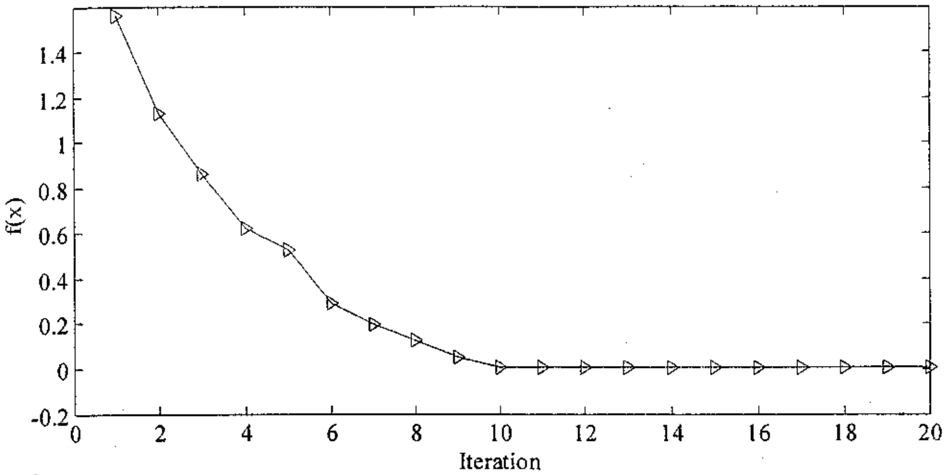
ดำเนินการคำนวณวนรอบไปเรื่อยๆ จนกว่าเงื่อนไขการหยุดจะเป็นจริง

กราฟในรูปที่ 7.4 แสดงการลู่เข้าสู่จุดคำตอบของกระบวนการ SA ที่ถูกคัดสรร โดยใช้การคำนวณ 20 รอบและให้ผลเฉลยที่ดี อย่างไรก็ตาม ผลเฉลยจะลู่เข้าหรือไม่ขึ้นอยู่กับการปรับตั้งค่าพารามิเตอร์ให้เหมาะสม ซึ่งเป็นการลองผิดลองถูกเพื่อหาชุดพารามิเตอร์ที่เหมาะสมกับปัญหาเฉพาะปัญหาใดปัญหาหนึ่งเท่านั้น ถ้าปัญหาที่พิจารณาเปลี่ยนไป พารามิเตอร์ชุดดังกล่าวไม่

จำเป็นต้องให้ผลลัพธ์ที่ดี ดังนั้น จำเป็นต้องปรับแต่งพารามิเตอร์ใหม่ทุกครั้งสำหรับการเริ่มต้นแก้ปัญหา



รูปที่ 7.3 แผนผังการทำงานของการจำลองการอบอ่อน



รูปที่ 7.4 การลู่เข้าสู่คำตอบของเทคนิคการจำลองการอบอ่อน $x^* = [0.0021 \quad 0.0620]^T$

7.3 จีนเนติกอัลกอริทึม (Genetic Algorithms: GA)

จีนเนติกอัลกอริทึมเป็นการจำลองกระบวนการวิวัฒนาการในระดับยีน โดยการสร้างกลุ่มประชากรโครโมโซมแทนผลเฉลย กระบวนการที่ง่ายที่สุดอยู่ในรูปของการแปลงโครโมโซมในระบบเลขฐานสอง จากนั้น ประชาการในกลุ่มจะแข่งขันกันเพื่อความอยู่รอด โครโมโซมที่ถูกเลือกในแต่ละรุ่นการถ่ายทอด (generation) เท่านั้นที่มีสิทธิ์สร้างลูกหลานหรือทายาท (offspring) ในรุ่นถัดไปได้ การสร้างลูกหลานจะใช้การดำเนินการทางสายพันธุ์ (genetic operators) ซึ่งประกอบไป 1. ครอสโอเวอร์ (crossover) 2. การผ่าเหล่า (mutation) ลูกหลานหรือทายาทที่ถูกสร้างขึ้นจะแทนที่โครโมโซมต้นแบบโดยสมบูรณ์ โดยปกติ จีนเนติกอัลกอริทึมจะใช้ครอสโอเวอร์ในสัดส่วนที่สูง ประมาณ 60 – 70% ของจำนวนประชากรทั้งหมด ในขณะที่การผ่าเหล่าจะยอมให้เกิดได้น้อยเพียง 1 – 2 % เท่านั้น โดยใช้หลักการคัดเลือกตามธรรมชาติ (Darwin's natural selection) สามารถสร้างผลเฉลยที่ดีได้จากกระบวนการนี้ เนื่องจากเทคนิคนี้ ใช้การเข้ารหัสโครโมโซมด้วยระบบเลขฐานสอง ดังนั้น ก่อนที่จะกล่าวถึงกระบวนการต่างๆ จำเป็นต้องเข้าใจหลักการพื้นฐานดังกล่าวเสียก่อนดังนี้

1. <u>การเข้ารหัสโครโมโซม</u> (chromosome encoding) ในการแก้ปัญหาเวกเตอร์ค่าจริงการเข้ารหัสโครโมโซมเป็นการแปลงเลขจากฐานสิบไปเป็นฐานสองนั่นเอง การเข้ารหัสนี้อาศัยหลักการเดียวกันกับการแปลงสัญญาณจากแอนะล็อกไปเป็นดิจิตอล ถ้ากำหนดให้ตัวแปร x มีค่าพิสัยในช่วง x_{min} ถึง x_{max} กำหนดให้แบ่งช่วงจากจุดขอบทั้งสองเป็น n จุดเท่าๆ กัน ระยะห่างของข้อมูลที่ถูกแบ่งออกเป็นส่วนๆ สองจุดที่อยู่ติดกันสามารถคำนวณได้จาก $\Delta x = \dfrac{x_{max} - x_{min}}{n-1}$ เนื่องจากใช้เลขฐานสองดังนั้น $n = 2^m$ นั่นคือ $\Delta x = \dfrac{x_{max} - x_{min}}{2^m - 1}$ เรียก m ว่าจำนวนบิตของชุดข้อมูลหรือของสตริง ตัวอย่างเช่น แบ่งช่วง 2.0 ถึง 3.6 ออกเป็น 8 จุดข้อมูล จะใช้สตริงจำนวน

ทั้งสิ้น 3 บิต มีค่า $\Delta x = \dfrac{x_{max} - x_{min}}{2^m - 1} = \dfrac{3.6 - 2.0}{2^3 - 1} = 0.2286$ นำมาเขียนแจกแจงได้ดังตาราง ต่อไปนี้

No.	String Representative	Real Value
1	000	2.0000
2	001	2.2286
3	010	2.4572
4	011	2.6858
5	100	2.9144
6	101	3.1430
7	110	3.3716
8	111	3.0000

ในการแก้ปัญหาค่าเหมาะที่สุด ความละเอียดของจุดคำตอบมีความสำคัญ การใช้จำนวนบิตที่สูงจะได้ความละเอียดในการคำนวณที่สูงไปด้วย ซึ่งอาจจะใช้ 8 บิต 12 บิต 16 บิต หรืออื่นๆ สำหรับปัญหาหลายมิติตัวแปรแต่ละมิติจะถูกกำหนดความละเอียดด้วยจำนวนบิต เมื่อรวมจำนวนบิตของตัวแปรแต่ละมิติเข้าด้วยกันจะได้จำนวนบิตทั้งหมดของโครโมโซม โดยนำสตริงที่ได้จากการเข้ารหัสมาเขียนเรียงต่อกันได้โดยตรง ถ้ากำหนดให้ B_i นำเสนอสตริงที่ได้จากการแปลงตัวแปรค่าจริง x_i ใดๆ แล้ว โครโมโซม 1 ตัว ที่เป็นสมาชิกของกลุ่มประชากรเขียนได้โดย

Chromosome # k: $\quad B_1B_2\cdots B_i$

ตัวอย่างเช่น กำหนดให้

$x_1 \in [-1,1]$ 6 *bits*

$x_2 \in [0,2]$ 4 *bits*

$x_3 \in [4,12]$ 8 *bits*

จะได้สตริงโครโมโซมยาว 6 + 4 + 8 = 18 บิต ถ้ากำหนดค่าตัวแปรที่ถูกเข้ารหัสดังต่อไปนี้

$x_1 \rightarrow B_1 = 101101$

$x_2 \rightarrow B_2 = 0010$

$x_3 \rightarrow B_3 = 11011000$

$$Chrom = B_1B_2B_3 = \underbrace{101101}_{B_1}\underbrace{0010}_{B_2}\underbrace{11011000}_{B_3}$$

2. การดำเนินการทางสายพันธุ์ (genetic operators) เป็นการนำเอาสตริงโครโมโซมจำนวนตั้งแต่ 1 ตัวขึ้นไปมากระทำกันดังนี้

○ ครอสโอเวอร์ เป็นการนำเอาชิ้นส่วนของสตริงโครโมโซมสองตัวมาแลกเปลี่ยนกัน ดังต่อไปนี้

Chrom1 = 101001101101
Chrom2 = 001111100011

การแลกเปลี่ยนชิ้นส่วนโครโมโซมด้วยการดำเนินการครอสโอเวอร์มีหลายรูปแบบ ในที่นี้จะนำเสนอรูปแบบการครอสโอเวอร์จุดเดียวอย่างง่าย อันดับแรกต้องทำการสุ่มตำแหน่งที่จะเป็นจุดแบ่งของโครโมโซม ตัวอย่างนี้กำหนดให้มีค่าเท่ากับ 5 ดังนั้นโครโมโซมจะถูกแบ่งออกเป็นสองส่วน โดยบิตที่ 5 จะเป็นขอบเขตดังนี้

$$\text{Chrom1} = \underbrace{10100}_{a_1}\text{-}\underbrace{1101101}_{a_2}$$

$$\text{Chrom2} = \underbrace{00111}_{b_1}\text{-}\underbrace{1100011}_{b_2}$$

ดำเนินการแลกชิ้นส่วนระหว่าง Chrom1 กับ Chrom2 จะได้

$$\left.\begin{array}{l}\text{Chrom1} = \underbrace{10100}_{a_1}\text{-}\underbrace{1101101}_{a_2} \\ \text{Chrom2} = \underbrace{00111}_{b_1}\text{-}\underbrace{1100011}_{b_2}\end{array}\right\} \begin{array}{l}\text{Chrom3} = \underbrace{10100}_{a_1}\text{-}\underbrace{1100011}_{b_2} = 101001100011 \\ \text{Chrom4} = \underbrace{00111}_{b_1}\text{-}\underbrace{1101101}_{a_2} = 001111101101\end{array}$$

Chrom3 และ Chrom4 ที่ถูกสร้างขึ้นจะแทนที่โครโมโซมต้นแบบ Chrom1 และ Chrom2 อย่างไรก็ตาม การดำเนินการครอสโอเวอร์อาจจะเป็นแบบหลายจุด (multiple-point crossover) ซึ่งจะไม่กล่าวถึงในที่นี้

- <u>การผ่าเหล่า</u> เป็นการปรับเปลี่ยนข้อมูลของบิตสตริงตำแหน่งใดๆ หรือหลายตำแหน่งให้มีค่าเปลี่ยนไปจากเดิม ทำให้ได้สายพันธ์ใหม่ที่มีรหัสเลขฐานสองแตกต่างไปจากเดิม ดังต่อไปนี้

$$\text{Chrom1} = 101001101101$$

การปรับเปลี่ยนข้อมูลของบิตสตริงที่ตำแหน่งบิตใดๆ ทำได้โดยอาศัยการสุ่มตำแหน่งบิตที่จะเกิดการผ่าเหล่า เช่น Chrom1 ที่กำหนดเกิดการผ่าเหล่าโดยให้ผลการสุ่มเป็นตำแหน่งบิตที่ 7 หมายความว่า รหัสเลขฐานสองที่ตำแหน่งดังกล่าวต้องถูกเปลี่ยนแปลง ดังนี้

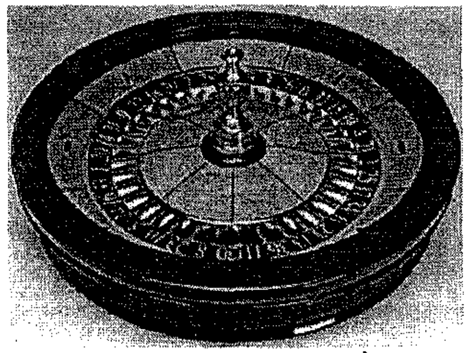
$$\text{Chrom1} = 101001\underline{1}01101$$
$$\downarrow$$
$$\text{Chrom2} = 101001\underline{0}01101$$

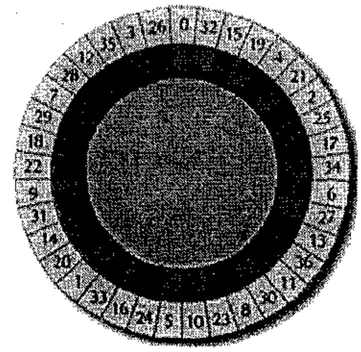
นอกจากตัวดำเนินการทั้งสองตัวแล้ว เพื่อให้รับประกันการอยู่รอดของสายพันธ์ที่ดี อาจจะใช้เทคนิคการทำสำเนาโครโมโซมที่ดีโดยถูกคัดเลือกเป็นพิเศษ (reproduction หรือ duplication) ให้อยู่รอดต่อไป

<u>3. การคัดเลือกสายพันธุ์</u> (genetic selection) การคัดเลือกโครโมโซมเดี่ยวหรือคู่ของโครโมโซมใดๆ จากกลุ่มประชากรเพื่อนำมาดำเนินการทางสายพันธุ์ใช้หลักของความน่าจะเป็นเข้าช่วย อย่างไรก็ตาม โครโมโซมแต่ละชุดจะมีโอกาสที่จะอยู่รอดสร้างรุ่นถ่ายทอดได้ไม่เท่ากัน โดยจะใช้การวัดความเหมาะสมของการอยู่รอดที่เรียกว่า ค่าความฟิต (fitness value) โดยอาจจะประเมินได้จากฟังก์ชันวัตถุประสงค์หรือปริมาณอื่นๆ การคัดเลือกสายพันธุ์ทำได้หลายวิธี เช่น แผนการวงล้อรูเล็ต (roulette-wheel scheme) หรือ แผนการทัวร์นาเมนต์ (tournament scheme)

- <u>แผนการวงล้อรูเล็ต</u> ใช้หลักการสุ่มเพื่อสร้างศูนย์รวมการจับคู่ (mating pool) ตามหลักของการเล่นเกมส์รูเล็ต ดังรูปที่ 7.5 ถ้าเริ่มต้นด้วยจำนวนประชากร n ชุด จะสุ่มสมาชิกทีละตัวออกมาทั้งสิ้น n ตัว เพื่อบรรจุลงในศูนย์รวมการจับคู่ดังกล่าว โดยการจัดแบ่งโครโมโซมแต่ละตัว

เข้าไปยังช่องสล็อตของวงล้อรูเล็ต โครโมโซมแต่ละตัวจะมีสัดส่วนไม่เท่ากันในวงล้อ โดยโครโมโซมที่มีค่าความฟิตเหมาะที่สุดจะได้สิทธิ์จับจองพื้นที่สล็อตมากกว่า





รูปที่ 7.5 วงล้อรูเล็ต

พิจารณากลุ่มประชากรโครโมโซมจำนวน 6 ชุดพร้อมค่าความฟิต (ในที่นี้ใช้ค่าฟังก์ชันวัตถุประสงค์) ดังนี้

$$
\begin{array}{lll}
\mathit{Chrom}\#1: & 101011 & \Rightarrow f_1 = 4.2 \\
\mathit{Chrom}\#2: & 001111 & \Rightarrow f_2 = 2.3 \\
\mathit{Chrom}\#3: & 101010 & \Rightarrow f_3 = 3.8 \\
\mathit{Chrom}\#4: & 011011 & \Rightarrow f_4 = 5.6 \\
\mathit{Chrom}\#5: & 101101 & \Rightarrow f_5 = 1.9 \\
\mathit{Chrom}\#6: & 101111 & \Rightarrow f_2 = 4.4
\end{array}
$$

สำหรับปัญหาค่าต่ำสุดความน่าจะเป็นในการถูกเลือกของโครโมโซมแต่ละตัวกำหนดได้หลายวิธี ในที่นี้จะนำเสนอหลักการที่ง่าย เช่น P_i [Chrom#i] = $\dfrac{f_i^{-1}}{\sum_{j=1}^{n} f_j^{-1}}$

นั่นคือ

$$\mathit{Chrom}\#1: \quad P_1 = \frac{f_1^{-1}}{\sum f^{-1}} = \frac{0.2381}{1.8682} = 0.1274$$

$$\mathit{Chrom}\#2: \quad P_2 = \frac{f_2^{-1}}{\sum f^{-1}} = \frac{0.4348}{1.8682} = 0.2327$$

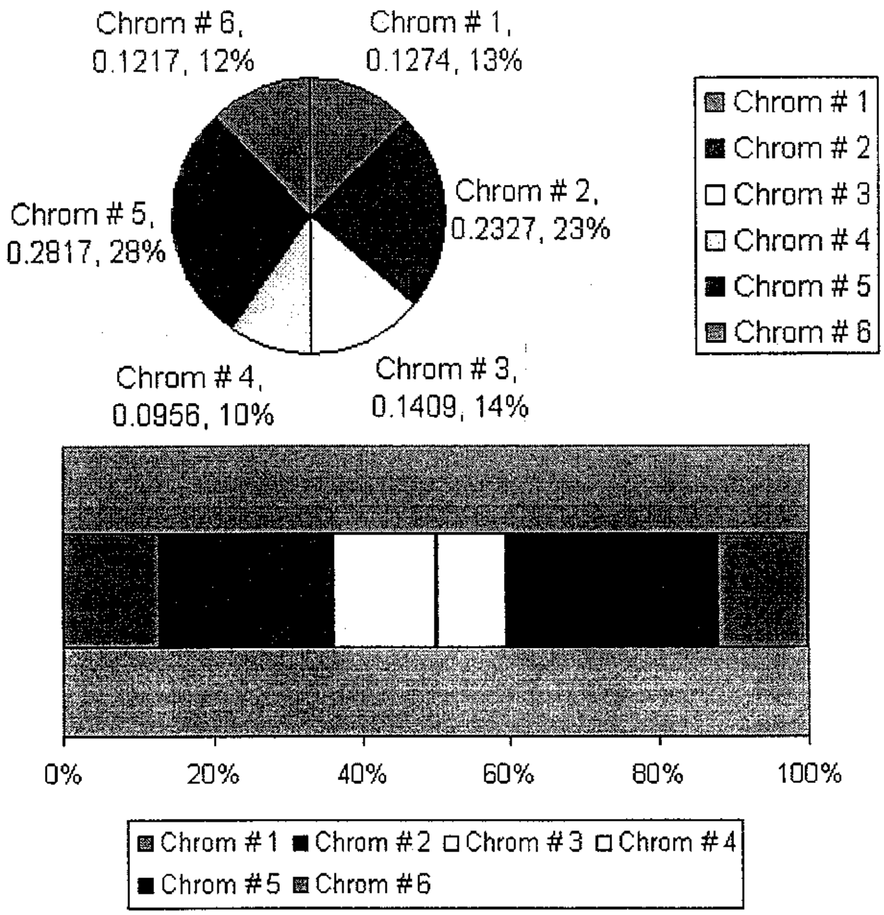
$$\mathit{Chrom}\#3: \quad P_3 = \frac{f_3^{-1}}{\sum f^{-1}} = \frac{0.2632}{1.8682} = 0.1409$$

$$\mathit{Chrom}\#4: \quad P_4 = \frac{f_4^{-1}}{\sum f^{-1}} = \frac{0.1786}{1.8682} = 0.0956$$

$$\mathit{Chrom}\#5: \quad P_5 = \frac{f_5^{-1}}{\sum f^{-1}} = \frac{0.5263}{1.8682} = 0.2817$$

$$Chrom\#6: \qquad P_6 = \frac{f_6^{-1}}{\sum f^{-1}} = \frac{0.2273}{1.8682} = 0.1217$$

แบ่งวงล้อรูเล็ตออกเป็น 6 ส่วนตามค่าความน่าจะเป็นที่คำนวณได้ หรืออาจจะแบ่งในรูปของช่วง 0% - 100% (ช่วงปิด [0,1]) ดังนี้



รูปที่ 7.6 การแบ่งพื้นที่หรือวงล้อรูเล็ตเพื่อคัดเลือกสายพันธุ์

ทำการสุ่มค่าตัวเลขจาก 0 – 1 (0% - 100%) ทีละชุด เช่น สุ่มครั้งแรกได้ 0.4256 ซึ่งตรงกับตำแหน่งของ Chrom#3 ดังนั้น ทำสำเนา Chrom#3 ลงใน mating pool ทำซ้ำการสุ่มจนได้จำนวนสมาชิกในศูนย์รวมการจับคู่ครบถ้วนตามจำนวนที่ต้องการ เพื่อนำมาดำเนินการทางสายพันธุ์ต่อไป

○ <u>แผนการทัวร์นาเมนต์</u> ใช้หลักการสุ่มโครโมโซมจากกลุ่มประชากรมาทีละคู่ จากนั้นเปรียบเทียบค่าความฟิตระหว่างโครโมโซมทั้งสอง คัดโครโมโซมที่มีค่าความฟิตดีที่สุดบรรจุลงในศูนย์รวมการจับคู่ ดำเนินการสุ่มคู่โครโมโซมดังกล่าวจำนวน n ครั้ง เพื่อสร้างศูนย์รวมการจับคู่ที่สมบูรณ์ เทคนิคนี้ง่ายกว่า และลดขั้นตอนหลายประการลงได้ ตัวอย่างเช่น

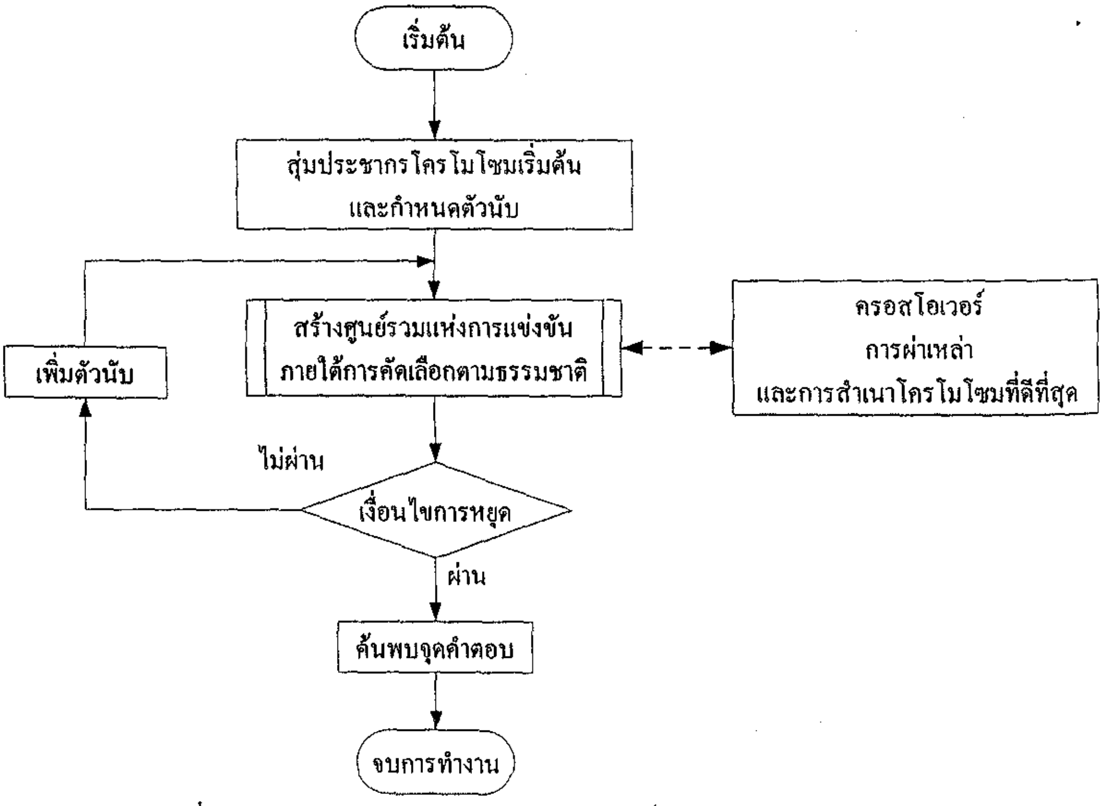
$$\begin{array}{lll} Chrom\#1: & 101011 & \Rightarrow f_1 = 4.2 \\ Chrom\#2: & 001111 & \Rightarrow f_2 = 2.3 \\ Chrom\#3: & 101010 & \Rightarrow f_3 = 3.8 \\ Chrom\#4: & 011011 & \Rightarrow f_4 = 5.6 \\ Chrom\#5: & 101101 & \Rightarrow f_5 = 1.9 \\ Chrom\#6: & 101111 & \Rightarrow f_2 = 4.4 \end{array}$$

สุ่มโครโมโซมสองตัวใดๆ จากกลุ่มประชากรที่กำหนดตามแผนการทัวร์นาเมนต์จะได้ว่า

$$\left| \begin{array}{lll} Chrom\#2: & 001111 & \Rightarrow f_2 = 2.3 \\ Chrom\#5: & 101101 & \Rightarrow f_5 = 1.9 \end{array} \right\rangle \; Chrom\#5 \; survives$$

จากการสุ่มเลือกคู่ Chrom#2 และ Chrom#5 จะพบว่า Chrom#5 ให้ค่าฟังก์ชันวัตถุประสงค์ที่ดีกว่า ดังนั้น Chrom#5 ถูกคัดเลือกให้บรรจุในศูนย์รวมการจับคู่

อย่างไรก็ตาม แผนการทั้งสองที่ได้นำเสนอนี้ไม่ได้รับประกันการอยู่รอดของโครโมโซมที่ดีที่สุด เนื่องจากมีโอกาสที่โครโมโซมที่ดีจะไม่ถูกเลือก เพื่อป้องกันการสูญเสียสายพันธุ์ที่ดีไป การเลือกทำสำเนาโครโมโซมตัวที่ดีที่สุดเพื่อบรรจุลงในศูนย์รวมการเลือกคู่เป็นวิธีหนึ่งที่แก้ปัญหานี้ได้ เมื่อตำแหน่งในศูนย์รวมการเลือกคู่ถูกแทนที่ด้วยสำเนาโครโมโซมที่ดีที่สุด ดังนั้นจะเหลือจำนวนโครโมโซมที่ต้องสุ่มเลือกเพียง $n-1$ เท่านั้น พิจารณาการค้นหาด้วยจีนเนติกอัลกอริทึมดังต่อไปนี้



รูปที่ 7.7 แผนผังการทำงานการหาค่าเหมาะที่สุดด้วยจีนเนติกอัลกอริทึม

ตัวอย่างที่ 7.2 ดำเนินการแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้โดยใช้จีนเนติกอัลกอริทึม

$$\text{Minimize} \qquad f(x) = x_1^2 + x_2^2$$

วิธีทำ ผลการคำนวณที่แสดงต่อไปนี้ได้จากการสร้างตัวเลขสุ่มด้วย MATLAB

กำหนดขอบเขตของตัวแปรดังนี้

$$\left.\begin{array}{ll} x_1 \in [-1,1] & 6\ bits \\ x_2 \in [-1,1] & 6\ bits \end{array}\right\} \begin{array}{c} S = b_{11}b_{12}b_{13}b_{14}b_{15}b_{16}b_{21}b_{22}b_{23}b_{24}b_{25}b_{26} \\ b_{ij} = \{0,1\} \end{array}$$

กำหนดจำนวนประชากรทั้งสิ้น 10 ตัว ใช้การดำเนินการครอสโอเวอร์ 80% และใช้การผ่าเหล่า 5%

Initial population

p_1 = 000011010111 ⇒ $f_1 = 0.89141$
p_2 = 100000011100 ⇒ $f_2 = 0.012598$
p_3 = 100100100111 ⇒ $f_3 = 0.077098$
p_4 = 000111111001 ⇒ $f_4 = 1.2603$
p_5 = 110000110011 ⇒ $f_5 = 0.6576$
p_6 = 110100101011 ⇒ $f_6 = 0.55682$
p_7 = 001101010001 ⇒ $f_7 = 0.55682$
p_8 = 101000100010 ⇒ $f_8 = 0.079113$
p_9 = 000100000110 ⇒ $f_9 = 1.4175$
p_{10} = 010001011010 ⇒ $f_{10} = 0.24238$

ผลจากการสร้างกลุ่มประชากรเริ่มต้น p_2 ให้ค่าความฟิตดีที่สุด ดังนั้น จะถูกคัดลอกสำเนาไปยังศูนย์รวมการจับคู่ สมาชิกโครโมโซมที่เหลืออีก 9 ตัวจะถูกคัดเลือกภายใต้การแข่งขัน อาจจะใช้แผนการวงล้อรูเล็ตหรือแผนการทัวร์นาเมนต์ก็ได้

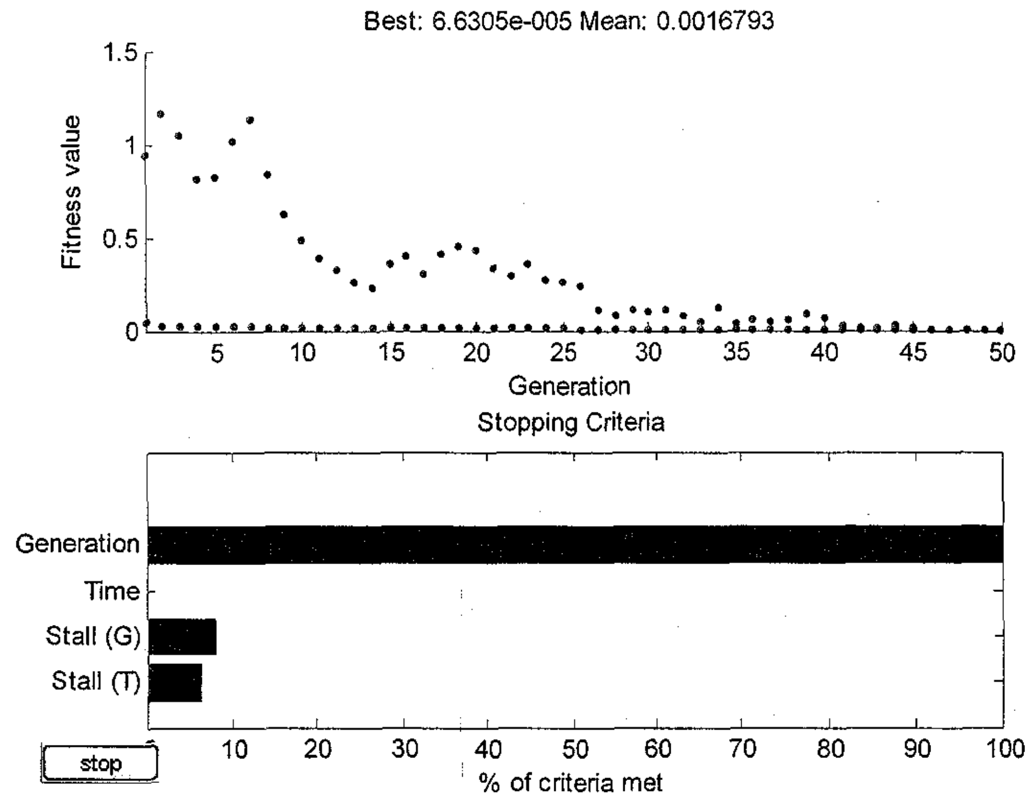
<u>รอบการคำนวณที่ 1</u>:

เมื่อสร้างศูนย์รวมการจับคู่ได้ครบแล้ว ขั้นต่อไป ได้แก่ การดำเนินการทางสายพันธ์ กำหนดให้ความน่าจะเป็นในการเกิดครอสโอเวอร์มีค่าเป็น 80% หมายความว่า สมาชิก 8 ตัวในศูนย์รวมการจับคู่ จะถูกเลือกเพื่อนำมาดำเนินการครอสโอเวอร์ โดยปกตินิยมใช้การคัดเลือกจากการจัดลำดับขั้น (ranking) ตามค่าความฟิต ผลการดำเนินการครอสโอเวอร์จะได้

Population of Generation # 1

p_1 = 011111100000 ⇒ $f_1 = 0.00050391$
p_2 = 001101010001 ⇒ $f_2 = 0.55682$
p_3 = 100000011100 ⇒ $f_3 = 0.012598$
p_4 = 100000010011 ⇒ $f_4 = 0.15772$
p_5 = 100100100111 ⇒ $f_5 = 0.077098$
p_6 = 100000011101 ⇒ $f_6 = 0.0065508$
p_7 = 101000100010 ⇒ $f_7 = 0.079113$
p_8 = 101000100011 ⇒ $f_8 = 0.08516$
p_9 = 100011011010 ⇒ $f_9 = 0.042832$
p_{10} = 001101101000 ⇒ $f_{10} = 0.41774$

โดยใช้หลักการที่ดังกล่าวแก้ปัญหาจะได้ผลเฉลยโดยมีกราฟการลู่เข้าดังรูปต่อไปนี้



รูปที่ 7.8 การลู่เข้าของจุดคำตอบโดยใช้จีนเนติกอัลกอริทึม $x^* = [7.70\times10^{-4} \quad -8.11\times10^{-3}]^T$

กราฟในรูปที่ 7.8 เป็นการประยุกต์ MATLAB's GADS TOOLBOX เพื่อใช้งาน อย่างไรก็ตาม SCILAB ยังไม่มีกล่องเครื่องมือสำเร็จรูปที่มีประสิทธิภาพเทียบเท่าได้ ดังนั้น ผู้แต่งได้พัฒนาชุดคำสั่งเพื่อแก้ปัญหาค่าเหมาะที่สุดนี้โดยใช้ SCILAB ขึ้น โดยพยายามเลียนแบบการใช้งานและการแสดงผลของ MATLAB's GADS TOOLBOX การคำนวณได้ตัดขั้นตอนที่ซับซ้อนออกไปเพื่อให้เข้าใจได้ง่าย และเหมาะสำหรับใช้ประกอบการศึกษา ดังนั้น สมรรถนะในการค้นหาจุดคำตอบของกล่องเครื่องมือที่พัฒนาขึ้นนี้ยังไม่สูงมากนัก

กล่องเครื่องมือนี้ ให้ชื่อว่า SGA TOOLBOX สำหรับ SCILAB โดยที่ SGA ย่อมาจาก simple genetic algorithms ประกอบด้วยไฟล์ทั้งสิ้น 11 ไฟล์ การเริ่มต้นใช้งานให้ดำเนินการ execute ไฟล์ ชื่อ ga_initial.sce เพื่อโหลดฟังก์ชันไฟล์ที่เกี่ยวข้องทั้งหมดและกำหนด seed number จาก ฟังก์ชัน getdate() เพื่อเป็นการสุ่มลำดับของตัวเลขสุ่มแบบเทียม pseudorandom number ที่ใช้ ไฟล์ทั้งหมดมีรายชื่อดังต่อไปนี้

sga.sci	ฟังก์ชันหลัก
fga.sci	ฟังก์ชันประเมินค่าความฟิตของฟังก์ชันวัตถุประสงค์
re2bin.sci	ฟังก์ชันแปลงรหัสตัวแปรค่าจริงเป็นรหัสตัวแปรไบนารี
bin2re.sci	ฟังก์ชันแปลงรหัสตัวแปรไบนารีไปเป็นรหัสตัวแปร
pop_ga.sci	ฟังก์ชันสร้างประชากรเริ่มต้น

randp.sci ฟังก์ชันสุ่มค่าประชากรเริ่มต้น

rand1.sci ฟังก์ชันสุ่มค่าแบบกระจายสม่ำเสมอจากช่วงข้อมูลที่กำหนด

xcross.sci ฟังก์ชันดำเนินการครอสโอเวอร์

mutation.sci ฟังก์ชันดำเนินการผ่าเหล่า

delrow.sci ฟังก์ชันลบแถวที่กำหนดในเมตริกซ์ที่เลือก

เนื่องจากไฟล์ที่เกี่ยวข้องมีจำนวนมาก ดังนั้น ชุดคำสั่งที่เขียนจะแสดงไว้ในภาคผนวก ในที่นี้จะนำเสนอตัวอย่างการใช้งานและผลการรันโปรแกรมเท่านั้น

ขั้นตอนที่ถือว่าสำคัญมากประการหนึ่งของ SGA คือ การเข้ารหัสตัวแปรไบนารี ดังแสดงในตัวอย่างต่อไปนี้

ตัวอย่างที่ 7.3 ต้องการแปลงข้อมูลค่าจริงของตัวแปร $x_1 = 0.25$, $x_2 = 2.68$ และ $x_3 = -1.778$ ให้เป็นตัวแปรไบนารี โดยตัวแปรทั้งสามมีขอบเขตดังต่อไปนี้

$$-1 \le x_1 \le 1, \quad 0 \le x_2 \le 5, \quad -5 \le x_3 \le 2$$

ถ้ากำหนดให้ค่าความละเอียดในระดับบิตที่ใช้สำหรับตัวแปรทั้งสาม มีค่าเท่ากับ 6 12 และ 10 บิต ตามลำดับ

วิธีทำ ใช้ฟังก์ชัน re2bin.sci ดังนี้

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 7.3</u>

// SCILAB source program

```
// ผลการรันโปรแกรม
--> xlimit1 = [-1 1 6];
--> x1 = 0.25;
--> bx1 = re2bin(x1,xlimit1)
 bx1  =
   1.   0.   1.   0.   0.   0.
--> xlimit2 = [0 5 12];
--> x2 = 2.68;
--> bx2 = re2bin(x2,xlimit2)
 bx2  =
   1.   0.   0.   0.   1.   0.   0.   1.   0.   0.   1.   1.
--> xlimit3 = [-5 2 10];
--> x3 = -1.778;
--> bx3 = re2bin(x3,xlimit3)
 bx3  =
   0.   1.   1.   1.   0.   1.   0.   1.   1.   1.
```

ไฟล์ bin2re.sci ใช้สำหรับแปลงตัวแปรรหัสไบนารีกลับไปเป็นตัวแปรค่าจริง อย่างไรก็ตาม จากผลการทดสอบต่อไปนี้ ความคลาดเคลื่อนที่เกิดขึ้น จะสัมพันธ์กับจำนวนบิตที่ใช้ ถ้าจำนวนบิตที่ใช้ในการเข้ารหัสมีค่าสูง ผลลัพธ์ที่ได้จากการแปลงจะมีความถูกต้องมากกว่าการเข้ารหัสด้วยจำนวนบิตต่ำ

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 7.3</u>

// SCILAB source program

```
// ผลการรันโปรแกรม
--> xx1 = bin2re(bx1,xlimit1)
 xx1  =
   0.2698413
--> xx2 = bin2re(bx2,xlimit2)
 xx2  =
   2.6800977
--> xx3 = bin2re(bx3,xlimit3)
 xx3  =
 - 1.7771261
--> abs([(x1-xx1)/x1;(x2-xx2)/x2;(x3-xx3)/x3]*100)
 ans  =
   7.9365079                    // 6 บิต คลาดเคลื่อน 7.94%
   0.0036448                    // 12 บิต คลาดเคลื่อน 0.0036%
   0.0491507                    // 10 บิต คลาดเคลื่อน 0.049%
```

ขั้นตอนการสร้างประชากรเริ่มต้น (creating a set of initial population) ของปัญหาถือเป็นสิ่งที่ต้องดำเนินการในขณะเริ่มการคำนวณ ฟังก์ชัน pop_ga.sci ทำหน้าที่นี้

ตัวอย่างที่ 7.4 ต้องการสร้างประชากรเริ่มต้นประกอบด้วยสมาชิกทั้งสิ้นจำนวน 20 โครโมโซม โดยใช้ตัวแปร $-1 \le x_1 \le 1, -1 \le x_2 \le 1$ มีความละเอียด 8 บิตเท่ากันทั้งสองตัวแปร

วิธีทำ ใช้ฟังก์ชัน pop_ga.sci ดังนี้

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 7.4</u>

// SCILAB source program

```
// ผลการรันโปรแกรม
--> xlimit=[-1 1 8;-1 1 8];
--> ps = pop_ga(2,20,xlimit);
```

```
-->ps
ps  =
  0.  0.  0.  0.  0.  0.  0.  1.  1.  0.  0.  1.  1.  0.  0.  0.
  1.  0.  0.  1.  0.  1.  0.  1.  1.  1.  1.  0.  1.  0.  1.  1.
  0.  0.  1.  0.  0.  1.  1.  0.  0.  1.  0.  0.  1.  0.  0.  0.
  1.  1.  0.  0.  1.  1.  0.  0.  1.  0.  0.  1.  1.  0.  1.  0.
  0.  1.  1.  0.  1.  1.  1.  1.  0.  0.  1.  1.  0.  1.  0.  0.
  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  1.  1.  0.
  1.  0.  1.  1.  1.  0.  1.  1.  0.  0.  1.  1.  1.  0.  0.  0.
  0.  1.  0.  1.  0.  1.  1.  0.  1.  0.  1.  1.  0.  1.  1.  0.
  1.  0.  0.  1.  0.  1.  0.  0.  1.  0.  0.  1.  1.  1.  1.  0.
  0.  0.  1.  0.  0.  0.  1.  1.  1.  1.  0.  1.  0.  0.  0.  1.
  1.  0.  1.  0.  0.  1.  1.  1.  0.  0.  0.  1.  0.  0.  1.  1.
  0.  1.  0.  0.  0.  0.  1.  0.  1.  0.  1.  1.  1.  0.  1.  1.
  0.  0.  0.  0.  0.  0.  0.  1.  1.  1.  1.  1.  0.  0.  0.  1.
  0.  1.  1.  1.  0.  1.  0.  0.  1.  0.  0.  1.  0.  0.  0.  0.
  1.  0.  0.  1.  0.  1.  0.  0.  1.  0.  0.  1.  1.  0.  1.  1.
  0.  1.  1.  1.  0.  0.  1.  1.  1.  1.  1.  1.  1.  0.  0.  0.
  0.  0.  1.  0.  1.  0.  1.  1.  0.  0.  1.  0.  1.  1.  0.  0.
  1.  1.  0.  0.  1.  0.  1.  1.  1.  0.  0.  0.  1.  1.  1.  0.
  0.  1.  0.  1.  1.  1.  1.  0.  0.  0.  1.  0.  0.  0.  1.  0.
  0.  0.  0.  0.  1.  1.  1.  0.  1.  0.  0.  0.  0.  0.  0.  1.
```

ตัวอย่างที่ 7.5 ต้องการประเมินค่าความฟิต (โดยใช้ฟังก์ชันวัตถุประสงค์) สำหรับประชากรเริ่มต้นที่ถูกสร้างขึ้นในตัวอย่างที่ 7.4 เมื่อกำหนดฟังก์ชันวัตถุประสงค์ดังแสดงในตัวอย่างที่ 7.2

วิธีทำ ใช้ฟังก์ชัน fga.sci ดังนี้

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 7.5</u>

// SCILAB source program

```
// กำหนดฟังก์ชันวัตถุประสงค์
=========================
          f_unc01.sci
=========================
function f = f_unc01(x)
x1=x(1);
```

```
x2=x(2);
f=x1^2+x2^2;
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('f_unc01.sci');
--> f = fga(f_unc01,ps,xlimit);
-->f
 f  =
    1.0212995
    0.7393156
    0.6822299
    0.4031988
    0.3673972
    0.8328181
    0.5322568
    0.2886582
    0.0830757
    0.9349327
    0.8201461
    0.4504421
    1.7768243
    0.0248827
    0.0723722
    0.9028220
    0.8681276
    0.3635832
    0.6068128
    0.7925875
```

ตัวอย่างที่ 7.6 ดำเนินการแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้โดยใช้จีนเนติกอัลกอริทึม

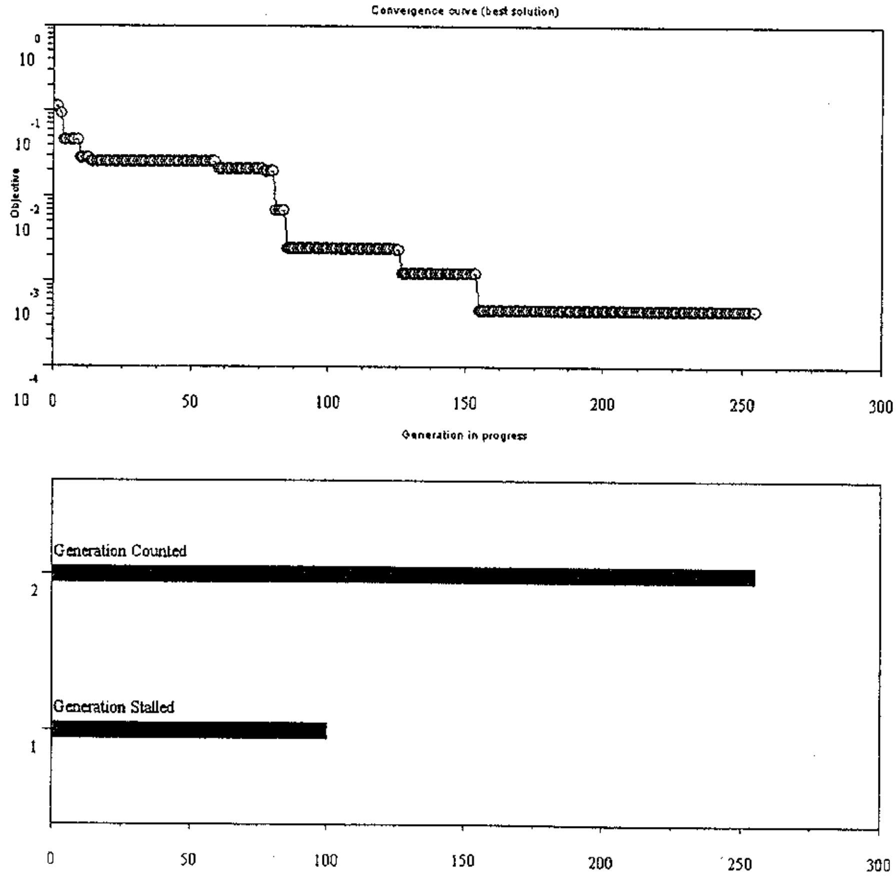
$$\text{Minimize} \qquad f(x) = x_1^2 + x_2^2$$

วิธีทำ ผลการคำนวณที่แสดงต่อไปนี้ได้จาก SCILAB SGA TOOLBOX ที่ผู้แต่งได้พัฒนาขึ้น กำหนดขอบเขตของตัวแปรดังตัวอย่างที่ 7.2 จะได้ผลดังต่อไปนี้

โปรแกรม SCILAB สำหรับตัวอย่างที่ 7.6

// SCILAB source program

```
// ผลการรันโปรแกรม
-->Ne = 2;
-->Np = 20;
-->xlimit = [-1 1 24;-1 1 24];
-->xprob = 0.8;
-->mprob = 0.05;
-->opt = [1000 100 0];
-->[pg,xg,fg] = sga(f_unc01,Ne,Np,xlimit,xprob,mprob,opt)
 fg  =
   0.0004477
 xg  =
 - 0.0151283    0.0147938
 pg =  01111110000100000100011010000001111001001 1000011
```



รูปที่ 7.9 การลู่เข้าของจุดคำตอบโดยใช้ SCILAB's SGA TOOLBOX

7.4 กำหนดการวิวัฒนาการ (Evolutionary Programming: EP)

กระบวนการวิวัฒนาการ (evolution process) เป็นการสร้างสิ่งมีชีวิตหรือสมาชิกในกลุ่มประชากรให้มีความซับซ้อนมากยิ่งขึ้น โดยได้รับอิทธิพลมาจากสิ่งแวดล้อมด้วยความน่าจะเป็นที่แตกต่างกัน การคัดเลือกตามธรรมชาติจะมีสมาชิกในกลุ่มประชากรจำนวนหนึ่งที่มีความเหมาะสมที่สุดที่สามารถอยู่รอดเพื่อถ่ายทอดคุณลักษณะไปสู่ทายาทรุ่นถัดไปได้

เบรเมอร์แมน (Bremermann 1958) ได้ทำการทดลองเพื่อนำไปสู่ข้อสรุปที่ว่า หลักการวิวัฒนาการเป็นวิธีการที่มีประสิทธิภาพในการทำความเข้าใจ การเรียนรู้ของปัญหาเพื่อค้นหาวิธีการแก้ปัญหาที่ดีที่สุดต่อไป ซึ่งเป็นหลักการสำคัญของกระบวนแก้ปัญหาค่าเหมาะที่สุดนั่นเอง การทดลองใช้การเข้ารหัสเลขฐานสองในการจำลองโครงสร้างของยีน เพื่อใช้เป็นตัวแทนสมาชิกของสมาชิกในกลุ่มประชากร และทำการทดสอบค่าความฟิตของสมาชิกเพื่อนำไปใช้ในกระบวนการคัดเลือกสมาชิกที่อยู่รอดต่อไป จากการทดลองพบว่ากระบวนการเกิดการผ่าเหล่าที่เหมาะสมนั้นต้องให้มีการเปลี่ยนแปลงเพียงหนึ่งบิทต่อสมาชิกหนึ่งตัวในแต่ละรุ่นเท่านั้น

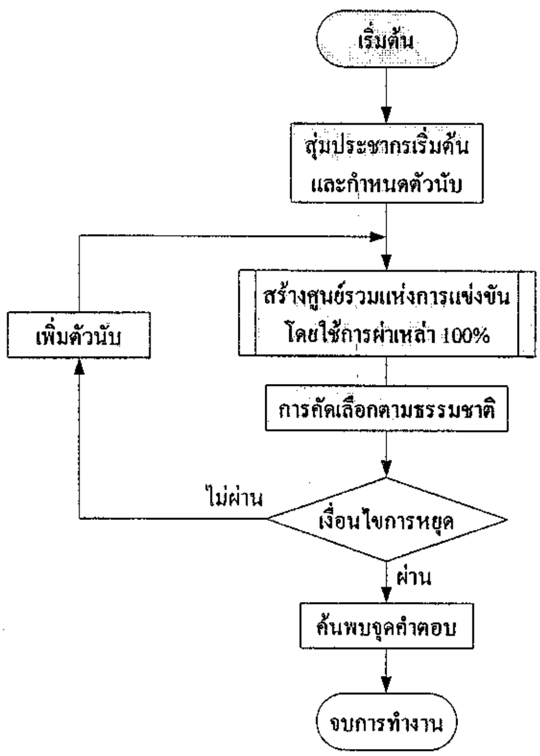
ในปี ค.ศ. 1962 เบรเมอร์แมนทำการทดลองเพื่อขยายผลจากเดิม โดยพิจารณาปัญหา การหาค่าต่ำสุดของฟังก์ชันค่าจริง $f(x_1,x_2,\ldots,x_n)$ การทดลองเลือกใช้ฟังก์ชันเชิงเส้นเพื่อให้ง่ายในการวิเคราะห์กระบวนการวิวัฒนาการ ปัญหาที่พิจารณานั้นอยู่ในรูป Minimize $\|\boldsymbol{Ax}-\boldsymbol{b}\|_2$ พบว่า ถ้าใช้เฉพาะการดำเนินการมิวเตชันแล้ว กระบวนการวิวัฒนาการจะสามารถนำไปสู่จุดต่ำสุดที่ต้องการได้อย่างรวดเร็ว นอกจากนี้ เบรเมอร์แมนได้แนะนำว่า ในกระบวนการเกิดการผ่าเหล่าอาจใช้ความน่าจะเป็นร่วมในการพิจารณาได้ แต่จะทำให้กระบวนการค้นหาจุดต่ำสุดนั้นช้าลง ต่อมาเบรเมอร์แมนและรอจสัน (Bremermann and Rogson) ได้ทำการทดลองโดยการใช้การแปลงแบบอื่นในการสร้างสมาชิกใหม่ของกลุ่มประชากร เช่น การครอสโอเวอร์ (crossover) แต่ผลที่ได้จากการทดลองไม่ดีกว่าการทดลองเดิม

ในปี ค.ศ. 1967 บาร์ริเซลลี (Barricelli) ได้ทำการทดลองกระบวนการวิวัฒนาการ โดยใช้การผ่าเหล่าและครอสโอเวอร์เพื่อเปรียบเทียบกัน ผลที่ได้พบว่า การทำครอสโอเวอร์นั้นไม่ได้ทำให้ความเร็วของการหาค่าเหมาะที่สุดเพิ่มขึ้นแต่อย่างใด

กำหนดการวิวัฒนาการคล้ายกับจีนเนติกอัลกอริทึม อย่างไรก็ตามความแตกต่างที่ชัดเจนได้แก่ กำหนดการวิวัฒนาการเน้นไปที่การผ่าเหล่าเป็นหลัก โดยไม่ใช้การครอสโอเวอร์เลย นอกจากนี้ กำหนดการวิวัฒนาการไม่ใช้การเข้ารหัสเลขฐานสอง ทำให้ง่ายกว่าในเรื่องของการเขียนโปรแกรมและความเร็วในการประมวลผล

ฟอเจล (Fogel) ได้พัฒนาปัญญาประดิษฐ์โดยใช้การจำลองกระบวนการวิวัฒนาการที่เรียกว่า กำหนดการวิวัฒนาการที่มีพื้นฐานมาจากการสร้างระบบที่มีพฤติกรรมที่ฉลาด โดยสามารถประเมินและทำนายสภาวะแวดล้อม ตลอดจนการให้การตอบสนองที่มีความเหมาะสม ในปี ค.ศ. 1962-1965 ฟอเจลได้เริ่มต้นทดลองกับระบบที่มีลักษณะไม่ต่อเนื่องที่เรียกว่า เครื่องจักรไฟไนต์

สเตต (finite state machine) และ ได้พัฒนาอย่างต่อเนื่องจนกระทั่งปี ค.ศ. 1991 ฟอเจลได้เสนอโครงร่างของกำหนดการวิวัฒนาการ เพื่อใช้เป็นมาตรฐานในการแก้ปัญหาค่าเหมาะที่สุด ดังรายละเอียดต่อไปนี้ เริ่มต้นด้วยการสุ่มค่าประชากรเริ่มต้นของกลุ่มประชากรพร้อมทั้งคำนวณ ค่าความฟิตตามหลักความน่าจะเป็น แล้วทำการสร้างทายาทหรือรุ่นถ่ายทอด (offspring) แบบสุ่มโดยใช้การผ่าเหล่าของประชากรเริ่มต้นเป็นหลัก ใช้ค่าความฟิตในการคัดเลือกสมาชิกที่จะอยู่รอดในทายาทรุ่นถ่ายทอดถัดไป กระบวนการหาค่าเหมาะที่สุดโดยกำหนดการวิวัฒนาการมีขั้นตอนดังต่อไปนี้



รูปที่ 7.10 แผนผังการทำงานการหาค่าเหมาะที่สุดด้วยกำหนดการวิวัฒนาการ

1) <u>*Vector representation*</u> : ในการหาค่าเหมาะที่สุดของฟังก์ชันค่าจริง โดยกำหนดฟังก์ชันวัตถุประสงค์เป็น $f(\boldsymbol{p}) : \Re^n \to \Re$ โดยเวกเตอร์ $\boldsymbol{p}$ เป็นสมาชิกของกลุ่มประชากรที่จะถูกพัฒนาต่อไป

2) <u>*Initialization*</u> : ประชากรเริ่มต้น p_i ; $i = 1,2,\ldots,N_P$ ถูกเลือกแบบสุ่มโดยใช้การสุ่มแบบสม่ำเสมอ (uniform random)

3) <u>*Creation of offspring*</u> : p_i' จะถูกสร้างจาก p_i โดยที่

$p_i'(j) = p_i(j) + N(0,\sigma_j^2)$ ซึ่งจะเรียกการดำเนินการนี้ว่า การผ่าเหล่า โดยที่

$$\sigma_j = \beta \frac{f_i}{f_{min}}\left[p_{max}(j) - p_{min}(j)\right]$$

4) <u>*Competition & Selection*</u> : สร้างศูนย์รวมการแข่งขัน (competing pool) จาก p_i และ p_i' สมาชิกทั้งหมดในศูนย์รวมการแข่งขันจะมีจำนวนเป็น 2 เท่าของประชากร

เริ่มต้น จากนั้นจะทำการเลือกสมาชิกที่มีค่าความฟิตดีที่สุดจำนวน N_P ให้อยู่รอดเพื่อใช้เป็นประชากรเริ่มต้นในรุ่นถ่ายทอดถัดไป

5) *Stopping rule* : กระบวนการในการสร้างรุ่นถ่ายทอดใหม่นั้นจะดำเนินการไปจนกระทั่งไม่สามารถปรับปรุงค่าความฟิตของกลุ่มประชากรได้อีกแล้ว หรือเมื่อจำนวนรุ่นถ่ายทอดถึงค่าสูงสุดที่ตั้งไว้

พิจารณาได้จากแผนผังการทำงานต่อไปนี้

ตัวอย่างที่ 7.7 ดำเนินการแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้โดยใช้กำหนดการวิวัฒนาการ

$$\text{Minimize} \qquad f(\mathbf{x}) = x_1^2 + x_2^2$$

วิธีทำ ผลการคำนวณที่แสดงต่อไปนี้ได้จากการสร้างตัวเลขสุ่มด้วยโปรแกรมคอมพิวเตอร์ กำหนดจำนวนประชากรทั้งสิ้น 5 ตัว สร้างกลุ่มประชากรเริ่มต้นได้

Initial Population

p	x_1	x_2	f_p
1	0.8576	0.7397	1.2826
2	0.8375	-0.2122	0.7464
3	0.9852	-0.0743	0.9761
4	0.0167	0.1768	0.0315
5	0.4778	-0.0598	0.2318

การผ่าเหล่าทำได้โดยการสร้างตัวแปรสุ่มบวกเข้าไปกับสมาชิกแต่ละตัว

<u>รอบการคำนวณที่ 1</u>:

Offspring of Generation # 1

p	x_1	x_2	f_p
1	1.0000	-0.5937	1.3525
2	1.0000	-0.2115	1.0447
3	0.7949	-0.7022	1.1249
4	0.0173	0.1484	0.0223
5	0.5059	-0.1775	0.2875

สร้างศูนย์รวมการแข่งขันได้เป็น

Competing Pool of Generation # 1

p	x_1	x_2	f_p
1	0.8576	0.7397	1.2826
2	0.8375	-0.2122	0.7464
3	0.9852	-0.0743	0.9761
4	0.0167	0.1768	0.0315
5	0.4778	-0.0598	0.2318
6	1.0000	-0.5937	1.3525
7	1.0000	-0.2115	1.0447
8	0.7949	-0.7022	1.1249
9	0.0173	0.1484	0.0223
10	0.5059	-0.1775	0.2875

จัดเรียงลำดับตามค่าความฟิต (ฟังก์ชันวัตถุประสงค์) จะได้

Ranking of Generation # 1

p	x_1	x_2	f_p
9	0.0173	0.1484	0.0223
4	0.0167	0.1768	0.0315
5	0.4778	-0.0598	0.2318
10	0.5059	-0.1775	0.2875
2	0.8375	-0.2122	0.7464
3	0.9852	-0.0743	0.9761
7	1.0000	-0.2115	1.0447
8	0.7949	-0.7022	1.1249
1	0.8576	0.7397	1.2826
6	1.0000	-0.5937	1.3525

สมาชิก 5 อันดับแรกเท่านั้นที่อยู่รอด

ในรอบการคำนวณที่ 1 นี้ ได้ผลเฉลยเป็น

$\mathbf{x}_1 = [0.0173 \quad 0.1484]^T$ ให้ค่า $f(\mathbf{x}_1) = 2.2331\times10^{-2}$

<u>รอบการคำนวณที่ 2</u>:

Initial Population for Generation 2

p	x_1	x_2	f_p
1	0.0173	0.1484	0.0223
2	0.0167	0.1768	0.0315
3	0.4778	-0.0598	0.2318
4	0.5059	-0.1775	0.2875
5	0.8375	-0.2122	0.7464

Offspring of Generation # 2

p	x_1	x_2	f_p
1	-0.0156	0.1667	0.0280
2	0.0540	0.1385	0.0221
3	0.5541	-0.1367	0.3257
4	0.0894	-0.5136	0.2718
5	0.0312	0.1936	0.0385

สร้างศูนย์รวมการแข่งขันได้เป็น

Competing Pool of Generation # 2

p	x_1	x_2	f_p
1	0.0173	0.1484	0.0223
2	0.0167	0.1768	0.0315
3	0.4778	-0.0598	0.2318
4	0.5059	-0.1775	0.2875
5	0.8375	-0.2122	0.7464
6	-0.0156	0.1667	0.0280
7	0.0540	0.1385	0.0221
8	0.5541	-0.1367	0.3257
9	0.0894	-0.5136	0.2718
10	0.0312	0.1936	0.0385

จัดเรียงลำดับตามค่าความฟิต (ฟังก์ชันวัตถุประสงค์) จะได้

Ranking of Generation # 2

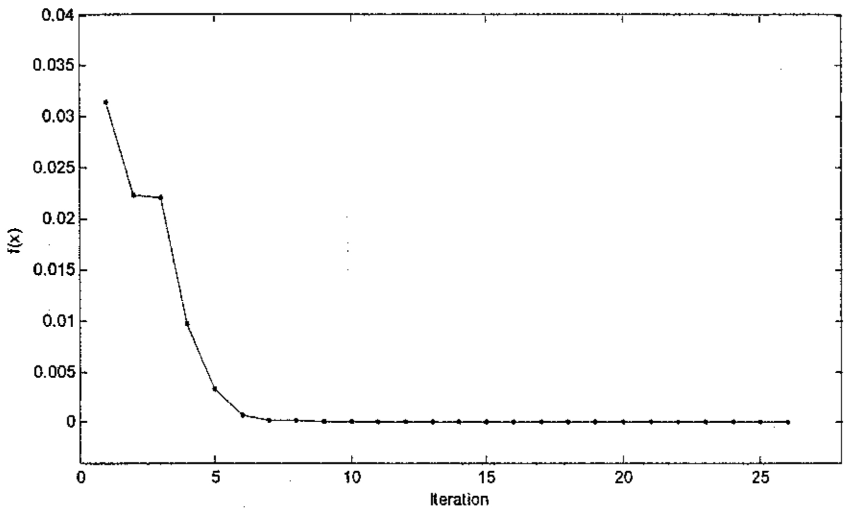
p	x_1	x_2	f_p
7	0.0540	0.1385	0.0221
1	0.0173	0.1484	0.0223
6	-0.0156	0.1667	0.0280

2	0.0167	0.1768	0.0315	
10	0.0312	0.1936	0.0385	สมาชิก 5 อันดับแรกเท่านั้นที่อยู่รอด
3	0.4778	-0.0598	0.2318	
9	0.0894	-0.5136	0.2718	
8	0.5541	-0.1367	0.3257	
4	0.5059	-0.1775	0.2875	
5	0.8375	-0.2122	0.7464	

ในรอบการคำนวณที่ 2 นี้ ได้ผลเฉลยเป็น

$$\boldsymbol{x}_2 = [0.0540 \quad 0.1385]^T \qquad \text{ให้ค่า} \quad f(\boldsymbol{x}_1) = 2.210\times10^{-2}$$

ดำเนินการคำนวณวนรอบจะได้ผลเฉลยและการลู่เข้าดังต่อไปนี้



รูปที่ 7.11 การลู่เข้าของจุดคำตอบโดยใช้กำหนดการวิวัฒนาการ $x^* = [-0.0011 \quad -0.0001]^T$

กล่องเครื่องมือ (TOOLBOX) สำหรับการคำนวณด้วยกำหนดการวิวัฒนาการนี้ ให้ชื่อว่า EP TOOLBOX สำหรับ SCILAB โดยที่ EP ย่อมาจาก Evolutionary Programming ประกอบด้วยไฟล์ทั้งสิ้น 7 ไฟล์ การเริ่มต้นใช้งานให้ดำเนินการ execute ไฟล์ ชื่อ ep_initial.sce เพื่อโหลดฟังก์ชันไฟล์ที่เกี่ยวข้องทั้งหมดและกำหนด seed number จาก ฟังก์ชัน getdate() เพื่อเป็นการสุ่มลำดับของตัวเลขสุ่มแบบเทียม pseudorandom number ที่ใช้ ไฟล์ทั้งหมดมีรายชื่อดังต่อไปนี้

epmain.sci	ฟังก์ชันหลัก
calc_f.sci	ฟังก์ชันประเมินค่าความฟิตของฟังก์ชันวัตถุประสงค์
offspr.sci	ฟังก์ชันแปลงรหัสตัวแปรค่าจริงเป็นรหัสตัวแปรไบนารี
survive.sci	ฟังก์ชันแปลงรหัสตัวแปรไบนารีไปเป็นรหัสตัวแปร
randp.sci	ฟังก์ชันสุ่มค่าประชากรเริ่มต้น
rand1.sci	ฟังก์ชันสุ่มค่าแบบกระจายสม่ำเสมอจากช่วงข้อมูลที่กำหนด

เนื่องจากไฟล์ที่เกี่ยวข้องมีจำนวนมาก ดังนั้น ชุดคำสั่งที่เขียนจะแสดงไว้ในภาคผนวก ในที่นี้จะนำเสนอตัวอย่างการใช้งานและผลการรันโปรแกรมเท่านั้น

ตัวอย่างที่ 7.8 ดำเนินการแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้โดยใช้กำหนดการวิวัฒนาการด้วยกล่องเครื่องมือ EP ที่พัฒนาขึ้น

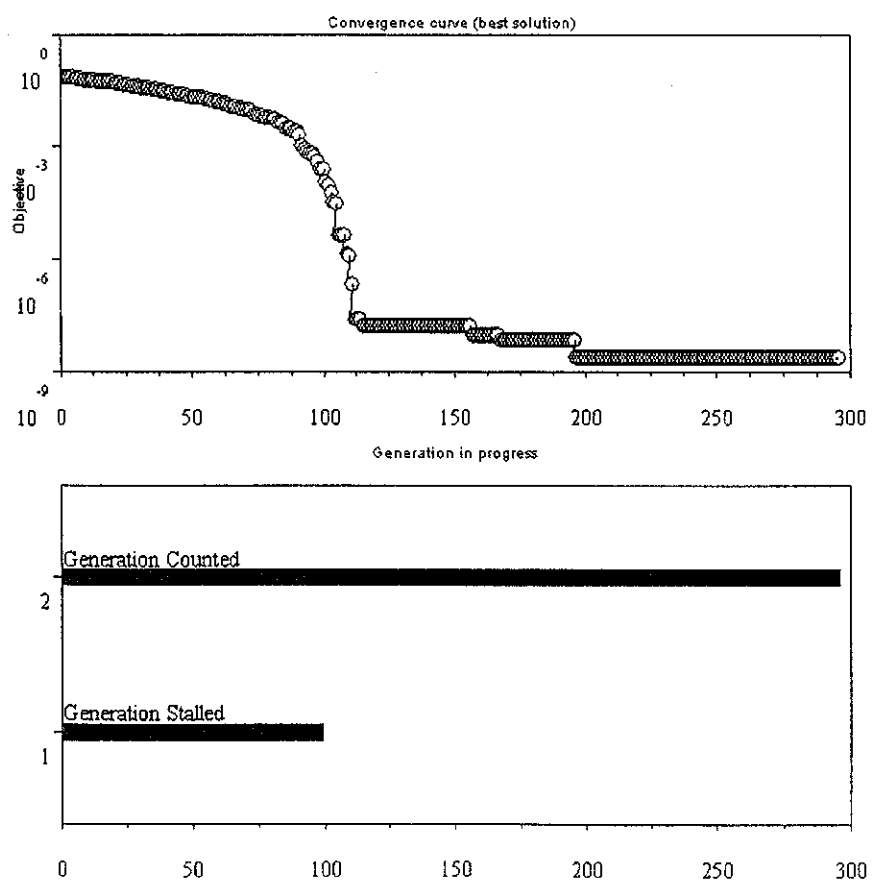
$$\text{Minimize} \qquad f(x) = x_1^2 + x_2^2$$

วิธีทำ

โปรแกรม SCILAB สำหรับตัวอย่างที่ 7.8

// SCILAB source program

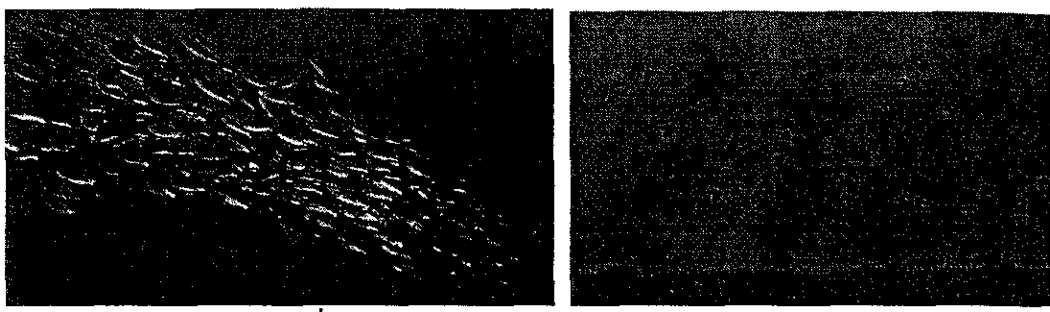
```
// ผลการรันโปรแกรม
--> exec('ep_initial.sce');
--> Ne = 2; Np = 20; xbeta = 0.02; xlimit = [-1 1;-1 1]; opt = [1000 100 0 4];
--> [xg,fg] = epmain(f_unc01,Ne,Np,xlimit,xbeta,opt);
--> [xg fg]
 ans  =
  - 0.0000074  - 0.0000490    2.458D-09
```



รูปที่ 7.12 การลู่เข้าของจุดคำตอบโดยใช้กำหนดการวิวัฒนาการสำหรับตัวอย่างที่ 7.8

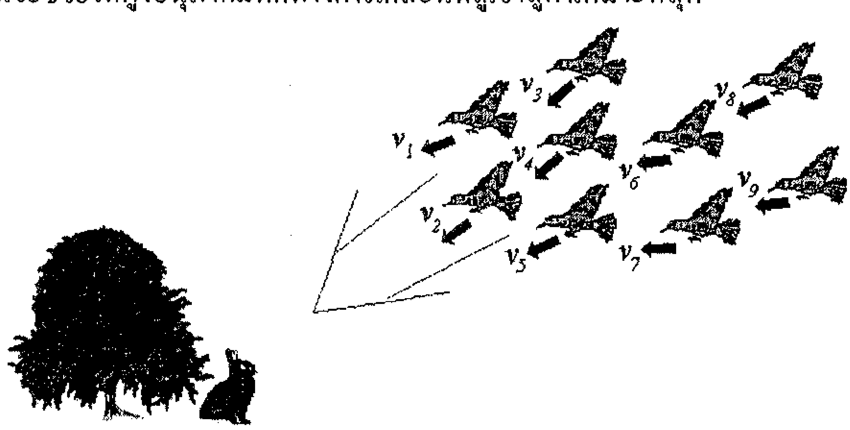
7.5 การหาค่าเหมาะที่สุดของฝูงอนุภาค (Particle Swarm Optimization: PSO)

ฝูงอนุภาคหรือพาร์ทิเคิลสวอร์ม เป็นการจำลองระบบหรือโครงสร้างทางสังคมของกลุ่มสิ่งมีชีวิตอย่างง่าย การรวมกลุ่มทางสังคมของสิ่งมีชีวิตเพื่อวัตถุประสงค์เฉพาะบางประการ เช่น การหาอาหารเพื่อการอยู่รอดของกลุ่ม เป็นส่วนสำคัญที่ช่วยผลักดันให้ประชากรส่วนใหญ่ในกลุ่มหรือสังคม หรือทั้งหมดมีพฤติกรรมที่สอดคล้องกัน กลุ่มสิ่งมีชีวิตที่มีพฤติกรรมในลักษณะนี้ ได้แก่ ฝูงผึ้ง (bee swarm) ฝูงปลา (fish school) ฝูงนก (bird flock) หรืออาณานิคมมด (ant colony) เป็นต้น ดังแสดงในรูปที่ 7.13



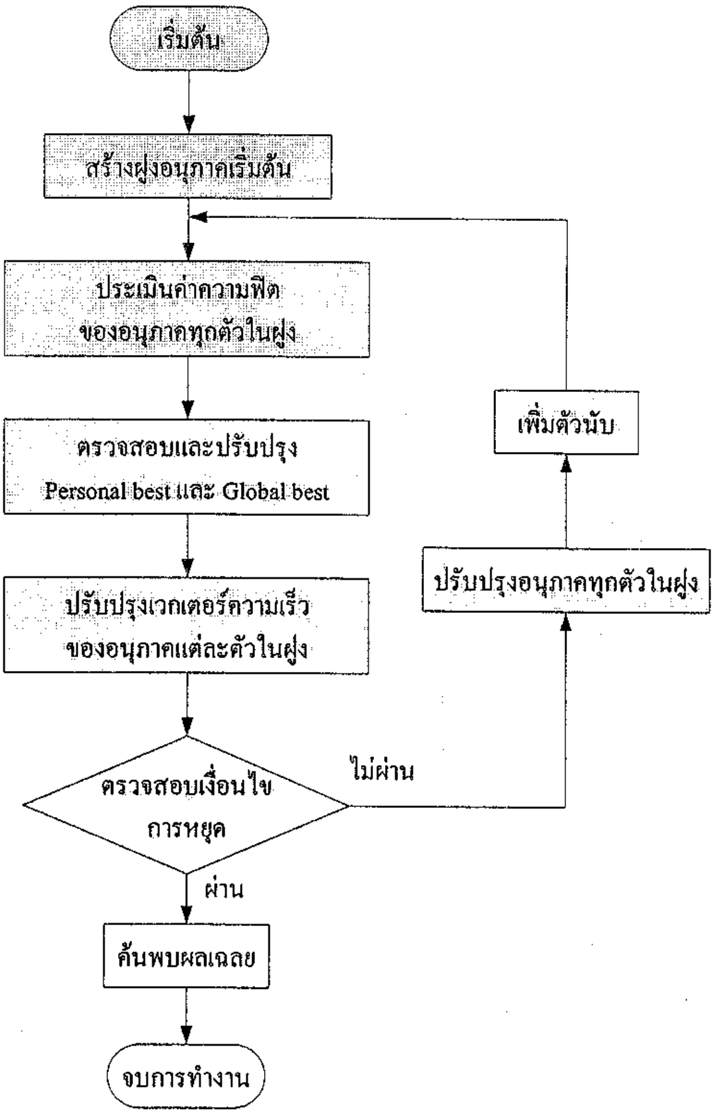
รูปที่ 7.13 ฝูงปลาและฝูงนกนำเสนอฝูงอนุภาค

พิจารณารูปการล่าเหยื่อของฝูงนกในรูปที่ 7.14 นกหนึ่งตัวนำเสนออนุภาคหนึ่งตัว ในการค้นหาเหยื่อ นกทั้งฝูงจะบินไปในทิศทางที่คล้อยตามกัน นกตัวที่อยู่ใกล้แหล่งอาหารมากที่สุด (มีค่าความฟิตดีที่สุด) จะมีระยะห่างของฝูงกับแหล่งอาหารน้อยที่สุด ดังนั้น กลยุทธ์ที่ดีที่สุด คือ นกตัวที่เหลือต้องคล้อยตามนกตัวดังกล่าว การจำลองฝูงอนุภาคอาศัยหลักการนี้ โดยกำหนดอนุภาคเพื่อใช้แทนผลเฉลยพร้อมด้วยการประเมินค่าความฟิต นอกจากนี้ คุณสมบัติที่สำคัญในการจำลองการหาอาหารของฝูงนก ได้แก่ เวกเตอร์ความเร็ว (particle's velocity) ของอนุภาคแต่ละตัวใช้เป็นตัวกำหนดทิศทางในการเคลื่อนที่ของอนุภาคนั้นๆ อนุภาคทุกตัวในฝูงจะต้องถูกปรับปรุงทิศทางในการเคลื่อนที่ให้สอดคล้องกับทิศทางการเคลื่อนที่ของอนุภาคตัวที่มีค่าความฟิตดีที่สุด ผลจากการดำเนินการนี้จะช่วยให้ฝูงอนุภาคมีทิศทางการเคลื่อนที่ลู่เข้าสู่ค่าเหมาะที่สุด



รูปที่ 7.14 การหาอาหารของฝูงนก

โดยมีรายละเอียดของอัลกอริทึมดังแผนผังการทำงานในรูปต่อไปนี้



รูปที่ 7.15 แผนผังการทำงานของการหาค่าเหมาะที่สุดด้วยฝูงอนุภาค

กำหนดให้ฝูงอนุภาคที่พิจารณาประกอบด้วยอนุภาคทั้งสิ้น n ตัว อนุภาคแต่ละตัวจะได้รับการปรับปรุงระหว่างกระบวนการวนรอบ ดังนี้

$$\boldsymbol{x}_i^{(k+1)} = \boldsymbol{x}_i^{(k)} + \boldsymbol{v}_i^{(k+1)}$$

$$\boldsymbol{v}_i^{(k+1)} = \boldsymbol{v}_i^{(k)} + \alpha_i\left(\boldsymbol{x}_i^{lbest} - \boldsymbol{x}_i^{(k)}\right) + \beta_i\left(\boldsymbol{x}^{gbest} - \boldsymbol{x}_i^{(k)}\right)$$

โดยที่ $\boldsymbol{x}_i^{(k)}$ แทนอนุภาคซึ่งนำเสนอผลเฉลยตัวที่ i ในรอบการคำนวณที่ k

$\boldsymbol{v}_i^{(k)}$ แทนเวกเตอร์ความเร็วของอนุภาคตัวที่ i ในรอบการคำนวณที่ k

α_i, β_i แทนตัวเลขสุ่มแบบสม่ำเสมอ (uniform random) ในช่วง [0,1]

$\boldsymbol{x}_i^{lbest}$ แทนผลเฉลยที่ดีที่สุดที่อนุภาคตัวที่ i ค้นพบ (personal best)

$\boldsymbol{x}^{gbest}$ แทนผลเฉลยที่ดีที่สุดที่ฝูงอนุภาคค้นพบ (global best)

ดำเนินการคำนวณวนรอบภายใต้เงื่อนไขการหยุดที่เหมาะสม จะได้ผลเฉลยของปัญหาค่าเหมาะที่สุด

ตัวอย่างที่ 7.9 ดำเนินการแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้โดยใช้ฝูงอนุภาค

$$\text{Minimize} \qquad f(\boldsymbol{x}) = x_1^2 + x_2^2$$

วิธีทำ โดยใช้การแก้ปัญหาด้วยคอมพิวเตอร์ จะได้ผลการคำนวณดังต่อไปนี้

สุ่มประชากรอนุภาคเริ่มต้น p_0 โดยใช้จำนวนอนุภาคทั้งสิ้น 5 ตัว จะได้

i	$x_{1,i}$	$x_{2,i}$	$f(x)$	
1	-0.9831	-0.5126	1.2292	← *lbest* for particle#1
2	-0.5697	0.7835	0.9384	← *lbest* for particle#2
3	-0.0579	-0.9041	0.8207	← *lbest* for particle#3
4	0.1661	-0.3025	0.1191	← *lbest* for particle#4 (*gbest*)
5	-0.9097	-0.7538	1.3959	← *lbest* for particle#5

กำหนดให้ความเร็วเริ่มต้น $\boldsymbol{v}$ เป็นเวกเตอร์ศูนย์

<u>รอบการคำนวณที่ 1</u>:

ปรับปรุงเวกเตอร์ความเร็ว จะได้ว่า

$$\boldsymbol{v}^{(1)} = \boldsymbol{v}^{(0)} + \alpha\left(\boldsymbol{x}^{lbest} - \boldsymbol{x}^{(0)}\right) + \beta\left(\boldsymbol{x}^{gbest} - \boldsymbol{x}^{(0)}\right)$$

$$\boldsymbol{v}^{(1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} + 0.436 \times \left\{ \begin{bmatrix} -0.9831 & -0.5126 \\ -0.5697 & 0.7835 \\ -0.0579 & -0.9041 \\ 0.1661 & -0.3025 \\ -0.9097 & -0.7538 \end{bmatrix} - \begin{bmatrix} -0.9831 & -0.5126 \\ -0.5697 & 0.7835 \\ -0.0579 & -0.9041 \\ 0.1661 & -0.3025 \\ -0.9097 & -0.7538 \end{bmatrix} \right\}$$

$$+0.161 \times \left\{ \begin{bmatrix} 0.1661 & -0.3025 \\ 0.1661 & -0.3025 \\ 0.1661 & -0.3025 \\ 0.1661 & -0.3025 \\ 0.1661 & -0.3025 \end{bmatrix} - \begin{bmatrix} -0.9831 & -0.5126 \\ -0.5697 & 0.7835 \\ -0.0579 & -0.9041 \\ 0.1661 & -0.3025 \\ -0.9097 & -0.7538 \end{bmatrix} \right\}$$

* 0.436 และ 0.161 เป็นตัวเลขสุ่มแบบสม่ำเสมอใดๆ ในช่วง [0,1] นั่นคือ

$$\boldsymbol{v}^{(1)} = \begin{bmatrix} 0.1850 & 0.0338 \\ 0.1185 & -0.1748 \\ 0.0361 & 0.0969 \\ 0 & 0 \\ 0.1732 & 0.0727 \end{bmatrix}$$

จะได้ว่า

$$x^{(1)} = x^{(0)} + v^{(1)} = \begin{bmatrix} -0.9831 & -0.5126 \\ -0.5697 & 0.7835 \\ -0.0579 & -0.9041 \\ 0.1661 & -0.3025 \\ -0.9097 & -0.7538 \end{bmatrix} + \begin{bmatrix} 0.1850 & 0.0338 \\ 0.1185 & -0.1748 \\ 0.0361 & 0.0969 \\ 0 & 0 \\ 0.1732 & 0.0727 \end{bmatrix}$$

$$= \begin{bmatrix} -0.7981 & -0.4788 \\ -0.4512 & 0.6087 \\ -0.0218 & -0.8072 \\ 0.1661 & -0.3025 \\ -0.7365 & -0.6811 \end{bmatrix}$$

นั่นคือ

i	$x_{1,i}$	$x_{2,i}$	$f(x)$	
1	−0.7981	−0.4788	0.8662	← *lbest* for particle#1
2	−0.4512	0.6087	0.5741	← *lbest* for particle#2
3	−0.0218	−0.8072	0.6521	← *lbest* for particle#3
4	0.1661	−0.3025	0.1191	← Still *lbest & gbest*
5	−0.7365	−0.6811	1.0064	← *lbest* for particle#5

<u>รอบการคำนวณที่ 12</u>:

ปรับปรุงเวกเตอร์ความเร็ว จะได้ว่า

$$v^{(2)} = \begin{bmatrix} 0.1850 & 0.0338 \\ 0.1185 & -0.1748 \\ 0.0361 & 0.0969 \\ 0 & 0 \\ 0.1732 & 0.0727 \end{bmatrix} + 0.299 \times \left\{ \begin{bmatrix} -0.7981 & -0.4788 \\ -0.4512 & 0.6087 \\ -0.0218 & -0.8072 \\ 0.1661 & -0.3025 \\ -0.7365 & -0.6811 \end{bmatrix} - \begin{bmatrix} -0.7981 & -0.4788 \\ -0.4512 & 0.6087 \\ -0.0218 & -0.8072 \\ 0.1661 & -0.3025 \\ -0.7365 & -0.6811 \end{bmatrix} \right\} + \ldots$$

$$\ldots + 0.309 \times \left\{ \begin{bmatrix} 0.1661 & -0.3025 \\ 0.1661 & -0.3025 \\ 0.1661 & -0.3025 \\ 0.1661 & -0.3025 \\ 0.1661 & -0.3025 \end{bmatrix} - \begin{bmatrix} -0.7981 & -0.4788 \\ -0.4512 & 0.6087 \\ -0.0218 & -0.8072 \\ 0.1661 & -0.3025 \\ -0.7365 & -0.6811 \end{bmatrix} \right\}$$

$$v^{(2)} = \begin{bmatrix} 0.5697 & 0.1042 \\ 0.3648 & -0.5384 \\ 0.1111 & 0.2982 \\ 0 & 0 \\ 0.5333 & 0.2237 \end{bmatrix}$$

จะได้ว่า

$$x^{(2)} = x^{(1)} + v^{(2)} = \begin{bmatrix} -0.7981 & -0.4788 \\ -0.4512 & 0.6087 \\ -0.0218 & -0.8072 \\ 0.1661 & -0.3025 \\ -0.7365 & -0.6811 \end{bmatrix} + \begin{bmatrix} 0.5697 & 0.1042 \\ 0.3648 & -0.5384 \\ 0.1111 & 0.2982 \\ 0 & 0 \\ 0.5333 & 0.2237 \end{bmatrix}$$

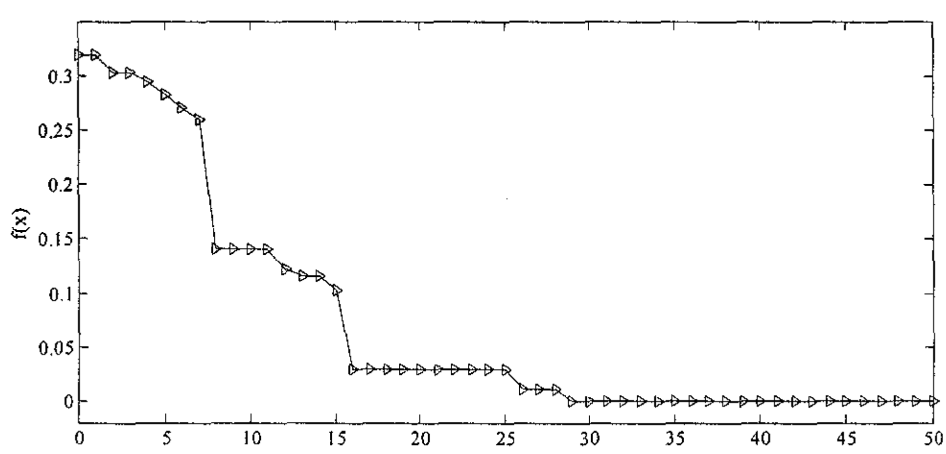
$$= \begin{bmatrix} -0.2284 & -0.3746 \\ -0.0865 & 0.0703 \\ 0.0892 & -0.5090 \\ 0.1661 & -0.3025 \\ -0.2032 & -0.4574 \end{bmatrix}$$

นั่นคือ

i	$x_{1,i}$	$x_{2,i}$	$f(x)$	
1	−0.2284	−0.3746	0.1925	← *lbest* for particle#1
2	−0.0865	0.0703	0.0124	← *lbest* for particle#2 & *gbest*
3	0.0892	−0.5090	0.2670	← *lbest* for particle#3
4	0.1661	−0.3025	0.1191	← Still *lbest* for particle#4
5	−0.2032	−0.4574	0.2505	← *lbest* for particle#5

จะพบว่า ในรอบนี้ *gbest* เกิดการเปลี่ยนแปลงและสามารถลดฟังก์ชันวัตถุประสงค์ให้มีค่าเหลือเพียง 0.0124 เท่านั้น เมื่อดำเนินการวนรอบต่อไปจะได้ผลเฉลยที่เหมาะที่สุด

ทดลองรันโปรแกรมโดยใช้จำนวนอนุภาคทั้งสิ้น 5 ตัวเริ่มต้นแบบสุ่ม จะได้ผลการลู่เข้าของคำตอบเป็นดังรูปต่อไปนี้



รูปที่ 7.16 การลู่เข้าของจุดคำตอบโดยใช้ฝูงอนุภาค $x^* = [-0.0084 \quad 0.0131]^T$

กล่องเครื่องมือ (TOOLBOX) สำหรับการคำนวณโดยใช้การหาค่าเหมาะที่สุดของฝูงอนุภาคนี้ ให้ชื่อว่า Particleswarm (PSO) TOOLBOX สำหรับ SCILAB ประกอบด้วยไฟล์ทั้งสิ้น 3 ไฟล์ การเริ่มต้นใช้งานให้ดำเนินการ execute ไฟล์ ชื่อ pso_initial.sce เพื่อโหลด

ฟังก์ชันไฟล์ที่เกี่ยวข้องทั้งหมดและกำหนด seed number จาก ฟังก์ชัน getdate() เพื่อเป็นการสุ่มลำดับของตัวเลขสุ่มแบบเทียม pseudorandom number ที่ใช้ ไฟล์ทั้งหมดมีรายชื่อดังต่อไปนี้

psomain.sci ฟังก์ชันหลัก

randx.sci ฟังก์ชันสุ่มค่าอนุภาคเริ่มต้น

ชุดคำสั่งที่เขียนจะแสดงไว้ในภาคผนวก ในที่นี้จะนำเสนอตัวอย่างการใช้งานและผลการรันโปรแกรมเท่านั้น

ตัวอย่างที่ 7.10 ดำเนินการแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้โดยใช้ฝูงอนุภาค

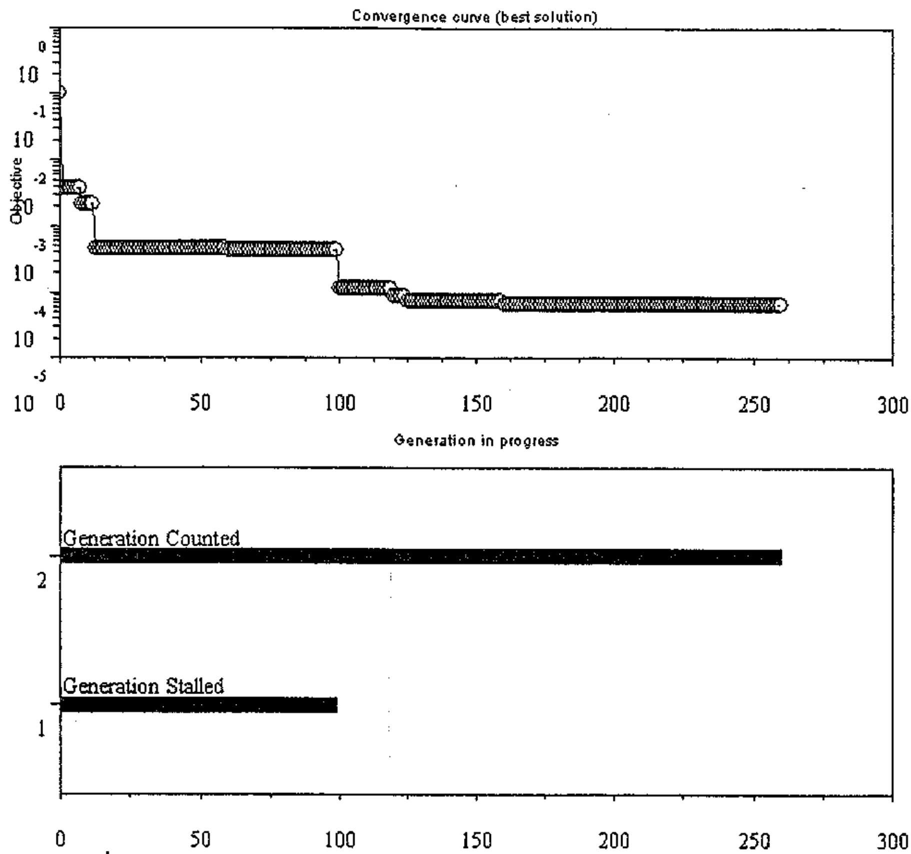
Minimize $f(x) = x_1^2 + x_2^2$

วิธีทำ โดยใช้การแก้ปัญหาด้วย SCILAB's PSO TOOLBOX จะได้ผลการคำนวณดังต่อไปนี้

โปรแกรม SCILAB สำหรับตัวอย่างที่ 7.10

// SCILAB source program

```
// ผลการรันโปรแกรม
--> exec('pso_initial.sce');
--> Ne = 2; Np = 5; Vmax = 10; xlimit = [-1 1;-1 1]; opt = [1000 100 0];
--> [gbest,fgbest,k] = psomain(f_unc01,Ne,Np,xlimit,Vmax,opt);
--> gbest
 gbest  =
   0.0006871    0.0081428
--> fgbest
 fgbest  =
   0.0000668
```



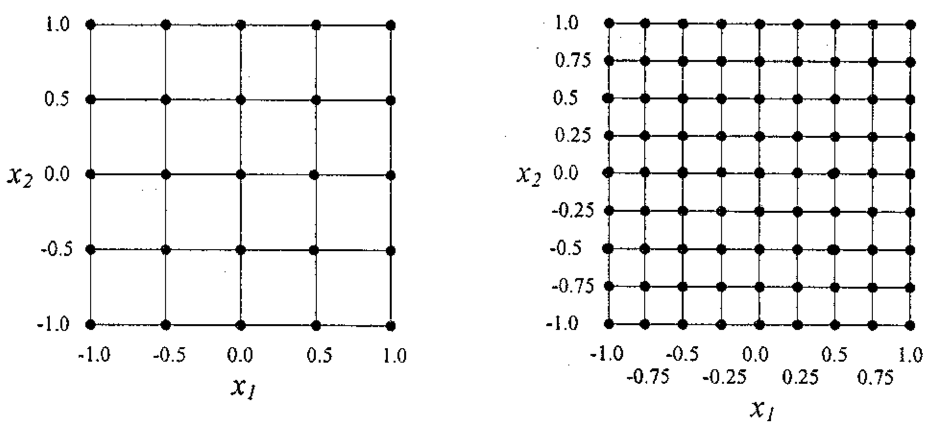
รูปที่ 7.17 การลู่เข้าของจุดคำตอบโดยใช้ฝูงอนุภาคสำหรับตัวอย่างที่ 7.10

7.6 การค้นหาตาบู (Tabu Search)

ในปี 1986 โกลฟ์เฟอร์ (Glover) นำเสนอการค้นหาตาบูเพื่อใช้แก้ปัญหาหาเหมาะที่สุดแบบคอมบิเนทอเรียล (combinatorial optimization) โดยใช้หลักการค้นหาจากเซตข้างเคียงของจุดคำตอบ (neighborhood search) และการกำหนดตาบูลิสต์ (tabu list: TL) กระบวนการค้นหาจะดำเนินการเหมือนกับเทคนิคการปีนเขา (hill-climbing method) การค้นหาจะลู่เข้าสู่จุดต่ำสุดเฉพาะถิ่น (local minimum) ในท้ายที่สุด กระบวนการค้นหาจะไม่สามารถปรับปรุงผลเฉลยได้อีก ปรากฏการณ์แบบนี้ เรียกว่า กับดักของจุดต่ำสุดเฉพาะถิ่น (local minimum trap) การค้นหาตาบู ได้นำเอากลยุทธ์ต่างๆ มาใช้เพื่อให้กระบวนการค้นหาสามารถหลบหนีจากกับดักดังกล่าวได้ ที่รู้จักกันเป็นอย่างดี ได้แก่ เกณฑ์ความทะเยอทะยาน (aspiration criteria) นอกจากนี้มีกลุ่มวิจัยจากสถาบันต่างๆ ทั่วโลก ได้พัฒนาการค้นหาตาบูให้มีสมรรถนะดียิ่งขึ้น เช่น การค้นหาตาบูรีแอกทีฟ (reactive tabu search: RTS) นำโดย บาตติติ และ เทคชิโอลิ (Battiti & Tecchiolli) ในปี ค.ศ. 1994 การค้นหาตาบูเชิงความน่าจะเป็น (probabilistic tabu search: PTS) นำโดย โคเชตอฟ และจอนเชนอฟ (Kochetov & Goncharov) ในปี ค.ศ. 2000 นอกจากนี้ ในช่วงปี ค.ศ. 2000 – 2005 คณะนักวิจัยสาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี ประกอบด้วย สราวุฒิ สุจิตจร, อาทิตย์ ศรีแก้ว, ธนัดชัย กุลวรวานิชพงษ์, เดชา

พวงดาวเรือง และกองพัน อารีรักษ์ ได้ร่วมกันพัฒนาเทคนิคการค้นหาตาบูเชิงปรับตัว (adaptive tabu search: ATS) โดยเพิ่มกลไกรัศมีการค้นหาเชิงปรับตัว (adaptive search radius mechanism) และกลไกการย้อนรอย (black-tracking mechanism) อย่างไรก็ตาม ในชั้นนี้ผู้เขียนจะนำเสนอหลักการเบื้องต้นของเทคนิคการค้นหาตาบูเชิงปรับตัวที่เพิ่มกลไกรัศมีการค้นหาเชิงปรับตัวเท่านั้น โดยไม่พิจารณาการใช้งานกลไกการย้อนรอย สำหรับรายละเอียดของการค้นหาตาบูเชิงปรับตัว นักศึกษา นักวิจัยและผู้สนใจทั่วไปสามารถติดตามได้จากตำรา เรื่อง Integrated Intelligent Systems for Engineering Design ของสำนักพิมพ์ IOS Press ซึ่งผู้แต่งร่วมกับคณะวิจัยได้เขียนไว้โดยละเอียดในบทที่ 12 ชื่อว่า Adaptive Tabu Search and Applications in Engineering Design ซึ่งจะมีรายละเอียดของการค้นหาตาบูเชิงปรับตัวที่สมบูรณ์เพื่อนำไปใช้ในงานวิจัยต่อไป

การค้นหาตาบูประกอบด้วยโครงสร้างหลักที่สำคัญ 4 ประการ ได้แก่ 1. ปริภูมิการค้นหา (search space) 2. เซตข้างเคียง (neighborhood set) 3. หน่วยความจำของการค้นหา (search memories) และ 4. เกณฑ์ความทะเยอทะยาน (aspiration criteria)

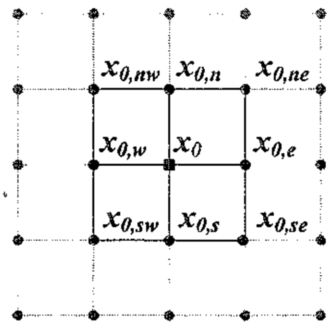


ความละเอียด = 0.5 ความละเอียด = 0.25

รูปที่ 7.18 การสร้างปริภูมิการค้นหา

- <u>การสร้างปริภูมิการค้นหา</u> (search space creation) เป็นการแบ่งปริภูมิการค้นหาออกเป็นที่ไม่ต่อเนื่อง (discrete search space) โดยมีความละเอียดเป็นตัวกำหนด พิจารณาลักษณะการสร้างปริภูมิการค้นหาได้จากรูปที่ 7.18

- <u>การสร้างเซตข้างเคียง</u> (neighborhood set creation) เป็นการสร้างเซตของผลเฉลยข้างเคียงของจุดคำตอบปัจจุบัน (current solution: x_0) จากรูปที่ 7.15 ในกรณีของปัญหาตัวแปรสองมิติ เมื่อปริภูมิการค้นหาถูกกำหนดค่าความละเอียดที่แน่นอนแล้ว จุดคำตอบที่อยู่ข้างจุด x_0 ทั้ง 8 จุด คือ สมาชิกในเซตข้างเคียง ดังรูป 7.19



x_0 นำเสนอผลเฉลยปัจจุบัน (current solution)

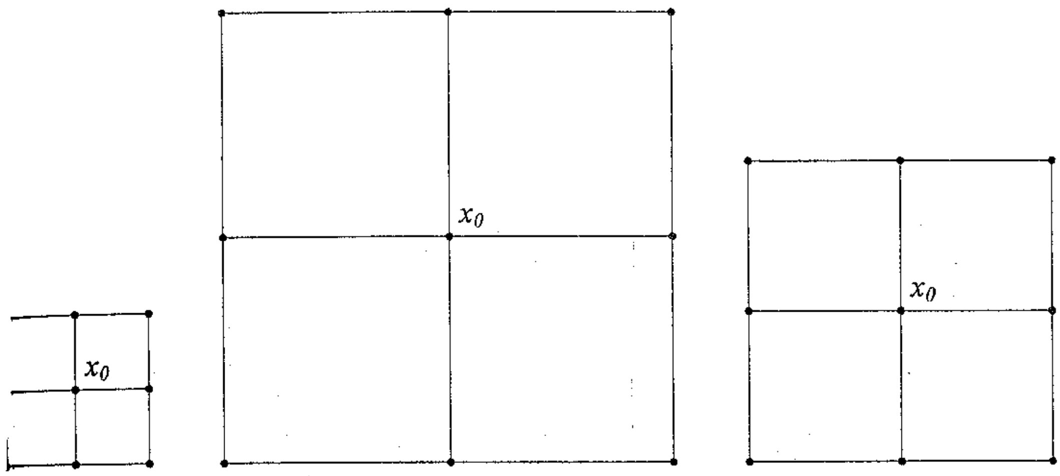
$x_{0,n}, x_{0,s}, x_{0,e}, x_{0,w}, x_{0,ne}, x_{0,nw}, x_{0,se}, x_{0,sw}$, นำเสนอสมาชิกของเซตข้างเคียงทั้ง 8 จุด $\aleph(x_0)$

รูปที่ 7.19 การสร้างเซตข้างเคียง

- <u>หน่วยความจำของการค้นหา</u> (search memory) เป็นองค์ประกอบที่สำคัญเพื่อช่วยให้กระบวนการค้นหาหลุดจากกับดักของจุดต่ำสุดเฉพาะถิ่นได้ (local minimum) โดยทั่วไปแล้วหน่วยความจำของการค้นหาแบ่งได้เป็น 2 ประเภท ได้แก่ หน่วยความจำระยะสั้น (short-term memory) ทำหน้าที่เก็บเซตของจุดคำตอบที่ถูกปรับปรุงในรอบการค้นหาที่ผ่านมา เพื่อกำหนดให้เป็นทิศทางต้องห้าม เรียกว่า ตาบูลิสต์ (tabu list) และหน่วยความจำระยะยาว (long-term memory) โดยจะเก็บเซตของจุดคำตอบเช่นกัน แต่จะมีช่วงเวลาในการเลือกเก็บที่ยาวนานมากกว่า ข้อมูลดังกล่าวจะใช้เพื่อเริ่มการค้นหาใหม่ หลังจากที่จุดต่ำสุดเฉพาะถิ่นใดๆ ถูกค้นพบแล้ว หน่วยความจำต่างๆ เหล่านี้มีคุณสมบัติแบบเข้าก่อนออกทีหลัง (first-in, last out) หรือนิยมเรียกว่า สแต็ก (stack)
- <u>เกณฑ์ความทะเยอทะยาน</u> (aspiration criteria) เป็นกลยุทธ์ที่ถูกกำหนดขึ้นมาเพื่อใช้ปลดล็อกการติดกับดักของจุดต่ำสุดเฉพาะถิ่น เซตคำตอบในตาบูลิสต์จะเป็นตัวกำหนดทิศทางต้องห้ามไม่ให้ค้นหา ผลของเกณฑ์ความทะเยอทะยานนี้จะไปปลดเงื่อนไขดังกล่าว นั่นคือ จะยอมให้การค้นหาไปในทิศทางต้องห้ามได้ แต่อาจจะต้องกำหนดกฎเกณฑ์บางอย่างกำกับไว้ เช่น เมื่อไม่สามารถปรับปรุงจุดคำตอบได้อีกแล้วเมื่อใช้การคำนวณวนอย่างน้อย 10 รอบการค้นหา เป็นต้น เกณฑ์ดังกล่าวไม่มีข้อกำหนดที่ตายตัว ผู้แก้ไขปัญหาสามารถกำหนดได้อย่างอิสระ

กลไกรัศมีการค้นหาเชิงปรับตัว (adaptive search radius mechanism) เป็นกลยุทธ์รูปแบบหนึ่งที่ถูกนำมาใช้ เมื่อกระบวนการค้นหาไม่สามารถปรับปรุงผลเฉลยได้เป็นจำนวนรอบที่มากกว่าที่กำหนด กลไกนี้จะปรับลดรัศมีการค้นหาซึ่งหมายถึงขอบเขตของเซตข้างเคียงลง โดยขอบเขตของเซตข้างเคียงที่ลดลง จะส่งผลให้การสร้างเซตข้างเคียงดังกล่าวมีโอกาสค้นพบผลเฉลยที่ดีกว่าผลเฉลยในปัจจุบัน อย่างไรก็ตาม อัตราการปรับลดค่าของรัศมีไม่มีกฎที่แน่ชัด ขึ้นอยู่กับปัญหาและต้องกำหนดโดยผู้ใช้งาน เช่น อาจจะปรับให้รัศมีการค้นหาลดลง 80% ของค่าเดิม เมื่อไม่สามารถปรับปรุงจุดคำตอบได้ในการคำนวณ 100 รอบการคำนวณติดกัน เป็นต้น พิจารณาได้ดังรูปที่ 7.20-ก นำเสนอรัศมีการค้นหาปกติสำหรับการค้นหาตาบูแบบดั้งเดิม การค้นหาด้วยหลักการนี้เสียเวลาในการคำนวณมากเกินไป ถ้าการค้นหามีคุณสมบัติลาดชัน การข้ามจุดหรือขยายขอบเขต

การสร้างเซตข้างเคียงให้กว้างขึ้น ดังรูปที่ 7.20-ข ย่อมทำให้ได้ผลเฉลยที่รวดเร็วขึ้น อย่างไรก็ตาม เมื่อใช้รัศมีการค้นหาที่กว้าง จะทำให้คำตอบลู่เข้าเร็ว แต่เมื่อผลเฉลยที่ถูกปรับปรุงมีค่าเข้าใกล้จุดต่ำสุดเฉพาะถิ่น การใช้รัศมีที่กว้างขึ้น ย่อมสุ่มจุดคำตอบข้ามจุดต่ำสุดเฉพาะถิ่น และไม่สามารถปรับปรุงจุดคำตอบได้เลย ดังนั้น เมื่อคำตอบไม่สามารถปรับปรุงได้อีกเป็นจำนวนรอบที่มากกว่าค่าที่กำหนด นั่นหมายความว่าผลเฉลยที่ปรับปรุงเข้าใกล้จุดต่ำสุดเฉพาะถิ่น ให้ปรับลดรัศมีการค้นหาลงดังรูปที่ 7.20-ค



รัศมีการค้นหาปกติ
ก)

รัศมีการค้นหาเริ่มต้นสำหรับกลไกรัศมีการค้นหาเชิงปรับตัว
ข)

รัศมีการค้นหาที่ถูกปรับสำหรับกลไกรัศมีการค้นหาเชิงปรับตัว
ค)

รูปที่ 7.20 กลไกรัศมีการค้นหาเชิงปรับตัว

กระบวนการค้นหาสามารถสรุปเป็นขั้นตอนย่อยๆ ได้ดังนี้

<u>ขั้นที่ 1</u>: สุ่มเลือกคำตอบเริ่มต้น x_0 กำหนดพารามิเตอร์เริ่มต้น

ให้ $\boldsymbol{x_{lbest}} = \boldsymbol{x_0}, \boldsymbol{x_{gbest}} = \boldsymbol{x_0}, TL =$ เซตว่าง

<u>ขั้นที่ 2</u>: สร้างผลเฉลยแบบสุ่มจากเซตข้างเคียง $\aleph(x_0)$ ของ x_0 โดยที่ $x \in \aleph(x_0)$

<u>ขั้นที่ 3</u>: ค้นหาสมาชิกที่ให้ค่าความฟิตดีที่สุด $\boldsymbol{x_{nbest}}$ จากเซตข้างเคียงที่สร้างขึ้น

ถ้า $\boldsymbol{x_{nbest}}$ ให้ค่าความฟิตดีกว่า $\boldsymbol{x_{lbest}}$ จะได้ว่า $\boldsymbol{x_{lbest}} = \boldsymbol{x_{nbest}}$ เก็บ $\boldsymbol{x_{nbest}}$ ไว้ในตาบูลิสต์

ถ้าไม่ใช่ ให้ตรวจสอบการติดกับดักเกณฑ์ความทะเยอทะยาน

- สอดคล้องตามเกณฑ์ความทะเยอทะยาน
  - * ถ้าจุดต่ำสุดเฉพาะถิ่นถูกค้นพบข้ามไปขั้นที่ 5
  - ** ถ้าไม่ใช่ข้ามไปขั้นที่ 4
- ไม่สอดคล้องตามเกณฑ์ความทะเยอทะยาน ทำซ้ำขั้นที่ 2
- ตรวจสอบเงื่อนไขการปรับรัศมีการค้นหา

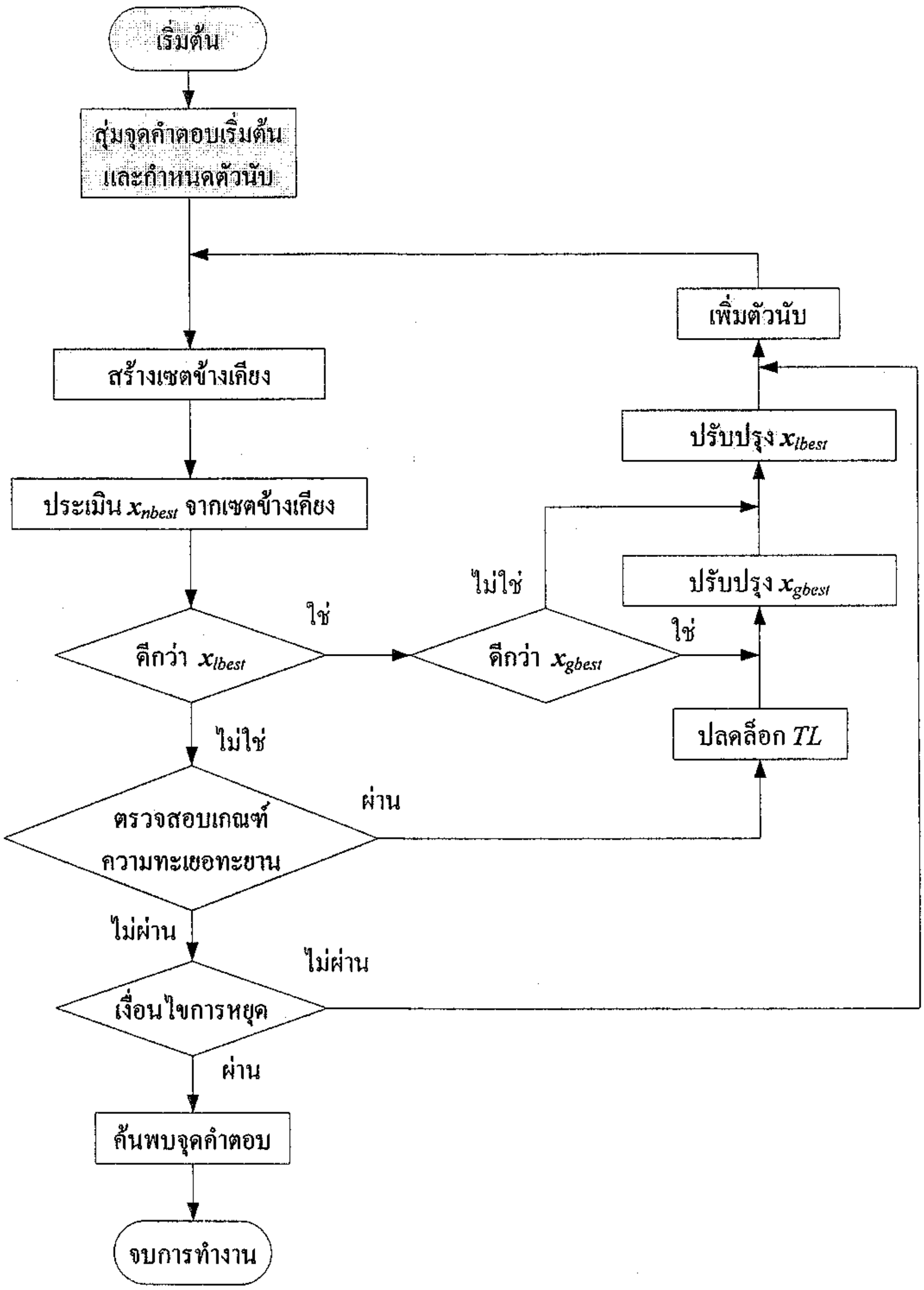
<u>ขั้นที่ 4</u>: ปลดเงื่อนไขตาบูลิสต์ ทำซ้ำขั้นที่ 2

<u>ขั้นที่ 5</u>: ตรวจสอบเงื่อนไขการหยุด

ถ้าไม่เป็นจริงให้เริ่มต้นการค้นหาจากหน่วยความจำระยะยาวที่เก็บไว้ ทำซ้ำขั้นที่ 2

<u>ขั้นที่ 6</u>: ค้นพบผลเฉลยของปัญหาการหาค่าเหมาะที่สุด

อาจจะสรุปเป็นแผนผังการทำงานได้ดังนี้



รูปที่ 7.21 แผนผังการทำงานของการหาค่าเหมาะที่สุดด้วยการค้นหาตาบู

ตัวอย่างที่ 7.11 ดำเนินการแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้โดยใช้การค้นหาตาบู

$$\text{Minimize} \qquad f(x) = x_1^2 + x_2^2$$

วิธีทำ กำหนดให้ความละเอียดของการเปลี่ยนแปลงค่าตัวแปรเท่ากับ 0.01

เนื่องจากปัญหานี้มีจุดต่ำสุดเพียงจุดเดียว ทำให้กับดักของจุดต่ำสุดเฉพาะถิ่นไม่มีผลต่อการค้นหา ดังนั้นจะไม่ใช้เกณฑ์ความทะเยอทะยาน

สุ่มจุดเริ่มต้น จะได้ $\boldsymbol{x_0} = [0.2 \;\; -0.4]^T$ ให้ค่า $f(x_0) = 0.2$

ให้ $\boldsymbol{x_{lbest}} = \boldsymbol{x_0}$,

<u>รอบการคำนวณที่ 1</u>:

สร้างเซตข้างเคียงโดยใช้การเปลี่ยนแปลงค่าตัวแปร 0.01 ในแต่ละมิติ จะได้ทั้งสิ้น 8 ตัว

i	x_1	x_2	*fobj*	*remark*
1	0.20	-0.39	0.1921	
2	0.20	-0.41	0.2081	
3	0.21	-0.39	0.3621	
4	0.21	-0.40	0.2041	
5	0.21	-0.41	0.2122	
6	0.19	-0.39	0.1882	*Neighborhood best*: $\boldsymbol{x_{lbest}}$
7	0.19	-0.40	0.1961	
8	0.19	-0.41	0.2042	

เก็บ $\boldsymbol{x_0} = [0.2 \;\; -0.4]^T$ ไว้ในตาบูลิสต์ $TL = \{\boldsymbol{x_0}\}$ จะได้ $\boldsymbol{x_1} = [0.19 \;\; -0.39]^T$

<u>รอบการคำนวณที่ 2</u>:

สร้างเซตข้างเคียงโดยใช้การเปลี่ยนแปลงค่าตัวแปร 0.01 ในแต่ละมิติ จะได้ทั้งสิ้น 8 ตัว

i	x_1	x_2	*fobj*	*remark*
1	0.19	-0.38	0.1805	
2	0.19	-0.4	0.1961	
3	0.20	-0.38	0.1844	
4	0.20	-0.39	0.1921	
5	0.20	-0.40	Don't care	*Tabu move*
6	0.18	-0.38	0.1768	*Neighborhood best*: $\boldsymbol{x_{lbest}}$
7	0.18	-0.39	0.1845	
8	0.18	-0.40	0.1924	

เก็บ $\boldsymbol{x_1} = [0.19 \;\; -0.39]^T$ ไว้ในตาบูลิสต์ $TL = \{\boldsymbol{x_0}, \boldsymbol{x_1}\}$ จะได้ $\boldsymbol{x_2} = [0.18 \;\; -0.38]^T$

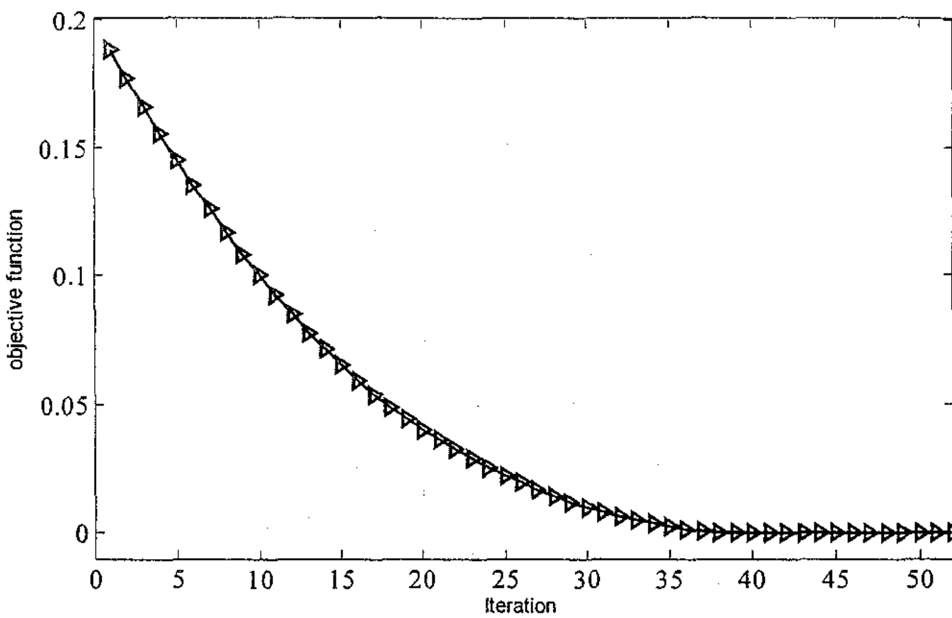
ทำการคำนวณวนรอบจะได้การลู่เข้าสู่จุดคำตอบของผลเฉลยดังกราฟในรูปที่ 7.22

กล่องเครื่องมือ (TOOLBOX) สำหรับการคำนวณโดยใช้การค้นหาแบบตาบูนี้ ให้ชื่อว่า Tabu Search (TS) TOOLBOX สำหรับ SCILAB ประกอบด้วยไฟล์ทั้งสิ้น 3 ไฟล์ การเริ่มต้นใช้งานให้ดำเนินการ execute ไฟล์ ชื่อ ts_initial.sce เพื่อโหลดฟังก์ชันไฟล์ที่เกี่ยวข้องทั้งหมด และกำหนด seed number จาก ฟังก์ชัน getdate() เพื่อเป็นการสุ่มลำดับของตัวเลขสุ่มแบบเทียม pseudorandom number ที่ใช้ ไฟล์ทั้งหมดมีรายชื่อดังต่อไปนี้

tsmain.sci ฟังก์ชันหลัก

randp.sci ฟังก์ชันสุ่มผลเฉลยเริ่มต้น

ชุดคำสั่งที่เขียนจะแสดงไว้ในภาคผนวก ในที่นี้จะนำเสนอตัวอย่างการใช้งานและผลการรันโปรแกรมเท่านั้น ดังแสดงในตัวอย่างที่ 7.12



รูปที่ 7.22 การลู่เข้าของจุดคำตอบโดยใช้การค้นหาตาบู $x^* = [0.0000 \quad 0.0000]^T$

ตัวอย่างที่ 7.12 ดำเนินการแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้โดยใช้การค้นหาตาบู

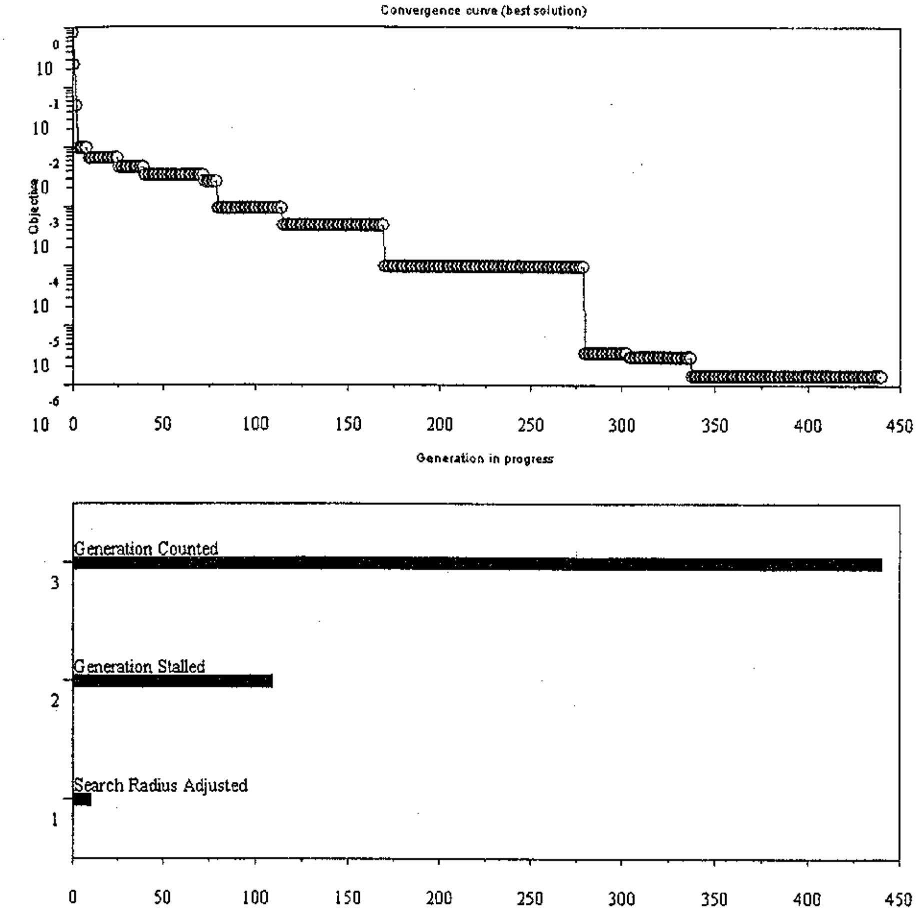
$$\text{Minimize} \qquad f(x) = x_1^2 + x_2^2$$

วิธีทำ โดยใช้โปรแกรม SCILAB จะได้ว่า

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 7.12</u>

```
// SCILAB source program
```

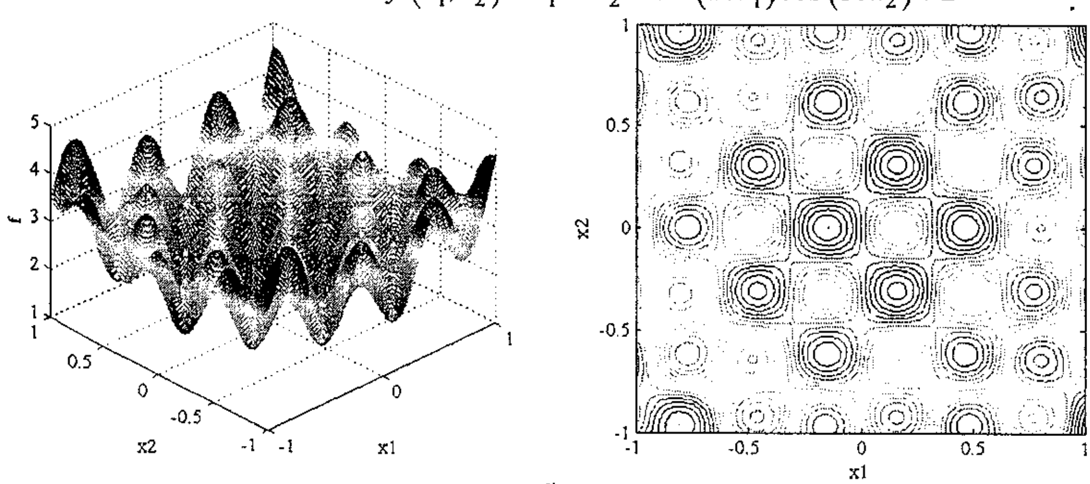
```
// ผลการรันโปรแกรม
--> exec('ts_initial.sce');
--> x0 = [-0.25 0.5]; xlimit = [-1 1 10000;-1 1 10000]; opt = [1000 100 10 0 1 0.2];
--> [xgbest,fgbest] = tsmain(f_unc01,x0,xlimit,opt);
--> xgbest
 xgbest  =
        - 0.0005001    0.0011001
--> fgbest
 fgbest  =
        0.0000015
```



รูปที่ 7.23 การลู่เข้าของจุดคำตอบโดยใช้การค้นหาตาบูเชิงปรับตัว

ตัวอย่างที่ 7.13 จงแก้ปัญหาค่าเหมาะที่สุดต่อไปนี้

$$\text{Minimize} \qquad f(x_1, x_2) = x_1^2 + x_2^2 + sin(10x_1)cos(10x_2) + 2$$



วิธีทำ โดยใช้ระเบียบวิธีการแก้ปัญหาดังต่อไปนี้

1. ระเบียบวิธีชันที่สุด (SDM)
2. ระเบียบวิธีคล้ายนิวตัน – ดีเอฟพี (DFP)
3. จีนเนติกอัลกอริทึม (GAs) Population Size = 20

4. กำหนดการวิวัฒนาการ (EP) Population Size = 20
5. การหาค่าเหมาะที่สุดของฝูงอนุภาค (PSO) Population Size = 20
6. การค้นหาตาบู (TS)

ดำเนินการแก้ปัญหาค่าเหมาะที่สุดโดยใช้การสุ่มผลเฉลยเริ่มต้นจำนวน 30 ครั้ง ในช่วงการค้นหา [-1,1] ภายใต้เงื่อนไขการหยุดที่ค่าความคลาดเคลื่อน 1×10^{-4} สำหรับ SDM และ DFP จำนวนรอบสูงสุด (maximum generation allowance) เท่ากับ 1000 รอบ จำนวนรอบที่ไม่สามารถปรับปรุงผลเฉลยได้ (maximum stalled generation) เท่ากับ 1000 รอบ สำหรับ GAs EP PSO และ TS

คำนวณเกรเดียนต์ได้เป็น

$$\nabla f(x_1,x_2)=\begin{bmatrix}2x_1+10\cos(10x_1)\cos(10x_2)\\2x_2-10\sin(10x_1)\sin(10x_2)\end{bmatrix}$$

เขียนเป็นฟังก์ชันไฟล์ได้ดังนี้

โปรแกรม SCILAB สำหรับตัวอย่างที่ 7.13

// SCILAB source program

```
// กำหนดฟังก์ชันวัตถุประสงค์และเกรเดียนต์
==============================
          exam7_13.sce
==============================
function f = fobj7_13(x)
f = x(1)^2 + x(2)^2 + sin(10*x(1))*cos(10*x(2)) + 2;
endfunction

function g = fgrad7_13(x)
g = [2*x(1) + 10*cos(10*x(1))*cos(10*x(2));
    2*x(2) - 10*sin(10*x(1))*sin(10*x(2))];
endfunction
```

○ ผลการแก้ปัญหาโดยใช้ระเบียบวิธีชันที่สุด (SDM)

```
// กำหนดฟังก์ชันสำหรับทดสอบผลการรันโปรแกรม
==============================
          runstp.sci
==============================
function [xg,fg,kg,tg] = runstp(fobj,fgrad,Ntrial)
xlimit = [-1 1; -1 1];
```

```
ftol = 1e-4;
vtol = 1e-4;
opt = 0;
for u=1:Ntrial
   x0 = randp(2,1,xlimit);
   tic;
   [xmin,fmin,k]=steepest(fobj,fgrad,x0',ftol,vtol,opt);
   t = toc;
   xg(u,1) = xmin(1);
   xg(u,2) = xmin(2);
   fg(u,1) = fmin;
   kg(u,1) = k;
   tg(u,1) = t;
   disp(['Trial # ' msprintf('%i',u) ' was successfully performed']);
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('exam7_13.sce');
--> [xg,fg,tg,kg] = runstp(fobj7_13,fgrad7_13,30);
--> [xg fg tg kg]
//      x1            x2           f(x1,x2)     time      Iter
ans  =
   0.1539934  - 0.3079926   1.1209495   0.000    3.
   0.7696644    0.9236626   2.4753184   0.000    3.
   1.0778303  - 6.314D-11   2.1852289   0.000    5.
 - 0.1539992    1.286D-09   1.0241902   0.000    4.
   0.1539934  - 0.3079925   1.1209495   0.015    4.
   0.4619156    0.6159147   1.6047095   0.000    3.
 - 0.4619683  - 0.3079695   1.3144609   0.000    4.
   0.1539934  - 0.3079917   1.1209495   0.000    3.
   0.1539934    0.3079926   1.1209495   0.000    4.
 - 0.1539759    0.6159620   1.4112202   0.000    3.
   0.1539934  - 0.3079926   1.1209495   0.000    4.
 - 0.4619685  - 0.3079694   1.3144609   0.000    3.
```

0.1539467	0.9238842	1.8949802	0.000	2.
- 0.7698177	0.6158187	1.9916726	0.016	4.
0.7696635	0.9236626	2.4753184	0.000	5.
0.1539462	0.9238841	1.8949802	0.015	3.
0.7696639	0.9236621	2.4753184	0.000	3.
0.1539465	- 0.9238839	1.8949802	0.016	3.
0.7699074	0.3079221	1.7014688	0.000	4.
- 0.1539760	- 0.6159618	1.4112202	0.016	4.
- 0.1539992	- 5.267D-11	1.0241902	0.000	3.
- 0.4619686	0.3079693	1.3144609	0.015	3.
- 0.7698170	0.615819	1.9916726	0.000	3.
0.7699081	0.3079221	1.7014688	0.000	4.
1.0776558	- 0.6156700	2.5720777	0.000	4.
- 1.0777869	- 0.3078493	2.2819433	0.015	5.
- 0.1539759	0.6159619	1.4112202	0.000	4.
- 0.1539992	- 2.425D-10	1.0241902	0.016	4.
0.7696630	- 0.9236628	2.4753184	0.000	4.
- 0.153976	0.6159619	1.4112202	0.016	3.

○ ผลการแก้ปัญหาโดยใช้ระเบียบวิธีคล้ายนิวตัน – ดีเอฟพี (DFP)

```
// กำหนดฟังก์ชันสำหรับทดสอบผลการรันโปรแกรม
============================
            rundfp.sci
============================
function [xg,fg,tg,kg] = rundfp(fobj,fgrad,Ntrial)
xlimit = [-1 1; -1 1];
ftol = 1e-4;
vtol = 1e-4;
opt = 0;
B0 = eye(2,2);
for u=1:Ntrial
    x0 = randp(2,1,xlimit);
    tic;
    [xmin,fmin,k]=dfp(fobj,fgrad,x0',B0,ftol,vtol,opt);
    t = toc();
```

```
  xg(u,1) = xmin(1);
  xg(u,2) = xmin(2);
  fg(u,1) = fmin;
  kg(u,1) = k;
  tg(u,1) = t;
  disp(['Trial # ' msprintf('%i',u) ' was successfully performed']);
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('exam7_13.sce');
--> [xg,fg,tg,kg] = rundfp(fobj7_13,fgrad7_13,30);
--> [xg fg tg kg]
//      x1          x2          f(x1,x2)    time    Iter
ans  =
   0.4619156    0.6159148    1.6047095   0.000    7.
   0.461986   - 8.164D-09    1.2177088   0.000    5.
 - 0.4618250  - 0.9238113    2.0884319   0.000    5.
 - 0.1539760  - 0.6159619    1.4112202   0.000    4.
   0.1539462    0.9238841    1.8949802   0.000    5.
 - 0.4619685    0.3079694    1.3144609   0.015    5.
 - 0.4618254    0.9238112    2.0884319   0.000    5.
 - 0.4619688    0.3079691    1.3144609   0.016    5.
 - 0.1539992  - 1.477D-08    1.0241902   0.000    7.
 - 0.1539759    0.6159619    1.4112202   0.000    6.
 - 0.7698177    0.6158186    1.9916726   0.000    4.
   0.4619155    0.6159148    1.6047095   0.000    7.
 - 0.1539992    1.895D-08    1.0241902   0.016    7.
 - 0.4618252    0.9238106    2.0884319   0.000    4.
   0.4619156  - 0.6159147    1.6047095   0.016    5.
 - 0.4618252    0.9238112    2.0884319   0.000    6.
 - 0.1539760  - 0.6159619    1.4112202   0.000    4.
   0.4619155  - 0.6159147    1.6047095   0.000    5.
 - 0.1539758    0.6159619    1.4112202   0.000    6.
 - 0.4619685    0.3079693    1.3144609   0.000    6.
```

- 0.1539759	- 0.6159619	1.4112202	0.015	6.
- 0.4619688	- 0.3079694	1.3144609	0.000	5.
0.1539462	- 0.9238841	1.8949802	0.016	6.
0.4619860	2.819D-08	1.2177088	0.000	5.
1.0778308	- 0.0000008	2.1852289	0.016	6.
0.1539934	- 0.3079926	1.1209495	0.000	6.
0.4619860	4.477D-08	1.2177088	0.015	5.
0.7699085	0.3079224	1.7014688	0.000	4.
0.7699089	- 0.3079217	1.7014688	0.016	5.
0.1539462	- 0.9238841	1.8949802	0.000	6.

○ <u>ผลการแก้ปัญหาโดยใช้จีนเนติกอัลกอริทึม (GAs)</u>

```
// กำหนดฟังก์ชันสำหรับทดสอบผลการรันโปรแกรม
=============================
            runga.sci
=============================
function [xg,fg,tg,kg] = runga(fobj,Ntrial)
xlimit = [-1 1 10; -1 1 10];
opt = [100 20 0];
xprop = 0.8;
mprop = 0.01;
Ne = 2; Np = 20;
for u=1:Ntrial
   tic;
   [pg,xmin,fmin,z] = ga0(fobj,Ne,Np,xlimit,xprop,mprop,opt)
// ga0.sci เหมือนกับ sga.sci ต่างกันตรงที่ไม่ให้มีการแสดงผลด้วยการวาดกราฟ
   t = toc();
   xg(u,1) = xmin(1);
   xg(u,2) = xmin(2);
   fg(u,1) = fmin;
   kg(u,1) = z;
   tg(u,1) = t;
   disp(['Trial # ' msprintf('%i',u) ' was successfully performed']);
end
endfunction
```

ผลการรันโปรแกรม

```
-> exec('exam7_13.sce');
-> [xg,fg,tg,kg] = runga(fobj7_13,30);
-> [xg fg tg kg]
      x1            x2          f(x1,x2)     time      Iter
ans =
   0.1358749    0.3235582    1.1498647    0.204    21.
 - 0.1632454  - 0.0166178    1.042575     0.266    35.
   0.1339198    0.3470186    1.2171298    0.281    37.
 - 0.4897361  - 0.3313783    1.3812474    0.156    20.
 - 0.1280547    0.0635386    1.249257     0.187    23.
   0.1417400    0.3079179    1.1285701    0.313    42.
   0.4349951    0.0087977    1.2578791    0.328    44.
 - 0.1319648  - 0.0459433    1.1513406    0.172    20.
 - 0.1652004  - 0.0263930    1.0657969    0.281    37.
   0.4467253    0.0400782    1.3079402    0.172    20.
 - 0.1808407  - 0.0205279    1.0816274    0.312    43.
 - 0.1495601  - 0.0009775    1.0252426    0.515    72.
 - 0.1573803    0.0107527    1.0306641    0.266    36.
   0.1612903  - 0.3235582    1.1360006    0.297    40.
 - 0.1749756    0.0342131    1.10479      0.359    48.
 - 0.1495601    0.0459433    1.1307078    0.219    28.
   0.1436950  - 0.3059629    1.1265327    0.422    57.
 - 0.1280547  - 0.0400782    1.1357609    0.235    31.
 - 0.1495601    0.0048876    1.0264088    0.219    27.
   0.1769306    0.3509286    1.2396223    0.203    25.
   0.1671554  - 0.3313783    1.1575375    0.172    20.
 - 0.1476051  - 0.0615836    1.2129499    0.156    20.
 - 0.1593353    0.0303030    1.0721122    0.187    24.
 - 0.4565005    0.3079179    1.3159735    0.203    26.
   0.4486804  - 0.0146628    1.2373244    0.203    27.
 - 0.1339198  - 0.0166178    1.0583178    0.485    68.
   0.1691105  - 0.3470186    1.2093645    0.344    47.
   0.1339198    0.3431085    1.2028572    0.156    20.
```

```
 - 0.1827957    0.0303030    1.1112816    0.343    47.
   0.1202346  - 0.3079179    1.1781996    0.187    24.
```

○ <u>ผลการแก้ปัญหาโดยใช้กำหนดการวิวัฒนาการ (EP)</u>

```
// กำหนดฟังก์ชันสำหรับทดสอบผลการรันโปรแกรม
==============================
             runep.sci
==============================
function [xg,fg,tg,kg] = runep(fobj,Ntrial)
xlimit = [-1 1; -1 1];
opt = [100 20 0 4];
xbeta = 0.02;
Ne = 2;
Np = 20;
for u=1:Ntrial
    tic;
    [xmin,fmin,k] = ep0(fobj,Ne,Np,xlimit,xbeta,opt)
// ep0.sci เหมือนกับ epmain.sci ต่างกันตรงที่ไม่ให้มีการแสดงผลด้วยการวาดกราฟ
    t = toc();
    xg(u,1) = xmin(1);
    xg(u,2) = xmin(2);
    fg(u,1) = fmin;
    kg(u,1) = k;
    tg(u,1) = t;
    disp(['Trial # ' msprintf('%i',u) ' was successfully performed']);
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('exam7_13.sce');
--> [xg,fg,tg,kg] = runep(fobj7_13,30);
--> [xg fg tg kg]
//      x1            x2          f(x1,x2)    time      Iter
ans  =
    0.4621299    0.0000621    1.2177101    0.203    82.
```

```
  0.1540384  - 0.3081252   1.1209505   0.203   100.
- 0.1538917    0.6160998   1.4112215   0.125    59.
  0.1541469    0.3081460   1.1209519   0.204   100.
- 0.1540425    0.0001375   1.0241912   0.11     56.
- 0.1539389    0.6158387   1.411221    0.14     65.
- 0.1539954  - 0.0000078   1.0241902   0.156    79.
- 0.1540413    0.0001197   1.024191    0.188   100.
  0.4618922  - 0.0000562   1.2177094   0.172    90.
  0.4619759  - 0.6160439   1.6047105   0.172    91.
- 0.1539445  - 0.0000205   1.0241903   0.203   100.
- 0.1540660    0.0001054   1.024191    0.125    60.
  0.1539290  - 0.3078451   1.1209508   0.187   100.
  0.4620275    0.0001624   1.2177102   0.172    87.
  0.1519611  - 0.3094154   1.1212633   0.203   100.
- 0.4620119  - 0.3078564   1.3144616   0.141    70.
- 0.1538800    0.6159499   1.4112207   0.11     55.
  0.1539732    0.3080233   1.1209496   0.188    93.
- 0.4620394    0.3080554   1.3144615   0.187    97.
- 0.1540239  - 0.6159488   1.4112203   0.125    66.
- 0.4619129  - 0.3082837   1.314466    0.171    87.
- 0.1540146  - 0.0000290   1.0241902   0.203   100.
- 0.4620848  - 0.3079474   1.3144616   0.156    78.
  0.1540909  - 0.3079938   1.12095     0.203   100.
- 0.1539279  - 0.0000144   1.0241904   0.188    93.
- 0.1539745  - 0.6159533   1.4112202   0.203   100.
- 0.1539529    0.6159987   1.4112203   0.125    62.
- 0.1538570  - 0.0000014   1.0241912   0.188    93.
  0.4620132  - 0.0001245   1.2177096   0.171    92.
- 0.1541356  - 0.0000248   1.0241911   0.203   100.
```

○ <u>ผลการแก้ปัญหาโดยใช้การหาค่าเหมาะที่สุดของฝูงอนุภาค (PSO)</u>

```
// กำหนดฟังก์ชันสำหรับทดสอบผลการรันโปรแกรม
==========================
        runpso.sci
==========================
```

```
function [xg,fg,tg,kg] = runpso(fobj,Ntrial)
xlimit = [-1 1; -1 1];
opt = [100 20 0 4];
Vmax = 15;
Ne = 2;
Np = 20;
for u=1:Ntrial
    tic;
    [xmin,fmin,k] = pso0(fobj,Ne,Np,xlimit,Vmax,opt)
// pso0.sci เหมือนกับ psomain.sci ต่างกันตรงที่ไม่ให้มีการแสดงผลด้วยการวาดกราฟ
    t = toc();
    xg(u,1) = xmin(1);
    xg(u,2) = xmin(2);
    fg(u,1) = fmin;
    kg(u,1) = k;
    tg(u,1) = t;
    disp(['Trial # ' msprintf('%i',u) ' was successfully performed']);
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('exam7_13.sce');
--> [xg,fg,tg,kg] = runpso(fobj7_13,30);
--> [xg fg tg kg]
//      x1            x2           f(x1,x2)    time      Iter
ans  =
  - 0.1578482  - 0.0042116   1.0258501   0.047   52.
  - 0.1554861    0.0012090   1.0243774   0.063   50.
    0.1531581    0.3028504   1.1223274   0.047   39.
  - 0.1585532    0.0051209   1.0265847   0.062   50.
  - 0.1560298    0.0022755   1.0246645   0.031   35.
    0.1483760    0.3208523   1.1309774   0.047   42.
  - 0.1511793    0.0037332   1.0253048   0.046   37.
  - 0.1542775  - 0.0034212   1.0247908   0.079   65.
  - 0.1542204    0.0004870   1.0242048   0.093   83.
```

- 0.1534541	- 0.0003562	1.0242118	0.062	61.
- 0.1515184	- 0.0001686	1.0245052	0.11	100.
- 0.1366084	0.0118421	1.0465399	0.063	62.
0.1544734	0.3064114	1.1210885	0.078	68.
- 0.1522411	0.0085609	1.0280789	0.078	66.
- 0.1632382	0.0127985	1.0368697	0.047	35.
0.1565720	0.3097188	1.1214391	0.063	63.
- 0.1501822	0.0059743	1.026748	0.047	37.
- 0.1584105	0.0128627	1.0336083	0.063	62.
- 0.1561604	- 0.0107162	1.0302793	0.063	57.
0.1529688	0.3035945	1.1219847	0.062	42.
- 0.4709729	- 0.3036141	1.3195554	0.031	33.
- 0.1573102	0.0034403	1.0253527	0.047	36.
- 0.1531858	0.0002349	1.0242267	0.063	50.
- 0.1514062	- 0.0060468	1.026394	0.062	60.
- 0.1525911	0.0082035	1.0277181	0.078	75.
0.1489847	- 0.311759	1.1229519	0.094	72.
0.1544937	0.3079830	1.1209622	0.062	62.
0.1617194	- 0.3030851	1.125209	0.031	35.
- 0.1541576	- 0.0017244	1.024343	0.047	52.
- 0.1681952	0.0035055	1.0350839	0.063	56.

- <u>ผลการแก้ปัญหาโดยใช้การค้นหาตาบู (TS)</u>

```
// กำหนดฟังก์ชันสำหรับทดสอบผลการรันโปรแกรม
=============================
                  runts.sci
=============================
function [xg,fg,tg,kg] = runts(fobj,Ntrial)
xlimit = [-1 1 10000; -1 1 10000];
opt = [1000 100 20 0 1 10];
x0 = [0 0];
Ne = 2;
Np = 20;
for u=1:Ntrial
   tic;
```

```
    [xmin,fmin,k] = ts0(fobj,x0,xlimit,opt);
// ts0.sci เหมือนกับ tsmain.sci ต่างกันตรงที่ไม่ให้มีการแสดงผลด้วยการวาดกราฟ
    t = toc();
    xg(u,1) = xmin(1);
    xg(u,2) = xmin(2);
    fg(u,1) = fmin;
    kg(u,1) = k;
    tg(u,1) = t;
    disp(['Trial # ' msprintf('%i',u) ' was successfully performed']);
end
endfunction
```

```
// ผลการรันโปรแกรม
--> exec('exam7_13.sce');
--> [xg,fg,tg,kg] = runts(fobj7_13,30);
--> [xg fg tg kg]
//      x1            x2           f(x1,x2)     time      Iter
ans  =
  - 0.1759176  - 0.0079008   1.0517648   0.093   436.
  - 0.2083208  - 0.0555056   1.3057609   0.063   285.
    0.2611261    0.3813381   1.8176959   0.031   121.
  - 0.3951395  - 0.3785379   1.7202291   0.016   121.
    0.1685169    0.3321332   1.1612482   0.047   207.
    0.1609161  - 0.2965297   1.1300482   0.093   396.
  - 0.1481148    0.0299030   1.071047    0.109   503.
  - 0.2139214  - 0.0097010   1.2070651   0.078   345.
    0.1585159    0.2967297   1.1284285   0.047   251.
  - 0.8465847    0.3203320   3.6363374   0.031   121.
    0.4589459    0.0011001   1.218239    0.14    586.
    0.1583158    0.3503350   1.2125938   0.032   121.
  - 0.3401340  - 0.0031003   2.3724138   0.031   121.
  - 0.1789179  - 0.0005001   1.0557749   0.078   351.
    0.1687169    0.3089309   1.1320245   0.109   468.
    0.1509151    0.2889289   1.1397547   0.078   297.
  - 0.1501150    0.0087009   1.0288082   0.141   569.
```

- 0.1559156	- 0.0027003	1.0247492	0.125	582.
- 0.1179118	0.0189019	1.1064529	0.094	416.
- 0.1625163	0.0263026	1.0629225	0.078	358.
- 0.1617162	0.0003000	1.0272314	0.125	554.
- 0.1575158	0.0003000	1.0248253	0.063	316.
0.0937094	- 0.2715272	1.348796	0.047	233.
- 0.1543154	0.0333033	1.0802283	0.062	265.
0.1571157	0.3223322	1.1319215	0.078	324.
0.1667167	- 0.3323332	1.1592727	0.078	312.
0.1213121	0.3335334	1.2067729	0.047	230.
- 0.1743174	- 0.0247025	1.075723	0.078	381.
- 0.1585159	- 0.0041004	1.0260877	0.078	378.
- 0.1531153	- 0.0005001	1.0242427	0.125	574.

จะได้ผลการเปรียบเทียบสมรรถนะดังตารางต่อไปนี้ และกราฟในรูปที่ 7.24 – 7.26

สมรรถนะในการค้นหาค่าต่ำสุด

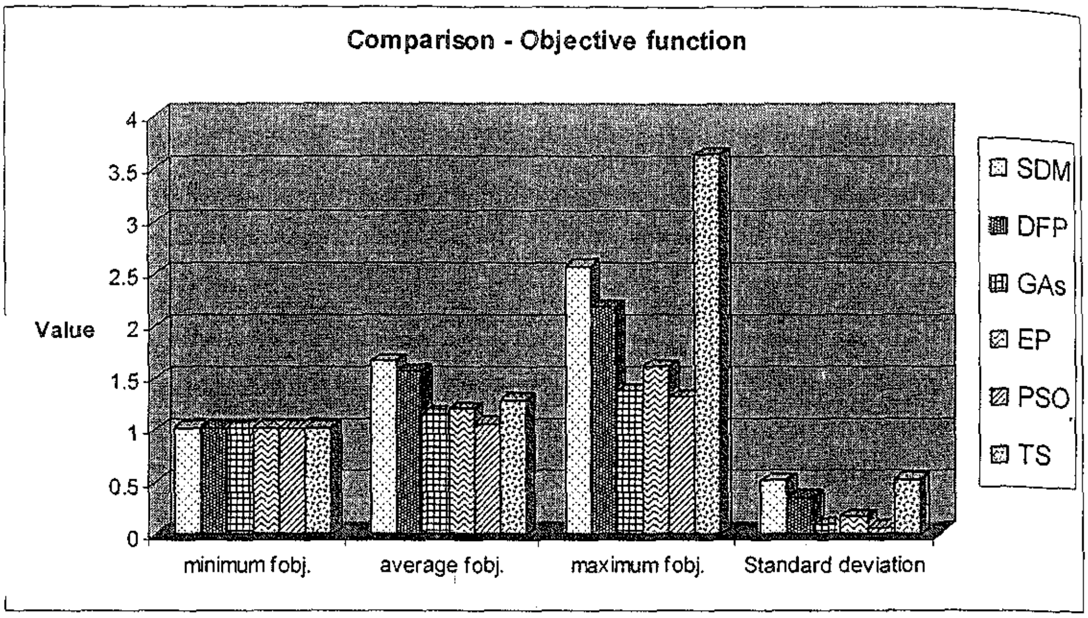
Method	minimum fobj.	average fobj.	maximum fobj.	Standard deviation
SDM	1.0242	1.6627	2.5721	0.5192
DFP	1.0242	1.5725	2.1852	0.3496
GAs	1.0252	1.1582	1.3812	0.0919
EP	1.0242	1.2048	1.6047	0.1661
PSO	1.0242	1.0632	1.3196	0.0647
TS	1.0242	1.2896	3.6363	0.5278

จำนวนรอบการค้นหาที่ใช้ * TS กำหนด Max Iter = 1000 รอบ

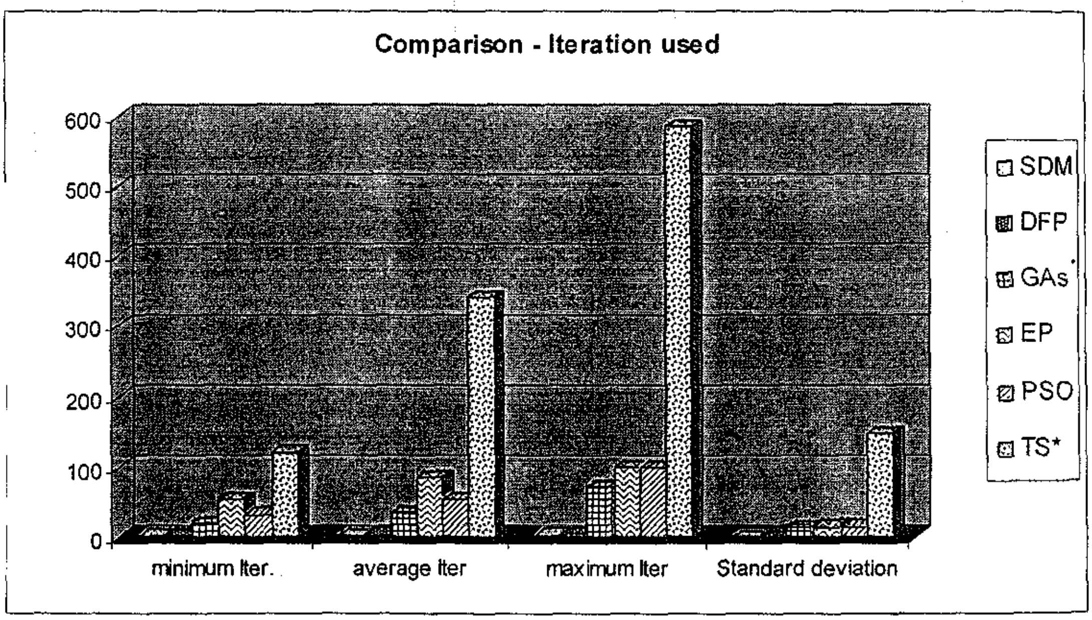
Method	minimum Iter.	average Iter	maximum Iter	Standard deviation
SDM	2	3.6	5	0.7
DFP	4	5.4	7	0.9
GAs	20	34.3	72	14.3
EP	55	85.2	100	15.9
PSO	33	54.6	100	16.1
TS*	121	340.7	586	148.3

ความเร็วในการค้นหา (วินาที)

Method	minimum time	average time	maximum time	Standard deviation
SDM	0.0000	0.0047	0.0160	0.0073
DFP	0.0000	0.0047	0.0160	0.0073
GAs	0.1560	0.2614	0.5150	0.0959
EP	0.1100	0.1708	0.2040	0.0317
PSO	0.0310	0.0610	0.1100	0.0186
TS	0.0160	0.0765	0.1410	0.0343



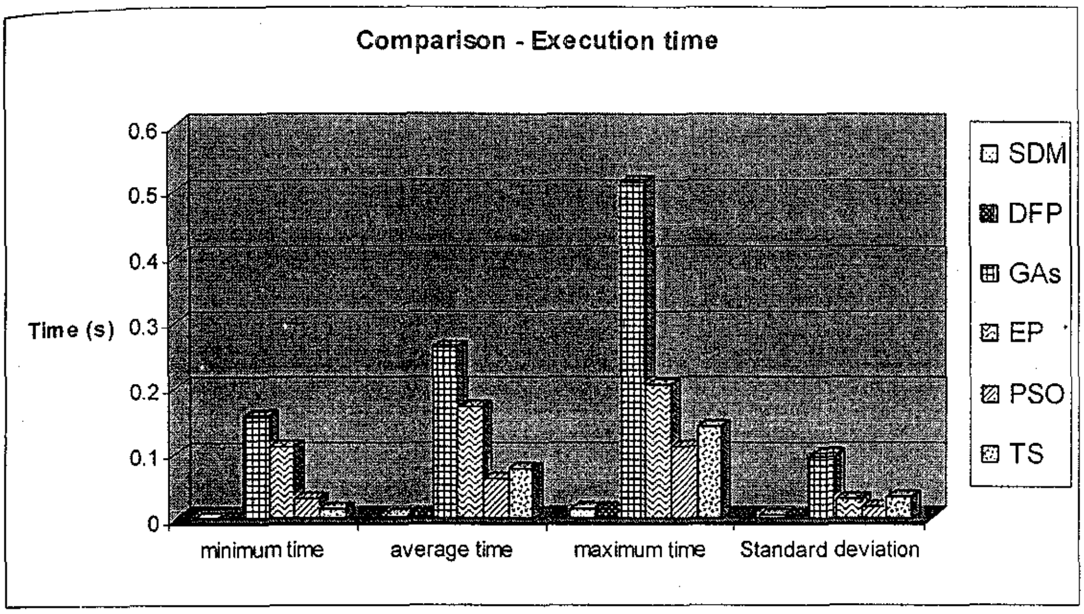
รูปที่ 7.24 การเปรียบเทียบค่าฟังก์ชันวัตถุประสงค์ของผลเฉลยที่ได้



รูปที่ 7.25 การเปรียบเทียบค่าจำนวนรอบของการค้นหาที่ใช้

จะเห็นได้ชัดเจนว่า เทคนิคชาญฉลาดให้ผลเฉลยที่ดีกว่า แต่มีปัญหาเรื่องความเร็วในการคำนวณ GA มีความคลาดเคลื่อนของผลเฉลยน้อยที่สุด ในขณะที่ EP ให้ความคลาดเคลื่อนที่ไม่ดีเท่ากับ GA แต่ใช้เวลาในการคำนวณโดยเฉลี่ยน้อยกว่า GA อย่างไรก็ตาม PSO ให้ผลเฉลยโดยเฉลี่ยดีที่สุดและใช้เวลาในการคำนวณน้อยที่สุดด้วยเมื่อเปรียบเทียบเฉพาะ GA EP และ PSO ถึงแม้ SDM และ BFGS จะใช้เวลาคำนวณที่รวดเร็วกว่า EP และ GA แต่เมื่อเปรียบเทียบกับ PSO แล้ว เร็วกว่าเพียงเล็กน้อยเท่านั้น สำหรับ TS ใช้เวลาในการค้นหาเร็วกว่า GA และ EP แต่ช้ากว่า PSO นอกจากนี้ ยังมีปัญหาเรื่องความแน่นอนของการค้นพบจุดคำตอบ เนื่องจาก TS ที่ใช้

แก้ปัญหานี้เป็น TS ที่ปรับปรุงสมรรถนะเพียงเล็กน้อยเท่านั้น จะเห็นได้ว่า การเลือกใช้เทคนิคการแก้ปัญหาค่าเหมาะที่สุดในทางปฏิบัติมีปัจจัยหลายประการเป็นตัวกำหนด ผู้ออกแบบระบบต้องพิจารณาถึงเงื่อนไขแวดล้อมต่าง ๆ ให้ดี จะทำให้การแก้ปัญหาค่าเหมาะที่สุดดำเนินไปได้อย่างมีประสิทธิภาพ



รูปที่ 7.26 การเปรียบเทียบเวลาของการค้นหาที่ใช้

7.7 สรุป

เมื่อปัญหาค่าเหมาะที่สุดที่มีความซับซ้อนและเป็นปัญหาแบบมัลติโมดอล (multimodal problem) ซึ่งมีจุดต่ำสุดหลายจุดในปริภูมิค้นหาและมีความไม่เชิงเส้นสูง ระเบียบวิธีกำหนดการทางคณิตศาสตร์ที่ได้นำเสนอมานั้น อาจจะไม่สามารถค้นหาจุดต่ำสุดโดยรวมของปัญหาได้ ดังแสดงในตัวอย่างที่ 7.6 ทำให้เทคนิคชาญฉลาดถูกนำมาใช้งานกันอย่างแพร่หลาย ถึงแม้เทคนิคเหล่านี้จะใช้เวลาค้นหาที่ยาวนานมาก ด้วยสมรรถนะของคอมพิวเตอร์ที่เพิ่มสูงขึ้นมากในปัจจุบัน ทำให้การแก้ปัญหาด้วยเทคนิคชาญฉลาด อยู่ในเกณฑ์ที่ยอมรับได้ในสำหรับการประยุกต์ใช้งานบางประเภท ถึงแม้เทคนิคชาญฉลาดมีหลากหลายชนิด ในช่วงเวลา 2 – 3 ปี ที่ผ่านมานี้ (ประมาณปี ค.ศ. 2003 – 2006) การหาค่าเหมาะที่สุดด้วยฝูงอนุภาค (PSO) ได้รับความสนใจจากนักวิจัยทั่วโลกเป็นอย่างมากเนื่องจากสมรรถนะในการค้นหาคำตอบที่ดีไม่ด้อยไปกว่า GA หรือ EP แต่ใช้เวลาในการคำนวณน้อยกว่ามาก สำหรับปัญหาที่กำหนดในบทนี้ ใช้เวลาน้อยกว่าระเบียบวิธีทางคณิตศาสตร์เพียงเล็กน้อยเท่านั้น ณ เวลาที่เรียบเรียงเอกสารนี้ อาจจะถือได้ว่า PSO เป็นเทคนิคชาญฉาดที่ได้รับการเลือกใช้เป็นลำดับแรก ในกรณีของการค้นหาตาบู (TS) เนื่องจากผู้เขียนเป็นส่วนหนึ่งในการพัฒนาการค้นหาตาบูเชิงปรับตัว (ATS) ซึ่งมีสมรรถนะที่ดีกว่า TS แบบดั้งเดิมมาก อย่างไรก็ตาม สำหรับการศึกษาในชั้นนี้ จะไม่นำ ATS มาพิจารณา

การแก้ปัญหาด้วยเทคนิคชาญฉลาดนั้น จะต้องใช้การปรับแต่งค่าพารามิเตอร์ให้เหมาะสมกับปัญหาแต่ละปัญหา หมายความว่า เมื่อปัญหาเปลี่ยนไปต้องปรับแต่งพารามิเตอร์ใหม่ทุกครั้งและต้องทำการทดสอบซ้ำหลายครั้ง (เป็นจำนวนมาก) เพื่อให้แน่ใจถึงความเที่ยงตรงในการให้ผลเฉลยจากชุดพารามิเตอร์ที่ปรับแต่ง ก่อนนำไปใช้งาน

7.8 แบบฝึกหัดท้ายบท

จงแก้ปัญหาต่อไปนี้โดยใช้ระเบียบวิธีชันที่สุด ระเบียบวิธีนิวตัน ระเบียบวิธีควอไซนิวตัน จีนเนติกอัลกอริทึม กำหนดการวิวัฒนาการ การหาค่าเหมาะที่สุดด้วยฝูงอนุภาค และการค้นหาตาบูแบบดั้งเดิม เพื่อเปรียบเทียบผลการค้นหาจุดต่ำสุด จำนวนรอบที่ใช้ และเวลาในการคำนวณ โดยให้ใช้ขนาดประชากรเท่ากับ 30 เท่ากัน ให้ทำซ้ำการทดสอบ 50 ครั้ง ใช้ความคลาดเคลื่อนไม่เกิน 1×10^{-4} หรือใช้จำนวนรอบในการคำนวณไม่เกิน 1000 รอบ ให้แสดงกราฟการลู่เข้า รวมกับการสร้างทางเดินของจุดคำตอบบนเส้นคอนทัวร์ของปัญหา สำหรับการค้นหาตาบู ให้ใช้ค่าความละเอียดของตัวแปรค้นหา 0.0001

1. $f(x_1,x_2)=3(1-x_1)^2 e^{-x_1^2-(x_2+1)^2}-10\left(\frac{x_1}{5}-x_1^3-x_2^5\right)e^{-x_1^2-x_2^2}-\frac{e^{-(x_1+1)^2-x_2^2}}{3}$

   โดยที่ $-3\le x_1,x_2\le 3$
2. $f(x_1,x_2)=x_1\,sin\,x_1+x_2\,sin\,5x_2$

   โดยที่ $0\le x_1\le 10,\ 4\le x_2\le 6$
3. $f(x_1,x_2)=100\left(x_2-x_1^2\right)^2+(1-x_1)^2$

   โดยที่ $-3\le x_1,x_2\le 3$
4. $f(x_1,x_2)=-\frac{1}{x_1^2+x_2^2+2}$

   โดยที่ $-3\le x_1,x_2\le 3$
5. $f(x_1,x_2)=x_1^2+2x_2^2-0.3\,cos\,3\pi x_1-0.4\,cos\,4\pi x_2+0.7$

   โดยที่ $-1\le x_1,x_2\le 1$
6. $f(x_1,x_2)=\left(x_1^2+x_2^2\right)^{1/4}\left[0.1+sin^2 50\left(x_1^2+x_2^2\right)^{1/10}\right]$

   โดยที่ $-\frac{1}{2}\le x_1,x_2\le\frac{1}{2}$
7. $f(x_1,x_2,x_3,x_4)=x_1^4+x_2^2+(x_3-0.2)^2+3(x_4-0.2)^4$

   โดยที่ $-1\le x_1,x_2,x_3,x_4\le 1$

บทที่ 8 การประยุกต์เทคนิคการหาค่าเหมาะที่สุดสำหรับเครื่องจักรกลไฟฟ้า (Applied Optimization Techniques for Electrical Machines)

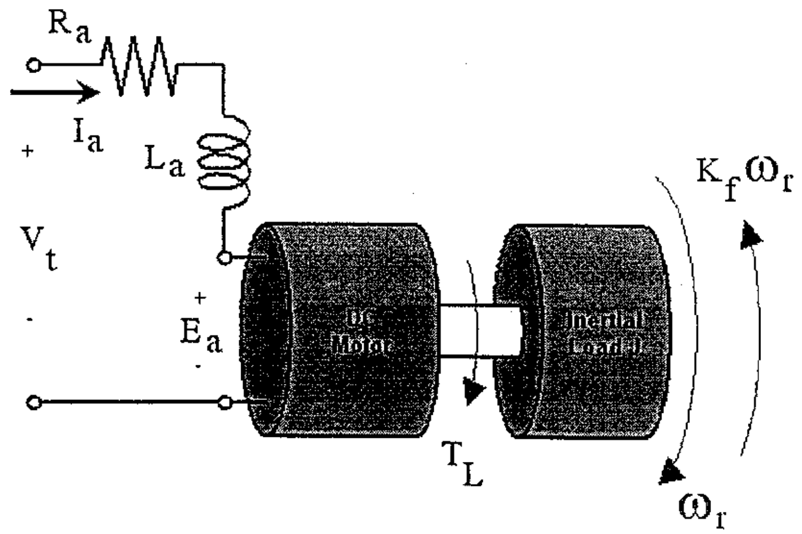
"Everything is vague to a degree you do not realize till you have tried to make it precise"

8.1 เกริ่นนำ

บทนี้เป็นการประยุกต์ใช้เทคนิคการหาค่าเหมาะที่สุดเพื่อแก้ปัญหาทางด้านวิศวกรรมไฟฟ้าทางด้านเครื่องจักรกลไฟฟ้า รายละเอียดของการคำนวณทีละขั้นจะไม่ถูกนำเสนออีกต่อไป ในบทนี้จะนำเสนอรูปแบบการสร้างปัญหา และการประยุกต์หลักการแก้ปัญหาค่าเหมาะที่สุดเข้ามาใช้งาน ซึ่งจะสิ้นสุดที่การสร้างฟังก์ชันวัตถุประสงค์และการเขียนเงื่อนไขบังคับของปัญหา โดยอาศัยการโปรแกรมด้วยคอมพิวเตอร์ผ่านซอฟแวร์ SCILAB ด้วยเทคนิคการแก้ปัญหาที่ได้นำเสนอไว้ อย่างไรก็ตาม เนื่องด้วยความซับซ้อนของปัญหา และความยุ่งยากในการคำนวณค่าเกรเดียนต์ของฟังก์ชันวัตถุประสงค์ ในบทนี้จะเน้นไปที่การแก้ปัญหาด้วยเทคนิคการคำนวณชาญฉลาดเท่านั้น ดังแสดงในรายละเอียดต่อไปนี้

8.2 การปรับแต่งพารามิเตอร์เหมาะที่สุดสำหรับมอเตอร์ไฟฟ้าดีซี

พิจารณาการขับเคลื่อนมอเตอร์ไฟฟ้าดีซีดังรูปที่ 8.1



รูปที่ 8.1 การควบคุมมอเตอร์ไฟฟ้าดีซีด้วยแรงดันอาร์เมเจอร์

เมื่อให้การกระตุ้นจากขดลวดสนามมีค่าคงที่เพื่อให้ง่ายต่อการพิจารณา จะได้ความสัมพันธ์ของตัวแปรสถานะและพารามิเตอร์ของมอเตอร์ตามสมการต่อไปนี้ [1]

$$v_t(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + E_a(t) \qquad (8.1)$$

$$T_a - T_L = J\frac{d\omega_r(t)}{dt} + K_f \omega_r(t) \qquad (8.2)$$

โดยที่

$$T_a = K_T i_a(t) \tag{8.3}$$

$$E_a(t) = K_E \omega_r(t) \tag{8.4}$$

โดยใช้เทคนิคการคำนวณค่าอนุพันธ์เชิงตัวเลขด้วยระเบียบวิธีผลต่างย้อนหลัง (Backward difference) [2] จะได้ว่า

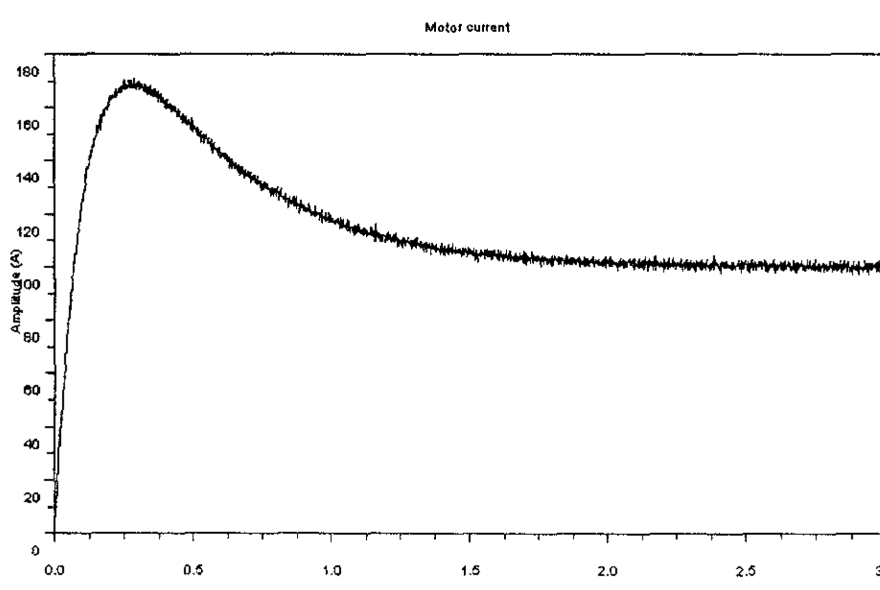
$$v_t(t) = R_a i_a(t) + \frac{L_a}{\Delta t}\{i_a(t) - i_a(t-\Delta t)\} + K_E \omega_r(t)$$

$$K_T i_a(t) - T_L = \frac{J}{\Delta t}\{\omega_r(t) - \omega_r(t-\Delta t)\} + K_f \omega_r(t)$$

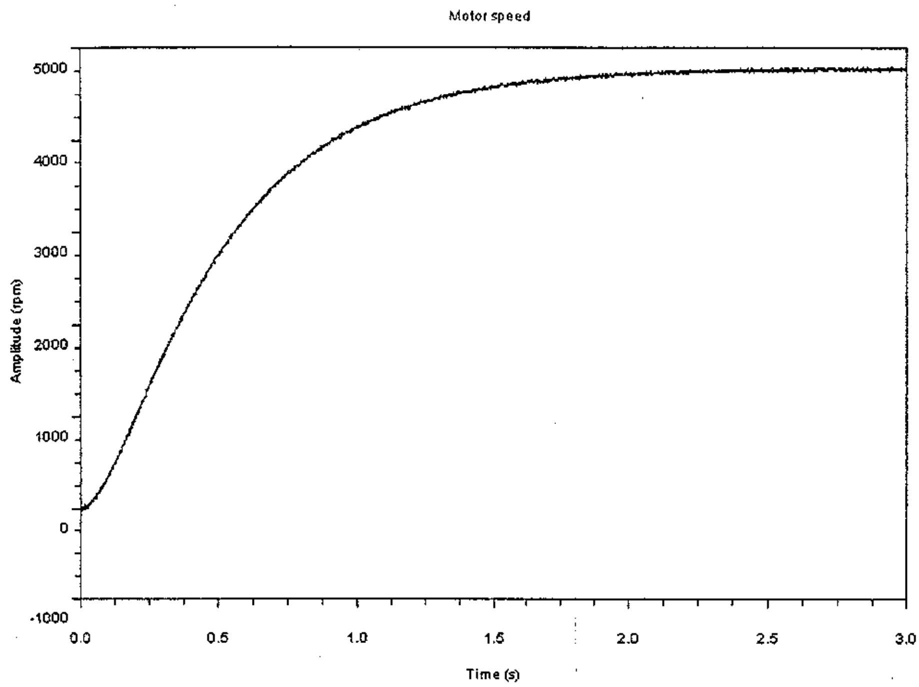
นั่นคือ

$$\begin{bmatrix} \frac{L_a}{\Delta t}+R_a & K_E \\ -K_T & \frac{J}{\Delta t}+K_f \end{bmatrix}\begin{bmatrix} i_a(t) \\ \omega_r(t) \end{bmatrix} = \begin{bmatrix} v_t(t)+\frac{L_a}{\Delta t}i_a(t-\Delta t) \\ -T_L(t)+\frac{J}{\Delta t}\omega_r(t-\Delta t) \end{bmatrix} \tag{8.5}$$

ตัวอย่างที่ 8.1 กำหนดให้ผลทดสอบการวัดกระแสและความเร็วรอบของมอเตอร์ไฟฟ้าดีซีที่สภาวะไร้โหลดตัวหนึ่งมีค่าดังกราฟในรูปที่ 8.2 และ 8.3 จงใช้เทคนิคการหาค่าเหมาะที่สุดเพื่อปรับแต่งค่าพารามิเตอร์ของมอเตอร์ไฟฟ้าดีซีให้มีความคลาดเคลื่อนน้อยที่สุดเมื่อเทียบกับผลการวัด



รูปที่ 8.2 กระแสของมอเตอร์ที่ได้จากการทดสอบ จำนวน 3001 จุดข้อมูล



รูปที่ 8.3 ความเร็วรอบของมอเตอร์ที่ได้จากการทดสอบ จำนวน 3001 จุดข้อมูล

วิธีทำ มอเตอร์ที่ทดสอบมีพิกัดแรงดันดีซีอยู่ที่ 100 V การทดสอบดำเนินการที่ค่าแรงดันพิกัดจากสมการที่ 8.5 จะพบว่า มีพารามิเตอร์ที่ไม่ทราบค่าและต้องปรับแต่งดังนี้ คือ R_a, L_a, K_T, K_E, K_f และ J อย่างไรก็ตาม ขอบเขตของตัวแปรดังกล่าวสามารถประเมินได้โดยการทดสอบแบบดั้งเดิม ตัวอย่างเช่น กรณีความต้านทานสามารถประเมินได้จากผลการทดสอบวัดค่าความต้านทานของขดลวดโดยตรง กรณีความเหนี่ยวนำสามารถประเมินได้จากค่ายอดของกระแสมอเตอร์ขณะเริ่มเดินเครื่อง หรือกรณีของค่าโมเมนต์ความเฉื่อยของมอเตอร์สามารถประเมินได้จากผลทดสอบการหน่วง (retardation test) [3] เป็นต้น ดังนั้น กำหนดช่วงการค้นหาของพารามิเตอร์ทั้ง 6 ตัว ดังนี้

$R_a \in [0.1,1.0]\ \Omega$
$L_a \in [10.0,100.0]$ mH
$K_T \in [0.05,0.5]$ N·m/A
$K_E \in [0.05,0.5]$ V·s/rad
$K_f \in [0.01,0.1]$ N·m·s
$J \in [0.01,0.1]$ kg·m^2·s^{-2}

ฟังก์ชันวัตถุประสงค์ใช้การประเมินความคลาดเคลื่อนของค่าที่วัดได้กับผลการจำลอง โดยพิจารณาที่จุดทดสอบทั้ง 3001 จุด ด้วยผลรวมของค่าความคลาดเคลื่อนกำลังสอง (sum square error: SSE) ฟังก์ชันวัตถุประสงค์นี้ต้องการตัวแปรอินพุตจำนวน 6 ตัว และให้ค่าเอาต์พุตเป็นค่า SSE กลับคืนไป ภายในตัวฟังก์ชันจะดำเนินการจำลองผลกระแสมอเตอร์และความเร็วรอบจากการปรับตั้งค่าพารามิเตอร์ดังกล่าว จากนั้นนำค่ามาคำนวณ SSE ดังสมการที่ 8.6 โดยที่ข้อมูลผล

การทดสอบกระแสและความเร็วรอบของมอเตอร์ถูกบันทึกไว้ในไฟล์ชื่อ “dcmot01.dat” ตัวแปรชื่อ Ia1 และ RPM10 ซึ่งเป็นตัวแปรขนาด 3001×1 ดังรายละเอียดต่อไปนี้

$$SSE = \alpha \sum_{k=1}^{3001} \left| I_a^{sim}(k) - I_a^{test}(k) \right|^2 + \beta \sum_{k=1}^{3001} \left| \omega_r^{sim}(k) - \omega_r^{test}(k) \right|^2 \qquad (8.6)$$

เมื่อ α และ β แทนตัวประกอบการถ่วงน้ำหนักมีค่าในช่วง 0.0 – 1.0 โดยที่ $\alpha + \beta = 1.0$

ปัญหาในสมการที่ 8.6 เรียกว่า ปัญหาค่าเหมาะที่สุดหลายวัตถุประสงค์ (multi-objective optimization problem) [4-6] ประกอบด้วยค่าความคลาดเคลื่อนระหว่างค่าที่วัดได้กับค่าที่ได้จากการจำลองผลของสัญญาณกระแส และสัญญาณความเร็วรอบ เนื่องจากผลการทดสอบต้องการปรับแต่งพารามิเตอร์เพื่อให้สอดคล้องกับสัญญาณทั้งสองมากที่สุด ดังนั้น จำเป็นต้องมีการคิดน้ำหนักหรือความสำคัญของสัญญาณแต่ละตัวโดยใช้ตัวปรับคูณ α และ β ซึ่งค่าทั้งสองนี้ไม่จำเป็นต้องเท่ากัน ทดลองปรับแต่งพารามิเตอร์โดยกำหนดให้ $\alpha = 1.0$ และ $\beta = 0.0$ นั่นคือ การเน้นการปรับแต่งพารามิเตอร์ให้สอดคล้องกับรูปคลื่นกระแสเพียงอย่างเดียว พิจารณาได้จากเขียนฟังก์ชันวัตถุประสงค์โดยใช้หลักการของฟังก์ชันปรับโทษต่อไปนี้

<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 8.1</u>

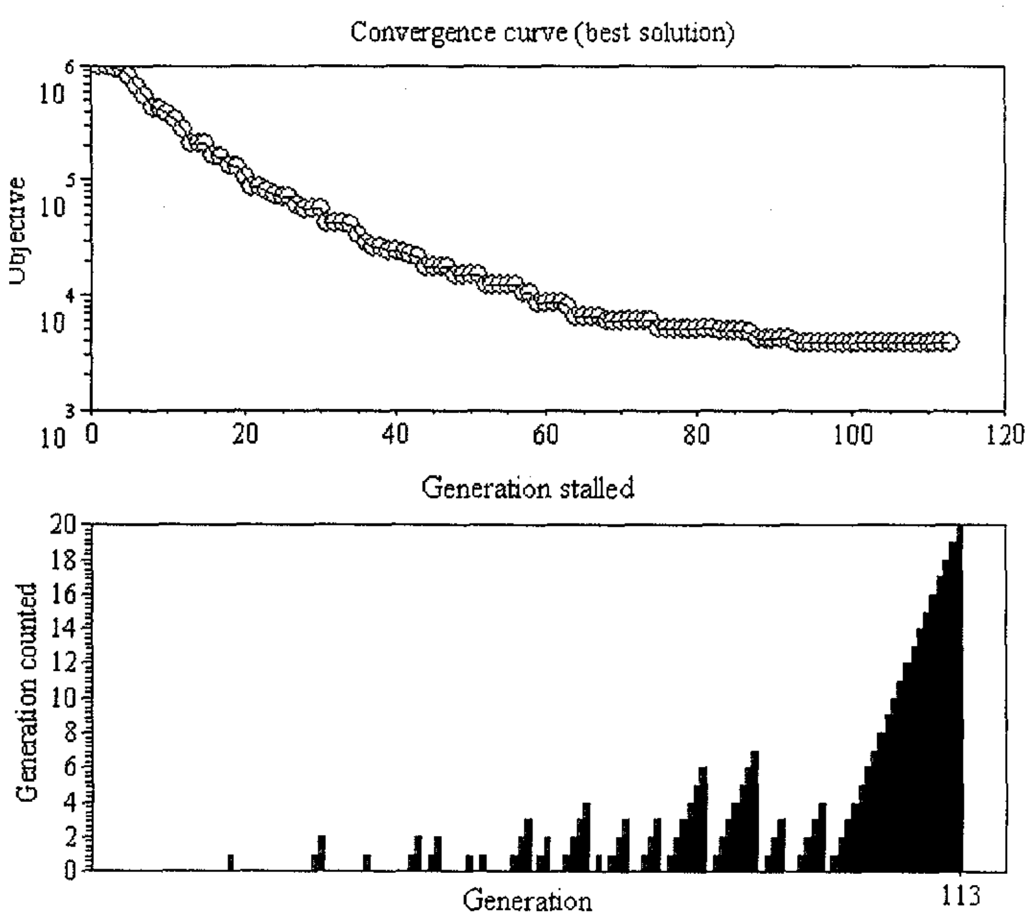
// SCILAB source program

```
// กำหนดฟังก์ชันวัตถุประสงค์                    fdcmot.sci
========================
            fdcmot.sci
========================
function f = fdcmot(x)
alpha1 = 1.0; beta1 = 0;
TSTEP = 1e-3;
TSPAN = [0 3];
t = TSPAN(1):TSTEP:TSPAN(2);
x0 = [0;0];
FSTEP = 1/TSTEP;
Ia(1) = x0(1);Wr(1) = x0(2);
RPM(1) = Wr(1)*60/2/%pi;
Vs = 100;
rho = 1000;
xlimit = [0.1 1;0.01 0.1;0.05 0.5;0.05 0.5;0.01 0.1;0.01 0.1];
Ra = x(1);
La = x(2);
```

```
KT = x(3);
KE = x(4);
J = x(5);
KF = x(6);
A = [Ra+La/TSTEP   KE;
   -KT         KF+J/TSTEP];
INV_A = inv(A);
TL = 0;
U = eye(2);
load('C:\SCILAB\Work\Scilab programs\Motor Applications\dcmot01.dat');
f = alpha1*(Ia1(1)-Ia(1))^2 + beta1*(RPM10(1)-RPM(1))^2;
for k=2:length(t)
   if t(k)>3 then
      TL = 5;
   end
   B = [Vs + La/TSTEP*Ia(k-1);
      J/TSTEP*Wr(k-1)-TL];
   x1 = INV_A*B;
   Ia(k) = x1(1,1);Wr(k) = x1(2,1); RPM(k) = Wr(k)*60/2/%pi;
   f = f + alpha1*(Ia1(k)-Ia(k))^2 + beta1*(RPM10(k)-RPM(k))^2;
end
for i=1:6
   P(i) = x(i)-xlimit(i,1);
   P(i+6) = xlimit(i,2)-x(i);
end
SSP=0;
for i=1:length(P)
   if P(i)>=0 then
      P(i)=0;
   end
   SSP=SSP+P(i)^2;
end
f = f + rho*SSP;
endfunction
```

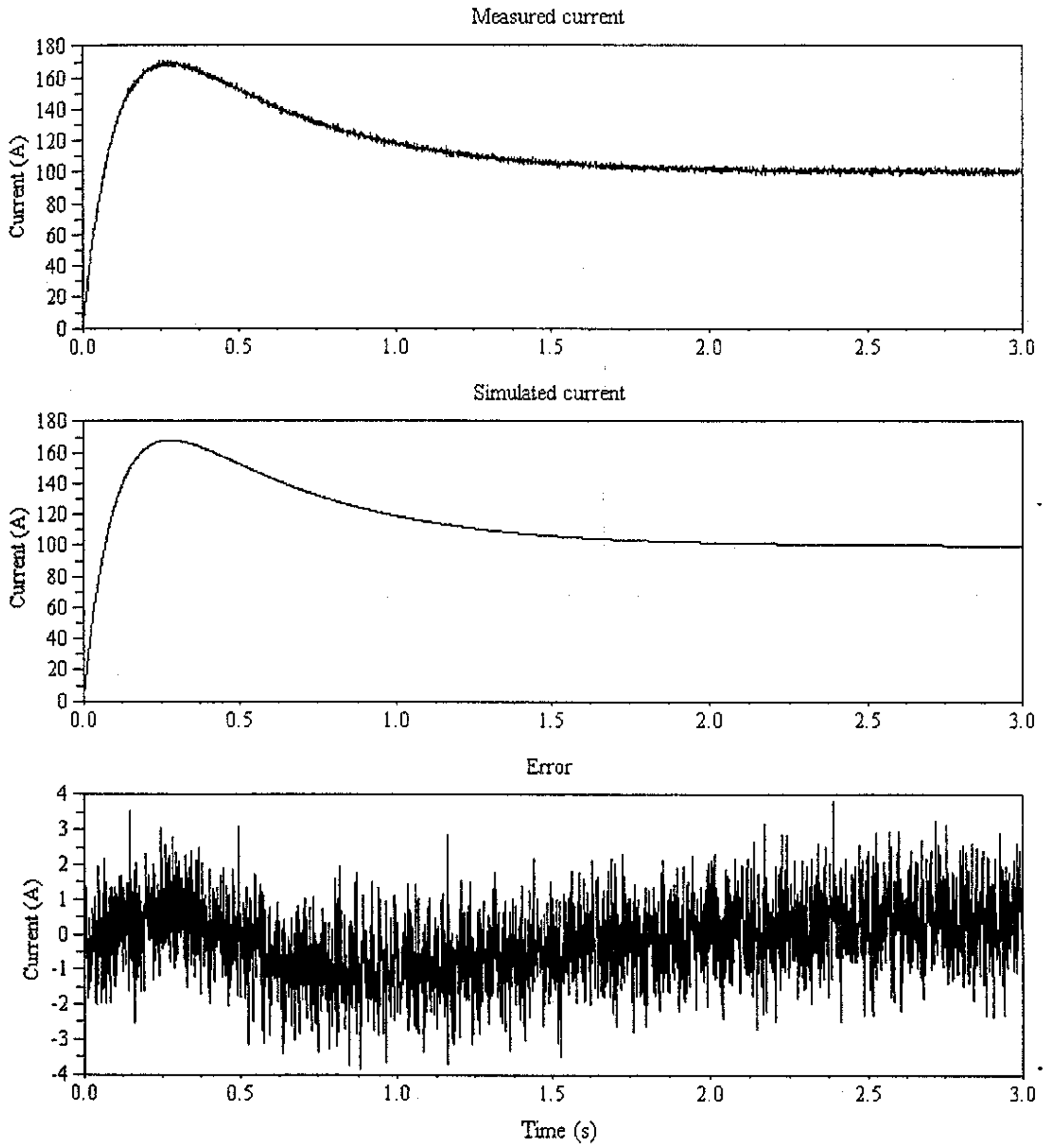
แก้ปัญหาค่าเหมาะที่สุดนี้ โดยใช้เทคนิคกำหนดการวิวัฒนาการดังนี้

```
// ผลการรันโปรแกรม
--> exec('fdcmot.sci');                    ประกาศฟังก์ชันวัตถุประสงค์
--> exec('ep_initial.sci');                ประกาศฟังก์ชันเพื่อการคำนวณ
--> Ne = 6; Np = 10; xbeta = 0.08;
--> xlimit = [0.1 1;0.01 0.1;0.05 0.5;0.05 0.5;0.01 0.1;0.01 0.1];
--> opt = [500 20 0 4];
--> [xg,fg] = epmain(fdcmot,Ne,Np,xlimit,xbeta,opt);
-->xg
 xg  =
   0.5057995    0.0497272    0.1237064    0.2540175    0.0662938    0.0628262
-->fg
 fg  =
   3887.5711
```

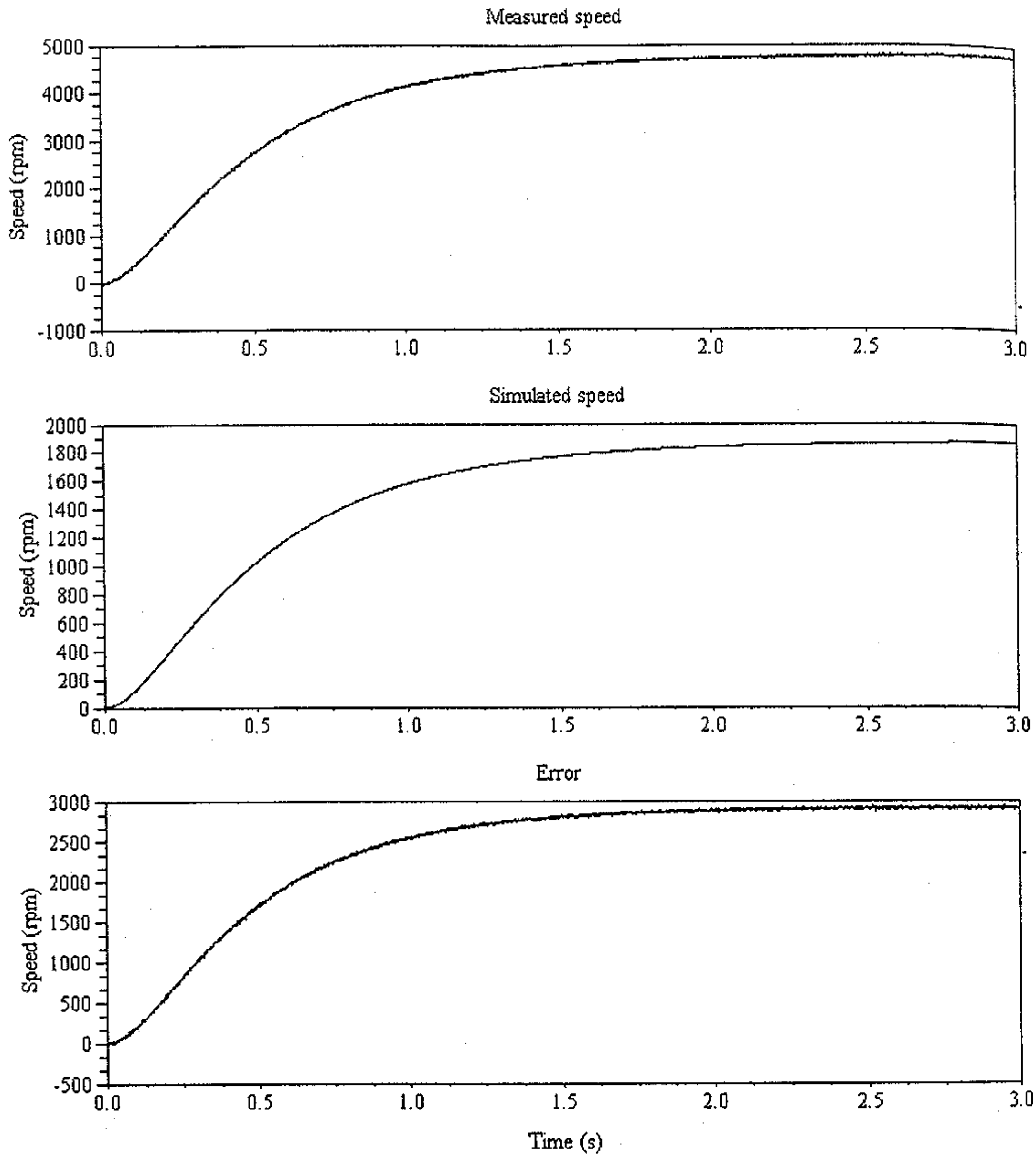


รูปที่ 8.4 อัตราการลู่เข้าของกำหนดการวิวัฒนาการ กรณี $\alpha = 1.0$ และ $\beta = 0.0$

เนื่องจากการกำหนดตัวประกอบการถ่วงน้ำหนักในกรณีนี้ เน้นไปที่การปรับแต่งจากค่ากระแสของมอเตอร์เท่านั้น จะเห็นได้ว่า กราฟในรูปที่ 8.5 และ 8.6 เปรียบเทียบระหว่างกระแสและความเร็วที่วัดได้และที่จำลองขึ้นมีความคลาดเคลื่อนต่างกันอย่างเห็นได้ชัดเจน



รูปที่ 8.5 กราฟการเปรียบเทียบกระแสของมอเตอร์ที่วัดได้กับผลการจำลอง

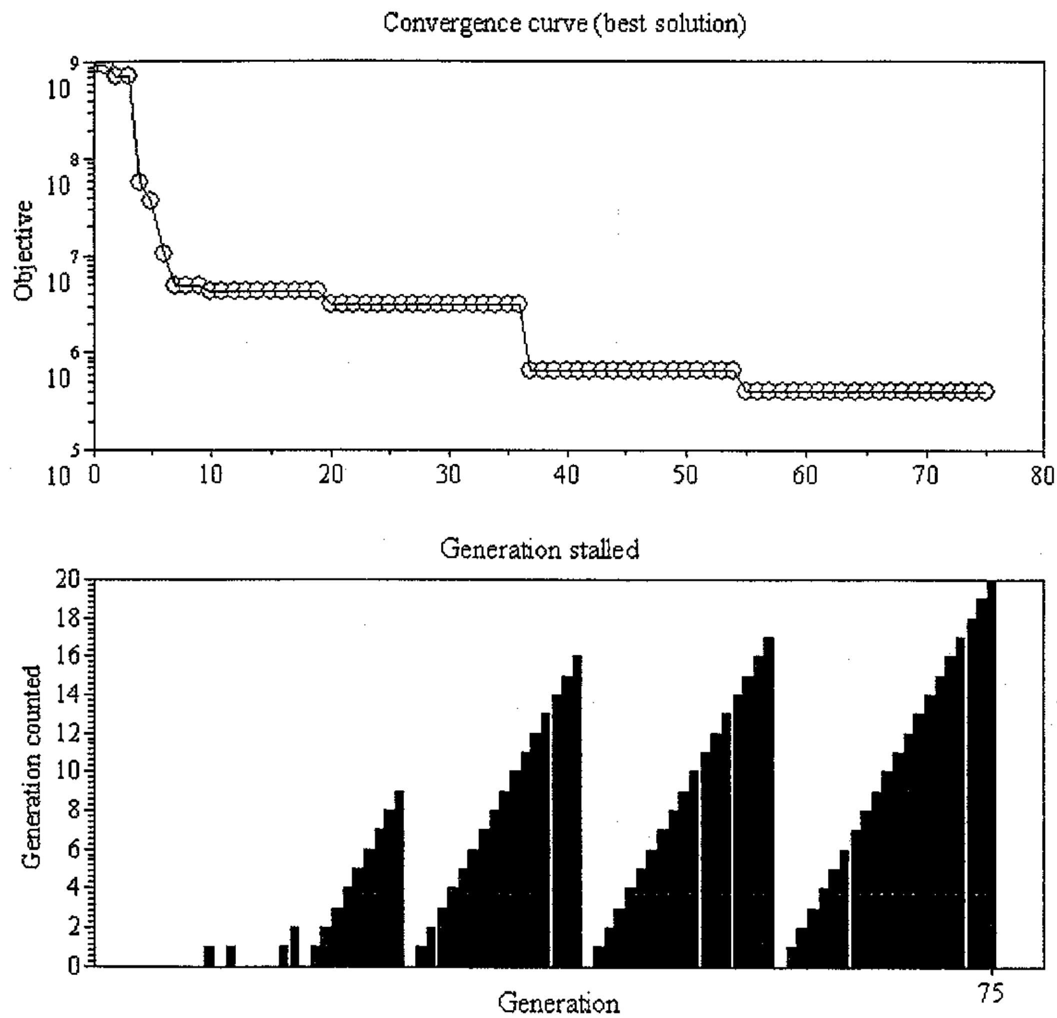


รูปที่ 8.6 กราฟการเปรียบเทียบความเร็วรอบของมอเตอร์ที่วัดได้กับผลการจำลอง

ปรับเปลี่ยนความสำคัญของการปรับแต่งพารามิเตอร์ใหม่ โดยเน้นไปที่ค่าความเร็วรอบ ดังนั้น กรณีนี้ กำหนดให้ $\alpha = 0.0$ และ $\beta = 1.0$ รันโปรแกรมใหม่อีกครั้ง ได้ผลดังต่อไปนี้

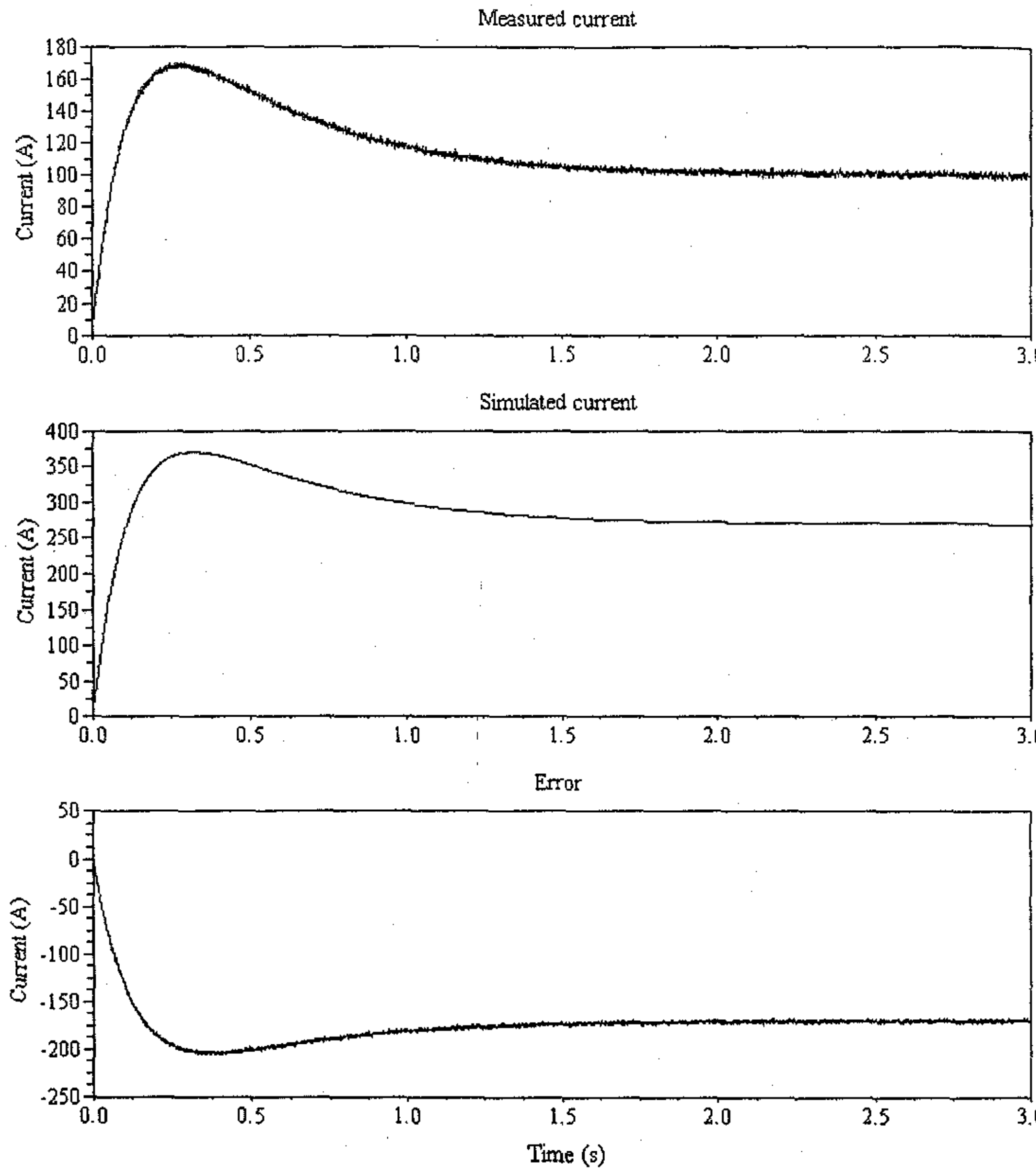
```
// ผลการรันโปรแกรม
--> exec('fdcmot.sci');                    ประกาศฟังก์ชันวัตถุประสงค์
--> exec('ep_initial.sci');                ประกาศฟังก์ชันเพื่อการคำนวณ
--> Ne = 6; Np = 10; xbeta = 0.2;
--> xlimit = [0.1 1;0.01 0.1;0.05 0.5;0.05 0.5;0.01 0.1;0.01 0.1];
--> opt = [500 20 0 4];
--> [xg,fg] = epmain(fdcmot,Ne,Np,xlimit,xbeta,opt);
```

```
-->xg
 xg  =
   0.2329601    0.0248975    0.0905771    0.0746078    0.0374219    0.0489028
-->fg
 fg  =
   397742.85
```

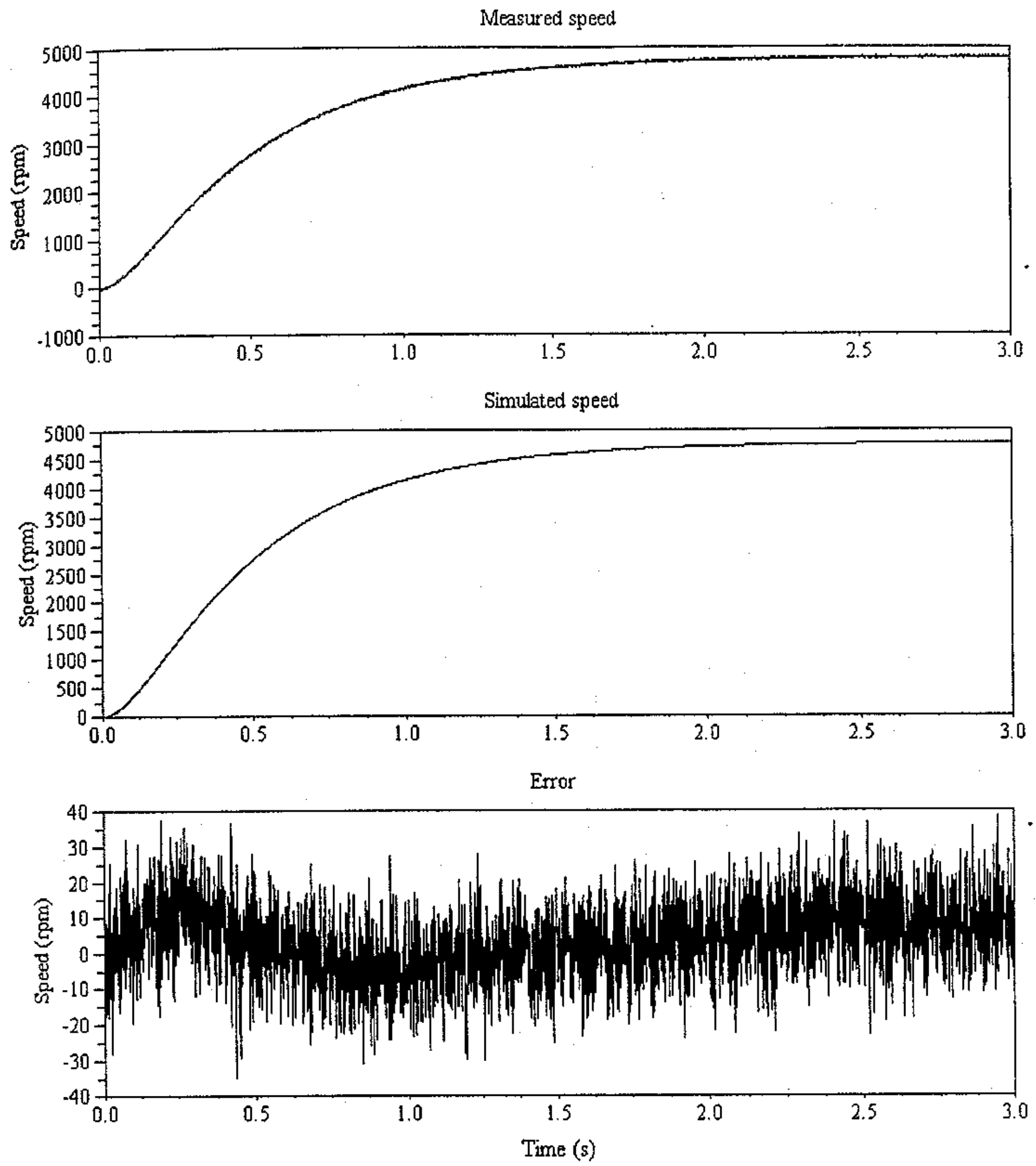


รูปที่ 8.7 อัตราการลู่เข้าของกำหนดการวิวัฒนาการ กรณี $\alpha = 0.0$ และ $\beta = 1.0$

เนื่องจากการกำหนดตัวประกอบการถ่วงน้ำหนักในกรณีนี้ เน้นไปที่การปรับแต่งจากค่าความเร็วรอบของมอเตอร์เท่านั้น จะเห็นได้ว่า กราฟในรูปที่ 8.5 และ 8.6 เปรียบเทียบระหว่างกระแสและความเร็วที่วัดได้และที่จำลองขึ้นมีความคลาดเคลื่อนต่างกันอย่างเห็นได้ชัดเจน



รูปที่ 8.8 กราฟการเปรียบเทียบกระแสของมอเตอร์ที่วัดได้กับผลการจำลอง

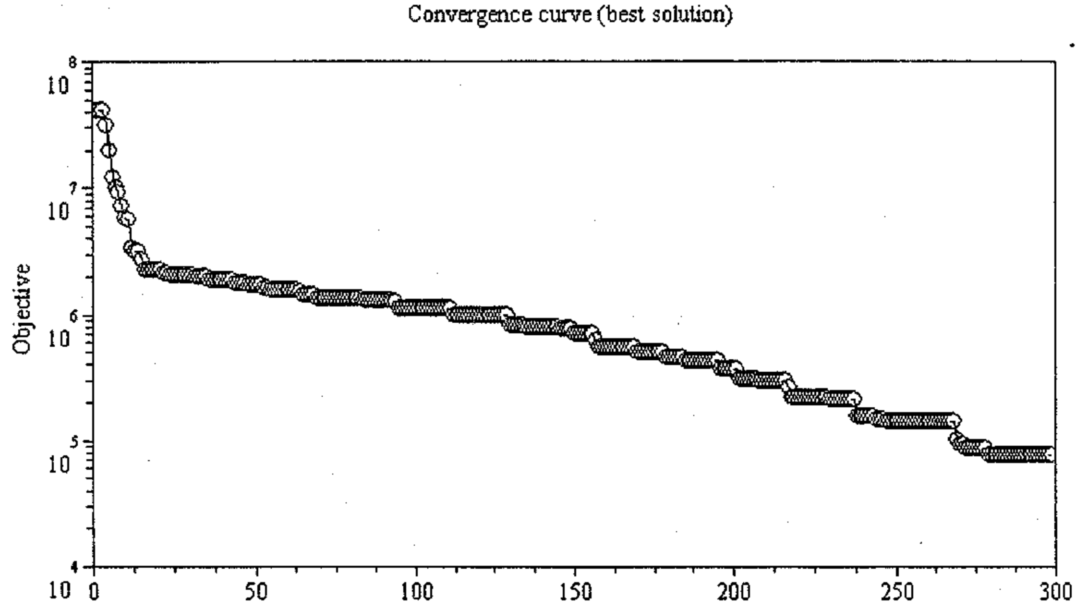


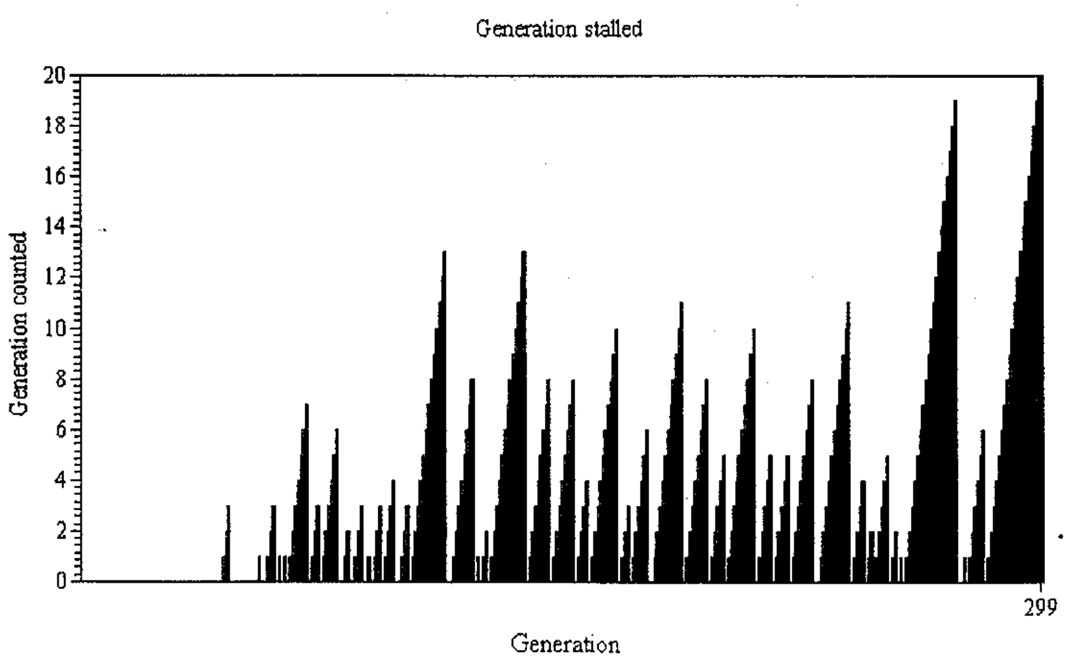
รูปที่ 8.9 กราฟการเปรียบเทียบความเร็วรอบของมอเตอร์ที่วัดได้กับผลการจำลอง

ปรับเปลี่ยนความสำคัญของการปรับแต่งพารามิเตอร์ใหม่ โดยเน้นความสำคัญทั้งกระแสมอเตอร์และความเร็วรอบ ในกรณีนี้ ค่าตัวประกอบการถ่วงน้ำหนักต้องมีค่าเหมาะสม ซึ่งจะมีค่าแตกต่างกันออกไปในที่นี้ ให้ α =0.95 และ β = 0.05 รันโปรแกรมใหม่อีกครั้ง ได้ผลดังต่อไปนี้

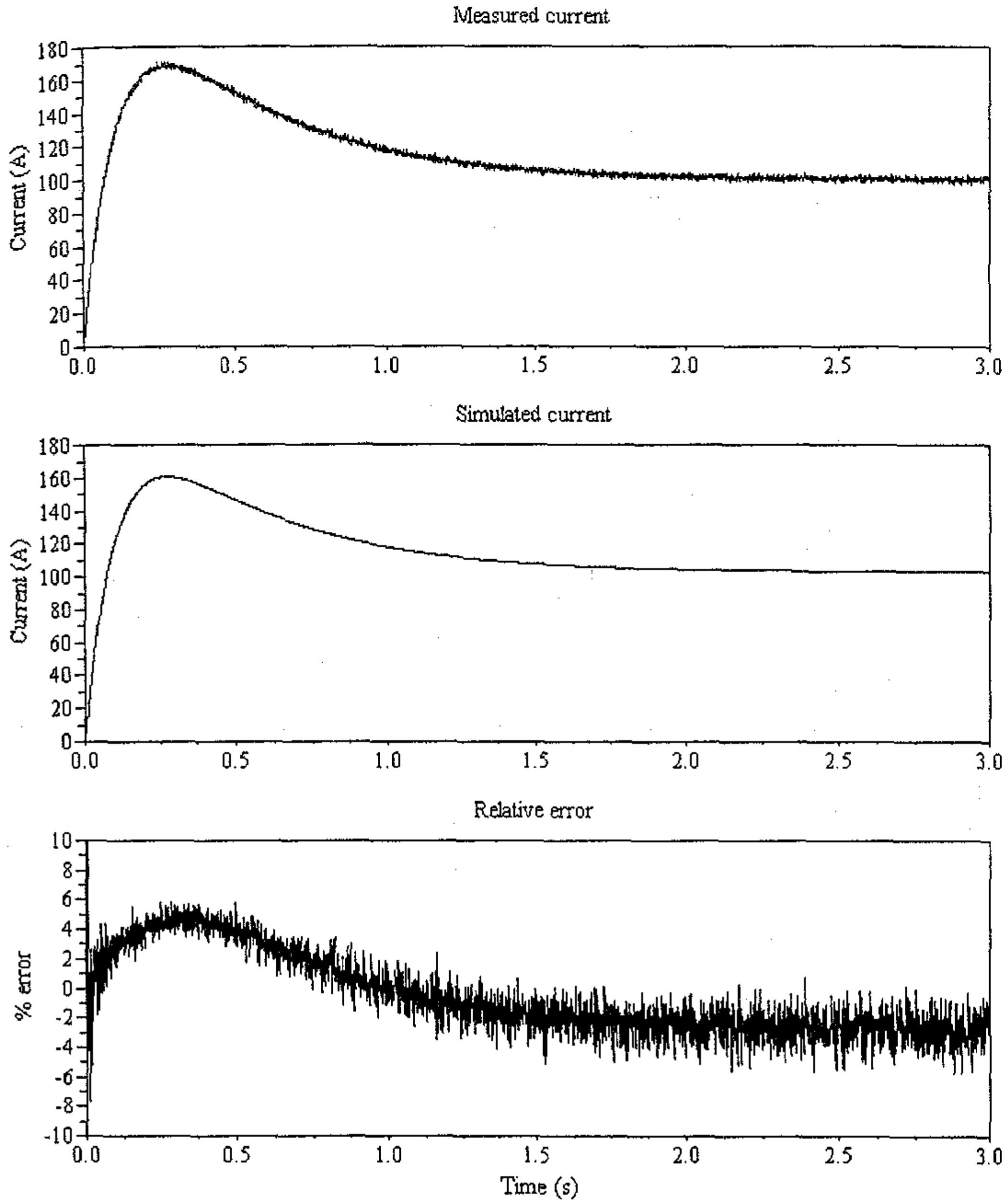
```
// ผลการรันโปรแกรม
--> exec('fdcmot.sci');                    ประกาศฟังก์ชันวัตถุประสงค์
--> exec('ep_initial.sci');                ประกาศฟังก์ชันเพื่อการคำนวณ
--> Ne = 6; Np = 10; xbeta = 0.1;
--> xlimit = [0.1 1;0.01 0.1;0.05 0.5;0.05 0.5;0.01 0.1;0.01 0.1];
--> opt = [500 20 0 4];
```

```
--> [xg,fg] = epmain(fdcmot,Ne,Np,xlimit,xbeta,opt);
-->xg
 xg  =
   0.5329245   0.0501212   0.3439865   0.0901721   0.0642641   0.0704620
-->fg
 fg  =
   78228.795
```

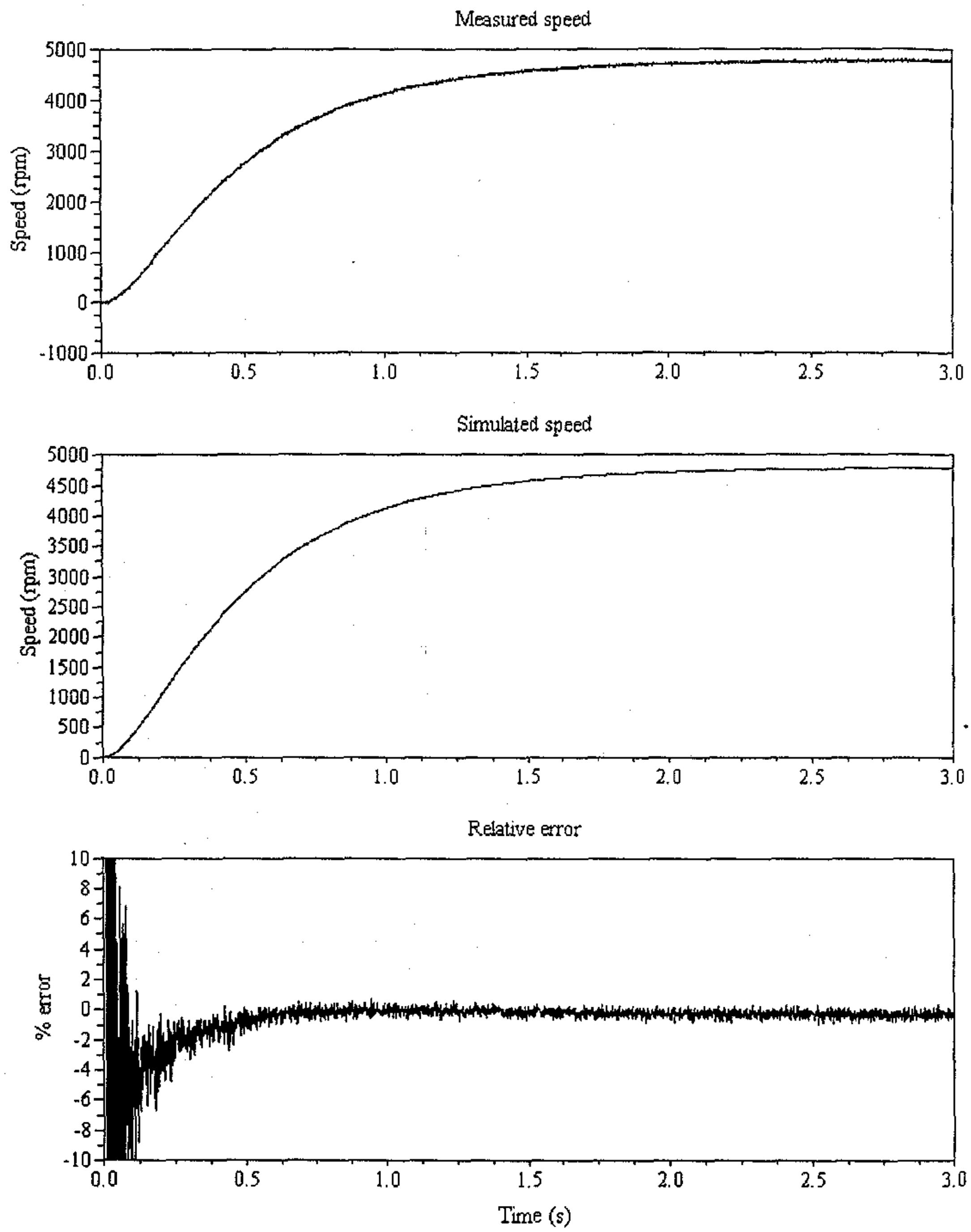




รูปที่ 8.10 อัตราการลู่เข้าของกำหนดการวิวัฒนาการ กรณี $\alpha = 0.95$ และ $\beta = 0.05$



รูปที่ 8.11 กราฟการเปรียบเทียบกระแสของมอเตอร์ที่วัดได้กับผลการจำลอง



รูปที่ 8.12 กราฟการเปรียบเทียบความเร็วรอบของมอเตอร์ที่วัดได้กับผลการจำลอง

8.3 การออกแบบตัวควบคุมสำหรับการควบคุมแรงดันอาร์เมเจอร์ของมอเตอร์ไฟฟ้าดีซี

เมื่อทราบพารามิเตอร์ของมอเตอร์ไฟฟ้าดีซีตามที่ได้กล่าวมา ขั้นตอนต่อไปจะกล่าวถึงการควบคุมการขับเคลื่อนมอเตอร์ไฟฟ้าดีซีดังรูปที่ 8.13 การควบคุมนี้อาศัยหลักการปรับค่าแรงดันที่จ่ายให้กับมอเตอร์ที่ขดลวดอาร์เมเจอร์ เรียกการควบคุมแบบนี้ว่า การควบคุมแรงดันอาร์เมเจอร์ [7] เพื่อให้สามารถควบคุมความเร็วรอบของมอเตอร์ดีซีนี้ได้ ตลอดย่านการทำงานที่มีการเปลี่ยนแปลงค่าแรงบิดของโหลด ตัวควบคุมพีไอดี (PID controller) [8] ถูกนำมาใช้งาน อย่างไรก็ตาม ถึงแม้

ตัวควบคุมดังกล่าวจะช่วยควบคุมความเร็วรอบของมอเตอร์ได้ แต่การตอบสนองของมอเตอร์ดีซีต่อการเปลี่ยนแปลงโหลดแรงบิดต้องได้รับการประเมินในรูปของเวลาในการตอบสนอง ความคลาดเคลื่อนในสภาวะคงตัว เพื่อให้ระบบตอบสนองได้รวดเร็วที่สุด ให้เกิดการแกว่งของความเร็วรอบน้อยที่สุด ดังนั้นปัญหาที่กล่าวถึงในหัวข้อนี้จะเป็นการปรับแต่งพารามิเตอร์ของตัวควบคุมพีไอดีเพื่อให้ได้การตอบสนองที่ดีที่สุด

จากสมการที่ 8.1 และ 8.2 โดยประยุกต์ผลการแปลงลาปลาซ (Laplace transform) [9] เพื่อแปลงระบบสมการเชิงอนุพันธ์ให้เป็นสมการพีชคณิตในโดเมนเอส (s-domain) ดังต่อไปนี้

$$\mathcal{L}\left\{v_t(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + K_E \omega_r(t)\right\}$$

$$V_t(s) = \left[sL_a + R_a\right] I_a(s) + K_E \Omega_r(s) \tag{8.7}$$

$$\mathcal{L}\left\{K_T i_a(t) - T_L = J\frac{d\omega_r(t)}{dt} + K_f \omega_r(t)\right\}$$

$$K_T I_a(s) - T_L(s) = sJ\Omega_r(s) + K_f \Omega_r(s)$$

$$I_a(s) = \frac{1}{K_T} T_L(s) + \frac{1}{K_T}\left[Js + K_f\right]\Omega_r(s) \tag{8.8}$$

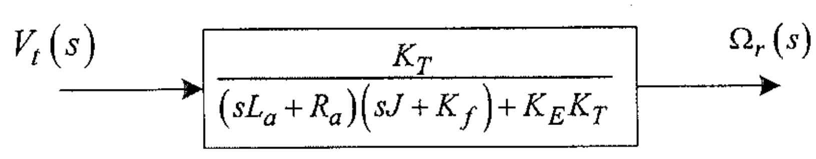
แทน (8.8) ลงใน (8.7) จะได้ว่า

$$V_t(s) = (sL_a + R_a)\left\{\frac{1}{K_T} T_L(s) + \frac{1}{K_T}(Js + K_f)\Omega_r(s)\right\} + K_E \Omega_r(s)$$

$$\Omega_r(s) = \frac{K_T}{(sL_a + R_a)(Js + K_f) + K_E K_T} V_t(s) - \frac{(sL_a + R_a)}{(sL_a + R_a)(Js + K_f) + K_E K_T} T_L(s) \tag{8.9}$$

โดยไม่พิจารณาผลจากการรบกวนโหลดเพื่อให้ง่ายสำหรับการออกแบบตัวควบคุมในเบื้องต้น นั่นคือ

$$\frac{\Omega_r(s)}{V_t(s)} = \frac{K_T}{(sL_a + R_a)(Js + K_f) + K_E K_T} \tag{8.10}$$



รูปที่ 8.13 ระบบควบคุมความเร็วรอบของมอเตอร์ไฟฟ้าดีซีแบบวงรอบเปิด

จะได้แผนภาพวงรอบเปิด (open-loop control block diagram) ของการควบคุมความเร็วรอบของมอเตอร์ดีซีด้วยแรงดันอาร์เมเจอร์ดังแสดงในรูปที่ 8.13

ตัวอย่างที่ 8.2 ให้ดำเนินการจำลองผลเพื่อหาผลตอบสนองของความเร็วรอบของมอเตอร์ดีซีตามสมการที่ 8.10 โดยใช้ SCILAB

วิธีทำ จะได้ชุดคำสั่งดังต่อไปนี้

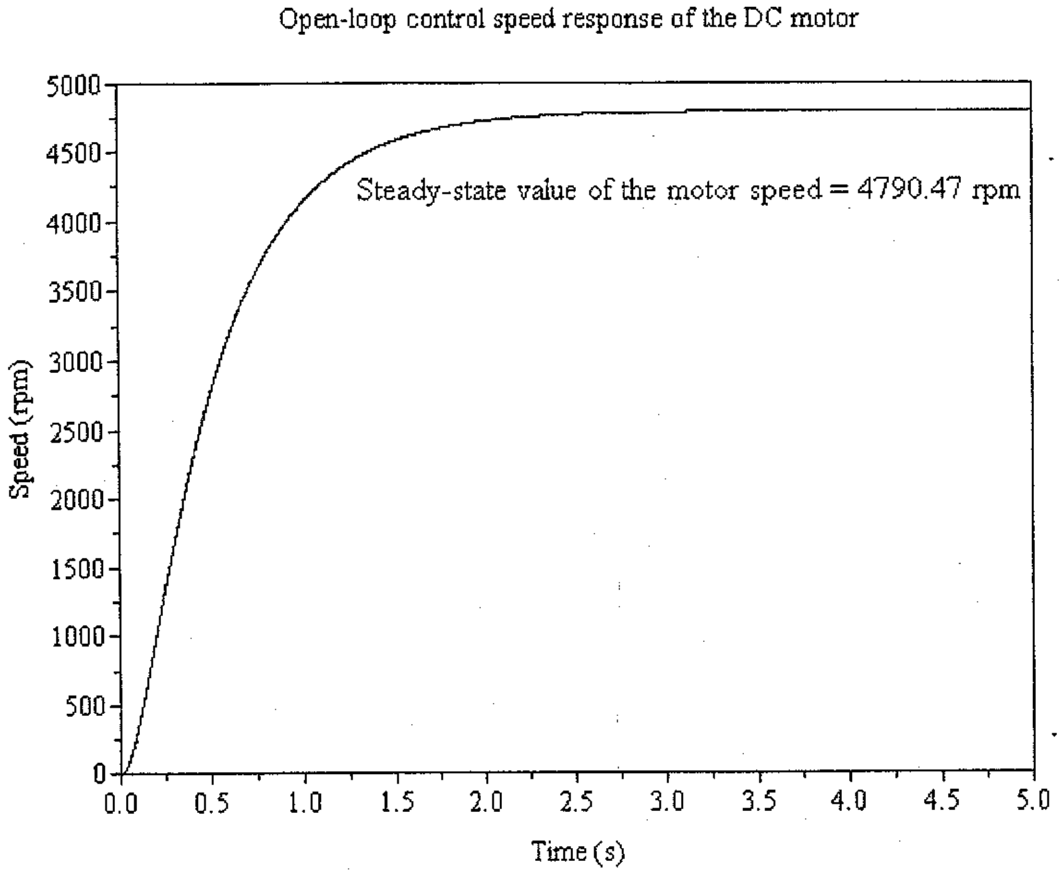
<u>โปรแกรม SCILAB สำหรับตัวอย่างที่ 8.2</u>

// SCILAB source program

```
// กำหนดฟังก์ชันสำหรับจำลองผล                                   otf_mdct.sce
=========================
           otf_mdc.sce
=========================
x = [0.5329245 0.0501212 0.3439865 0.0901721 0.0642641 0.0704620];
Ra = x(1);
La = x(2);
KT = x(3);
KE = x(4);
J = x(5);
KF = x(6);
Vt_mag = 100;
s = poly(0,'s');
numMDC = KT;
denMDC = (La*s+Ra)*(J*s+KF)+KE*KT;
OTF_MDC = Vt_mag*numMDC/denMDC;
t = 0:1e-3:5;
sp = csim('step',t,OTF_MDC);
plot(t,sp*60/2/%pi);
xtitle('Open-loop control speed response of the DC motor','Time (s)','Speed (rpm)');
```

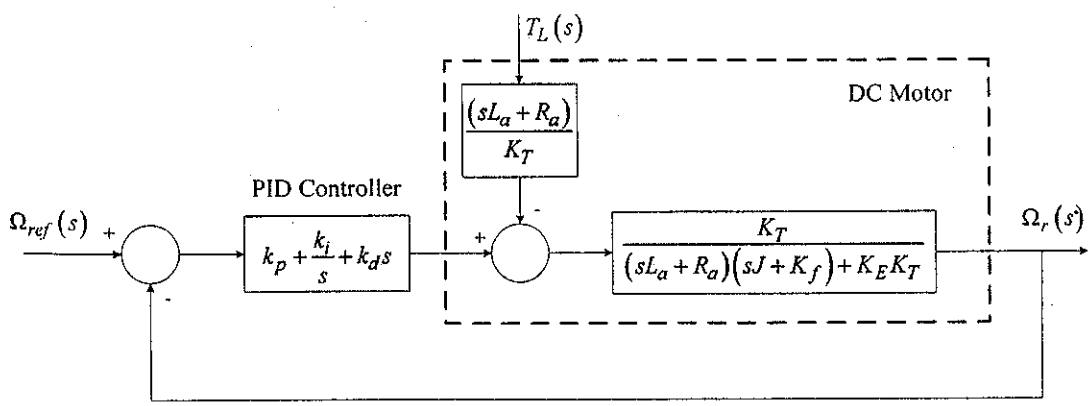
```
// ผลการรันโปรแกรม
--> exec('otf_mot.sci');
```

จะได้ผลการรันโปรแกรมดังกราฟในรูปที่ 8.14



รูปที่ 8.14 ผลตอบสนองของความเร็วรอบของมอเตอร์ไฟฟ้าดีซีแบบวงรอบเปิด

สำหรับปัญหานี้ จะเลือกใช้รูปแบบการควบคุมแรงดันอาร์เมเจอร์ในรูปของการอ้างอิงค่าความเร็วรอบของมอเตอร์ ดังแผนภาพในรูปที่ 8.15



รูปที่ 8.15 ระบบควบคุมความเร็วรอบของมอเตอร์ไฟฟ้าดีซีด้วยตัวควบคุมพีไอดี

โดยการกำหนดค่าโหลดแรงบิดให้เป็นศูนย์ การปรับแต่งพารามิเตอร์ของตัวควบคุมพีไอดีจะดำเนินการโดยอาศัยหลักการประมาณค่ากำลังสองน้อยที่สุด (least square estimation) โดยคำนวณผลต่างระหว่างความเร็วรอบอ้างอิงกับความเร็วรอบที่จำลองได้ ถ้าต้องการควบคุมความเร็วรอบของมอเตอร์ไฟฟ้ากระแสตรง ให้มีความคลาดเคลื่อนของการควบคุมความเร็วน้อยที่สุด โดยใช้

ตัวควบคุมพีไอดี ผลจากการควบคุมพีไอดีที่ออกแบบนี้ต้องทำให้ระบบมอเตอร์ดีซีนี้มีคุณสมบัติ $PO \le 15\%$, $ess \le 2\%$ $T_r \le 1$ s และ $T_s \le 2$ s จะได้ปัญหาค่าเหมาะที่สุดดังต่อไปนี้

$$\begin{aligned} \text{Minimize} \quad & SSE = \left\| \omega_{ref} - \omega\left(k_p, k_i, k_d\right) \right\|^2 \\ \text{Subject to} \quad & PO \le 15 \\ & ess \le 2 \\ & T_r \le 1 \\ & T_s \le 2 \\ & k_{min} \le k_p, k_i, k_d \le k_{max} \end{aligned}$$

เนื่องจากปัญหาเป็นชนิดมีเงื่อนไขบังคับ ดังนั้น แปลงปัญหาไปเป็นการหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไขโดยใช้ฟังก์ชันปรับโทษกำลังสอง จะได้ว่า

$$\begin{aligned} P(\boldsymbol{x}) = & \left\| \omega_{ref} - \omega\left(k_p, k_i, k_d\right) \right\|^2 + \rho\left\{ min(15 - PO, 0) \right\}^2 \\ & + \rho\left\{ min(2 - ess, 0) \right\}^2 + \rho\left\{ min(1 - Tr, 0) \right\}^2 \\ & + \rho\left\{ min(2 - T_s, 0) \right\}^2 \end{aligned}$$

โดยที่ $\omega_{ref} = 1$, $\rho = 10000$

จะพบว่า การประเมินสมรรถนะของการตอบสนองมีความสำคัญมาก ไม่ว่าจะเป็นค่า *MO* (การพุ่งเกินสูงสุด: maximum overshoot) *ess* (ความผิดพลาดในสถานะอยู่ตัว: steady-state error) *Tr* (เวลาขึ้น: rise time) หรือ *Ts* (เวลาตั้งตัว: settling time) ดังนั้น ต้องการฟังก์ชันที่ใช้ประเมินค่าดังกล่าวเสียก่อน ผู้เรียบเรียงได้สรุปฟังก์ชันประเมินสมรรถนะไว้ในฟังก์ชันไฟล์ชื่อ tspec.sci และฟังก์ชันสำหรับประเมินค่าฟังก์ชันการปรับโทษดังนี้

โปรแกรม SCILAB

// SCILAB source program

```
// กำหนดฟังก์ชันสำหรับจำลองผล                                  tspec.sci
========================
            tspec.sci
========================
function [MO,ess,Tr,Ts,Tmo] = tspec(x,t,Rref,errband)
N = length(x);
Ts = t(N);
ess = abs(Rref-x(N));
for u=1:N
    if abs((Rref-x(N-u+1)))>=errband*Rref then
        Ts = t(N-u+1);
```

```
      break;
   end
end
[xmax,imax] = max(abs(x));
if xmax<=Rref then
   MO = 0;
   Tmo = -1;
   Tr = 1e8;
else
   MO = xmax-Rref;
   Tmo = t(imax);
end
if xmax<0.9*Rref then
   Tr = 1e8;
else
   id1 = find(abs(x)>=0.1*Rref);
   id2 = find(abs(x)>=0.9*Rref);
   Tr = t(id2(1))-t(id1(1));
end
endfunction
```

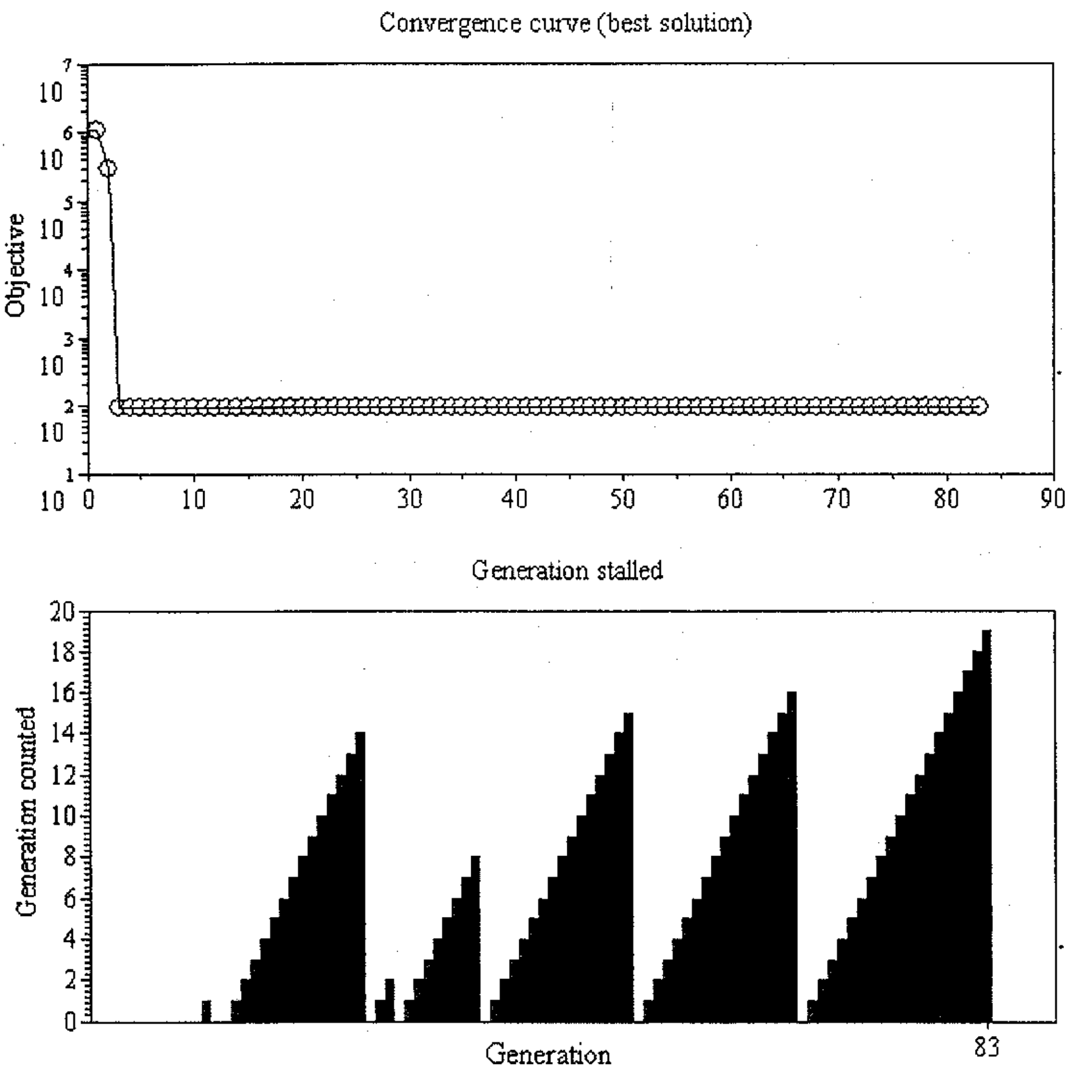
// กำหนดฟังก์ชันวัตถุประสงค์ *sim_mdc.sci*

```
========================
         sim_mdc.sci
========================
function f = sim_mdc(x)
kp = x(1);
ki = x(2);
kd = x(3);
param = [0.5329245 0.0501212 0.3439865 0.0901721 0.0642641 0.0704620];
Ra = param(1);
La = param(2);
KT = param(3);
KE = param(4);
J = param(5);
```

```
KF = param(6);
Vt_mag = 100;
RPM_ref = 4500;
s = poly(0,'s');
numMDC = KT;
denMDC = (La*s+Ra)*(J*s+KF)+KE*KT;
OTF_MDC = numMDC/denMDC;
TF_PID = kp + ki/s + kd*s;
NUM = Vt_mag/s + (2*%pi*RPM_ref/60)*TF_PID/s;
DEN = 1/OTF_MDC + TF_PID;
CTF_PIDMDC = (60/2/%pi)*NUM/DEN;
t = 0:1e-3:5;
RPM = csim('impulse',t,CTF_PIDMDC);
//plot(t,RPM);
//xtitle('Closed-loop control speed response of the DC motor','Time (s)','Speed (rpm)');
[PO,ess,Tr,Ts,Tmo]=tspec(RPM,t,RPM_ref,0.02);
//disp([PO ess Tr Ts])
SSE = sum((1-RPM/RPM_ref).^2);
constr = [PO-15;ess-2;Tr-1;Ts-2];
rho = 10000;
P = 0;
for k=1:4
    if constr(k)<0 then
    constr(k) = 0;
    end
    P = P + rho*constr(k)^2;
end
f = SSE + P;
endfunction
```

การออกแบบกรณีแรกนี้ จะใช้ตัวควบคุมพี (proportional controller หรือ P-controller) จะคิดเฉพาะค่า k_p ในช่วง 0 – 100 เท่านั้น โดยที่ $k_i = k_d = 0$ เพื่อทดลองปรับแต่งพารามิเตอร์ การทดสอบเลือกใช้ PSO ในการแก้ปัญหาค่าเหมาะที่สุด ดังนี้

```
// ผลการรันโปรแกรม
-->[gbest,fgbest,k] = psomain(sim_mdc,1,20,[0 100],5.0,[200 20 0]);
-->gbest
 gbest  =
   0.9259302
-->fgbest
 fgbest  =
   92.429362
```



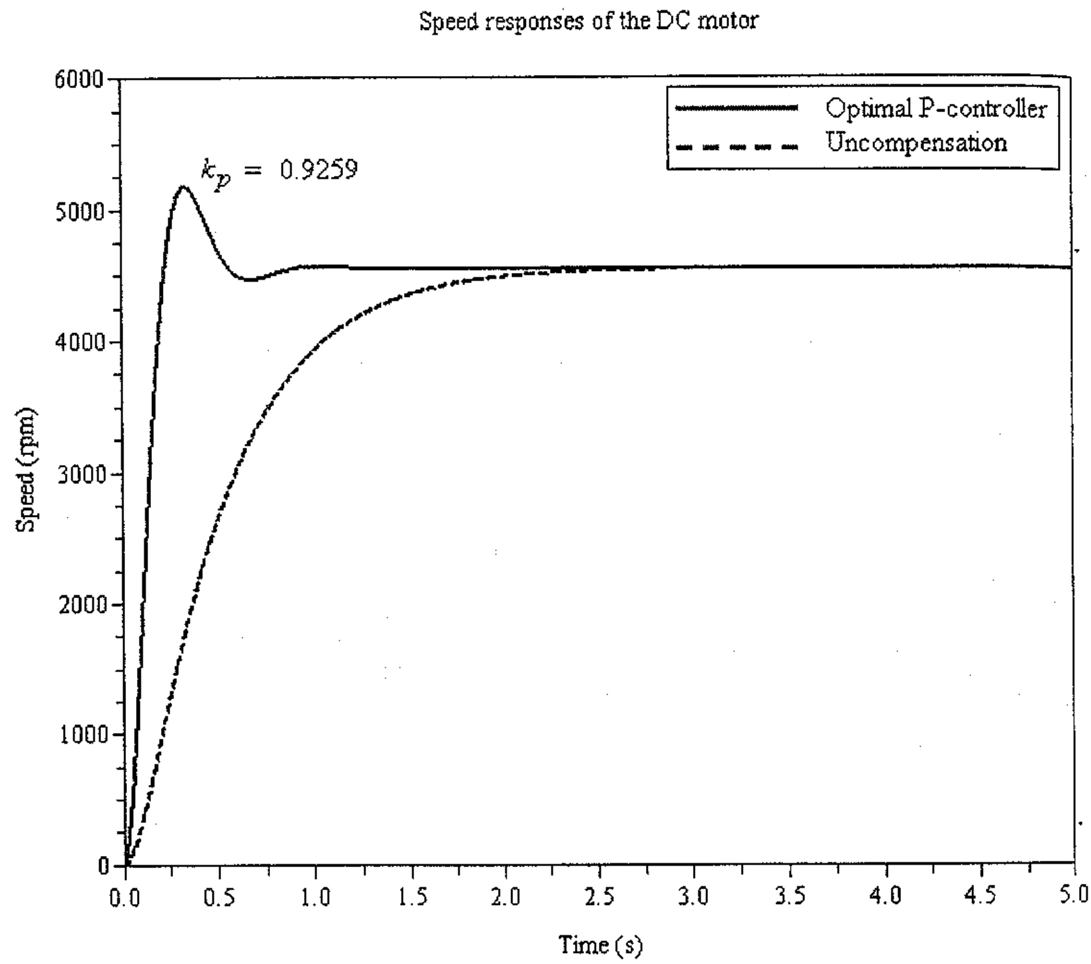
รูปที่ 8.16 อัตราการลู่เข้าของระเบียบวิธี PSO สำหรับการออกแบบตัวควบคุมพี

ผลการทดสอบได้ค่า k_p = 0.9259 ได้ค่าฟังก์ชันวัตถุประสงค์ 92.4294 และจะได้ค่าสมรรถนะตัวอื่น ๆ เป็นดังนี้

MO = 14.9798%
ess = 1.1438%

$Tr = 0.1530$ s
$Ts = 0.1550$ s

ได้ผลการตอบสนองความเร็วรอบเทียบกับกรณีการควบคุมแบบวงรอบเปิดที่ความเร็วอ้างอิง 4500 rpm ดังแสดงในรูปที่ 8.17

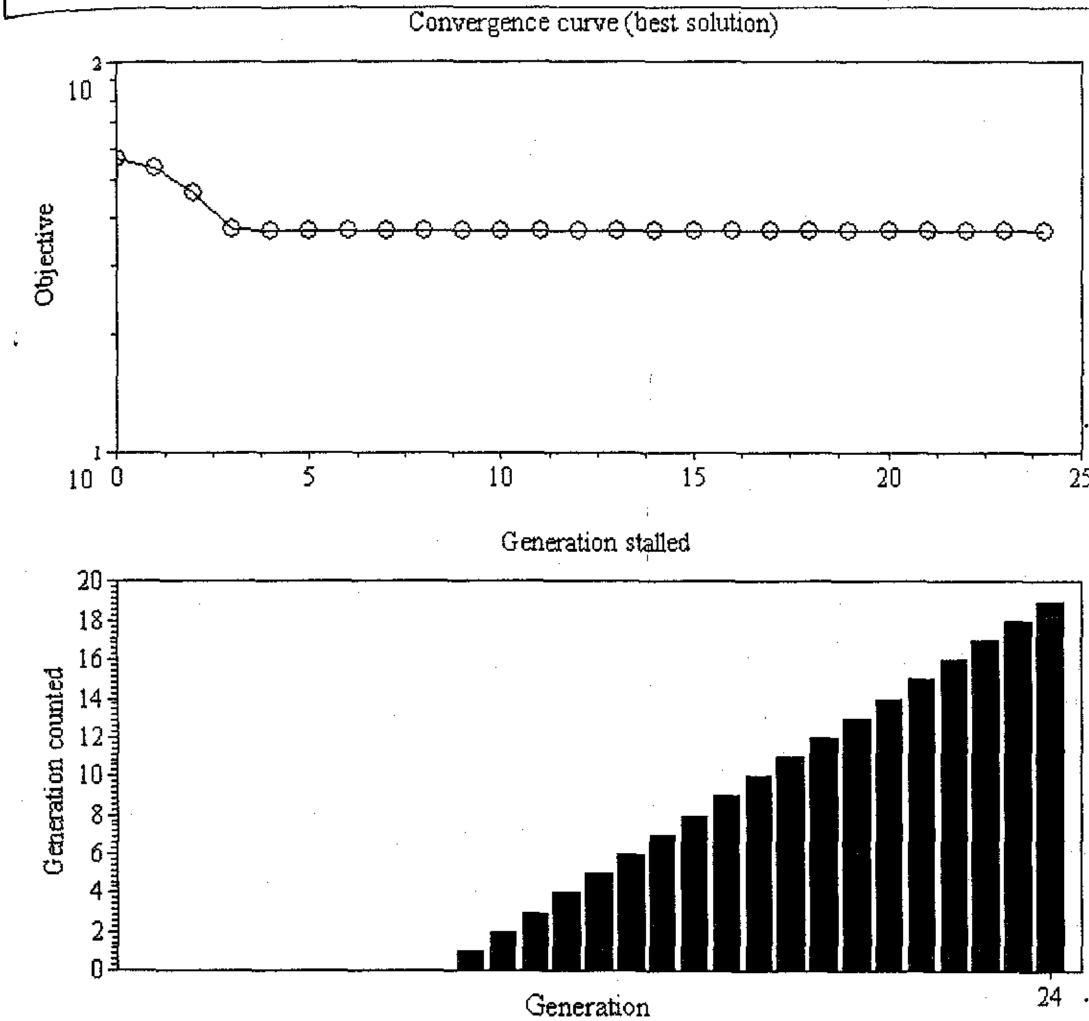


รูปที่ 8.17 ผลตอบสนองความเร็วรอบของมอเตอร์ที่ได้จากการออกแบบตัวควบคุมพี

จะพบว่า ตัวควบคุมพี ให้ผลการตอบสนองที่ดีในระดับหนึ่ง อย่างไรก็ตาม การพุ่งเกิน และความผิดพลาดในสถานะคงตัวมีค่าสูง ดังนั้น เพื่อปรับปรุงให้สมรรถนะของระบบดียิ่งขึ้น จะใช้ตัวควบคุมพีไอดี (proportional-plus-integral-plus-derivative controller หรือ PID-controller) จะคิดค่า k_p ในช่วง 0 – 2 k_i ในช่วง 0.1 – 1.0 และ k_d ในช่วง 0.01 – 0.1 เท่านั้น เพื่อทดลองปรับแต่งพารามิเตอร์ การทดสอบเลือกใช้ PSO ในการแก้ปัญหาค่าเหมาะที่สุด ดังนี้

```
// ผลการรันโปรแกรม
-->[gbest,fgbest,k] = psomain(sim_mdc,3,10,[0 2;0.1 1;0.01 0.1],1.0,[200 20 0]);
-->gbest
 gbest  =
```

```
 2.   1.   0.1
-->fgbest
fgbest  =
  36.912523
```

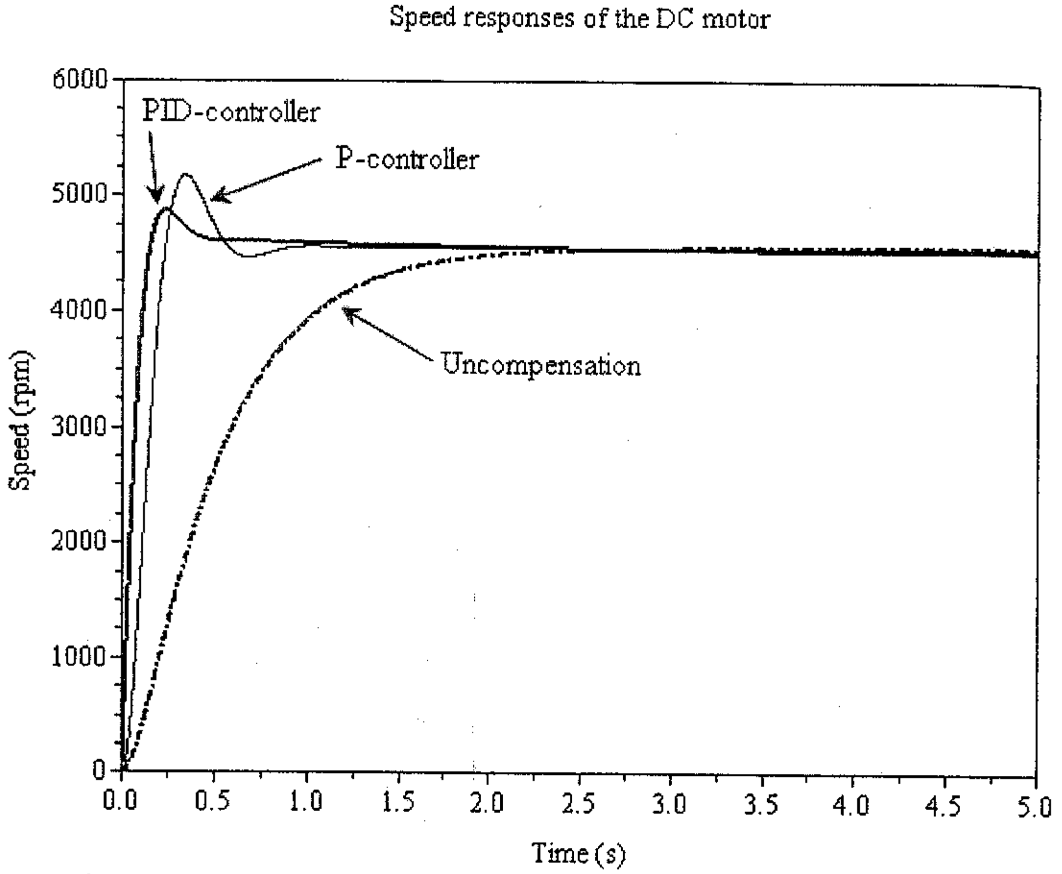


รูปที่ 8.18 อัตราการลู่เข้าของระเบียบวิธี PSO สำหรับการออกแบบตัวควบคุมพีไอดี

ผลการทดสอบได้ค่า $k_p = 2.0$ $k_i = 1.0$ และ $k_d = 0.1$ ได้ค่าฟังก์ชันวัตถุประสงค์ 36.912523 และจะได้ค่าสมรรถนะตัวอื่น ๆ เป็นดังนี้

$MO = 8.2762\%$
$ess = 0.291\%$
$Tr = 0.105$ s
$Ts = 0.951$ s

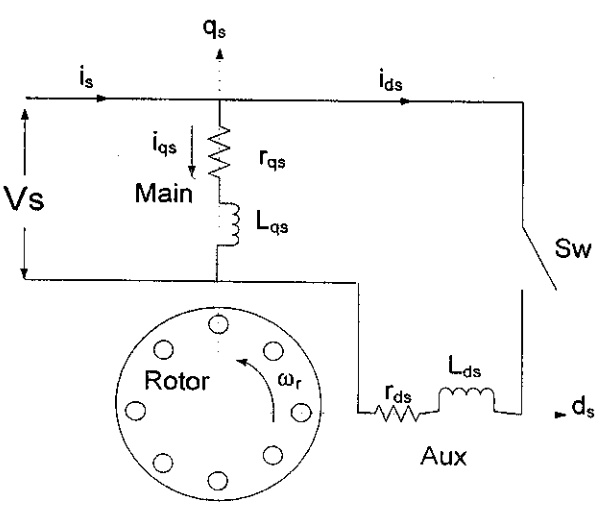
ได้ผลการตอบสนองความเร็วรอบเทียบกับกรณีการควบคุมแบบวงรอบเปิด และกรณีใช้ตัวควบคุมพีที่ความเร็วอ้างอิง 4500 rpm ดังแสดงในรูปที่ 8.19



รูปที่ 8.19 ผลตอบสนองความเร็วรอบของมอเตอร์ที่ได้จากการออกแบบตัวควบคุมพีไอดี

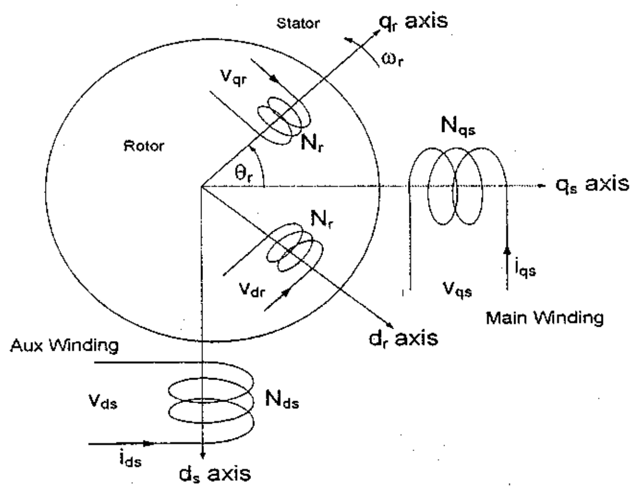
8.4 การปรับแต่งพารามิเตอร์สำหรับมอเตอร์เหนี่ยวนำเฟสเดียว

มอเตอร์ไฟฟ้าเหนี่ยวนำแตกต่างจากมอเตอร์ดีซีทั้งในด้านโครงสร้าง ความซับซ้อนของแบบจำลอง สมรรถนะและการควบคุม เริ่มต้นพิจารณาจากส่วนประกอบของมอเตอร์เหนี่ยวนำเฟสเดียวแบบแยกเฟส (split-phase motor) [10] ดังรูปที่ 8.20



รูปที่ 8.20 โครงสร้างของมอเตอร์ไฟฟ้าเหนี่ยวนำเฟสเดียวแบบแยกเฟส

การหาแบบจำลองทางคณิตศาสตร์ของมอเตอร์เหนี่ยวนำเฟสเดียวแบบนี้จะใช้แบบจำลองแบบพลวัตซึ่งต้องอาศัยการพิจารณา มอเตอร์แบบ 2 แกน ซึ่งประกอบด้วยแกนดี (direct axis หรือ d-axis) และแกนคิว (quadrature axis หรือ q-axis) ซึ่งทั้ง 2 แกนจะตั้งฉากกัน ดังนั้น ด้วยลักษณะการวางตัวของชุดขดลวดหลักและชุดขดลวดช่วย ทำให้เกิดการสร้างสนามแม่เหล็กที่ตั้งฉากกัน โดยสามารถพิจารณาโดยใช้ทฤษฎีกรอบอ้างอิง (reference-frame theory) [11] ในที่นี้จะพิจารณาที่แกนอ้างอิงหยุดนิ่ง (stationary reference frame) จากรูปที่ 8.21 ปริมาณทางไฟฟ้าที่โรเตอร์และสเตเตอร์ จะประกอบด้วยองค์ประกอบบนแกนดีและองค์ประกอบบนแกนคิว ซึ่งค่าแรงดันของชุดขดลวดหลักและขดลวดช่วยจะมีมุมเฟสต่างกัน 90 องศาทางไฟฟ้า



รูปที่ 8.21 แผนภาพขดลวดสเตเตอร์และโรเตอร์ของมอเตอร์เหนี่ยวนำแบบแยกเฟส

เมื่อนำมาพิจารณาที่แกนอ้างอิงหยุดนิ่ง จะทำการย้ายเวกเตอร์แรงดัน และกระแส ในโรเตอร์ (v_{qr}, v_{dr} , i_{qr} , i_{dr}) ไปอยู่บนสเตเตอร์ โดยใช้การแตกเวกเตอร์เข้าสู่แกนอ้างอิงดังกล่าว จากรูปกำหนดให้แกน qr ทำมุมกับแกน qs เท่ากับ θ_r ดังนั้นจะได้ความสัมพันธ์ดังนี้

$$\begin{bmatrix} v_{qr}^{s} \\ v_{dr}^{s} \end{bmatrix} = \begin{bmatrix} cos\theta_r & cos\left(\frac{\pi}{2}-\theta_r\right) \\ cos\left(\frac{\pi}{2}+\theta_r\right) & cos\theta_r \end{bmatrix} \begin{bmatrix} v_{qr}^{r} \\ v_{dr}^{r} \end{bmatrix} \tag{8.11}$$

ในบทความนี้จะพิจารณากรณีโรเตอร์สมมาตร (symmetrical cage rotor) โดยค่าจำนวนรอบประสิทธิ์ผล (Effective turn: N) ค่า ความเหนี่ยวนำรั่วของโรเตอร์ (rotor's leakage inductances) และค่าความต้านทานของโรเตอร์ (rotor's resistances) ในแนวแกนดีและคิวมีค่าเท่ากัน $N_{qr} = N_{dr}$, $L_{lqr} = L_{ldr}$ และ $r_{qr} = r_{ds} = r_r$ [12] จะได้สมการแรงดันที่โรเตอร์ในรูปของเมตริกซ์เป็น

$$\begin{bmatrix} V_{qr}^{r} \\ V_{dr}^{r} \end{bmatrix} = \begin{bmatrix} r_{qr} + pL_{qrqr} & pL_{qrdr} \\ pL_{drqr} & r_{dr} + pL_{drdr} \end{bmatrix} \begin{bmatrix} i_{qr}^{r} \\ i_{dr}^{r} \end{bmatrix} + \begin{bmatrix} L_{qrqs} & L_{qrds} \\ L_{drqr} & L_{drds} \end{bmatrix} p \begin{bmatrix} i_{qs}^{s} \\ i_{ds}^{s} \end{bmatrix} \tag{8.12}$$

ทำการย้าย แรงดัน และ กระแส ที่อยู่บนโรเตอร์ ไปยังสเตเตอร์ ด้วยหลังการถ่ายโอนแกน โดยค่าความเหนี่ยวนำร่วมระหว่างแกนดีและคิวของสเตเตอร์จะมีค่าเท่ากับศูนย์ จากสมการแรงดันของ ทั้งโรเตอร์ และสเตเตอร์ เมื่อพิจารณาด้วยแกนอ้างอิงหยุดนิ่งโดยในที่นี้จะทำการจำลองผลการขับเคลื่อนมอเตอร์เหนี่ยวนำเฟสเดียวแบบจ่ายแรงดันให้ขดลวดหลักเพียงชุดเดียวจึงจำเป็นต้องการย้ายตัวแปรที่อยู่บนแกนดีของสเตเตอร์ แกนดีและคิวของโรเตอร์ไปอยู่บนแกนคิวของสเตเตอร์ ด้วยความสัมพันธ์ของจำนวนรอบประสิทธิ์ภาพ สุดท้ายจะได้แบบจำลองปริภูมิสถานะ (state-space) บนแกนอ้างอิงหยุดนิ่ง ดังนี้ [13–17]

$$\begin{bmatrix} v_{qs}^{s} \\ v_{ds}^{'s} \\ v_{qr}^{'s} \\ v_{dr}^{'s} \end{bmatrix} = \begin{bmatrix} r_{qs} & 0 & -\omega_r L_{mqs}\sin\theta_r & -\omega_r L_{mqs}\cos\theta_r \\ 0 & r_{ds}' & \omega_r L_{dsqr}\cos\theta_r & -\omega_r L_{mqs}\cos\theta_r \\ -\omega_r L_{mqs}\sin\theta_r & \omega_r L_{mqs}\cos\theta_r & r_r' & 0 \\ -\omega_r L_{mqs}\cos\theta_r & -\omega_r L_{mqs}\cos\theta_r & 0 & r_r' \end{bmatrix} \begin{bmatrix} i_{qs}^{s} \\ i_{ds}^{'s} \\ i_{qr}^{'s} \\ i_{dr}^{'s} \end{bmatrix}$$
$$+ \begin{bmatrix} L_{lqs}' + L_{mqs} & 0 & L_{mqs}\cos\theta_r & -L_{mqs}\sin\theta_r \\ 0 & \left(L_{lds}' + L_{mqs}\right) & \omega_r L_{dsqr}\sin\theta_r & \omega_r L_{dsqr}\cos\theta_r \\ L_{mqs}\cos\theta_r & L_{mqs}\sin\theta_r & \left(L_{lr}' + L_{mqs}\right) & 0 \\ -L_{mqs}\sin\theta_r & \cos\theta_r & 0 & \left(L_{lr}' + L_{mqs}\right) \end{bmatrix} p \begin{bmatrix} i_{qs}^{s} \\ i_{ds}^{'s} \\ i_{qr}^{'s} \\ i_{dr}^{'s} \end{bmatrix} \tag{8.13}$$

จากสมการปริภูมิสถานะ นั่นคือ

$$\frac{d}{dt}[i] = [A][i] + [B][v] \tag{8.14}$$

พิจารณาการเคลื่อนที่ของโรเตอร์โดยใช้กฎการเคลื่อนที่ของนิวตัน จะได้ว่า

$$\frac{P}{2} \cdot T_e(t) - \frac{P}{2} \cdot T_L(t) - B_m \omega_r(t) = J_m \alpha = J_m \frac{d\omega_r}{dt} \tag{8.15}$$

จัดรูปใหม่ได้ดังสมการที่ 8.15

$$\frac{d\omega_r}{dt} = \frac{P}{2J_m} T_e(t) - \frac{P}{2J_m} T_L - \frac{B_m}{J_m} \omega_r(t) \tag{8.16}$$

$$\frac{d\theta_r}{dt} = \omega_r \tag{8.17}$$

นำสมการที่ 8.16 – 8.17 มาเขียนในรูปเมตริกซ์ได้

$$\begin{bmatrix} \dfrac{d\omega_r}{dt} \\ \dfrac{d\theta_r}{dt} \end{bmatrix} = \begin{bmatrix} -\dfrac{B_m}{J_m} & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_r \\ \theta_r \end{bmatrix} + \begin{bmatrix} \dfrac{P}{2J_m} \\ 0 \end{bmatrix} \begin{bmatrix} T_e - T_L \end{bmatrix} \tag{8.18}$$

เมื่อ แรงบิดทางแม่เหล็กไฟฟ้า T_e มีค่าเท่ากับ

$$T_e = \left(\frac{P}{2}\right) L_{mqs}\, i_{qs}\,(-i_{qr}\sin\theta_r - i_{dr}\cos\theta_r) + L_{mqs}\, i_{ds}\,(i_{qr}\cos\theta_r - i_{dr}\sin\theta_r) \tag{8.19}$$

ดังนั้น สร้างสมการปริภูมิสถานะของมอเตอร์เหนี่ยวนำเฟสเดียวได้ดังนี้

$$\begin{bmatrix} \dfrac{d[i]_{4x1}}{dt} \\ \dfrac{d\omega_r}{dt} \\ \dfrac{d\theta_r}{dt} \end{bmatrix} = \begin{bmatrix} [A]_{4x4} & \cdots\ 0\ \cdots & \\ \vdots & -\dfrac{B_m}{J_m} & 0 \\ 0 & & \\ \vdots & 1 & 0 \end{bmatrix} \begin{bmatrix} [i]_{4x1} \\ \omega_r \\ \theta_r \end{bmatrix} + \begin{bmatrix} [B]_{4x4} & \cdots\ 0\ \cdots & \\ \vdots & \dfrac{P}{2J_m} & 0 \\ 0 & & \\ \vdots & 0 & 0 \end{bmatrix} \begin{bmatrix} [v]_{4x1} \\ T_e - T_L \\ 0 \end{bmatrix} \tag{8.20}$$

ด้วยหลักการเดียวกันกับการปรับแต่งพารามิเตอร์ของมอเตอร์ดีซี พารามิเตอร์ของมอเตอร์เหนี่ยวนำเฟสเดียวสามารถปรับแต่งได้ อย่างไรก็ตาม ตัวแปรที่เกี่ยวข้องประกอบด้วยตัวแปรทั้งสิ้น 10 ตัว ได้แก่

ค่าความต้านทานของขดลวดหลัก (r_{qs})

ค่าความต้านทานของขดลวดช่วย (r_{ds})

ค่าความต้านทานของโรเตอร์ (r_r)

ค่าความเหนี่ยวนำของขดลวดหลัก (L_{lqs})

ค่าความเหนี่ยวนำของขดลวดช่วย (L_{lds})

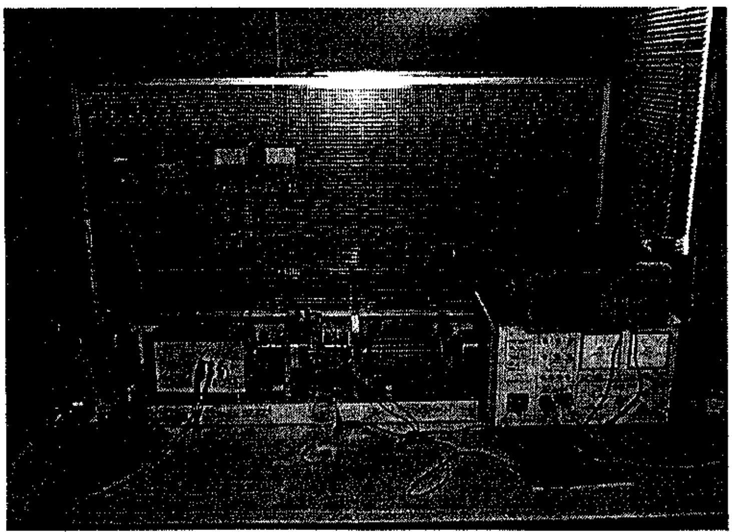
ค่าความเหนี่ยวนำของขดลวดโรเตอร์ (L_{lr})

ค่าความเหนี่ยวนำสนามแม่เหล็ก (L_{mqs})

ค่าโมเมนต์ความเฉื่อยของโรเตอร์ (J)

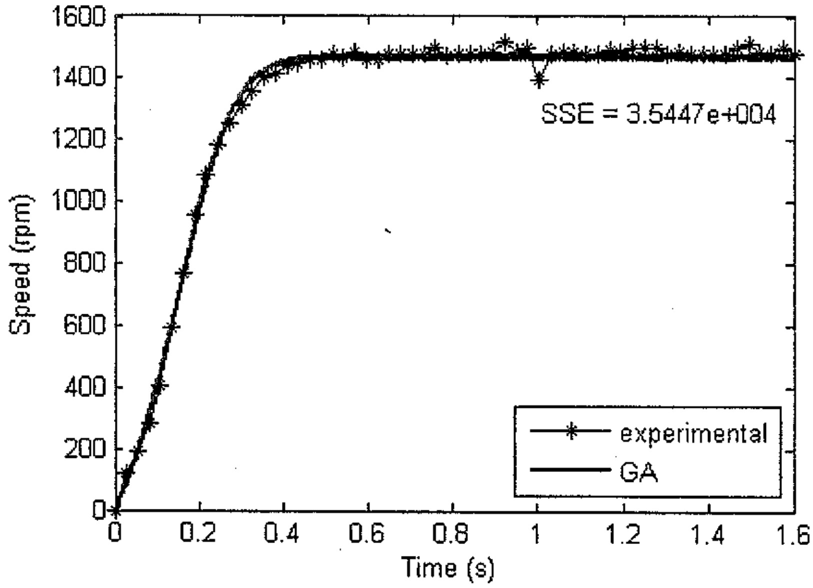
ค่าสัมประสิทธิ์แรงเสียดทาน (B)

ปรับแต่งพารามิเตอร์โดยอ้างอิงจากผลการทดสอบในรูปที่ 8.22 [17]

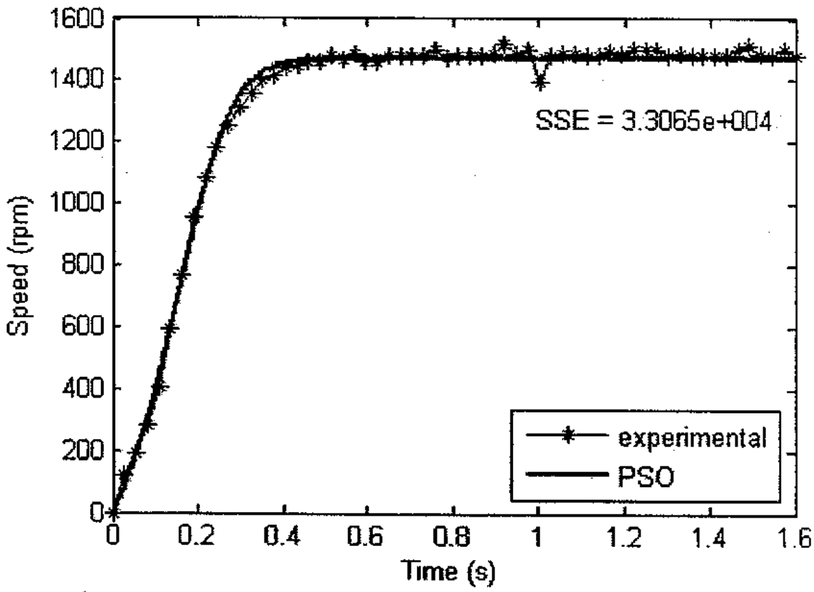


รูปที่ 8.22 มอเตอร์ทดสอบพิกัด ½ hp, 230 V, 2.6 A, 50 Hz, 4 ขั้ว

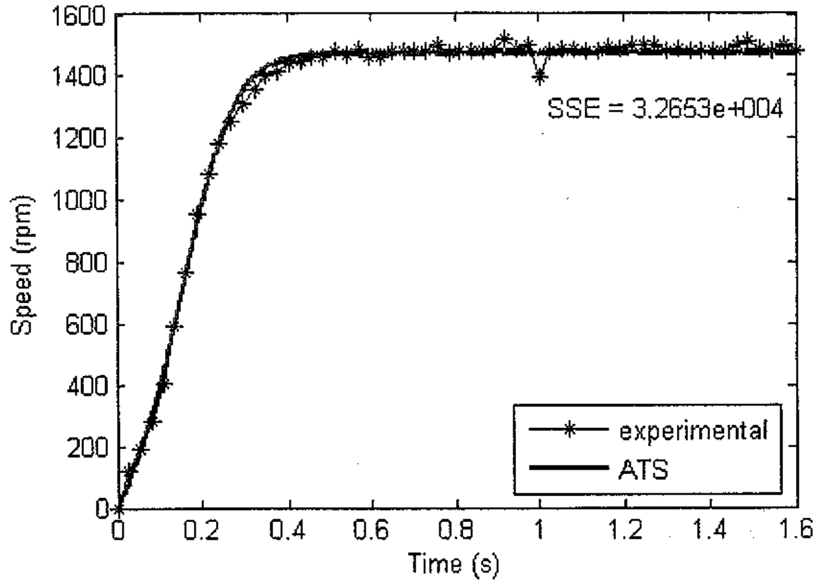
โดยใช้เทคนิคการหาค่าเหมาะที่สุดด้วยระเบียบวิธี GA, PSO และ ATS เปรียบเทียบกัน 3 วิธี จะได้ผลเฉลยค่าพารามิเตอร์ที่ดีที่สุดดังรูปที่ 8.23 – 8.25 และค่าพารามิเตอร์ที่เหมาะสมดังแสดงในตารางที่ 8.1



รูปที่ 8.23 ความเร็วรอบของมอเตอร์โดยใช้ GA ค้นหาพารามิเตอร์



รูปที่ 8.24 ความเร็วรอบของมอเตอร์โดยใช้ PSO ค้นหาพารามิเตอร์



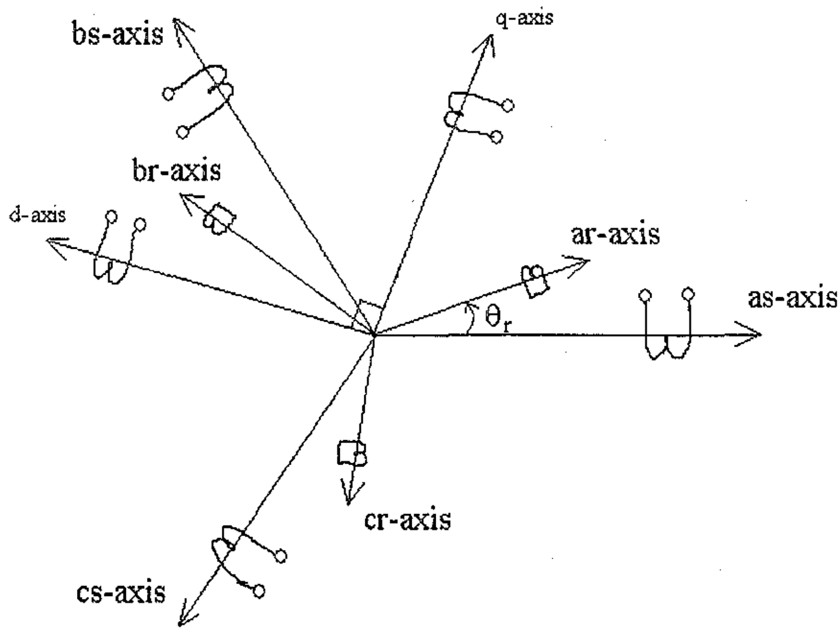
รูปที่ 8.25 ความเร็วรอบของมอเตอร์โดยใช้ ATS ค้นหาพารามิเตอร์

ตารางที่ 8.1 การเปรียบเทียบค่าพารามิเตอร์ที่ได้จากการค้นหาทั้งสามวิธี [17]

Parameters	Search Techniques		
	GA	PSO	ATS
$R_{qs}(\Omega)$	4.0000	2.0000	2.0010
$R_r(\Omega)$	28.0700	28.3517	28.3522
$R_{ds}(\Omega)$	8.0000	4.0000	3.9989
L_{lqs}(H)	0.0100	0.0005	0.0006
L_{lr}(H)	0.0010	0.0005	0.0006
L_{lds}(H)	0.5736	0.7041	0.7046
L_{mqs}(H)	10.0000	20.0000	19.9804
$J_m(N.m.s^2/rad)$	0.0100	0.0100	0.0088
$B_m(N.m.s/rad)$	0.0000	0.0010	0.0009

8.5 การปรับแต่งพารามิเตอร์สำหรับมอเตอร์เหนี่ยวนำ 3 เฟส

มอเตอร์ไฟฟ้าเหนี่ยวนำ 3 เฟส มีความซับซ้อน พฤติกรรมทางพลวัตอธิบายได้ด้วยทฤษฎีกรอบอ้างอิงเช่นเดียวกับมอเตอร์เฟสเดียว ในกรณีของมอเตอร์ 3 เฟส ขดลวดสเตเตอร์และโรเตอร์ในกรอบอ้างอิงปกติจะมี 3 ชุด การโอนย้ายปริมาณต่าง ๆ จะถูกนำมาใช้เช่นเดียวกัน แต่กรณีของมอเตอร์ 3 เฟส ขดลวดสเตเตอร์ทั้ง 3 เฟส จะถือว่ามีจำนวนขดลวดเท่ากัน และในโรเตอร์ก็เช่นเดียวกัน เนื่องจากสมการมีความไม่เป็นเชิงเส้นสูง ดังนั้น เพื่อหลีกเลี่ยงการนำเสนอในรูปสมการทางคณิตศาสตร์ที่มีความซับซ้อนมากจนเกินไป ในที่นี้จะนำเสนอสมการสถานะสุดท้ายของระบบมอเตอร์เหนี่ยวนำ 3 เฟส [11,18,19] ดังสมการต่อไปนี้



รูปที่ 8.26 แผนภาพเวกเตอร์บนแกนสเตเตอร์ แกนโรเตอร์และแกนดีคิว

$$\frac{d}{dt}[x]=[A][x]+[B][u] \tag{8.21}$$

โดยที่

$$[G_1]=\begin{bmatrix} L_{ls}+L_{ms} & -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & L_{ls}+L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} & L_{ls}+L_{ms} \end{bmatrix}$$

$$[G_2]=\begin{bmatrix} L_{ms}\cos\theta_r & L_{ms}\cos\left(\theta_r+\frac{2\pi}{3}\right) & L_{ms}\cos\left(\theta_r-\frac{2\pi}{3}\right) \\ L_{ms}\cos\left(\theta_r-\frac{2\pi}{3}\right) & L_{ms}\cos\theta_r & L_{ms}\cos\left(\theta_r+\frac{2\pi}{3}\right) \\ L_{ms}\cos\left(\theta_r+\frac{2\pi}{3}\right) & L_{ms}\cos\left(\theta_r-\frac{2\pi}{3}\right) & L_{ms}\cos\theta_r \end{bmatrix}$$

$$[G_3]=\begin{bmatrix} L_{ms}\cos\theta_r & L_{ms}\cos\left(\theta_r-\frac{2\pi}{3}\right) & L_{ms}\cos\left(\theta_r+\frac{2\pi}{3}\right) \\ L_{ms}\cos\left(\theta_r+\frac{2\pi}{3}\right) & L_{ms}\cos\theta_r & L_{ms}\cos\left(\theta_r-\frac{2\pi}{3}\right) \\ L_{ms}\cos\left(\theta_r-\frac{2\pi}{3}\right) & L_{ms}\cos\left(\theta_r+\frac{2\pi}{3}\right) & L_{ms}\cos\theta_r \end{bmatrix}$$

$$[G_4]=\begin{bmatrix} L_{lr}+L_{ms} & -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & L_{lr}+L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} & L_{lr}+L_{ms} \end{bmatrix}$$

$$[H_1]=\begin{bmatrix} -R_{ls} & 0 & 0 \\ 0 & -R_{ls} & 0 \\ 0 & 0 & -R_{ls} \end{bmatrix}$$

$$[H_2]=\begin{bmatrix} \omega_r L_{ms}\sin\theta_r & \omega_r L_{ms}\sin\left(\theta_r+\frac{2\pi}{3}\right) & \omega_r L_{ms}\sin\left(\theta_r-\frac{2\pi}{3}\right) \\ \omega_r L_{ms}\sin\left(\theta_r-\frac{2\pi}{3}\right) & \omega_r L_{ms}\sin\theta_r & \omega_r L_{ms}\sin\left(\theta_r+\frac{2\pi}{3}\right) \\ \omega_r L_{ms}\sin\left(\theta_r+\frac{2\pi}{3}\right) & \omega_r L_{ms}\sin\left(\theta_r-\frac{2\pi}{3}\right) & \omega_r L_{ms}\sin\theta_r \end{bmatrix}$$

$$[H_3]=\begin{bmatrix} \omega_r L_{ms}\sin\theta_r & \omega_r L_{ms}\sin\left(\theta_r-\frac{2\pi}{3}\right) & \omega_r L_{ms}\sin\left(\theta_r+\frac{2\pi}{3}\right) \\ \omega_r L_{ms}\sin\left(\theta_r+\frac{2\pi}{3}\right) & \omega_r L_{ms}\sin\theta_r & \omega_r L_{ms}\sin\left(\theta_r-\frac{2\pi}{3}\right) \\ \omega_r L_{ms}\sin\left(\theta_r-\frac{2\pi}{3}\right) & \omega_r L_{ms}\sin\left(\theta_r+\frac{2\pi}{3}\right) & \omega_r L_{ms}\sin\theta_r \end{bmatrix}$$

$$[H_4]=\begin{bmatrix} -R_{lr} & 0 & 0 \\ 0 & -R_{lr} & 0 \\ 0 & 0 & -R_{lr} \end{bmatrix}$$

$$[M]=\begin{bmatrix} -\frac{B_m}{J} & 0 \\ 1 & 0 \end{bmatrix}$$

$$[x]=\begin{bmatrix} i_{as} & i_{bs} & i_{cs} & i_{ar} & i_{br} & i_{cr} & \omega_r & \theta_r \end{bmatrix}^T$$

$$[u]=\begin{bmatrix} v_{as} & v_{bs} & v_{cs} & v_{ar} & v_{br} & v_{cr} & -\frac{P}{2J}\left(T_E-T_L\right) & 0 \end{bmatrix}^T$$

$$[G]=\begin{bmatrix} G_1 & G_2 \\ G_3 & G_4 \end{bmatrix}$$

$$[H]=\begin{bmatrix} H_1 & H_2 \\ H_3 & H_4 \end{bmatrix}$$

$$[A]=\begin{bmatrix} G & 0 \\ 0 & I \end{bmatrix}^{-1}\begin{bmatrix} H & 0 \\ 0 & M \end{bmatrix}$$

$$[B]=\begin{bmatrix} G & 0 \\ 0 & I \end{bmatrix}^{-1}$$

R_{ls} ค่าความต้านทานสเตเตอร์ต่อเฟส (Ω)

R_{lr} ค่าความต้านทานโรเตอร์ต่อเฟส (Ω)

L_{ls} ค่าความเหนี่ยวนำรั่วของขดลวดสเตเตอร์ต่อเฟส (H)

L_{lr} ค่าความเหนี่ยวนำรั่วของขดลวดสเตเตอร์ต่อเฟส (H)

L_{ms} ค่าความเหนี่ยวนำสนามแม่เหล็กของขดลวดสเตเตอร์ต่อเฟส (H)

θ_r ตำแหน่งของโรเตอร์ (rad)

ω_r ความเร็วรอบการหมุนของโรเตอร์ (rad/s)

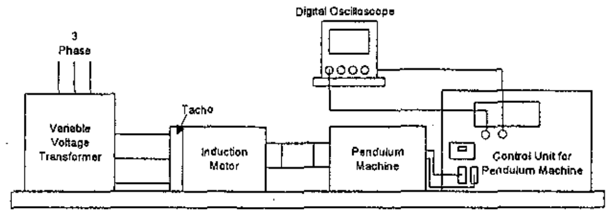
$[i_{as}, i_{bs}, i_{cs}]$ เวกเตอร์กระแสสเตเตอร์

$[i_{ar}, i_{br}, i_{cr}]$ เวกเตอร์กระแสโรเตอร์

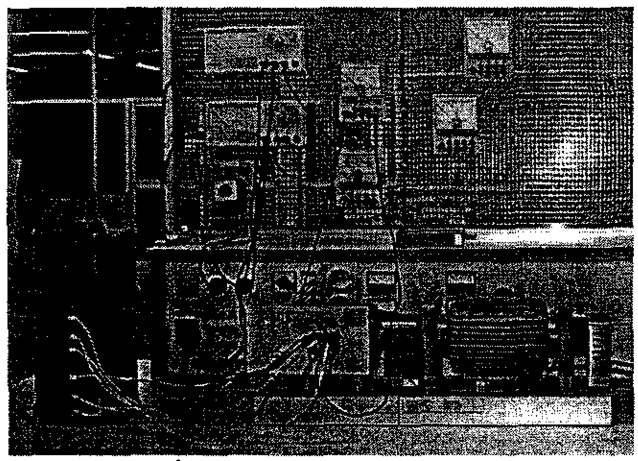
B_m สัมประสิทธิ์ความเสียดทานของโรเตอร์

J_m โมเมนต์ความเฉื่อยของโรเตอร์

ดำเนินการปรับแต่งพารามิเตอร์โดยอ้างอิงจากผลการทดสอบมอเตอร์เหนี่ยวนำ 3 เฟส ในรูปที่ 8.27 [19]

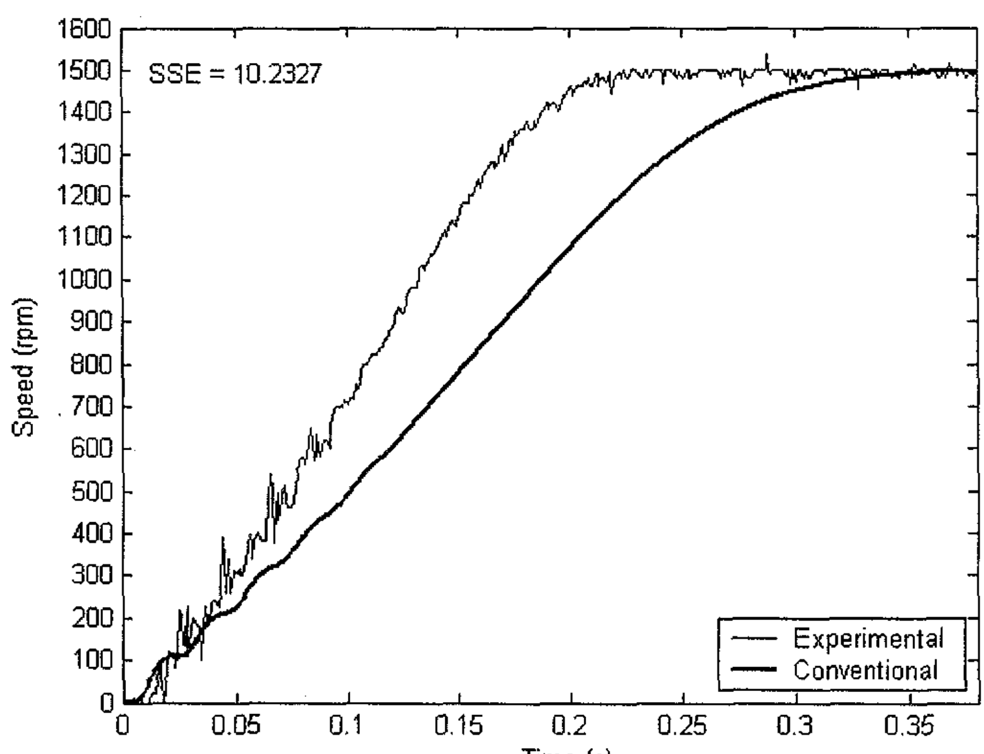


ก) แผนภาพชุดทดสอบ

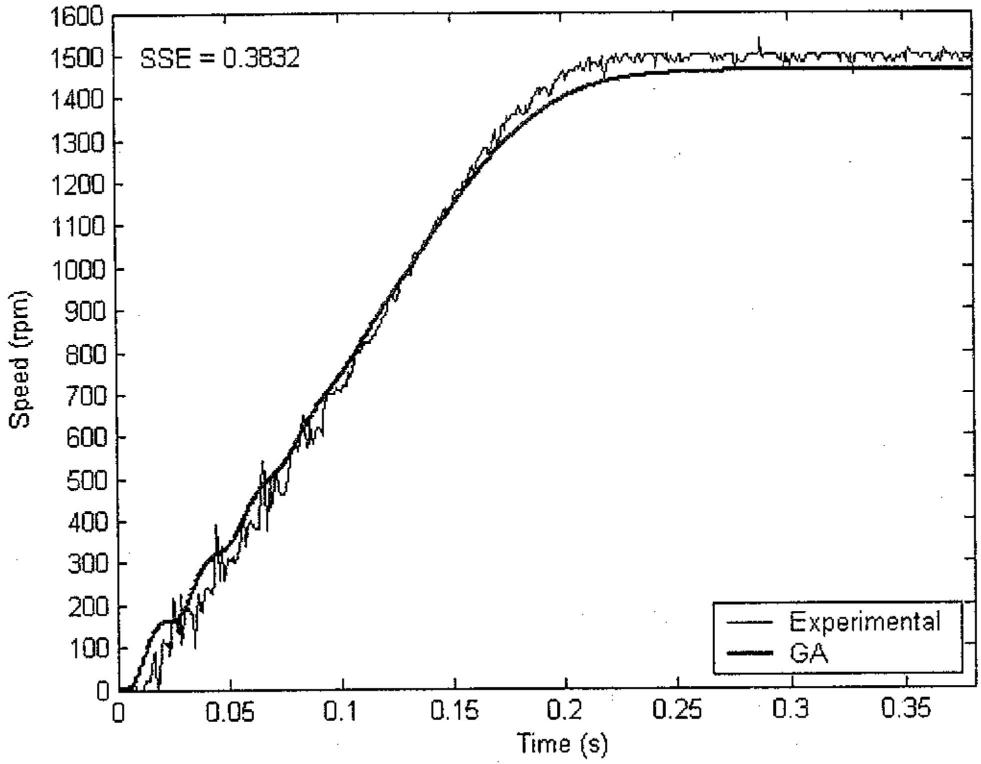


ข) ชุดทดสอบมอเตอร์เหนี่ยวนำ 3 เฟส พิกัด 0.37-kW, 400/230-V, 50-Hz

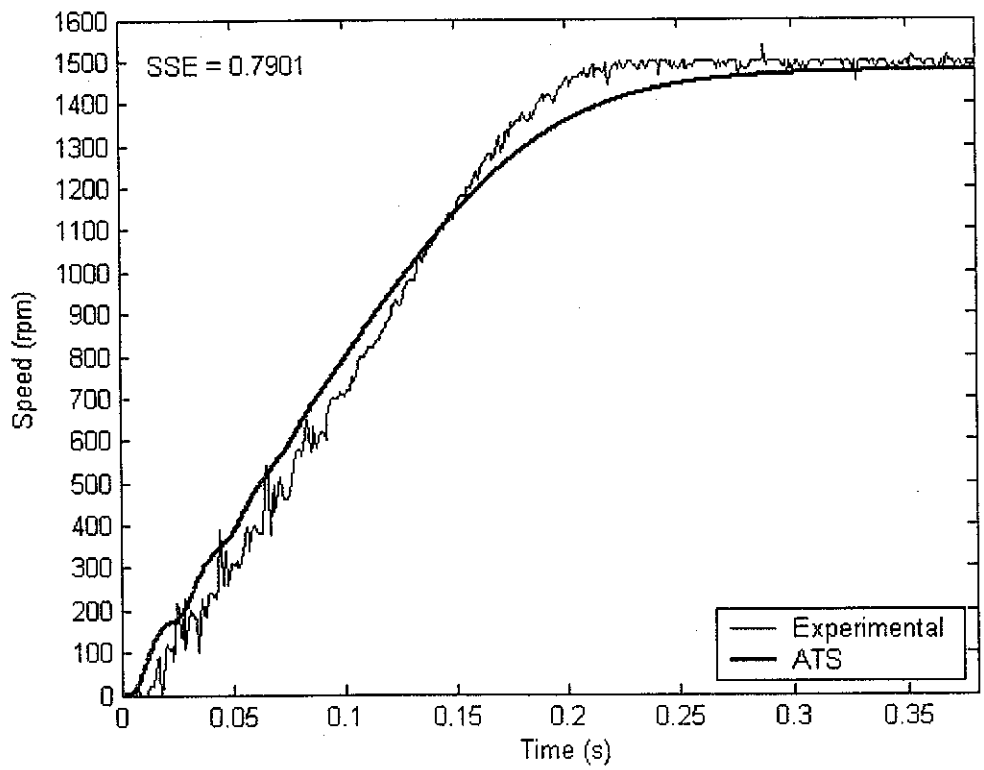
รูปที่ 8.27 ชุดทดสอบมอเตอร์เหนี่ยวนำ 3 เฟส



รูปที่ 8.28 ความเร็วรอบของมอเตอร์โดยใช้การคำนวณพารามิเตอร์แบบดั้งเดิม



รูปที่ 8.29 ความเร็วรอบของมอเตอร์โดยใช้การค้นหาพารามิเตอร์ด้วย GA



รูปที่ 8.30 ความเร็วรอบของมอเตอร์โดยใช้การค้นหาพารามิเตอร์ด้วย ATS

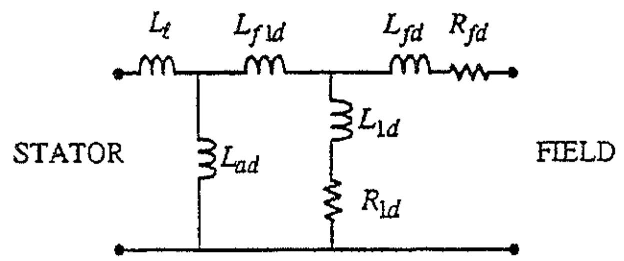
โดยใช้เทคนิคการหาค่าเหมาะที่สุดด้วยระเบียบวิธี GA และ ATS เปรียบเทียบกับผลการทดสอบพารามิเตอร์แบบดั้งเดิม (conventional test) [3,7] ได้แก่ การทดสอบในสภาวะไร้โหลด (no-load test) การทดสอบตรึงตัวหมุน (locked-rotor test) และการทดสอบการหน่วง (retardation test) โดยจะนำผลที่ได้จากทั้ง 3 กรณีมาเปรียบเทียบกันจะได้ผลเฉลยค่าพารามิเตอร์ในแต่ละกรณีดังรูปที่ 8.28 – 8.30 และค่าพารามิเตอร์ที่เหมาะสมดังแสดงในตารางที่ 8.2

ตารางที่ 8.2 การเปรียบเทียบค่าพารามิเตอร์ที่ได้จากการค้นหาทั้งสามวิธี [19]

Parameters	Conventional	GA	ATS
R_{ls} (Ω)	74.02	40.4249	41.3487
R_{lr} (Ω)	62.01	28.0440	37.8249
L_{ls} (H)	0.2087	0.0376	0.0371
L_{lr} (H)	0.2087	0.1394	0.0571
L_{ms} (H)	3.4377	1.6565	1.2332
B_m -$N{\cdot}m{\cdot}s/rad$	0.0000	0.0040	0.0017
J_m - $N{\cdot}m{\cdot}s^2/rad$	0.0025	0.0034	0.0041

8.6 การปรับแต่งพารามิเตอร์สำหรับเครื่องกำเนิดไฟฟ้าซิงโครนัส

เครื่องกำเนิดไฟฟ้าซิงโครนัส (synchronous generator) มีความสำคัญอย่างมากต่อการทำงานในระบบไฟฟ้ากำลัง การศึกษาพฤติกรรมทางพลวัตของเครื่องกำเนิดไฟฟ้านี้ช่วยให้เข้าใจถึงปรากฏการณ์ต่าง ๆ ที่เกิดขึ้นในขณะจ่ายโหลด ปัญหาด้านเสถียรภาพและความมั่นคงในระบบไฟฟ้าสามารถวิเคราะห์ภายใต้แบบจำลองที่เหมาะสม ภายใต้การแปลงปริมาณ 3 เฟส โดยใช้ทฤษฎีกรอบอ้างอิงเช่นเดียวกับในกรณีของมอเตอร์เหนี่ยวนำ แต่เลือกวางแกนอ้างอิงไว้บนแกนหมุนซึ่งหมุนด้วยความเร็วซิงโครนัส (synchronous reference frame) หรือนิยมเรียกว่า ผลการแปลงปาร์ก (Park's transform) [20] เพื่อให้การประเมินสมรรถนะของเครื่องกำเนิดไฟฟ้ามีความสอดคล้องกับพฤติกรรมที่เกิดขึ้น พารามิเตอร์ต่าง ๆ ของเครื่องกำเนิดไฟฟ้าเป็นปัจจัยที่สำคัญต่อการประเมินสมรรถนะดังกล่าว ในที่นี้ จะนำเสนอการประมาณค่าพารามิเตอร์ของเครื่องกำเนิดไฟฟ้าซิงโครนัสรูปแบบหนึ่ง ซึ่งอ้างอิงการปรับแต่งพารามิเตอร์ในโดเมนความถี่ (frequency domain) ผลการตอบสนองในโดเมนความถี่นี้จะถูกแยกออกมาเป็นสองแกน ได้แก่ แกนดีและแกนคิว แยกจากกัน เพื่อเป็นตัวอย่างในเบื้องต้น การประมาณค่าพารามิเตอร์ บนแกนดีจะถูกนำเสนอดังรูปที่ 8.31 [21]



รูปที่ 8.31 วงจรสมมูลของเครื่องกำเนิดไฟฟ้าซิงโครนัสบนแกนคิว

ฟังก์ชันถ่ายโอนของความเหนี่ยวนำปฏิบัติการ (d-axis operational inductance transfer function) ของเครื่องกำเนิดไฟฟ้าซิงโครนัสแสดงไว้ในสมการที่ 8.22 เมื่อพิจารณาค่าพารามิเตอร์จากการทดสอบแบบดั้งเดิม ทำให้รูปแบบฟังก์ชันถ่ายโอนดังกล่าวถูกปรับเปลี่ยนให้อยู่ในรูปของความเหนี่ยวนำชั่วครู่และสภาวะชั่วแวบ (transient and sub-transient inductances) ดังแสดงในสมการที่ 8.23 [21]

$$L_d(s) = L_d \frac{(1+T_1 s)(1+T_2 s)}{(1+T_3 s)(1+T_4 s)} \tag{8.22}$$

$$L_d(s) = L_d \frac{\left[1+\left(\tau'_{\mathrm{do}}\frac{L'_d}{L_d}\right)s\right]\left[1+\left(\tau''_{\mathrm{do}}\frac{L''_d}{L'_d}\right)s\right]}{\left(1+\tau'_{\mathrm{do}}s\right)\left(1+\tau''_{\mathrm{do}}s\right)} \tag{8.23}$$

โดยที่

L_l	ความเหนี่ยวนำรั่วของขดลวดสเตเตอร์
L_{ad}	ความเหนี่ยวนำเชื่อมโยงระหว่างขดลวดสเตเตอร์และโรเตอร์
L_{f1d}	ความเหนี่ยวนำเชื่อมโยงระหว่างขดลวดสนามและขดลวดหน่วง
L_{1d}	ความเหนี่ยวนำรั่วของขดลวดหน่วง
R_{1d}	ความต้านทานของขดลวดหน่วง
L_{fd}	ความเหนี่ยวนำรั่วของขดลวดสนาม
R_{fd}	ความต้านทานของขดลวดสนาม
L_d	ความเหนี่ยวนำรั่วซิงโครนัส
L'_d	ความเหนี่ยวนำในสภาวะชั่วครู่
L''_d	ความเหนี่ยวนำในสภาวะชั่วแวบ
τ'_{do}	ค่าคงตัวเวลาเปิดในขณะเปิดวงจรในสภาวะชั่วครู่
τ''_{do}	ค่าคงตัวเวลาเปิดในขณะเปิดวงจรในสภาวะชั่วแวบ

โดยการใช้ GA เข้าช่วยเพื่อปรับแต่งพารามิเตอร์ของฟังก์ชันถ่ายโอน ในที่นี้จะใช้ผลการทดสอบในโดเมนความถี่ ซึ่งเป็นผลการทดสอบเครื่องกำเนิดไฟฟ้าซิงโครนัสตามมาตรฐาน IEEE [22-25] มาใช้เป็นตัวอ้างอิง

เนื่องจาก ผลตอบสนองทางความถี่อยู่ในรูปแผนภาพโบด (bode diagram) [8] ซึ่งมีทั้งขนาดและเฟส การประมาณค่าจะต้องเทียบกับกราฟทั้ง 2 เส้นพร้อมกัน โดยใช้เทคนิคการถ่วงน้ำหนักของฟังก์ชันวัตถุประสงค์ด้วยตัวประกอบ W_{mag} และ W_{phase} ในลักษณะเดียวกับที่นำเสนอในหัวข้อ 8.2 จะได้ว่า

$$\mathrm{SSE} = \sum_{i=1}^{N}\left(\frac{y_{\mathrm{measured}} - y_{\mathrm{simulated}}}{y_{\mathrm{measured,max}}}\right)^2 \tag{8.24}$$

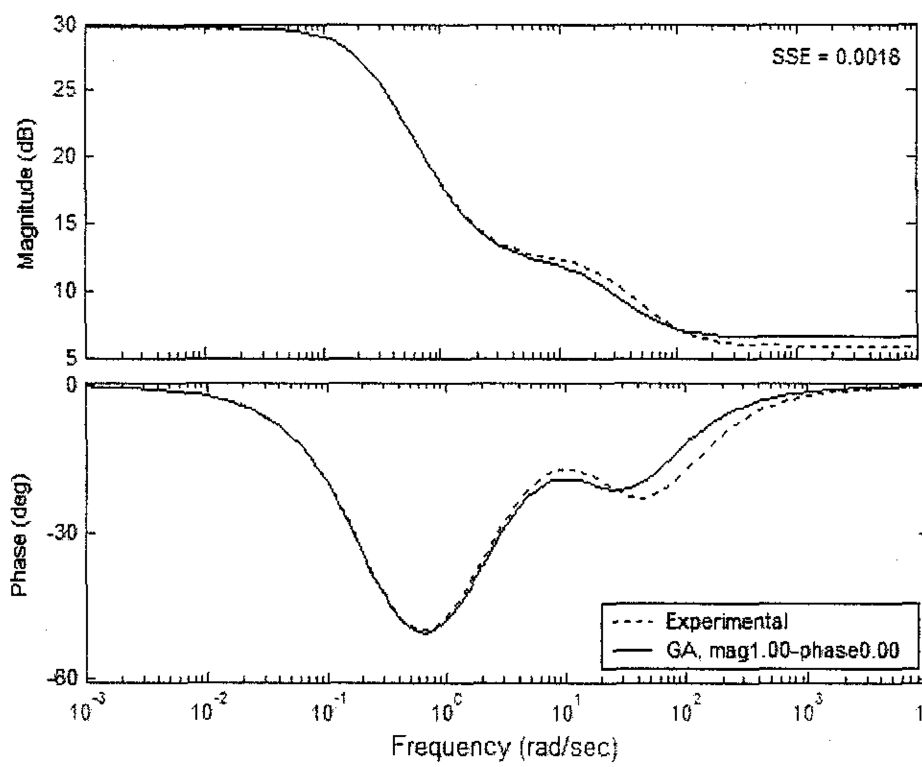
$$f_{\text{bi-obj}} = (W_{\text{mag}})\cdot \text{SSE}_{\text{mag}} + (W_{\text{phase}})\cdot \text{SSE}_{\text{phase}} \tag{8.25}$$

$$(W_{\text{mag}}) + (W_{\text{phase}}) = 1.0 \tag{8.26}$$

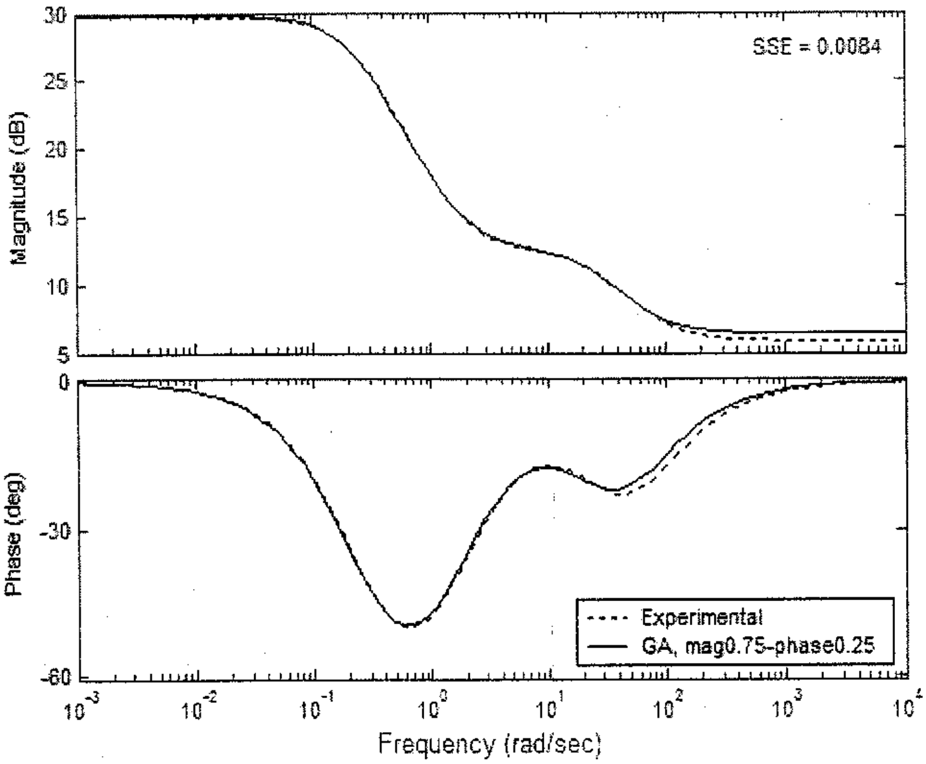
จะได้ผลการทดสอบดังแสดงในตารางที่ 8.3 เมื่อนำมาวาดกราฟการตอบสนองทางความถี่ของฟังก์ชันถ่ายโอนเปรียบเทียบกับข้อมูลทดสอบมาตรฐาน จะได้ดังกราฟในรูปที่ 8.32 – 8.36

ตารางที่ 8.3 ผลการเปรียบเทียบพารามิเตอร์ของกรณีทดสอบต่าง ๆ [5]

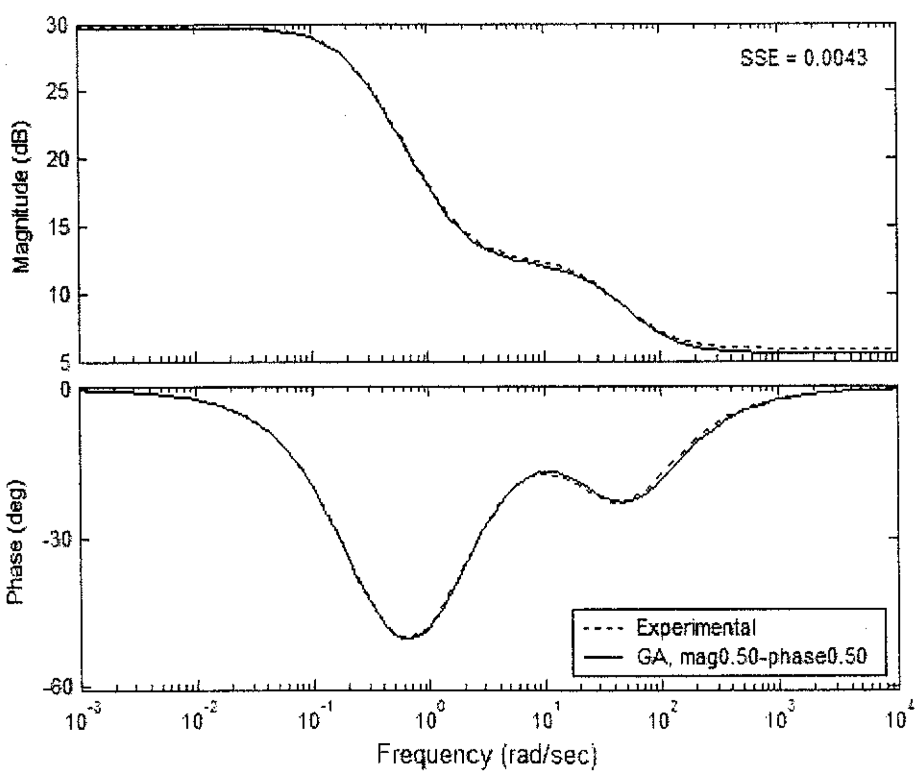
Parameters / Methods	L_d (p.u.)	L'_d (p.u.)	L''_d (p.u.)	τ'_{do} (s)	τ''_{do} (s)
Experimental	1.9700	0.2700	0.1270	4.3000	0.0310
GA, mag1.00-phase0.00	2.1273	0.2845	0.1477	4.3111	0.0427
GA, mag0.75-phase0.25	2.0991	0.2917	0.1428	4.4120	0.0355
GA, mag0.50-phase0.50	1.8957	0.2530	0.1183	4.3265	0.0284
GA, mag0.25-phase0.75	1.9803	0.2785	0.1295	4.4444	0.0315
GA, mag0.00-phase0.10	2.2632	0.3082	0.1449	4.2615	0.0327



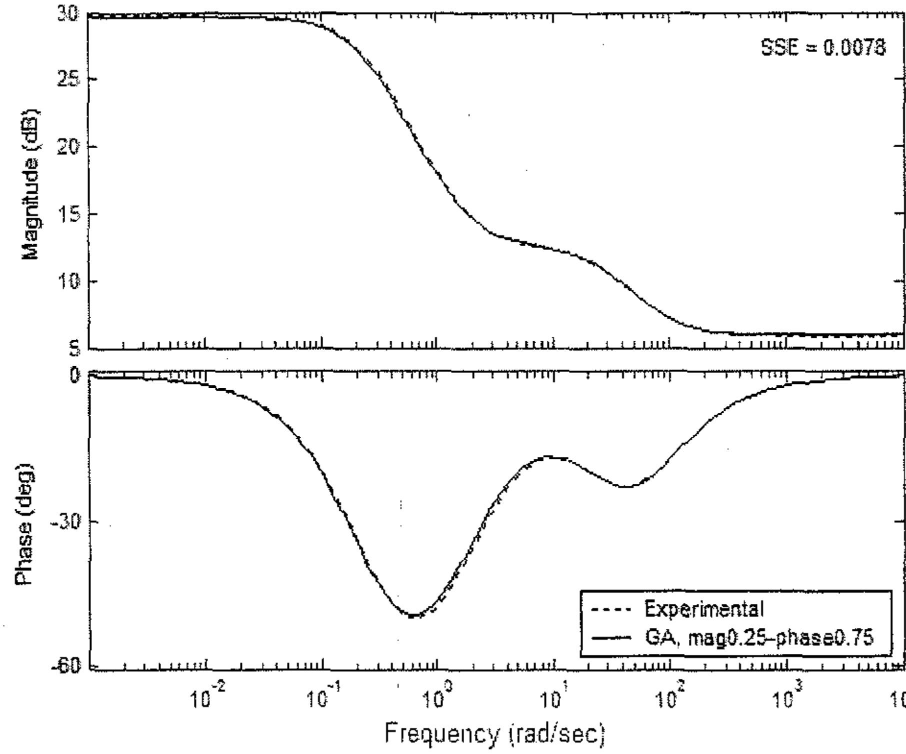
รูปที่ 8.32 ผลตอบสนองทางความถี่ กรณี W_{mag} = 1.0 และ W_{phase} = 0.0



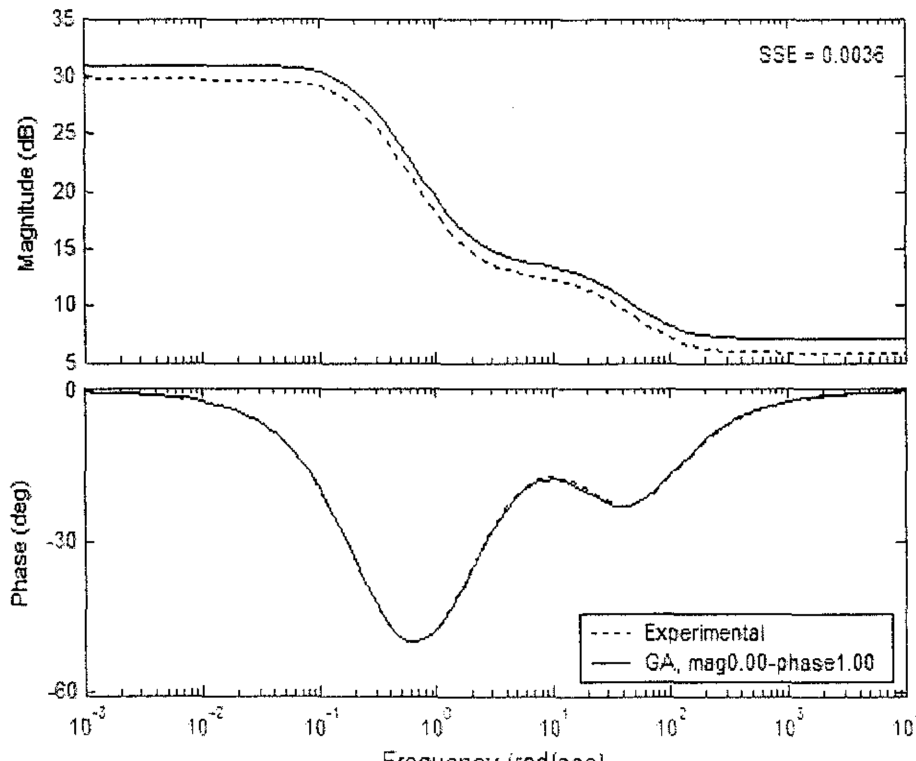
รูปที่ 8.33 ผลตอบสนองทางความถี่ กรณี $W_{mag} = 0.75$ และ $W_{phase} = 0.25$



รูปที่ 8.34 ผลตอบสนองทางความถี่ กรณี $W_{mag} = 0.5$ และ $W_{phase} = 0.5$



รูปที่ 8.35 ผลตอบสนองทางความถี่ กรณี $W_{mag} = 0.25$ และ $W_{phase} = 0.75$



รูปที่ 8.35 ผลตอบสนองทางความถี่ กรณี $W_{mag} = 0.0$ และ $W_{phase} = 1.0$

8.7 สรุป

บทนี้ นำเสนอรูปแบบการสร้างปัญหาค่าเหมาะที่สุดสำหรับการออกแบบในงานทางด้านวิศวกรรมไฟฟ้า เน้นไปที่การประยุกต์กับเครื่องจักรกลไฟฟ้าประเภทต่าง ๆ เพื่อให้เข้าใจถึงขั้นตอนการกำหนดตัวแปร ฟังก์ชันวัตถุประสงค์ การเลือกเงื่อนไขบังคับ ตลอดจนการเลือกใช้เทคนิคการแก้ปัญหา จะสังเกตได้อย่างชัดเจนว่า งานทางด้านการปรับแต่งพารามิเตอร์นี้ใช้เทคนิคการคำนวณชาญฉลาดทั้งหมดในทุกขั้นตอน เหตุผลหลักในการไม่เลือกใช้งานกำหนดการทางคณิตศาสตร์สืบเนื่องมาจากความยุ่งยากในการคำนวณค่าเกรเดียนต์ ถึงแม้ว่าจะมีเทคนิคการคำนวณเกรเดียนต์เชิงตัวเลข การคำนวณดังกล่าวอาจจะทำให้กระบวนการค้นหาผลเฉลยไม่มีเสถียรภาพ และโดยทั่วไปการคำนวณเกรเดียนต์เชิงตัวเลขของปัญหาที่มีตัวแปรหลายตัวคำนวณได้ช้า ยิ่งไปกว่านั้น ผู้เขียนได้รวบรวมงานวิจัยที่ใช้เทคนิคการหาค่าเหมาะที่สุดกับงานทางด้านเครื่องจักรกลไฟฟ้า เพื่อให้ผู้อ่านเข้าใจกระบวนการเชื่อมโยงระหว่างเทคนิคการหาค่าเหมาะที่สุดเพื่อใช้งานทางด้านวิศวกรรมไฟฟ้า ผู้เขียนหวังเป็นอย่างยิ่งว่า นักศึกษาและผู้สนใจจะได้แนวความคิดของการประยุกต์เทคนิคการหาค่าเหมาะที่สุดไปใช้ในการแก้ปัญหาในงานวิจัยทางวิศวกรรมได้

8.8 เอกสารอ้างอิง

[1] P.C. SEN, **Thyristor DC Drives**, Krieger, 1991

[2] L.V. FAUSETT, **Applied Numerical Analysis using MATLAB®**, Prentice-Hall, 1999

[3] P. Vas, **Parameter Estimation, Condition Monitoring, and Diagnosis of Electrical Machines**, Oxford University Press, 1993

[4] S.S. Rao, **Engineering optimization: Theory and practices**, Wiley-Interscience, 1996

[5] P. Pao-la-or, T. Kulworawanichpong & A. Oonsivilai, *Frequency Domain Parameter Estimation of a Synchronous Generator Using Bi-objective Genetic Algorithms*, **The 7Th WSEAS Int. Conf. on Power Systems (PE'07)**, Beijing, 12 – 14 September 2007

[6] ธนัดชัย กุลวรวานิชพงษ์, **การทำงานที่เหมาะสมในระบบไฟฟ้ากำลังโดยใช้การตัดสินใจแบบฟัซซี**, วิทยานิพนธ์มหาบัณฑิต, จุฬาลงกรณ์มหาวิทยาลัย, พ.ศ. 2543

[7] S.J. Chapman, **Electric Machinery Fundamentals**, McGraw-Hill, 2004

[8] G.F. Franklin, J.D. Powell & M. Workman, **Digital Control of Dynamic Systems**, Addison-Wesley, 1998

[9] C.R. Wylie & L.C. Barrett, **Advanced Engineering Mathematics**, McGraw-Hill, 1995

[10] นิรันดร์ แนวเงินดี, ธนัดชัย กุลวรวานิชพงษ์, *แบบจำลองและการจำลองผลมอเตอร์แบบแยกเฟสโดยพิจารณาการทำงานของสวิตช์แรงเหวี่ยงหนีศูนย์*, **การประชุมวิชาการวิศวกรรมไฟฟ้าแห่งประเทศไทย ครั้งที่ 28 (EECON28)**, ภูเก็ต, 20 – 21 ตุลาคม 2548

[11] P.C. Krause, **Analysis of Electric Machines**, McGraw-Hill, 1987

[12] D.W. Novotny & T.A. Lipo, **Vector Control and Dynamic of AC Drives**, Oxford University Press, 1996

[13] N. Naewngerndee, C. Sukchareon & T. Kulworawanichpong, *Optimizing voltage-frequency control strategy for single-phase induction motor drives*, **The 5th WSEAS Int. Conf. on Applications of Electrical Engineering (AEE'06)**, Prague, Czech Republic, 12 – 14 March 2006, pp. 84 – 89

[14] S. Raweekul, T. Kulworawanichpong, S. Sujitjorn, *Modelling and Simulation of Multiple Single - Phase Induction Motors in Parallel Operation*, **The 8th WSEAS Int. Conf. on Automatic Control, Modeling & Simulation (ACMOS '06)**, Prague, Czech Republic, 12-14 March 2006, pp. 195-200

[15] N. Naewngerndee, C. Sukcharoen & T. Kulworawanichpong, *Simulation of Single-Phase Induction Motor Drives with Non-sinusoidal Power Supply,* **The WSEAS Transactions on Systems**, Issue 5, Vol 5, pp. 1029-1034

[16] S. Raweekul, T. Kulworawanichpong, & S. Sujitjorn, *Parallel -Connected Single -Phase Induction Motor:Modelling and Simulation*, **The WSEAS Transactions on Circuits and Systems**, Issue 3, Vol 5, pp. 377-384

[17] D. Puangdownreong, N. Naewngerndee & T. Kulworawanichpong, *Application of Intelligent Search Techniques to Identify Single-Phase Induction Motor's Parameters,* **The 7Th WSEAS Int. Conf. on Simulation, Modeling and Optimization (SMO'07)**, Beijing, 12 – 14 September 2007

[18] R. KRISHNAN, **Electric motor drives: Modeling, analysis and control**, Prentice-Hall, 2001

[19] T. Kulworawanichpong, K-L. Areerak, K-N. Areerak, D. Pao-la-or, P. Puangdownreong, S. Sujitjorn, *Dynamic parameter identification of induction motors using intelligent search techniques*, **IASTED International Conference on Modelling, Identification and Control (MIC2005)**, Innsbruck, Austria, 16 – 18 February 2005

[20] P.M. Anderson & A.A. Fouad, **Power System Control and Stability**, Wiley, 2003

[21] P. Kundur, **Power System Stability and Control**, McGraw-Hill, 1994

[22] IEEE Power Engineering Society, *IEEE Std 115A-1987, IEEE Standard Procedures for obtaining synchronous machine parameters by standstill frequency response testing,* **The Institute of Electrical and Electronics Engineers**, 1987

[23] IEEE Power Engineering Society, *IEEE Std 115-1995, IEEE Guide: Test procedures for synchronous machines,* **The Institute of Electrical and Electronics Engineers**, 1996

[24] IEEE Power Engineering Society, *IEEE Std 1110-2002, IEEE Guide for synchronous generator modeling practices and applications in power system stability analyses,* **The Institute of Electrical and Electronics Engineers**, 2003

[25] I. Kamwa, P. Viarouge, H. Le-Huy, J. Dickinson, *A frequency-domain maximum likelihood estimation of synchronous machine high-order models using SSFR test data*, **IEEE Transactions on Energy Conversion**, Vol.7, No.3, 1992, pp. 525-536

บทที่ 9 การหาค่าเหมาะที่สุดในระบบไฟฟ้ากำลัง
(Power System Optimization)

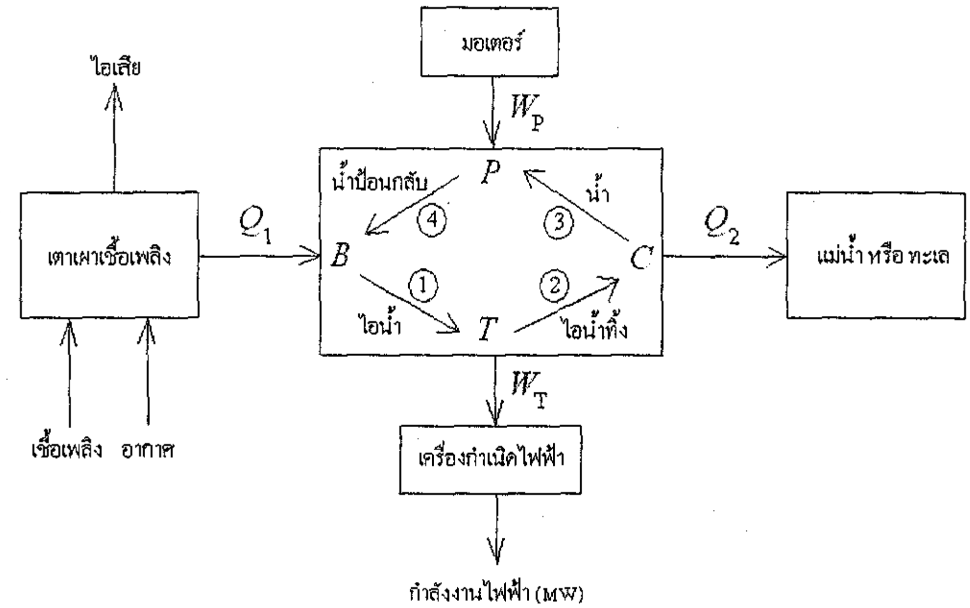
"a kindergarten student must obey his parents
a primary student must follow his teachers
a secondary student always listen to his friends
a bachelor degree student always trust his textbooks
a master degree student is very self-confident
a PhD student believes nothing"

9.1 เกริ่นนำ

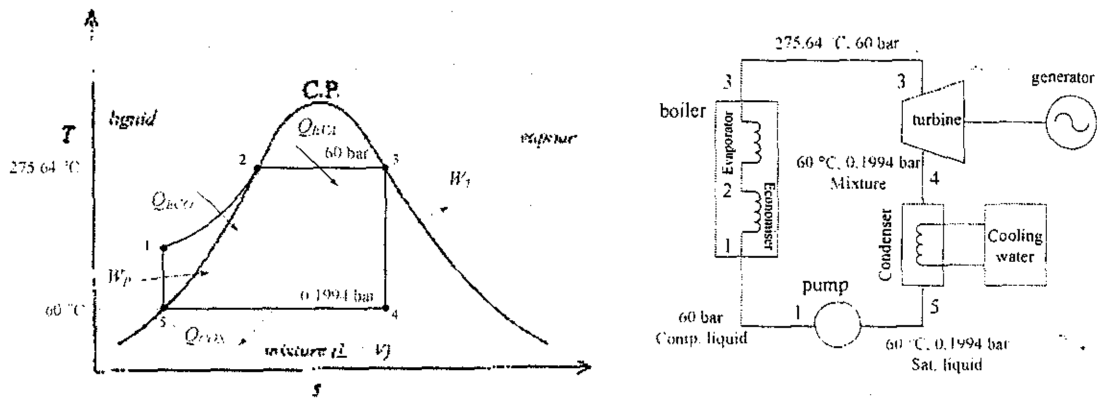
งานวิจัยทางด้านการวิเคราะห์ระบบไฟฟ้ากำลังมีความก้าวหน้าอย่างมาก การคำนวณด้วยเทคนิคเชิงตัวเลข และการหาค่าเหมาะที่สุดถูกนำมาประยุกต์ใช้ไม่น้อยกว่าครึ่งศตวรรษ อย่างไรก็ตาม ในช่วงเริ่มต้น ปัญหาด้านสมรรถนะของอุปกรณ์คำนวณทำให้เทคนิคทางคณิตศาสตร์ที่พึ่งพาระเบียบวิธีเชิงตัวเลขทำได้ยาก จนกระทั่งปัจจุบัน เมื่อมีคอมพิวเตอร์สมรรถนะสูงขึ้น เช่น เทคโนโลยีเพนเทียมโพรเซสเซอร์ ทำให้การแก้ปัญหาที่ซับซ้อนและมีขนาดใหญ่ทำได้ไม่ยากนัก และใช้เวลาคำนวณไม่นานเกินไป ในบทนี้ ผู้เขียนได้เรียบเรียงองค์ความรู้ทางด้านการวิเคราะห์ระบบไฟฟ้ากำลังเอาไว้โดยสังเขป ได้แก่ การจ่ายโหลดอย่างประหยัด (economic load dispatch) การไหลของกำลังไฟฟ้าเหมาะที่สุด (optimal power flow) การพยากรณ์โหลดทางไฟฟ้า (electric load forecasting) และการป้องกันระบบไฟฟ้ากำลัง (power system protection) ดังรายละเอียดต่อไปนี้

9.2 ปัญหาการจ่ายโหลดอย่างประหยัดของโรงไฟฟ้าพลังความร้อน

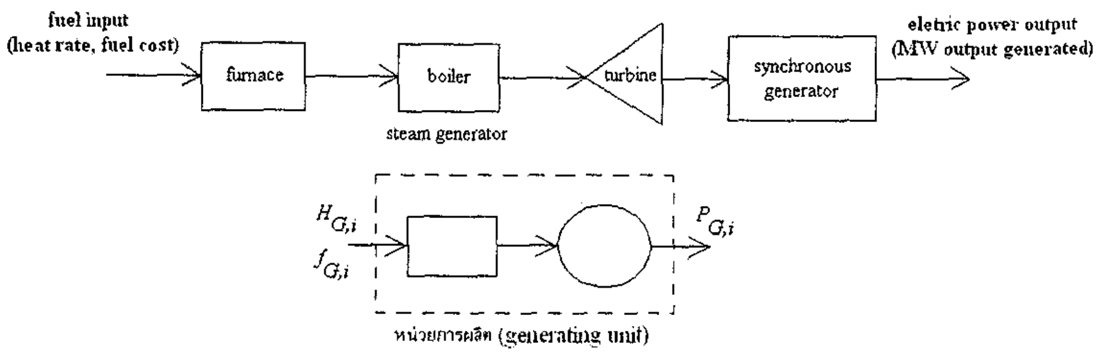
ระบบไฟฟ้ากำลังในปัจจุบันมีขนาดใหญ่และซับซ้อน ประกอบไปด้วยโรงไฟฟ้าจำนวนมาก ซึ่งโรงไฟฟ้าแต่ละชนิดมีประสิทธิภาพและต้นทุนการผลิตกำลังไฟฟ้าที่แตกต่างกัน การจ่ายโหลดโดยทั่วไป อาศัยหลักการสมดุลของกำลังงาน กล่าวคือ กำลังผลิตเท่ากับความต้องไฟฟ้า (generation = demand) ในกรณีของระบบไฟฟ้ากำลังที่ไม่ใช่ระบบในอุดมคติ การสูญเสียกำลังงานไฟฟ้าในรูปแบบต่างๆ เกิดขึ้นได้ในทุก ๆ ส่วน เช่น ในระบบผลิตของโรงไฟฟ้าเอง ในระบบส่งจ่ายซึ่งเกิดการสูญเสียกำลังงานในสายส่งต่างๆ ดังนั้น การวางแผนการผลิตให้กับโรงไฟฟ้าแต่ละโรงเป็นปัญหาที่สำคัญเพราะจะส่งผลกระทบต่อต้นทุนการผลิตโดยรวมของทั้งระบบ ซึ่งจะใช้ในการกำหนดค่าไฟฟ้านั่นเอง หรือถ้ามองในรูปของผลกำไร การผลิตกำลังไฟฟ้าที่ต้นทุนต่ำ จะทำให้ได้กำไรจากการขายไฟสูงสุด เมื่อให้อัตราค่าไฟฟ้ามีค่าคงที่ การวางแผนในลักษณะนี้ จำเป็นที่จะต้องทราบคุณลักษณะสมบัติของต้นทุนการผลิตของโรงไฟฟ้าแต่ละโรง (ในหน่วยของเงิน) ต่อหน่วย MW ที่ผลิตขึ้น แบบจำลองนี้จะอาศัยค่าอัตราความร้อนสุทธิของวัฏจักร (net cycle heat rate: H) ของโรงไฟฟ้า [1] โดยแบบจำลองดังกล่าวมีรายละเอียดโดยย่อดังนี้



รูปที่ 9.1 แผนภาพองค์ประกอบของโรงไฟฟ้าพลังความร้อน



รูปที่ 9.2 แผนภาพวัฏจักรแรนคินและกระบวนการของโรงไฟฟ้าพลังความร้อน



รูปที่ 9.3 แผนภาพการแปลงผันกำลังงานของกระบวนการที่เกิดขึ้นในโรงไฟฟ้าพลังความร้อน

รูปที่ 9.1 แสดงองค์ประกอบต่าง ๆ ที่จำเป็นของโรงไฟฟ้าพลังความร้อน องค์ประกอบย่อยแต่ละส่วนสามารถขยายให้เห็นจุดการทำงานด้วยเส้นโค้งเอนโทรปี-อุณหภูมิดังรูปที่ 9.2 เนื่องจากการศึกษาปัญหาการจ่ายโหลดอย่างประหยัดของโรงจักรไฟฟ้านั้น จะพิจารณาเฉพาะต้นทุนขอ

โรงไฟฟ้า ซึ่งประกอบด้วยค่าจ้างบุคลากร ค่าเชื้อเพลิง ค่าบำรุงรักษา และอื่น ๆ โดยสิ่งที่ได้จากโรงไฟฟ้าจะพิจารณาเฉพาะกำลังงานไฟฟ้าที่ผลิตขึ้นเท่านั้น กระบวนการหรือวัฏจักรทางความร้อนภายในโรงไฟฟ้าจะไม่นำรายละเอียดทั้งหมดมาพิจารณา ในที่นี้จะนำเสนอเป็นแผนภาพอย่างง่ายในรูปของการแปลงผันกำลังงานจากพลังงานเคมีที่สะสมอยู่ในเชื้อเพลิงมาเป็นพลังงานความร้อน และได้พลังงานไฟฟ้าในที่สุด ดังแสดงในรูปที่ 9.3 กระบวนการตรงกลางจะถูกประมาณด้วยฟังก์ชันคณิตศาสตร์ที่ไม่ซับซ้อนมากนัก เช่น สมการเส้นตรง สมการเส้นตรงแบบไม่ต่อเนื่อง สมการกำลังสอง สมการกำลังสาม หรือสมการไม่เชิงเส้นบางรูปแบบ เป็นต้น โดยปกติความสัมพันธ์ระหว่างเชื้อเพลิงอินพุตกับกำลังงานเอาต์พุตเขียนให้อยู่ในรูปสมการอย่างง่าย [2] ดังนี้

$$f_{G,i} = \alpha_i + \beta_i P_{G,i} + \gamma_i P_{G,i}^2 \tag{9.1}$$

โดยที่

$f_{G,i}$ อัตราค่าความร้อนสุทธิของวัฏจักรของหน่วยผลิตที่ i

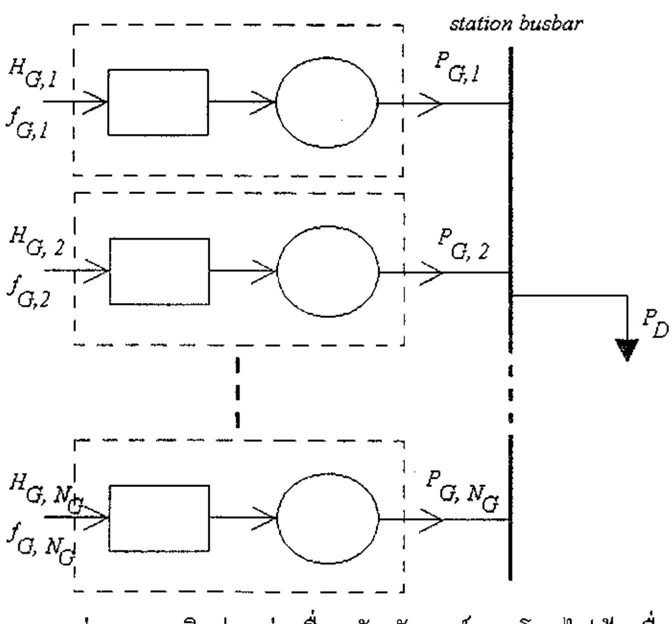
$P_{G,i}$ กำลังงานไฟฟ้าเอาต์พุตของหน่วยผลิตที่ i

$\alpha_i, \beta_i, \gamma_i$ สัมประสิทธิ์ของฟังก์ชันค่าความร้อน (heat-rate coefficients)

สำหรับโรงไฟฟ้าหนึ่งโรง สามารถมีหน่วยการผลิตได้หลายหน่วยและแต่ละหน่วยอาจจะใช้เชื้อเพลิงต่างชนิดกัน ดังนั้น ต้นทุนการผลิตโดยรวม (total generation cost) ของโรงไฟฟ้ามีค่าเป็น

$$F_T = \sum_{i=1}^{N_G} f_{G,i} = \sum_{i=1}^{N_G} \left(\alpha_i + \beta_i P_{G,i} + \gamma_i P_{G,i}^2 \right) \tag{9.2}$$

เมื่อ N_G คือ จำนวนหน่วยการผลิตย่อยทั้งหมดของโรงไฟฟ้า



รูปที่ 9.4 แผนภาพหน่วยการผลิตย่อยต่อเชื่อมกับบัสบาร์ของโรงไฟฟ้าเพื่อจ่ายโหลด

จะพบว่า แบบจำลองที่ใช้นี้เป็นเพียงรูปแบบหนึ่งเท่านั้น ที่ใช้การนำเสนอด้วยฟังก์ชันกำลังสอง อาจจะนำฟังก์ชันไม่เชิงเส้นอื่น ๆ แทนได้ เมื่อพิจารณาโรงไฟฟ้าพลังความร้อนแห่งหนึ่งที่ประกอบด้วยหน่วยการผลิตย่อย N_G หน่วย โดยหน่วยการผลิตทั้งหมด ถูกต่อเชื่อมเข้าไปที่บัสบาร์ของโรงไฟฟ้าเพื่อทำการจ่ายโหลด โดยไม่พิจารณาผลของระบบส่งจ่าย ดังนั้น จะสมมติให้โหลดทั้งหมดที่โรงไฟฟ้านี้จ่ายต่ออยู่ที่บัสบาร์ของโรงไฟฟ้า ดังรูปที่ 9.4 และจะได้เงื่อนไขการทำงานของโรงไฟฟ้าดังนี้

$$\text{Power Generation} = \text{Power Demand} \qquad \text{(neglecting losses)}$$

$$\sum_{i=1}^{N_G} P_{G,i} - P_D = 0$$

โดยมีเป้าหมายหรือที่เรียกว่า วัตถุประสงค์ (objective) เป็นต้นทุนการผลิต $F_T = f(P_{G,i})$ มีค่าต่ำที่สุด จะได้รูปแบบของปัญหาเป็น

$$\begin{aligned} &\text{Minimise} \qquad F_T = f(P_{G,i}) = \sum_{i=1}^{N_G} f_{G,i}(P_{G,i}) \\ &\text{subject to} \qquad \sum_{i=1}^{N_G} P_{G,i} - P_D = 0 \end{aligned} \tag{9.3}$$

พิจารณาโรงไฟฟ้าพลังความร้อนตัวอย่างแห่งหนึ่งประกอบไปด้วยหน่วยการผลิตย่อย 3 ชุด มีฟังก์ชันค่าเชื้อเพลิงเป็นชนิดไม่เชิงเส้นและไม่ราบเรียบ (non-smooth fuel cost) [3-5] ดังแสดงในตาราง ต่อไปนี้

ตารางที่ 9.1 สัมประสิทธิ์ของฟังก์ชันค่าเชื้อเพลิง

Unit	a	b	c	d	e	min MW	max MW
1	240	6.5	0.008	182	0.082	50	200
2	220	6.1	0.005	191	0.057	30	250
3	240	6.7	0.009	208	0.059	40	250

โดยที่

$$f(P_i) = a_i + b_i P_i + c_i P_i^2 + d_i \left| \sin e_i \left(P_i^{min} - P_i \right) \right| \qquad \text{฿/h} \tag{9.4}$$

กำหนดให้โรงไฟฟ้าแห่งนี้ได้รับการจัดสรรโหลดรายวันดังแสดงในตารางต่อไปนี้

ตารางที่ 9.2 โหลดรายวันของโรงไฟฟ้าตัวอย่างแบ่งเป็น 4 ช่วงเวลา ได้แก่ r s u และ v

Hour (j)	0.00 – 6.00 (r)	6.00 – 12.00 (s)	12.00 – 18.00 (u)	18.00 – 24.00 (v)
MW	247 MW	630 MW	454 MW	528 MW

จงแก้ปัญหาการจ่ายโหลดอย่างประหยัด โดยให้หน่วยการผลิตทั้งหมดออนไลน์ตลอดเวลา เพื่อให้ได้ต้นทุนการผลิตต่ำที่สุด

จะพบว่า ปัญหานี้ประกอบไปด้วยตัวแปรทั้งสิ้น 12 ตัว ได้แก่

$$P_{1,r};\ P_{1,s};\ P_{1,u};\ P_{1,v};\ P_{2,r};\ P_{2,s};\ P_{2,u};\ P_{2,v};\ P_{3,r};\ P_{3,s};\ P_{3,u};\ P_{3,v}$$

จะได้ปัญหาค่าเหมาะที่สุดดังต่อไปนี้

$$\text{Minimize} \quad F_T = \sum_{j=\{r,s,u,v\}} \sum_{i=1}^{3} \left\{ a_i + b_i P_{i,j} + c_i P_{i,j}^2 + d_i \left| sin\ e_i \left(P_i^{min} - P_{i,j} \right) \right| \right\}$$

$$\text{Subject to} \quad \sum_{i=1}^{3} P_{i,r} = 247$$

$$\sum_{i=1}^{3} P_{i,s} = 630$$

$$\sum_{i=1}^{3} P_{i,u} = 454$$

$$\sum_{i=1}^{3} P_{i,v} = 528$$

$$50 \le P_{1,j} \le 200; \qquad \forall j \in \{r,s,u,v\}$$

$$30 \le P_{2,j} \le 250; \qquad \forall j \in \{r,s,u,v\}$$

$$40 \le P_{3,j} \le 250; \qquad \forall j \in \{r,s,u,v\}$$

นำมาสร้างเป็นฟังก์ชันวัตถุประสงค์ได้ดังนี้

<u>โปรแกรม SCILAB สำหรับตัวอย่างเรื่องการจ่ายโหลดอย่างประหยัด</u>

// SCILAB source program

```
// กำหนดฟังก์ชันวัตถุประสงค์                    eld01.sci
=========================
          eld01.sci
=========================
function fobj = eld01(x)
p1r=x(1);
p1s=x(2);
p1u=x(3);
p1v=x(4);
p2r=x(5);
p2s=x(6);
p2u=x(7);
p2v=x(8);
p3r=x(9);
p3s=x(10);
p3u=x(11);
p3v=x(12);
```

```
rho=1e6;
a=[ 1   240   6.5   0.008  182   0.082  50    200
    2   220   6.1   0.005  191   0.057  30    250
    3   240   6.7   0.009  208   0.059  40    250];
fcost=0;
for u=1:4
    fc=0;
    for v=1:3
        fc=fc+a(v,2)+a(v,3)*x(3*(v-1)+u)+a(v,3)*x(3*(v-1)+u)^2+...
            ...+a(v,4)*abs(sin(a(v,4)*(a(v,5)-x(3*(v-1)+u))));
    end
    fcost=fcost+6*fc;
end
Cst_r=p1r+p2r+p3r-247;
Cst_s=p1s+p2s+p3s-630;
Cst_u=p1u+p2u+p3u-454;
Cst_v=p1v+p2v+p3v-528;
C(1)=min(p1r-50,0);C(2)=min(p1s-50,0);C(3)=min(p1u-50,0);C(4)=min(p1v-50,0);
C(5)=min(200-p1r,0);C(6)=min(200-p1s,0);C(7)=min(200-p1u,0);C(8)=min(200-p1v,0);
C(9)=min(p2r-30,0);C(10)=min(p2s-30,0);C(11)=min(p2u-30,0);C(12)=min(p2v-30,0);
C(13)=min(250-p2r,0);C(14)=min(250-p2s,0);C(15)=min(250-p2u,0);
C(16)=min(250-p2v,0);C(17)=min(p3r-40,0);C(18)=min(p3s-40,0);
C(19)=min(p3u-40,0);C(20)=min(p3v-40,0);C(21)=min(250-p3r,0);
C(22)=min(250-p3s,0);C(23)=min(250-p3u,0);C(24)=min(250-p3v,0);
CC=0;
for h=1:24
    CC=CC+C(h)^2;
end
fobj=fcost+rho*(Cst_r^2+Cst_s^2+Cst_u^2+Cst_v^2+CC);
endfunction
```

ในที่นี้ จะทำการแก้ปัญหาโดยใช้ระเบียบวิธีชันที่สุด โดยดัดแปลงชุดคำสั่งที่ได้นำเสนอไว้ในบทที่ 3 ทำการแก้ไขด้วยการเพิ่มกราฟแสดงผลระหว่างการคำนวณ และใช้เทคนิคการคำนวณเกรเดียนต์ของฟังก์ชันวัตถุประสงค์เชิงตัวเลข ดังนี้

```
// ฟังก์ชันหาค่าเหมาะที่สุดด้วยระเบียบวิธีชันที่สุด                    steepestx.sci
========================
          steepestx.sci
========================
function [xmin,fmin,k]=steepestx(fobj,x0,opt)
f0=fobj(x0);
g0=fgradnum_n(fobj,x0);
ferr=norm(g0);
k=0;
if opt(3)==1 then
   disp([k x0' f0 ferr]);
end
plot2d(0,f0,style=[-9],logflag='nl')
xset('font',2,3);
xtitle('Convergence curve (best solution)',' ','Objective');
Ferr=%inf;
while Ferr>opt(2)
   k = k+1;
   p=-g0;
   lambda=lsearch(fobj,x0,p);
   xmin=x0+lambda*p;
   fmin=fobj(xmin);
   gmin=fgradnum_n(fobj,xmin);
   ferr=norm(gmin);
   if opt(3)==1 then
      disp([k xmin' fmin ferr]);
   end
   plot2d(k,fmin,style=[-9],logflag='nl')
   plot2d([k-1 k],[f0 fmin],logflag='nl')
   Ferr = abs((fmin - f0)/f0);
   x0=xmin;f0=fmin;g0=gmin;
   if k>opt(1) then
       break;
   end
```

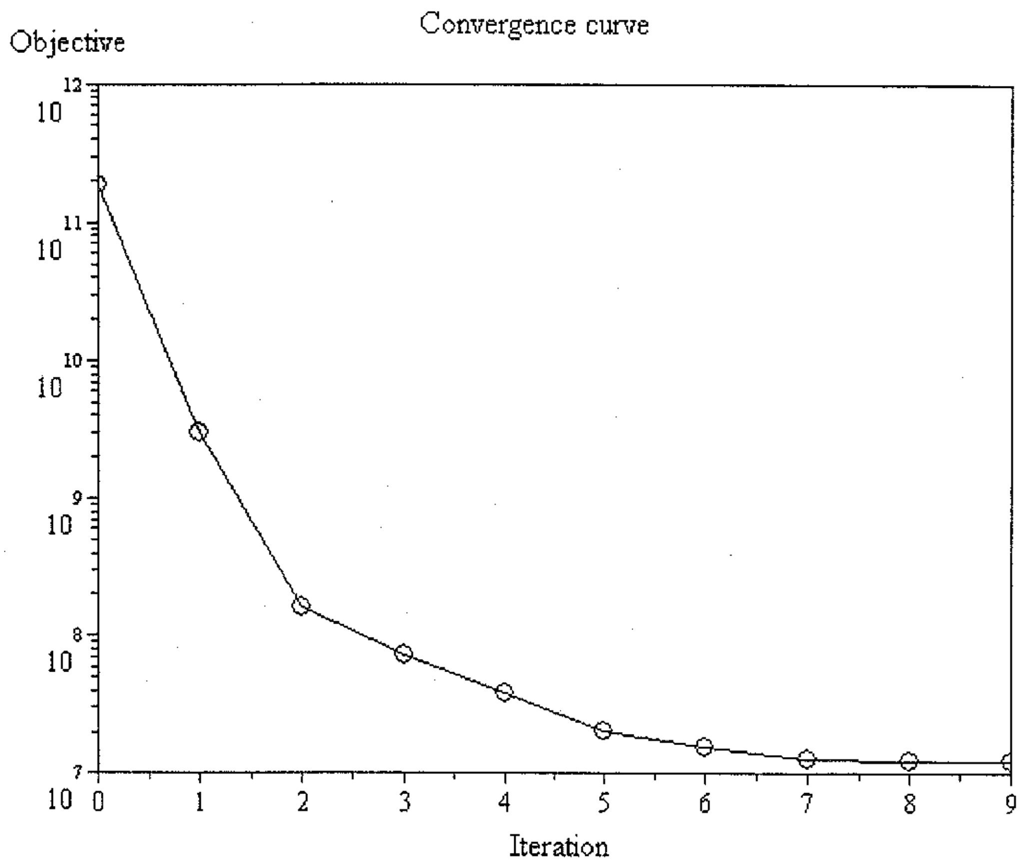
```
end
endfunction
```

```
// ฟังก์ชันคำนวณเกรเดียนต์ของฟังก์ชันวัตถุประสงค์เชิงตัวเลข          fgradnum_n.sci
========================
          fgradnum_n.sci
========================
function g = fgradnum_n(fobj,x)
n = length(x);
dx=1e-6;
for k=1:n
   x(k,1) = x(k,1) - dx;
   fn = fobj(x);
   x(k,1) = x(k,1) + 2*dx;
   fp = fobj(x);
   g(k) = (fp - fn)/(2*dx);
   x(k,1) = x(k,1) - dx;
end
endfunction
```

รันโปรแกรมเพื่อทดสอบผลจะได้ว่า

```
// ผลการรันโปรแกรม
-->[xmin,fmin,k]=steepestx(eld01,100*ones(12,1),[100 0.01 0]);
-->xmin
 xmin  =
   82.332924
   200.07456
   151.32862
   175.95971
   82.334015
   214.90284
   151.2982
   175.99263
   82.332379
   214.89882
```

```
   151.36001
   176.03003
-->fmin
 fmin  =
   12239848.
```

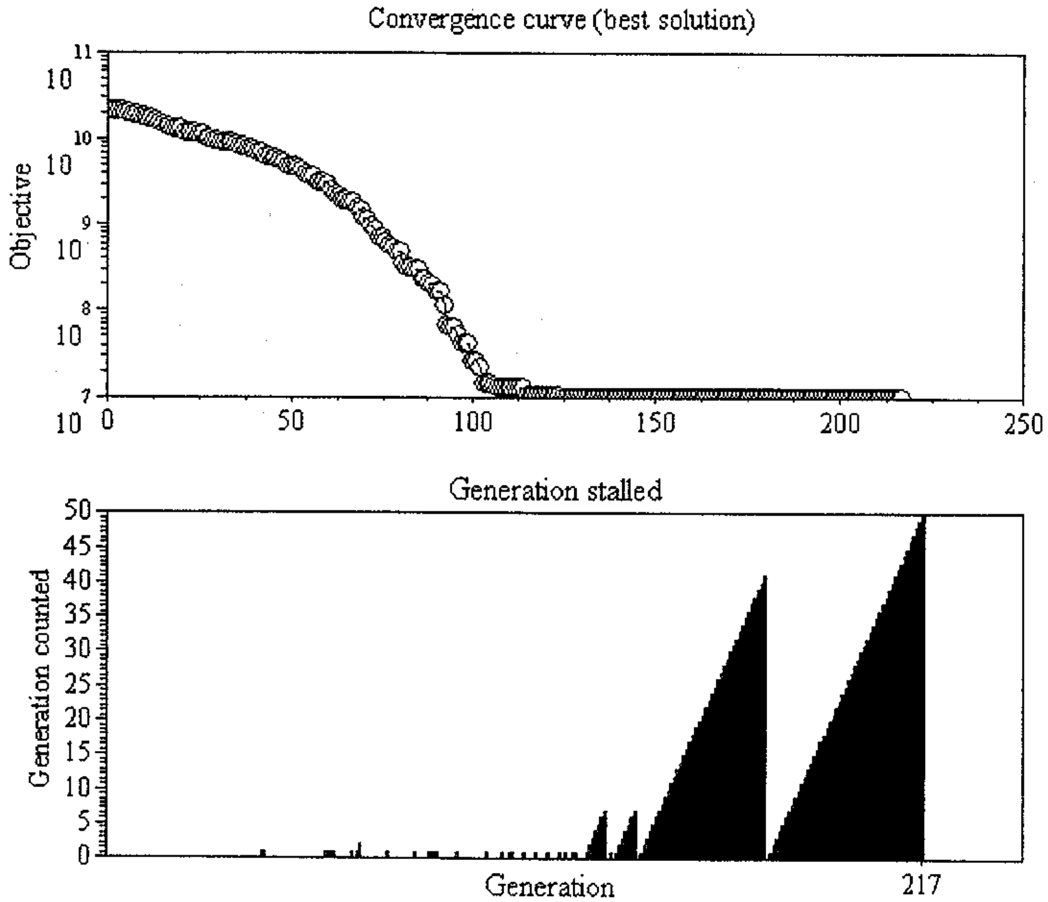


รูปที่ 9.5 แผนภาพการลู่เข้าของปัญหาการจ่ายโหลดอย่างประหยัดโดยใช้ระเบียบวิธีชันที่สุด

ทดลองแก้ปัญหาโดยใช้กำหนดการวิวัฒนาการหรือ EP จะได้ว่า

```
// ผลการรันโปรแกรม
-->xlimp1 = [50 200]; xlimp2 = [30 250]; xlimp3 = [40 250];
-->xlimit=[xlimp1;xlimp1;xlimp1;xlimp1;xlimp2;xlimp2;xlimp2;xlimp2;
xlimp3;xlimp3;xlimp3;xlimp3];
-->[xg,fg] = epmain(eld01,12,20,xlimit,0.15,[500 50 0 4]);
-->xg'
 ans  =
   52.043943
   133.01962
```

```
   83.195678
   99.534889
   88.609574
   248.59168
   123.08456
   179.19561
   105.94762
   248.37591
   247.55912
   249.17564
-->fg
 fg  =
   10021278.
```



รูปที่ 9.6 แผนภาพการลู่เข้าของปัญหาการจ่ายโหลดอย่างประหยัดโดยใช้กำหนดการวิวัฒนาการ

จากผลการทดสอบเปรียบเทียบกันระหว่างระเบียบวิธีทั้งสองที่นำเสนอนี้ จะได้ผลดังแสดงในตารางที่ 9.3 – 9.4

ตารางที่ 9.3 สรุปผลเฉลยที่ได้จากระเบียบวิธีชันที่สุด

Unit	0.00 – 6.00	6.00 – 12.00	12.00 – 18.00	18.00 – 24.00
1	82.33 MW	200.07 MW	151.33 MW	175.96 MW
2	82.33 MW	214.90 MW	151.30 MW	175.99 MW
3	82.33 MW	214.90 MW	151.36 MW	176.03 MW
Total	246.99 MW	629.87 MW	453.99 MW	527.98 MW
	Total cost = 12.24 million Baht/day			
	Generation used = 9		Execution time consumed = 0.766 s	

ตารางที่ 9.4 สรุปผลเฉลยที่ได้จากกำหนดการวิวัฒนาการ

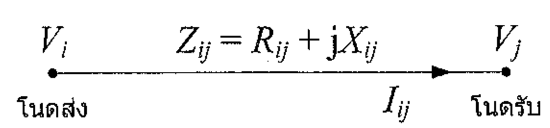
Unit	0.00 – 6.00	6.00 – 12.00	12.00 – 18.00	18.00 – 24.00
1	52.04 MW	133.02 MW	83.20 MW	99.53 MW
2	88.61 MW	248.59 MW	123.08 MW	179.20 MW
3	105.95 MW	248.38 MW	247.56 MW	249.18 MW
Total	246.6 MW	629.99 MW	453.84 MW	527.91 MW
	Total cost = 10.02 million Baht/day			
	Generation used = 217		Execution time consumed = 29.67 s	

จะเห็นได้ว่า ความแตกต่างที่เกิดขึ้นเมื่อเทียบเป็นสัดส่วน อยู่ที่ 22.2% เท่านั้น เมื่อนำมาแปลงเป็นต้นทุนที่ลดลงจะได้สูงถึง 2.22 ล้านบาทต่อวัน รายละเอียดเพิ่มเติมของการแก้ปัญหาการจ่ายโหลดอย่างประหยัดของโรงไฟฟ้าพลังความร้อนโดยใช้กำหนดการวิวัฒนาการสามารถค้นคว้าได้จากงานวิจัยของผู้แต่งในรายการอ้างอิง [6]

9.3 ปัญหาการชดเชยแรงดันตกในสายป้อนจำหน่าย

ระบบจ่ายกำลังไฟฟ้าผ่านสายป้อนในระดับแรงดันต่ำถึงแรงดันปานกลางเป็นระบบที่มีการลงทุนไม่สูงมากนัก ยกเว้นในกรณีของการจ่ายไฟให้กับโหลดที่สำคัญ เช่น โรงงานอุตสาหกรรม หรือโรงพยาบาล เป็นต้น [7]

การศึกษาเบื้องต้นนี้จะเน้นไปที่ระบบสายป้อนจำหน่ายที่จ่ายโหลด 3 เฟส แบบสมดุล ทำให้การวิเคราะห์ต่อเฟสสามารถนำมาใช้ได้ โดยปกติ สายป้อนหรือส่วนของสายป้อนมีระยะทางไม่ไกลมากนัก ทำให้แบบจำลองสายส่งระยะสั้นเพียงพอที่จะนำมาใช้งาน [8,9] ถ้ากำหนดให้ส่วนของสายป้อนดังรูปที่ 9.7 สามารถคำนวณแรงดันตกในส่วนของสายป้อนนี้ได้ดังสมการที่ 9.5 นี้



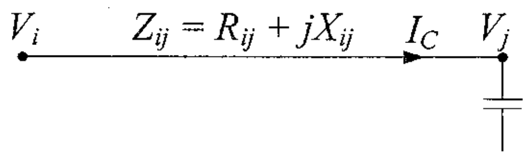
รูปที่ 9.7 ตัวอย่างส่วนของสายป้อน

$$VD = \|(R_{ij} + jX_{ij})I_{ij}\| = \|Z_{ij}\| \cdot \|I_{ij}\| \tag{9.5}$$

เพื่อให้ง่ายในการคำนวณแรงดันตกในสายป้อน การประมาณค่าแรงดันตกคร่อมส่วนของสายป้อนใดๆ อาจจะคำนวณได้โดยใช้ตัวประกอบแรงดันตก (K "drop" factor: K_{drop}) [8] นิยามได้จากสมการต่อไปนี้

$$K_{drop} = \frac{\%voltage\ drop}{kVA \cdot km} \tag{9.6}$$

K_{drop} ถือเป็นคุณสมบัติเฉพาะของสายส่งแต่ละชนิด บ่งชี้เปอร์เซ็นต์ของแรงดันตกต่อการจ่ายโหลด 1.0 kVA ต่อความยาว 1.0 km พิจารณาได้จากตัวอย่างต่อไปนี้



รูปที่ 9.7 ตัวอย่างส่วนของสายป้อนที่มีการชดชเยแรงดันตกด้วยตัวเก็บประจุ

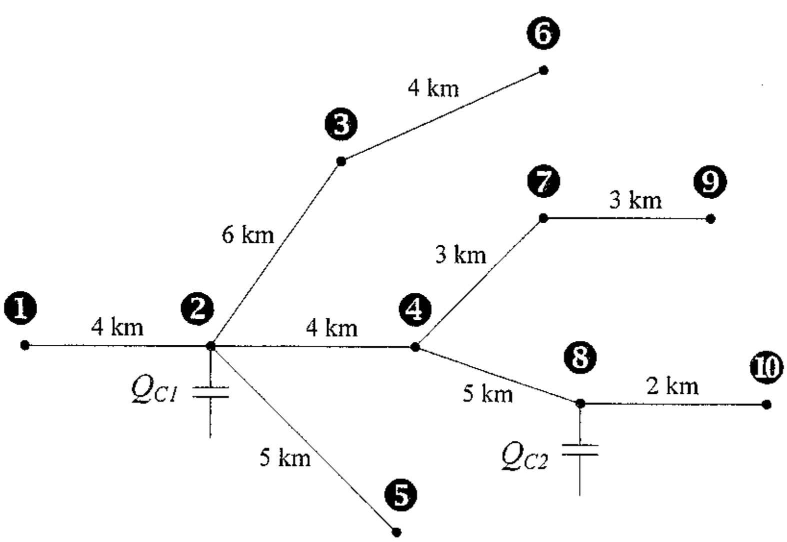
ปัญหาแรงดันตกนี้เกิดขึ้นกับสายป้อนจำหน่ายโดยทั่วไป การชดเชยแรงดันตกนิยมใช้การติดตั้งตัวเก็บประจุที่จุดโหลด เพื่อให้ง่ายในการคำนวณ การกำหนดขนาดตัวเก็บประจุที่เหมาะสมเพื่อชดเชยแรงดันตกนี้จะดำเนินการผ่านตัวประกอบการยกแรงดันขึ้น (K "rise" factor: K_{rise}) [8]

$$VD = \|(R_{ij} + jX_{ij})I_C\| = \|Z_{ij}\| \cdot \|I_C\| \tag{9.7}$$

เพื่อให้ง่ายในการคำนวณการเพิ่มขึ้นของแรงดัน การประมาณค่าแรงดันชดเชยในส่วนของสายป้อนใดๆ อาจจะคำนวณได้โดย

$$K_{rise} = \frac{\%\,voltage\ rise}{kvar \times km} \tag{9.8}$$

พิจารณาสายป้อนในระบบจ่ายกำลังไฟฟ้า 10 โนดดังต่อไปนี้ ในสภาวะปกติซึ่งไม่มีการชดเชย ทำการติดตั้งตัวเก็บประจุที่โนด 2 และ โนด 8 ดังรูปที่ 9.8



Node	P (kW)	Q (kvar)
1	0	0
2	1500	750
3	875	375
4	1000	750
5	875	250
6	1000	750
7	1250	875
8	1250	875
9	250	125
10	250	125

รูปที่ 9.8 ระบบสายป้อนจำหน่ายทดสอบขนาด 10 โนด [10]

ปัญหานี้จะพิจารณาขนาดการติดตั้งตัวเก็บประจุที่โนดทั้งสอง เพื่อให้การกระจายตัวของขนาดแรงดันไฟฟ้าที่โนดต่าง ๆ ในระบบมีค่าเข้าใกล้ 1.0 p.u. มากที่สุด [11] ดังนั้น ฟังก์ชันวัตถุประสงค์ที่ใช้ในที่นี้อยู่ในรูปของผลรวมของการเบี่ยงเบนขนาดแรงดันกำลังสอง (sum square of voltage deviation: SSVD) คำนวณได้จากสมการที่ 9.9 [12]

$$SSVD = \sum_{i=1}^{n}\left(1.0 - \|V_i\|\right)^2 \tag{9.9}$$

ดังนั้น ทำการแก้ปัญหานี้โดยใช้เทคนิคการหาค่าเหมาะที่สุดโดยใช้ค้นหาผลเฉลยที่ให้ค่า SSVD น้อยที่สุด พิจารณาจากชุดคำสั่งต่อไปนี้

<u>โปรแกรม SCILAB สำหรับตัวอย่างเรื่องการชดเชยแรงดันตก</u>

// SCILAB source program

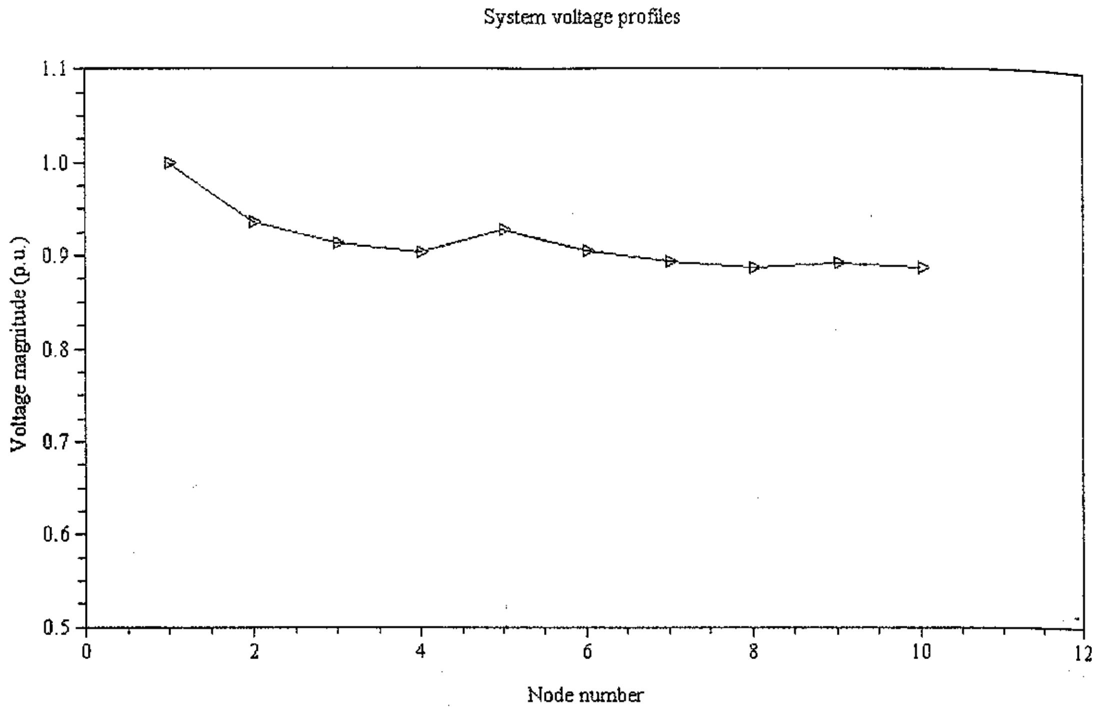
```
// กำหนดฟังก์ชันวัตถุประสงค์                    vdcalc10.sci
========================
        vdcalc10.sci
========================
function [SSVD,V] = vdcalc10(x)
Kdrop = 0.000168;  // % per kVA.km
// Linfo: line information
//   col-1   line number
//   col-2   from node
//   col-3   to-node
//   col-4   line length (km)
Linfo = [1  1   2   4
        2  2   3   6
        3  2   4   4
        4  2   5   5
        5  3   6   4
        6  4   7   3
        7  4   8   5
        8  7   9   3
        9  8   10  2];
// Ninfo: node information
// col-1   node number
// col-2   real power (kW)
// col-3   reactive power (kvar)
```

```
Ninfo = [1  0      0
        2  1500   750
        3  875    375
        4  1000   750
        5  875    250
        6  1000   750
        7  1250   875
        8  1250   875
        9  250    125
        10 250    125];
ss = Ninfo(:,2)+Ninfo(:,3)*%i;
LDsum(10) = ss(10);
LDsum(9) = ss(9);
LDsum(6) = ss(6);
LDsum(5) = ss(5);
LDsum(8) = ss(8) + LDsum(10);
LDsum(7) = ss(7) + LDsum(9);
LDsum(4) = ss(4) + LDsum(7) + LDsum(8);
LDsum(3) = ss(3) + LDsum(6);
LDsum(2) = ss(2) + LDsum(3) + LDsum(4) + LDsum(5);
LDsum(1) = ss(1) + LDsum(2);
//disp(LDsum)
QC = [2 x(1) 4;
     8 x(2) 13];
Nnum = size(Ninfo,1);
Lnum = size(Linfo,1);
V(1) = 1.0;
for k=2:Nnum
    for h=1:Lnum
        if Linfo(h,3)==k then
            i = Linfo(h,2);
            if k==2 then
                VRF = Kdrop/100*QC(1,3)*QC(1,2);
            elseif k==8 then
```

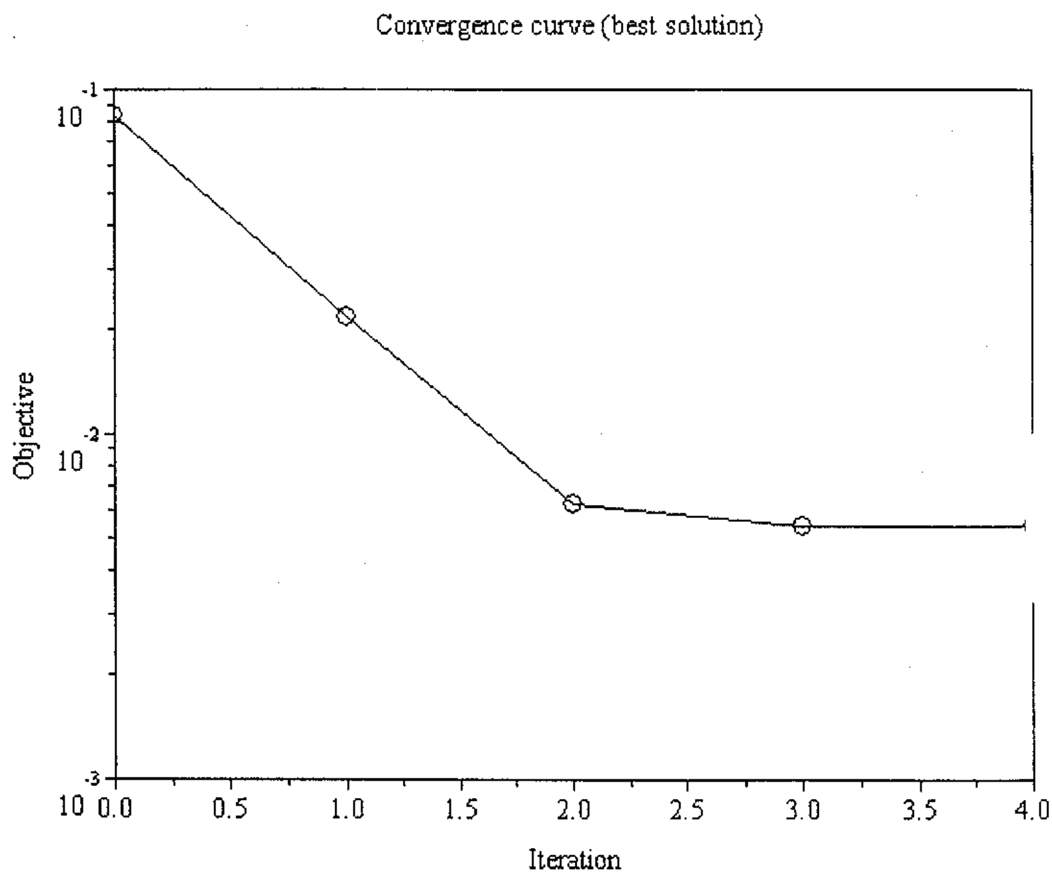
```
            VRF = Kdrop/100*QC(2,3)*QC(2,2);
         else
            VRF = 0;
         end
         VDF = Kdrop/100*Linfo(h,4)*abs(LDsum(k));
         V(k) = V(i) - VDF + VRF;
         break;
      end
   end
end
SSVD = 0;
for k=2:Nnum
   SSVD = SSVD + (1 - V(k))^2;
end
endfunction
```

เมื่อไม่มีการชดเชยจะได้ขนาดแรงดันที่โนดต่าง ๆ และค่า SSVD ดังต่อไปนี้

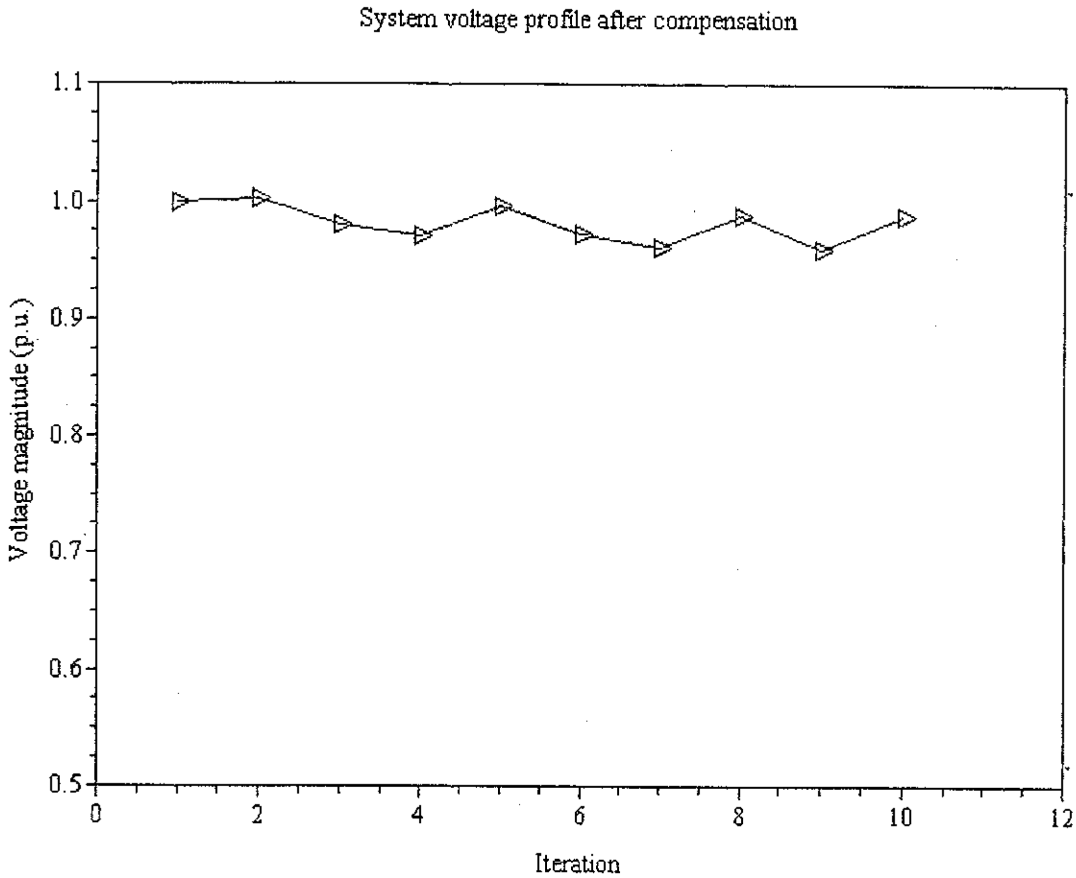
```
// ผลการรันโปรแกรม
-->[SSVD,V] = vdcalc10([0 0]);
-->SSVD
 SSVD  =
   0.0834005
-->V
 V  =
   1.0000000
   0.9356043
   0.9135633
   0.9029846
   0.9279601
   0.9051633
   0.8938986
   0.8878412
   0.8924898
   0.8869021
-->plot(V)
```



รูปที่ 9.9 การกระจายตัวของแรงดันที่โนดต่าง ๆ ของระบบในกรณีฐาน



รูปที่ 9.10 กราฟการลู่เข้าของผลเฉลยโดยใช้ระเบียบวิธีชันที่สุด

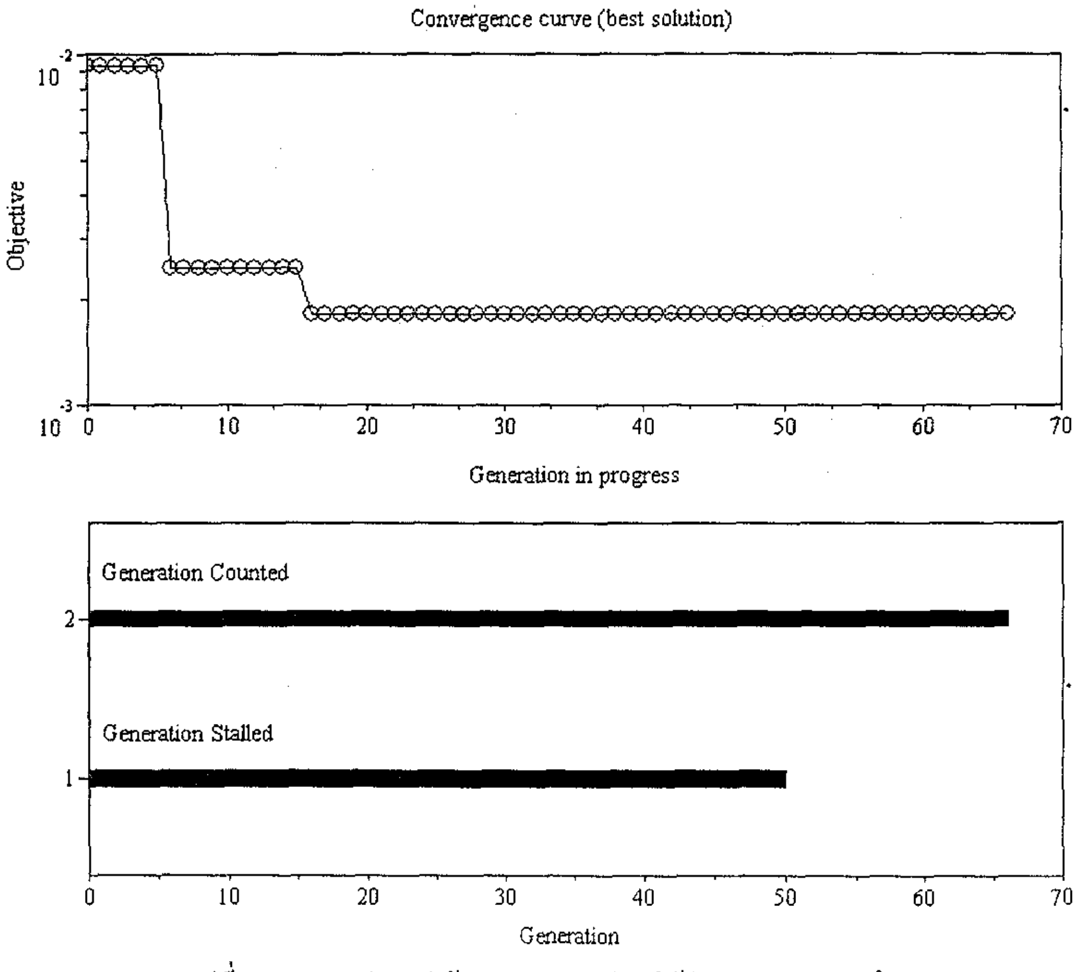


รูปที่ 9.11 การกระจายตัวของแรงดันที่โนดต่าง ๆ ของระบบกรณีที่มีการชดเชย

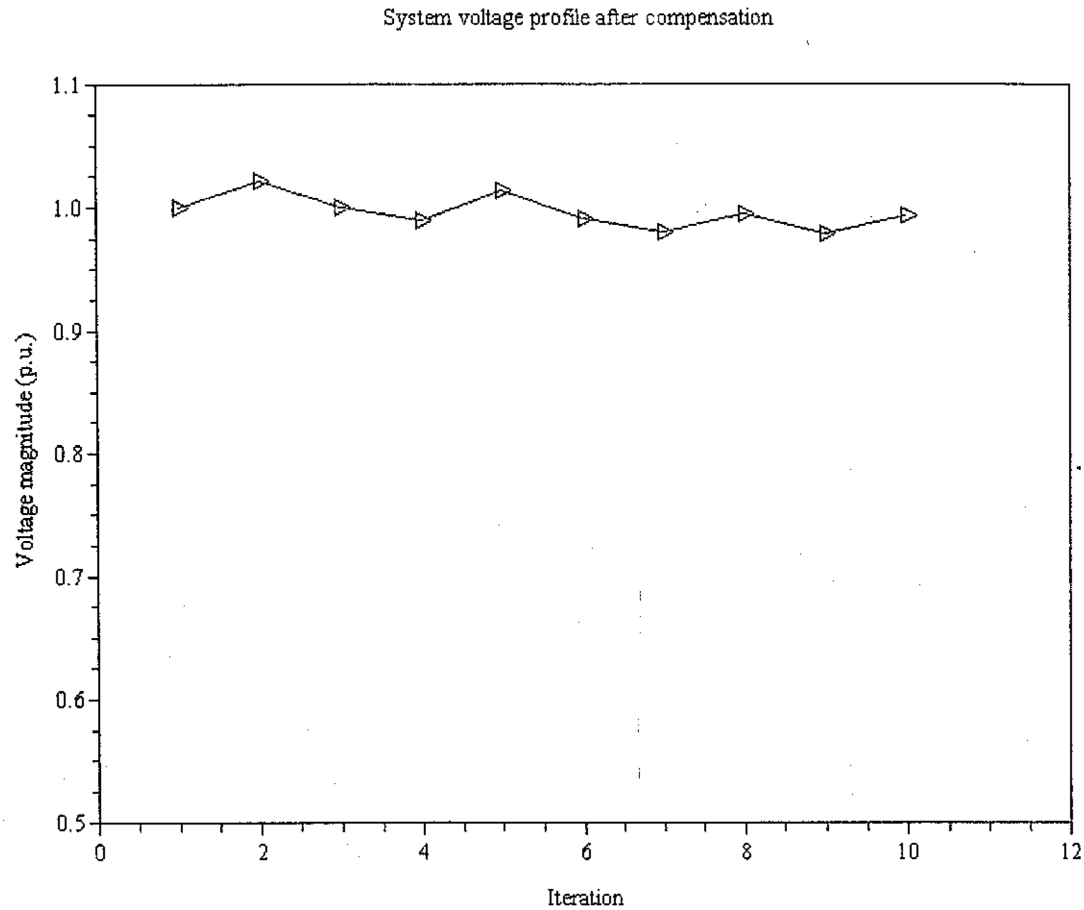
ทดลองแก้ปัญหาโดยใช้ระเบียบวิธีชันที่สุด จะได้ดังรูปที่ 9.10 และ 9.11

```
// ผลการรันโปรแกรม
-->[xmin,fmin,k]=steepestx(vdcalc10,[0;0],[100 1e-6 0]);
-->xmin
 xmin  =
    10024.406
    1534.4474
-->fmin
 fmin  =
    0.0054174
-->[SSVD,V] = vdcalc10(xmin);
-->SSVD
 SSVD  =
    0.0054174
-->fmin
 fmin  =
```

```
   0.0054174
-->V
 V  =
   1.0000000
   1.0029683
   0.9809273
   0.9703486
   0.9953242
   0.9725273
   0.9612626
   0.9887176
   0.9598539
   0.9877784
-->plot(V)
```



รูปที่ 9.12 กราฟการลู่เข้าของผลเฉลยโดยใช้จีนเนติกอัลกอริทึม



รูปที่ 9.13 การกระจายตัวของแรงดันที่โนดต่าง ๆ ของระบบกรณีที่มีการชดเชย

ทดลองแก้ปัญหาโดยใช้จีนเนติกอัลกอริทึม จะได้ดังรูปที่ 9.12 และ 9.13

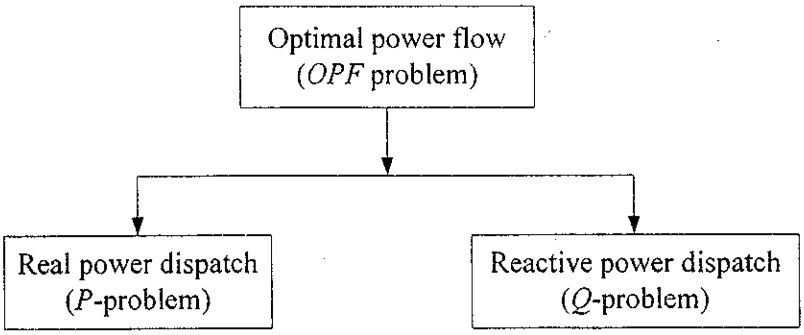
```
// ผลการรันโปรแกรม
-->[pg,xg,fg] = sga(vdcalc10,2,5,[0 15000 16;0 8000 16],0.8,0.05,[500 50 0]);
-->xg
 xg  =
   12769.284    949.23323
-->fg
 fg  =
   0.0018128
-->[SSVD,V] = vdcalc10(xg');
-->SSVD
 SSVD  =
   0.0018128
-->plot(V)
-->V
 V  =
```

```
1.0000000
1.0214138
0.9993728
0.9887941
1.0137697
0.9909728
0.9797082
0.9943821
0.9782994
0.9934429
```

9.4 ปัญหาการไหลของกำลังไฟฟ้าเหมาะที่สุด

ปัญหาการไหลของกำลังไฟฟ้าเหมาะที่สุด (optimal power flow: OPF) ถูกนำเสนอโดยคาร์เพนเทียร์ (Carpentier) [13] ในปี ค.ศ. 1962 การศึกษาการจ่ายโหลดอย่างประหยัดเป็นปัญหาค่าเหมาะที่สุดของต้นทุนการผลิตกำลังไฟฟ้า ภายใต้เงื่อนไขบังคับสมการหนึ่งชุด ได้แก่ กำลังผลิตเท่ากับโหลดบวกกำลังงานสูญเสีย ซึ่งกำลังงานสูญเสียนี้เป็นผลมาจากการไหลของกำลังไฟฟ้าในระบบส่งจ่ายนั่นเอง การปรับการไหลกำลังไฟฟ้าส่งผลต่อกำลังไฟฟ้าสูญเสียในระบบโดยตรง นั่นคือ ปัญหาการจ่ายโหลดอย่างประหยัดนั้น ถึงแม้จะพิจารณาเงื่อนไขหลายประการ แต่เงื่อนไขบังคับสมการจะใช้แบบจำลองกำลังงานสูญเสียในรูปอย่างง่าย เช่น ฟังก์ชันกำลังสอง ถึงแม้ว่า การแก้ปัญหาด้วยการจ่ายโหลดอย่างประหยัดง่ายกว่าปัญหาการไหลของกำลังไฟฟ้าเหมาะที่สุดมาก แต่ข้อมูลการไหลของกำลังไฟฟ้าผ่านสายส่ง (power flows) และกำลังงานสูญเสีย (power losses) ในระบบส่งจ่ายกำลังไฟฟ้าจะหายไป ส่งผลกระทบต่อปัญหาด้านการรักษาระดับแรงดันทำงานของระบบ ดังนั้น การแก้ปัญหาที่ใช้หลักการสมดุลของกำลังผลิตในรูปของการจ่ายโหลดอย่างประหยัดพิจารณาเฉพาะกำลังไฟฟ้าจริง นิยมเรียกสั้น ๆ ว่า ปัญหาการจัดสรรกำลังไฟฟ้าจริง (problem of real-power allocation) หรือปัญหาพี (*P*-problem) [6] เมื่อได้ผลการจัดสรรกำลังจริงจากหน่วยการผลิตของโรงไฟฟ้าต่าง ๆ แล้ว ปัญหาที่สำคัญสืบเนื่องกันมา คือ การควบคุมการไหลกำลังไฟฟ้าที่ผลิตจากโรงไฟฟ้าไปยังโหลดผ่านระบบโครงข่ายสายส่งกำลังไฟฟ้า โดยทำให้มีกำลังงานสูญเสียโดยรวมต่ำที่สุดและการกระจายของแรงดันที่บัสต่าง ๆ ยังอยู่ในขอบเขตที่กำหนด ปัญหาสืบเนื่องนี้เรียกว่า ปัญหาการจัดสรรกำลังไฟฟ้ารีแอกทีฟ (problem of reactive power allocation) หรือปัญหาคิว (*Q*-problem) โดยจะดำเนินการปรับเปลี่ยนตัวแปรควบคุม ได้แก่ ขนาดแรงดันของบัสควบคุมแรงดันต่างๆ ที่กระจายอยู่ในระบบไฟฟ้ากำลัง ค่าแท็บของหม้อแปลงในระบบ ค่าอุปกรณ์ชดเชยกำลังไฟฟ้ารีแอกทีฟที่บัสโหลด หรืออุปกรณ์ชดเชยแรงดันชนิดอนุกรม เป็นต้น ภายใต้การแก้ปัญหาค่าเหมาะที่สุด โดยหาจุดทำงานที่ให้กำลังงานสูญเสียน้อยที่สุด อาจจะเรียกอีกชื่อหนึ่งว่า

ปัญหากำลังสูญเสียน้อยที่สุด (loss minimization problem) ปัจจุบันนี้ ระบบไฟฟ้ากำลังซับซ้อนมากขึ้น ปัญหาแรงดันตกและเสถียรภาพแรงดันไฟฟ้าเข้ามามีบทบาทอย่างมากในการทำงานของระบบไฟฟ้ากำลังนอกเหนือจากเงื่อนไขของการทำงานอย่างประหยัด พิจารณาจากแผนภาพต่อไปนี้



รูปที่ 9.14 โครงสร้างของปัญหาการไหลของกำลังไฟฟ้าเหมาะที่สุด

ในอดีต เนื่องจากสมรรถนะของเครื่องคอมพิวเตอร์สำหรับการคำนวณมีประสิทธิภาพต่ำ ทำให้การแก้ปัญหาการไหลของกำลังไฟฟ้าเหมาะที่สุดโดยตรงมีความซับซ้อนสูง และใช้เวลานาน ดังนั้น การแยกคู่ปัญหาการไหลของกำลังไฟฟ้าเหมาะที่สุดออกเป็นปัญหาย่อยของการจัดสรรกำลังไฟฟ้าจริง และปัญหาย่อยของการจัดสรรกำลังไฟฟ้ารีแอกทีฟ (real and reactive power decomposition) ถูกนำมาเพื่อให้สามารถแก้ปัญหาได้ ดังนี้

- **ปัญหาพี (*P*-problem)**

Minimize	total production cost
Subject to	power flow equations
	generation limits

<u>ตัวแปรควบคุม</u> (control variables):

กำลังผลิตจากเครื่องกำเนิดไฟฟ้าของหน่วยการผลิตย่อยต่างๆ

<u>ตัวแปรคงที่</u> (fixed variables):

ขนาดแรงดันบัสควบคุม

ขนาดกำลังไฟฟ้าของตัวชดเชยต่าง ๆ

ค่าแท็บหม้อแปลง หรืออื่น ๆ

<u>ตัวแปรสถานะ</u> (state variables):

ขนาดแรงดันของบัสโหลด

มุมเฟสแรงดันของบัสต่าง ๆ

กำลังไฟฟ้ารีแอกทีฟของเครื่องกำเนิดไฟฟ้า

ถ้ากำลังงานสูญเสียในระบบสายส่ง อาศัยผลการคำนวณจากผลเฉลยแรงดันไฟฟ้าของการคำนวณการไหลของกำลังไฟฟ้า นั่นหมายความว่า ขนาดแรงดันบัสควบคุม ขนาดกำลังไฟฟ้าของตัวชดเชยต่าง ๆ ค่าแท็บหม้อแปลง หรืออื่น ๆ จะต้องถูกกำหนดให้คงที่ก่อนดำเนินการคำนวณ

เรียกตัวแปรเหล่านี้ว่า ตัวแปรคงที่ (fixed variables) โดยตัวแปรเหล่านี้จะถูกปรับในส่วนของการแก้ปัญหาคิว

- **ปัญหาคิว (*Q*-problem)**

Minimize	total losses
Subject to	power flow equations
	voltage limits
	reactive power limits
	transformer's tap limits

<u>ตัวแปรควบคุม</u>(control variables):

ขนาดแรงดันบัสควบคุม

ขนาดกำลังไฟฟ้าของตัวชดเชยต่าง ๆ

ค่าแท็บหม้อแปลง หรือ อื่น ๆ

<u>ตัวแปรคงที่</u> (fixed variables):

กำลังผลิตจากเครื่องกำเนิดไฟฟ้าของหน่วยการผลิตย่อยต่าง ๆ

<u>ตัวแปรสถานะ</u> (state variables):

ขนาดแรงดันของบัสโหลด

มุมเฟสแรงดันของบัสต่าง ๆ

กำลังไฟฟ้ารีแอกทีฟของเครื่องกำเนิดไฟฟ้า

ปัญหาย่อยทั้ง 2 ปัญหานี้ จำเป็นต้องทำควบคู่กันไปจนกระทั่งได้จุดคำตอบที่ไม่สามารถลดค่าฟังก์ชันวัตถุประสงค์ได้อีก การแก้ปัญหาแบบลำดับนี้ถูกนำมาใช้ในช่วงเวลาที่เครื่องช่วยคำนวณมีสมรรถนะต่ำ แต่ในปัจจุบัน คอมพิวเตอร์ได้รับการพัฒนาให้มีสมรรถนะสูงขึ้นมาก ทำให้การรวมปัญหาทั้ง 2 เข้าด้วยกันมีประสิทธิภาพมากกว่า หรือที่รู้จักกันในชื่อ ปัญหา OPF ปัญหานี้เป็นปัญหาเปิด การเลือกฟังก์ชันวัตถุประสงค์ มีได้หลากหลาย เช่น ต้นทุนการผลิต กำลังงานสูญเสีย การกระจายแรงดันไฟฟ้าในระบบ ความมั่นคงของระบบ หรืออื่น ๆ ส่วนตัวแปรควบคุมที่ใช้ ได้แก่ กำลังผลิตจากเครื่องกำเนิดไฟฟ้าของหน่วยการผลิตย่อยต่าง ๆ ขนาดของแรงดันที่บัสควบคุม ขนาดกำลังไฟฟ้าของตัวชดเชยต่าง ๆ ค่าแท็บหม้อแปลง หรืออื่น ๆ ต้องถูกปรับค่าไปพร้อม ๆ กัน [14]

อย่างไรก็ตาม เพื่อเป็นการศึกษาในเบื้องต้น จะใช้ต้นทุนการผลิตโดยรวมของระบบ (total production cost) เป็นฟังก์ชันวัตถุประสงค์ ดังนี้

Minimize	total production cost (หรืออื่น ๆ)
Subject to	power flow equations
	variable limits

<u>ตัวแปรควบคุม</u> (control variables):

กำลังผลิตจากเครื่องกำเนิดไฟฟ้าของหน่วยการผลิตย่อยต่าง ๆ

ขนาดแรงดันบัสควบคุม

ขนาดกำลังไฟฟ้าของตัวชดเชยต่าง ๆ

ค่าแท็บหม้อแปลง หรืออื่น ๆ

<u>ตัวแปรสถานะ</u> (state variables):

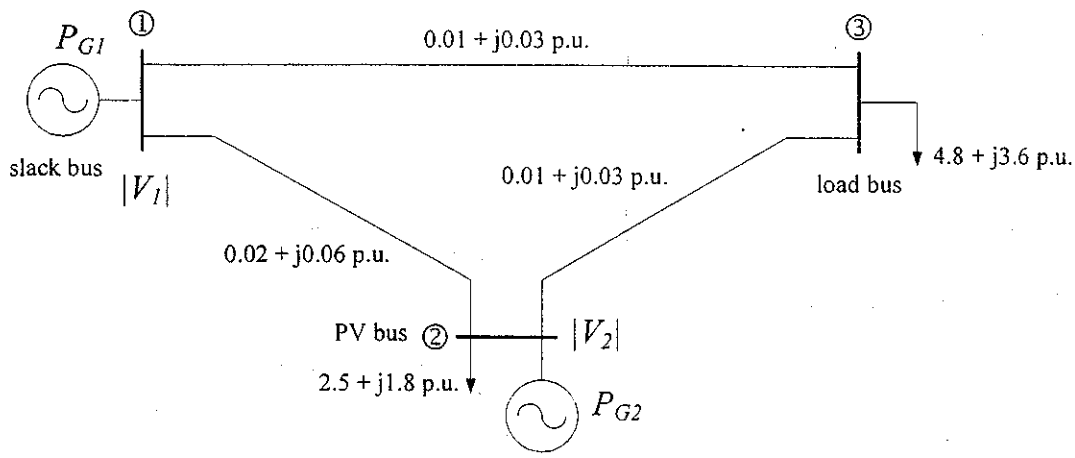
ขนาดแรงดันของบัสโหลด

มุมเฟสแรงดันของบัสต่าง ๆ

กำลังไฟฟ้ารีแอกทีฟของเครื่องกำเนิดไฟฟ้า

ในกรณีของปัญหา OPF นี้ ไม่มีตัวแปรคงที่

จากระบบไฟฟ้ากำลังต่อไปนี้ จงแก้ปัญหาการไหลของกำลังไฟฟ้าเหมาะที่สุดเพื่อค้นหาจุดทำงานที่เหมาะสม



รูปที่ 9.15 ระบบทดสอบสำหรับการแก้ปัญหา OPF

โดยเครื่องกำเนิดไฟฟ้าทั้งสองมีฟังก์ชันค่าเชื้อเพลิงดังนี้

$$f_{G1}(P_{G1}) = 561 + 7.92P_{G1} + 0.001562P_{G1}^2 \qquad \Re/h;\ 150 \le P_{G1} \le 600\ MW$$

$$f_{G2}(P_{G2}) = 310 + 7.85P_{G2} + 0.00194P_{G2}^2 \qquad \Re/h;\ 100 \le P_{G2} \le 400\ MW$$

$$-400 \le Q_{G1} \le +400\ Mvar; \qquad -400 \le Q_{G2} \le +400\ Mvar;$$

ดำเนินการวิเคราะห์ปัญหาจะได้ว่า

<u>ตัวแปรควบคุม</u> $P_{G1}, P_{G2}, |V_1|, |V_2|$

<u>ตัวแปรสถานะ</u> $|V_3|, \delta_2, \delta_3, Q_{G1}, Q_{G2}$

<u>ฟังก์ชันวัตถุประสงค์</u> (total production cost)

$$F_T = \left(561 + 7.92P_{G1} + 0.001562P_{G1}^2\right) + \left(310 + 7.85P_{G2} + 0.00194P_{G2}^2\right)$$

<u>เงื่อนไขบังคับสมการ</u> (สมการการไหลของกำลังไฟฟ้ารวมทั้งสิ้น 6 สมการ) [2,15,16]

$$P_{G1} - \left|Y_{11}V_1^2\right|\cos(\theta_{11}) - \left|Y_{12}V_1V_2\right|\cos(\theta_{12} + \delta_2 - \delta_1) - \left|Y_{13}V_1V_3\right|\cos(\theta_{13} + \delta_3 - \delta_1) = 0$$

$$Q_{G1} + \left|Y_{11}V_1^2\right|\sin(\theta_{11}) + \left|Y_{12}V_1V_2\right|\sin(\theta_{12} + \delta_2 - \delta_1) + \left|Y_{13}V_1V_3\right|\sin(\theta_{13} + \delta_3 - \delta_1) = 0$$

$$P_{G2} - 2.5 - |Y_{21}V_2V_1|\cos(\theta_{21}+\delta_1-\delta_2) - |Y_{22}V_2^2|\cos(\theta_{22}) - |Y_{23}V_2V_3|\cos(\theta_{23}+\delta_3-\delta_2) = 0$$

$$Q_{G2} - 1.8 + |Y_{21}V_2V_1|\sin(\theta_{21}+\delta_1-\delta_2) + |Y_{22}V_2^2|\sin(\theta_{22}) + |Y_{23}V_2V_3|\sin(\theta_{23}+\delta_3-\delta_2) = 0$$

$$-4.8 - |Y_{31}V_3V_1|\cos(\theta_{31}+\delta_1-\delta_3) - |Y_{32}V_3V_2|\cos(\theta_{32}+\delta_2-\delta_3) - |Y_{33}V_3^2|\cos(\theta_{33}) = 0$$

$$-3.6 + |Y_{31}V_3V_1|\sin(\theta_{31}+\delta_1-\delta_3) + |Y_{32}V_3V_2|\sin(\theta_{32}+\delta_2-\delta_3) + |Y_{33}V_3^2|\sin(\theta_{33}) = 0$$

<u>เงื่อนไขบังคับอสมการ</u> (variable limits รวม 10 อสมการ)

$P_{G1} - 600 \le 0$
$150 - P_{G1} \le 0$
$P_{G2} - 400 \le 0$
$100 - P_{G2} \le 0$
$Q_{G1} - 400 \le 0$
$-400 - Q_{G1} \le 0$
$Q_{G2} - 400 \le 0$
$-400 - Q_{G2} \le 0$
$V_3 - 1.1 \le 0$
$0.9 - V_3 \le 0$

กำหนดตัวแปรเพื่อแก้ปัญหาค่าเหมาะที่สุด

$$x = [P_{G1} \quad P_{G2} \quad V_1 \quad V_2 \quad V_3 \quad \delta_2 \quad \delta_3 \quad Q_{G1} \quad Q_{G2}]^T$$

กำหนดค่าเริ่มต้น

$$x^{(0)} = [400 \quad 330 \quad 1.0 \quad 1.0 \quad 1.0 \quad 0.0 \quad 0.0 \quad 300 \quad 240]^T$$

<u>โปรแกรม SCILAB สำหรับตัวอย่างเรื่องการไหลของกำลังไฟฟ้าเหมาะที่สุด</u>

// SCILAB source program

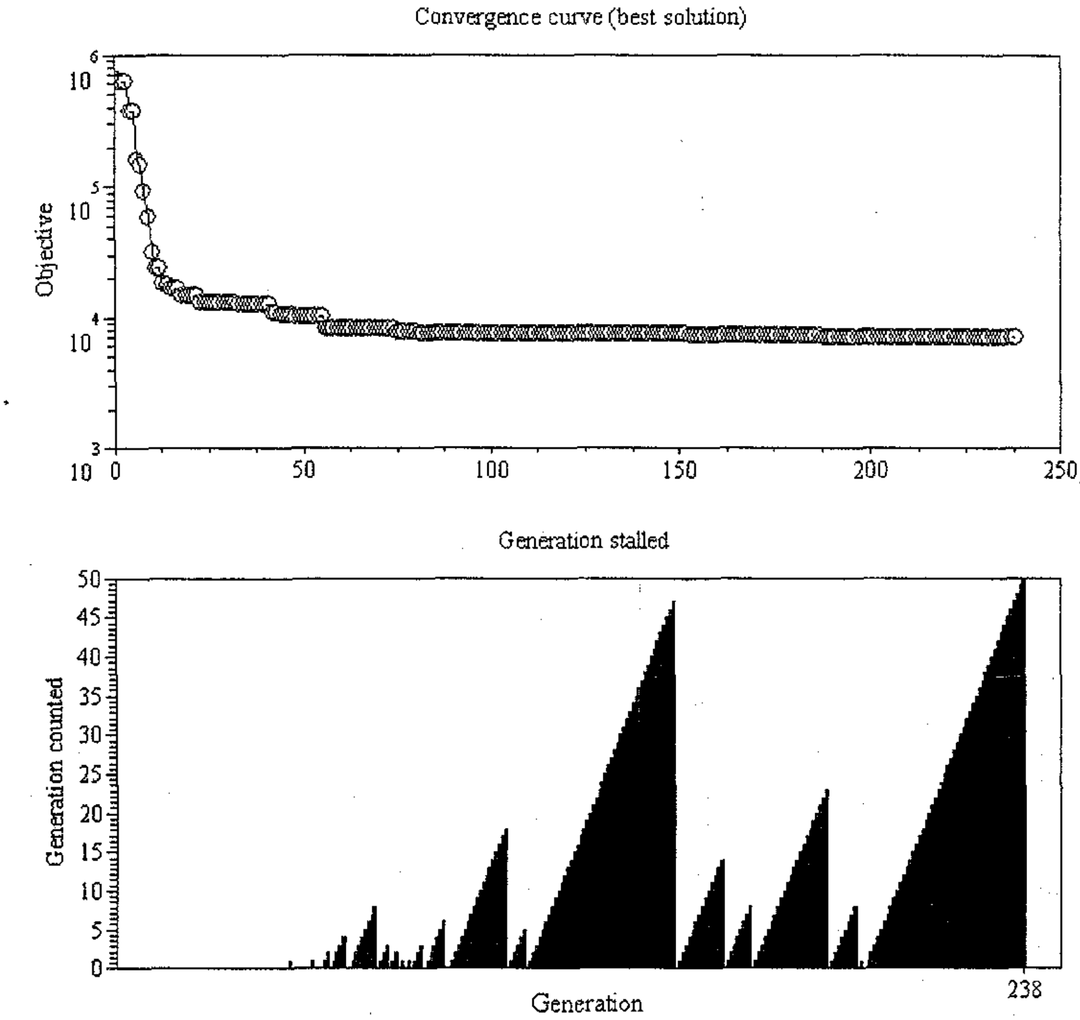
```
// กำหนดฟังก์ชันวัตถุประสงค์                    fopf01.sci
========================
           fopf01.sci
========================
function Fcost = fopf01(x)
// Bus Admittance Matrix
Ybus=[15-45*%i -5+15*%i -10+30*%i;-5+15*%i 10-30*%i -5+15*%i;-10+30*%i -5+15*%i
15-45*%i];
Ym = abs(Ybus);
Ya = [-71.6 108.4 108.4;108.4 -71.36 108.4;108.4 108.4 -71.6]*%pi/180;
Z1=0;
rho=1000;
// Control & State Variables
Pg1=x(1)/100;
```

```
Pg2=x(2)/100;
V1=x(3);
V2=x(4);
V3=x(5);
Z2=x(6);
Z3=x(7);
Qg1=x(8)/100;
Qg2=x(9)/100;
Fg1=561+7.92*x(1)+0.001562*x(1)^2;
Fg2=310+7.85*x(2)+0.00194*x(2)^2;
f=Fg1+Fg2;  // Obj. func.
// Equality Constraints
C(1,1)=abs(Pg1-Ym(1,1)*V1^2*cos(Ya(1,1))-Ym(1,2)*V1*V2*cos(Ya(1,2)+Z2-Z1)-
Ym(1,3)*V1*V3*cos(Ya(1,3)+Z3-Z1));
C(2,1)=abs(Qg1+Ym(1,1)*V1^2*sin(Ya(1,1))+Ym(1,2)*V1*V2*sin(Ya(1,2)+Z2-
Z1)+Ym(1,3)*V1*V3*sin(Ya(1,3)+Z3-Z1));
C(3,1)=abs(Pg2-2.5-Ym(2,1)*V2*V1*cos(Ya(2,1)+Z1-Z2)-Ym(2,2)*V2^2*cos(Ya(2,2))-
Ym(2,3)*V2*V3*cos(Ya(2,3)+Z3-Z2));
C(4,1)=abs(Qg2-1.8+Ym(2,1)*V2*V1*sin(Ya(2,1)+Z1-
Z2)+Ym(2,2)*V2^2*sin(Ya(2,2))+Ym(2,3)*V2*V3*sin(Ya(2,3)+Z3-Z2));
C(5,1)=abs(-4.8-Ym(3,1)*V3*V1*cos(Ya(3,1)+Z1-Z3)-Ym(3,2)*V3*V2*cos(Ya(3,2)+Z2-
Z3)-Ym(3,3)*V3^2*cos(Ya(3,3)));
C(6,1)=abs(-3.6+Ym(3,1)*V3*V1*sin(Ya(3,1)+Z1-Z3)+Ym(3,2)*V3*V2*sin(Ya(3,2)+Z2-
Z3)+Ym(3,3)*V3^2*sin(Ya(3,3)));
// Inequality Constraints
D=[Pg1-6;
   1.5-Pg1;
   Pg2-4;
   1-Pg2;
   Qg1-4;
   -4-Qg1;
   Qg2-4;
   -4-Qg1;
   V3-1.1;
```

```
  0.9-V3];
SSP=0;
for k=1:length(D)
   if D(k)<0 then
      D(k)=0;
   end
   SSP = SSP + rho*D(k)^2;
end
Fcost = f + SSP + rho*sum(C.^2);
endfunction
```

ทดลองแก้ปัญหาโดยใช้กำหนดการวิวัฒนาการ จะได้ดังรูปที่ 9.16

```
// ผลการรันโปรแกรม
-->xlimit=[[150 600;100 400; 0.9 1.1;0.9 1.1;0.9 1.1;-%pi/2 %pi/2;-%pi/2 %pi/2;
        -400 400;-400 400];
-->[xmin,fmin]=epmain(fopf01,9,20,xlimit,0.5,[500 50 0 4]);
-->xmin'
 ans  =
   373.94428
   228.86513
   1.0191976
   1.0083552
   0.9033024
 - 0.0542469
 - 0.0827438
   290.78572
   305.69325
-->fmin
 fmin  =
   7035.2328
```



รูปที่ 9.16 กราฟการลู่เข้าของผลเฉลยโดยใช้กำหนดการวิวัฒนาการ

9.5 ปัญหาการพยากรณ์โหลดทางไฟฟ้า

การวางแผนระบบไฟฟ้ากำลังต้องอาศัยข้อมูลจาการพยากรณ์ความต้องการไฟฟ้าของระบบ การทำนายค่าการเพิ่มขึ้นของโหลดได้ถูกต้องย่อมช่วยให้การวางแผนขยายระบบผลิต ระบบส่งจ่าย และระบบจำหน่ายมีความเหมาะสม [17-19] การพยากรณ์โหลดอาจจะแบ่งได้ตามช่วงเวลาการพยากรณ์เป็น i) การพยากรณ์โหลดระยะสั้น (short-term load forecasting) ที่มีช่วงเวลาการพยากรณ์แบบรายชั่วโมงจนถึงการพยากรณ์โหลดรายวัน ii) การพยากรณ์โหลดระยะกลาง (medium-term load forecasting) มีช่วงเวลาการพยากรณ์แบบรายสัปดาห์ หรือหลายสัปดาห์ และ iii) การพยากรณ์โหลดระยะยาว (long-term load forecasting) มีช่วงเวลาที่ยาวนานกว่า เช่น การพยากรณ์โหลดแบบเดือน รายฤดูกาล หรือรายปี เป็นต้น นอกจากนี้ การพยากรณ์ยังสามารถแบ่งได้เป็น การพยากรณ์การใช้พลังงานไฟฟ้า (forecast of energy consumption) การพยากรณ์โหลดสูงสุด (forecast of peak load) หรือ การพยากรณ์การขยายตัวของกลุ่มผู้ใช้ไฟ (spatial load forecast) ในส่วนนี้ จะนำเสนอการพยากรณ์ด้วยเทคนิคของบ็อกซ์-เจนกินส์ (Box-Jenkins techniques) [17]

ก่อนจะกล่าวถึงวิธีการนี้ จะนำเสนอหลักการวิเคราะห์ความถดถอย (auto-regressive analysis: AR) โดยมีตัวแบบดังนี้

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \varepsilon_t \tag{9.10}$$

ตัวแบบนี้ จะอาศัยหลักการพยากรณ์ค่าในอนาคตด้วยค่าในอดีตที่มีการถ่วงน้ำหนัก โดยใช้ค่าย้อนหลังได้หลายพจน์ตามความเหมาะสม ถ้าใช้ค่าย้อนหลัง p พจน์ เรียกว่า $AR(p)$ โดยมี ϕ_i แทนค่าสัมประสิทธิ์ ε_t แทนค่าสุ่ม สิ่งสำคัญสำหรับวิธีการนี้ คือ การคำนวณค่าสัมประสิทธิ์นั่นเอง อาจจะหาได้จากเทคนิคความคลาดเคลื่อนกำลังสองน้อยที่สุด (least square error: LSE) ดังนี้

เพื่อให้ง่าย จะพิจารณาตัวแบบ $AR(3)$ และไม่พิจารณาผลจากค่าตัวแปรสุ่ม นั่นคือ

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \phi_3 X_{t-3} \tag{9.11}$$

ถ้ากำหนดให้มีข้อมูลทั้งสิ้น n ($n > 3$) ชุด จะพบว่า ค่าแรกที่เริ่มพิจารณาคือ $t = 4$ นั่นเอง

$$\begin{aligned} X_4 &= \phi_1 X_3 + \phi_2 X_2 + \phi_3 X_1 \\ X_5 &= \phi_1 X_4 + \phi_2 X_3 + \phi_3 X_2 \\ &\vdots \\ X_n &= \phi_1 X_{n-1} + \phi_2 X_{n-2} + \phi_3 X_{n-3} \end{aligned}$$

คำนวณค่าความคลาดเคลื่อนยกกำลังสอง จะได้ว่า

$$SSE = \sum_{t=4}^{n} \left(X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} - \phi_3 X_{t-3} \right)^2 \tag{9.12}$$

ดังนั้น ค่าความคลาดเคลื่อนกำลังสองน้อยที่สุด

$$LSE = \min \quad \sum_{t=4}^{n} \left(X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} - \phi_3 X_{t-3} \right)^2 \tag{9.13}$$

ให้ SSE เป็นฟังก์ชันวัตถุประสงค์ สามารถคำนวณพารามิเตอร์ ϕ_1, ϕ_2, ϕ_3 ได้โดยใช้การแก้ปัญหาค่าเหมาะที่สุดแบบไม่มีเงื่อนไข

พิจารณาจากตัวอย่างการใช้พลังงานไฟฟ้าต่อไปนี้

ตารางที่ 9.5 ข้อมูลการใช้พลังงานแบ่งตามฤดูกาล (GWh)

Year	1965	1966	1967	1968	1969
1. Winter	874	866	843	906	952
2. Spring	679	700	719	703	745
3. Summer	616	603	594	634	635
4. Autumn	816	814	819	844	871

เขียนฟังก์ชันวัตถุประสงค์ได้เป็น

โปรแกรม SCILAB สำหรับตัวอย่างเรื่องการพยากรณ์โหลดทางไฟฟ้า

// SCILAB source program

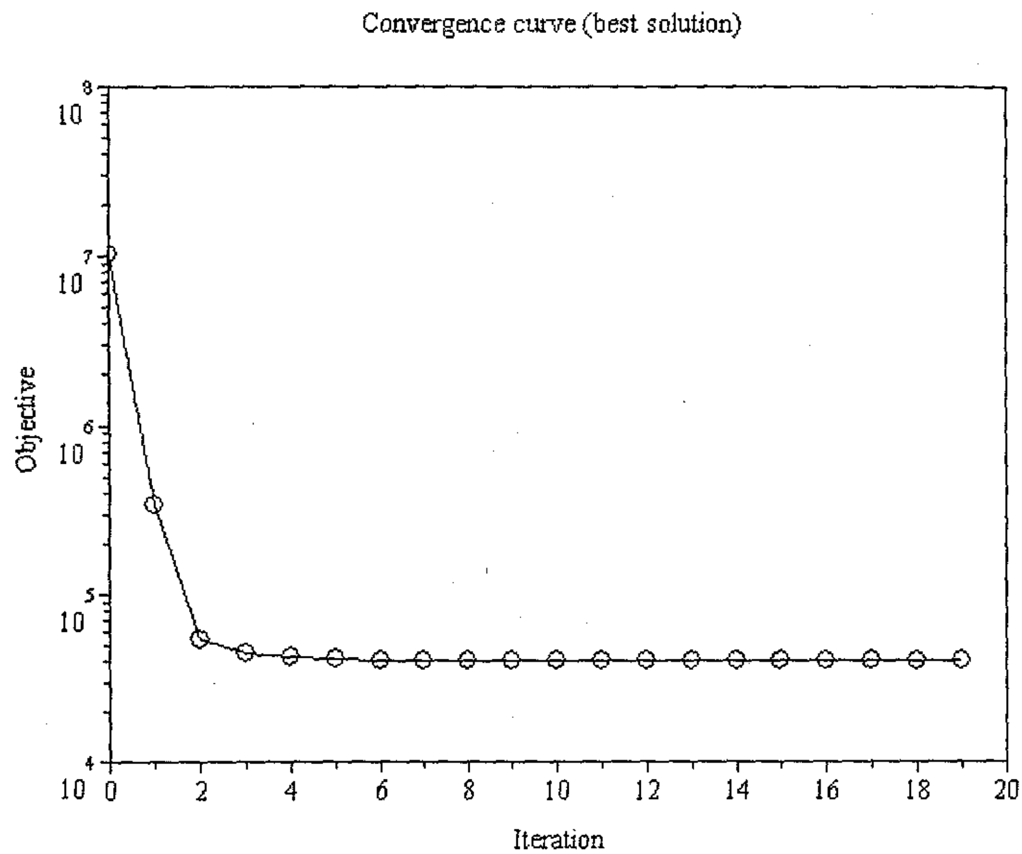
```
// กำหนดฟังก์ชันวัตถุประสงค์                                    forecast01.sci
========================
        forecast01.sci
========================
```

```
function f = AR3test(x)
Y = [874
   679
   616
   816
   866
   700
   603
   814
   843
   719
   594
   819
   906
   703
   634
   844
   952
   745
   635
   871];
f = 0;
for u=4:20
   f = f + (Y(u)-x(1)*Y(u-1)-x(2)*Y(u-2)-x(3)*Y(u-3))^2;
end
endfunction
```

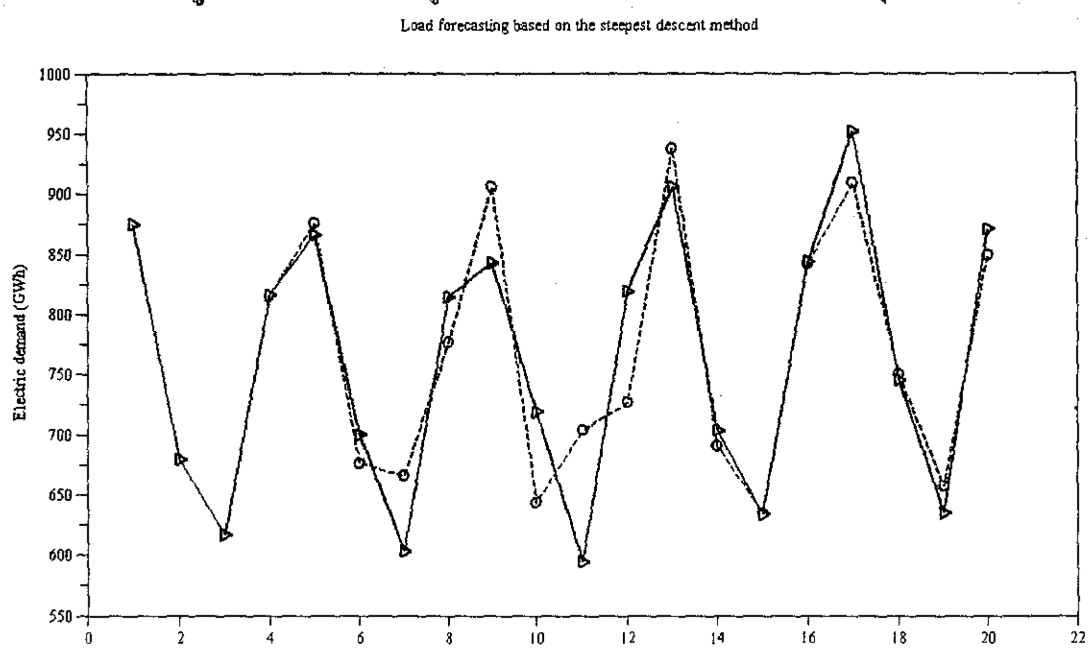
แก้ปัญหาโดยใช้ระเบียบวิธีชันที่สุด จะได้ผลดังกราฟในรูปที่ 9.17 และ 9.18

```
// ผลการรันโปรแกรม
-->[xmin,fmin,k]=steepestx(AR3test,[0;0;0],[200 1e-4 0]);
-->xmin
 xmin  =
   0.9601202
 - 0.9290002
   0.9778796
```

```
-->fmin
 fmin  =
   40274.495
```



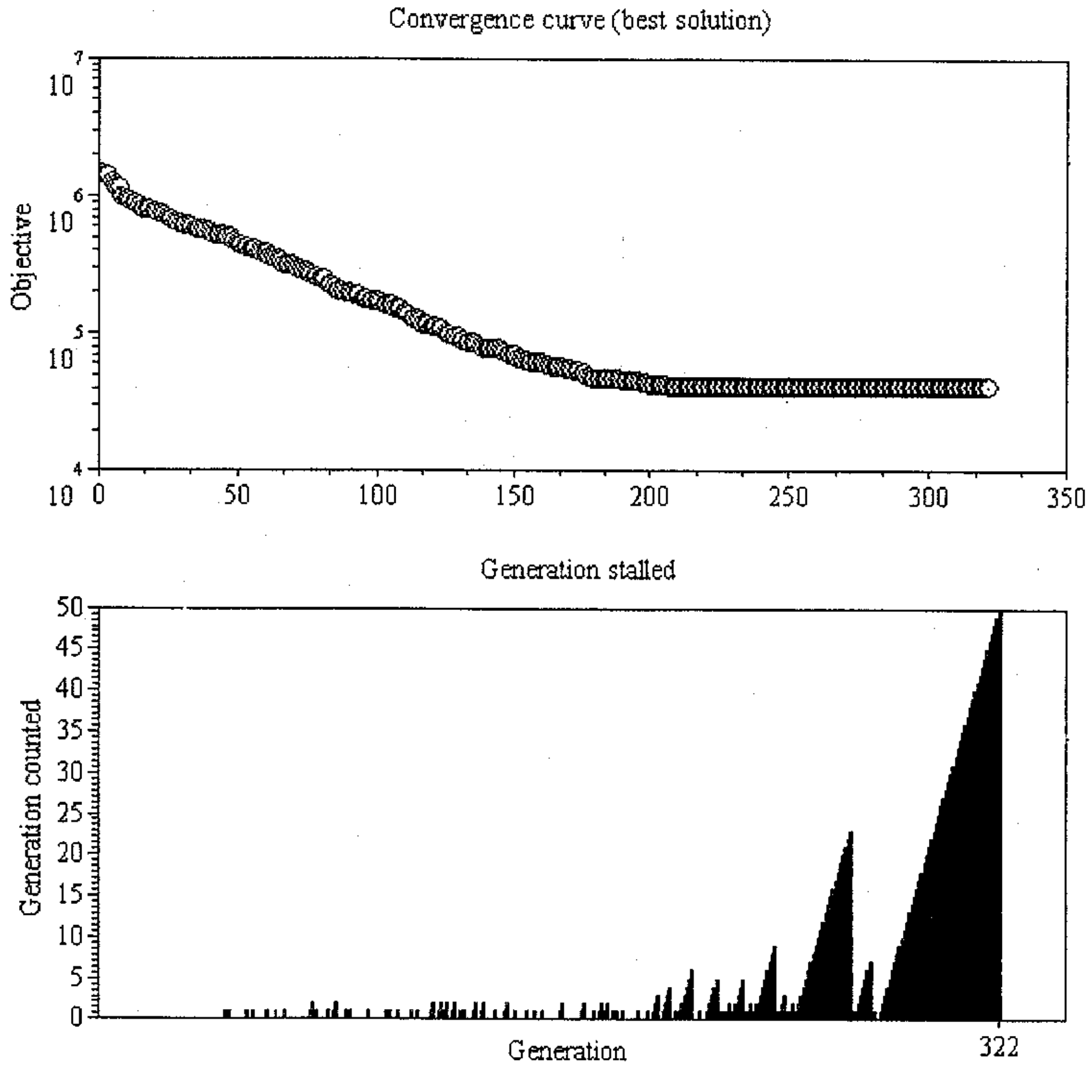
รูปที่ 9.17 กราฟการลู่เข้าของผลเฉลยโดยใช้ระเบียบวิธีชันที่สุด



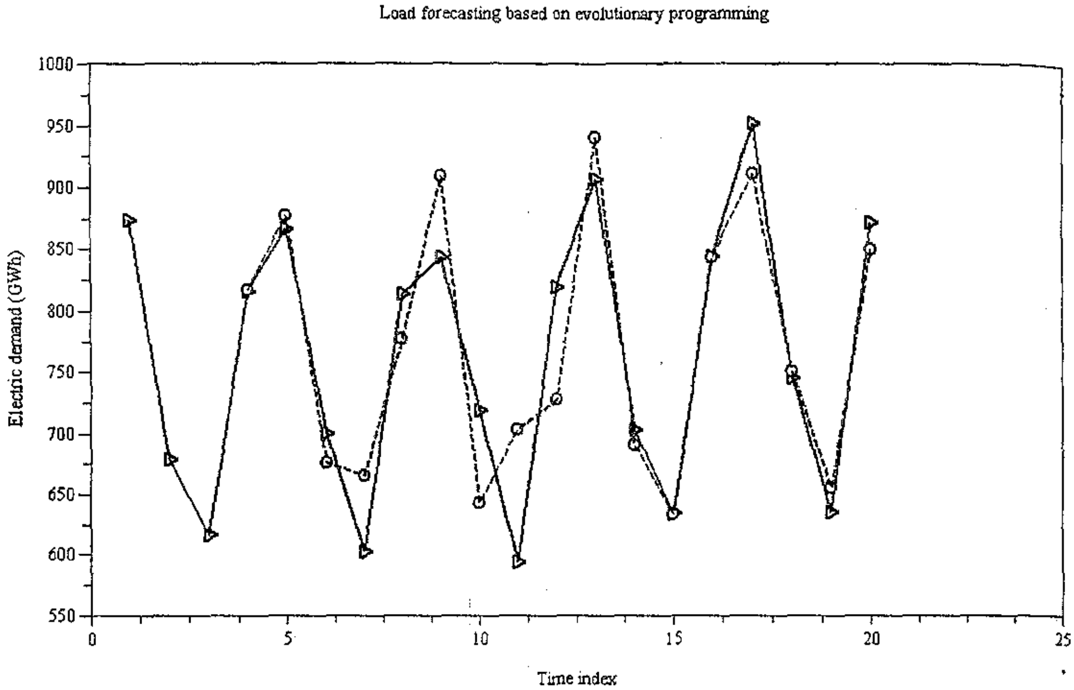
รูปที่ 9.18 การเปรียบเทียบตัวแบบการพยากรณ์โหลดโดยใช้ระเบียบวิธีชันที่สุด

แก้ปัญหาโดยใช้กำหนดการวิวัฒนาการ จะได้ผลดังกราฟในรูปที่ 9.19 และ 9.20

```
// ผลการรันโปรแกรม
-->[xg,fg] = epmain(AR3test,3,20,[-4 4;-4 4;-4 4],0.05,[500 50 0 4]);
-->fg
 fg  =
   40273.766
```



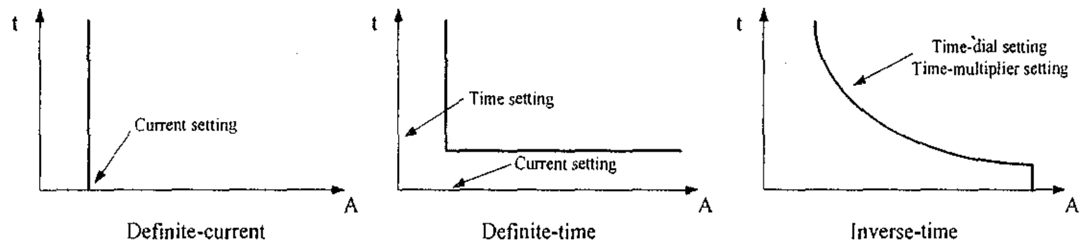
รูปที่ 9.19 กราฟการลู่เข้าของผลเฉลยโดยใช้กำหนดการวิวัฒนาการ



รูปที่ 9.20 การเปรียบเทียบตัวแบบการพยากรณ์โหลดโดยใช้กำหนดการวิวัฒนาการ

9.6 ปัญหาการจัดลำดับความสัมพันธ์การทำงานของรีเลย์ป้องกันในระบบไฟฟ้ากำลัง

รีเลย์ป้องกันเป็นอุปกรณ์ที่สำคัญในการป้องกันระบบไฟฟ้ากำลัง ทำหน้าที่ตรวจสอบความผิดพร่องในระบบไฟฟ้ากำลัง [20-23] การตรวจจับกระแสเกินในระบบไฟฟ้ากำลังเป็นฟังก์ชันการทำงานของรีเลย์ที่ใช้งานแพร่หลาย เพื่อป้องกันโหลดเกิน และความผิดพร่องชนิดต่าง ๆ รีเลย์กระแสเกินแบ่งตามคุณลักษณะการทำงานได้ 3 ชนิด ได้แก่ รีเลย์กระแสเกินชนิดกระแสจำกัด (definite-current O/C relay) รีเลย์กระแสเกินชนิดเวลาจำกัด (definite-time O/C relay) และรีเลย์กระแสเกินชนิดเวลาผกผัน (inverse-time O/C relay) รีเลย์กระแสเกินแต่ละชนิดมีกราฟการทำงานที่ต่างกันดังรูปที่ 9.21



รูปที่ 9.21 เส้นโค้งการทำงานของรีเลย์

ในความเป็นจริง รีเลย์กระแสเกินใด ๆ มีฟังก์ชันเวลาจำกัด กระแสจำกัดและเวลาผกผันอยู่ในตัว การปรับตั้งรีเลย์ดังกล่าวสามารถทำได้หลายฟังก์ชันพร้อมกัน ในที่นี้ จะกล่าวถึงหลักการปรับตั้งค่าการหน่วงเวลา และการใช้รีเลย์ชนิดกระแสจำกัดและเวลาจำกัดมาประกอบการปรับตั้งรีเลย์แบบทำงานทันที (instantaneous operation) การตั้งค่าการทำงานของรีเลย์ทำได้ดังนี้

1. <u>pickup setting</u> เป็นการตั้งค่าค่ากระแสขั้นต่ำเพื่อให้รีเลย์เริ่มต้นทำงาน การปรับตั้งค่า pickup นี้ ทำได้โดยการเลือกค่าแท็บที่เรียกว่า current tap setting (CTS) โดยที่รีเลย์จะถูกต่อกับด้านทุติยภูมิของ CT ดังนั้น ก่อนการปรับตั้งค่ากระแสดังกล่าว CT ratio ต้องถูกกำหนดเป็นอันดับแรก
2. <u>time-delay setting</u> (time-dial setting/time multiplier setting: TDS) เป็นการตั้งค่าการหน่วงเวลา หรืออีกนัยหนึ่ง TDS เป็นการเลือกเส้นโค้งการทำงานของรีเลย์

การเลือกค่า pickup เป็นสิ่งที่สำคัญประการหนึ่ง คำถามที่สำคัญ คือ จะใช้กฎอะไรในการเลือกค่าดังกล่าว จะเห็นได้ว่า รีเลย์กระแสเกินนี้ ต้องทำหน้าที่ยอมให้จ่ายโหลดในสภาวะปกติได้ นั่นคือ การตั้งค่า pickup ต้องสูงกว่าค่าโหลดสูงสุด (maximum load) ปัญหา คือ สูงกว่าด้วยตัวปรับคูณเท่าใดจึงเหมาะสม ตัวปรับคูณที่นิยมใช้ คือ ยอมให้เกิดโหลดเกินได้ไม่เกิน 25% นั่นคือ

$$I_{pickup} \geq 1.25 \times I_{load,max} \tag{9.14}$$

อาจจะใช้ตัวปรับคูณมากกว่านี้ได้ เช่น 1.3 หรือ 1.5 เป็นต้น

อย่างไรก็ตาม กฎนี้ อาจจะมีข้อยกเว้นในกรณีที่ fault level มีค่าค่อนข้างต่ำ นั่นคือ รีเลย์ต้องป้องกันความผิดพร่องที่มีความรุนแรงน้อยที่สุดได้ ดังนั้น ค่า pickup จะต้องน้อยกว่า minimum fault เสมอ

$$I_{pickup} \leq I_{fault,min} \tag{9.15}$$

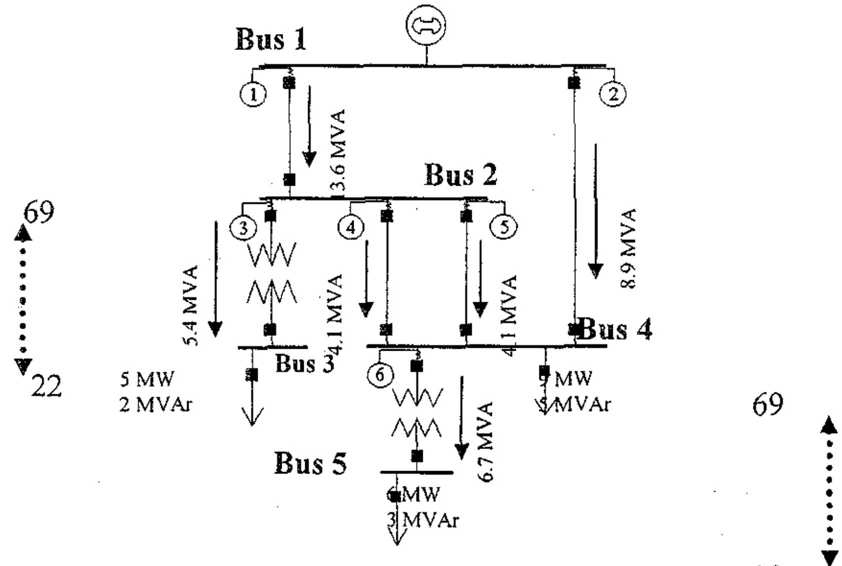
รวมเงื่อนไขเข้าด้วยกันจะได้

$$I_{pickup} = \min\{1.25 \times I_{max}, I_{fault,min}\} \tag{9.16}$$

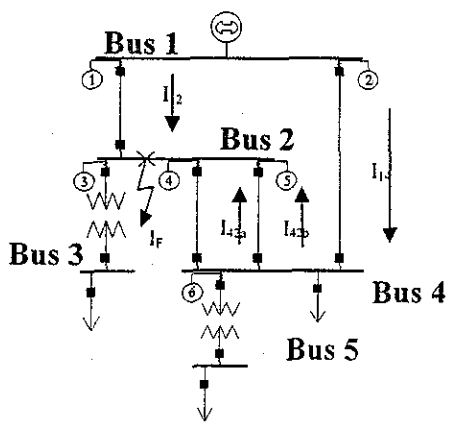
การปรับตั้งค่ากระแสจำกัด หรือที่เรียกว่า การปรับตั้งค่ากระแสตัดฉับพลัน (setting of instantaneous operation) มีความจำเป็นมาก นั่นคือ ตัวที่อยู่ใกล้จุดเกิดความผิดพร่องมากที่สุดต้องทำงานทันที ไม่มีกฎตายตัวสำหรับการปรับตั้งนี้ แต่มีข้อแนะนำดังนี้

- สำหรับสายส่งระหว่างสถานีไฟฟ้า ให้ปรับตั้งไว้ที่ค่าอย่างน้อย 125% ของระดับการลัดวงจรสูงสุดที่เกิดขึ้น ณ ตำแหน่งสถานีไฟฟ้าถัดไป
- สำหรับสายป้อนในระบบจำหน่ายทำได้ดังนี้ คือ ให้ปรับตั้งไว้ที่ 50% ของระดับการลัดวงจรสูงสุดที่ตำแหน่งติดตั้งรีเลย์ หรือให้ตั้งค่าไว้ที่ประมาณ 6 – 10 เท่า ของค่ากระแสพิกัดสายป้อน
- อย่างไรก็ตาม ในหลาย ๆ กรณี อาจจะไม่จำเป็นต้องตั้งค่ากระแสตัดฉับพลันก็ได้

พิจารณาระบบไฟฟ้ากำลัง [24] ดังรูปที่ 9.22 ซึ่งประกอบด้วยรีเลย์ป้องกันแบบดิจิตอล (digital relay) จำนวนทั้งสิ้น 6 ตัว ผลการคำนวณกระแสลัดวงจรและการกระจายตัวของกระแสลัดวงจรไปยังส่วนต่าง ๆ ของระบบ ได้นำเสนอไว้ในรูปที่ 9.23 และ 9.24 ในกรณีของการเกิดลัดวงจรที่บัส 2 และ 5 ตามลำดับ



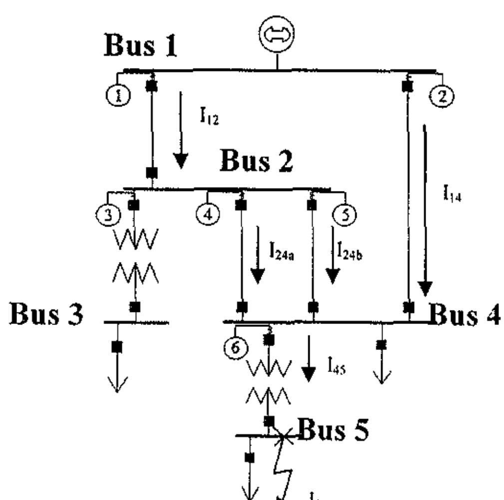
รูปที่ 9.22 ระบบไฟฟ้ากำลังทดสอบสำหรับการปรับตั้งรีเลย์ป้องกัน



Fault current distribution:

$I_F = 889$ A
$I_{12} = 635$ A
$I_{14} = 254$ A
$I_{42a} = 127$ A
$I_{42b} = 127$ A

รูปที่ 9.23 ผลการคำนวณกระแสลัดวงจรที่บัส 2



Fault current distribution:

$I_F = 2504$ A
$I_{12} = 460$ A
$I_{14} = 344$ A
$I_{24a} = 229$ A
$I_{24b} = 229$ A

รูปที่ 9.24 ผลการคำนวณกระแสลัดวงจรที่บัส 5

เพื่อให้การปรับตั้งรีเลย์ป้องกันสำหรับระบบไฟฟ้ากำลังตัวอย่างนี้ มีความเหมาะสมที่สุด จำเป็นต้องกำหนดฟังก์ชันวัตถุประสงค์และเงื่อนไขบังคับ ในที่นี้ใช้ผลรวมของเวลาทำงานของรีเลย์ทุกตัวเป็นฟังก์ชันวัตถุประสงค์ และให้ผลต่างของเวลาทำงานระหว่างรีเลย์สองตัวใด ๆ ที่อยู่ในตำแหน่งต้นน้ำ (upstream) กับปลายน้ำ (downstream) ซึ่งกันและกันมีค่าไม่เกินค่าเวลาส่วนต่าง (time grading margin) ของรีเลย์ ดังสมการต่อไปนี้

$$Minimize \qquad F_{obj} = \sum_{k=1}^{m}\left(\sum_{i=1}^{n}\frac{\alpha_i \times TDS_i}{\left(\frac{I_{i,k}}{I_{S,i}}\right)^{\gamma_i}-1}\right) \tag{9.17}$$

$$Subject\,to \qquad t_{u,j}-t_{d,j} > T_{gm} \qquad ; j=1,2,\ldots,J$$

โดยที่

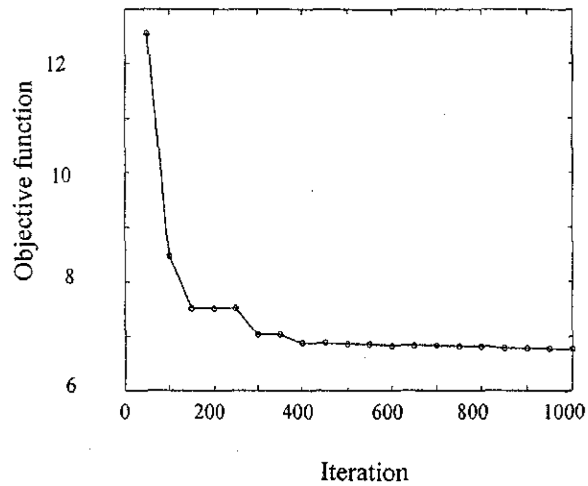
F_{obj}	แทนฟังก์ชันวัตถุประสงค์
α_i และ γ_i	ค่าคงตัวของคุณลักษณะการทำงานของรีเลย์ ตัวที่ i
TDS_i	แทนค่า time-dial setting ของรีเลย์ตัวที่ i
$I_{S,i}$	แทนค่ากระแส pickup ของรีเลย์ตัวที่ i
$I_{i,k}$	แทนค่ากระแสวงจรที่รีเลย์ตัวที่ i ตรวจจับได้สำหรับ กรณีลัดวงจรที่ k
T_{gm}	แทนค่าเวลาส่วนต่าง (time grading margin) ที่กำหนด
$T_{u,j}$และ $T_{d,j}$	แทนค่าเวลาการทำงานของรีเลย์ต้นน้ำและปลายน้ำตามลำดับ
n, m และ J	แทนจำนวนรีเลย์ที่ปรับตั้ง จำนวนกรณีที่เกิดลัดวงจร และจำนวนคู่ของรีเลย์ต้นน้ำและปลายน้ำ

รีเลย์ทุกตัวกำหนดให้เป็นรีเลย์แบบดิจิตอล มีค่าพารามิเตอร์การทำงานเป็นแบบเส้นโค้งผกผันมาตรฐาน α = 0.14 และ γ = 0.02 (standard inverse) [9] กำหนดขอบเขตการค้นหาในช่วง [0.05 1.00] สำหรับค่า *TDS* ของรีเลย์ทุกตัวที่ทำการปรับตั้ง โดยใช้การค้นหาตาบูเชิงปรับตัว จะได้ผลเฉลยเหมาะที่สุดดังตารางที่ 9.6 โดยมีกราฟที่เกี่ยวข้องแสดงในรูปที่ 9.25 – 9.26

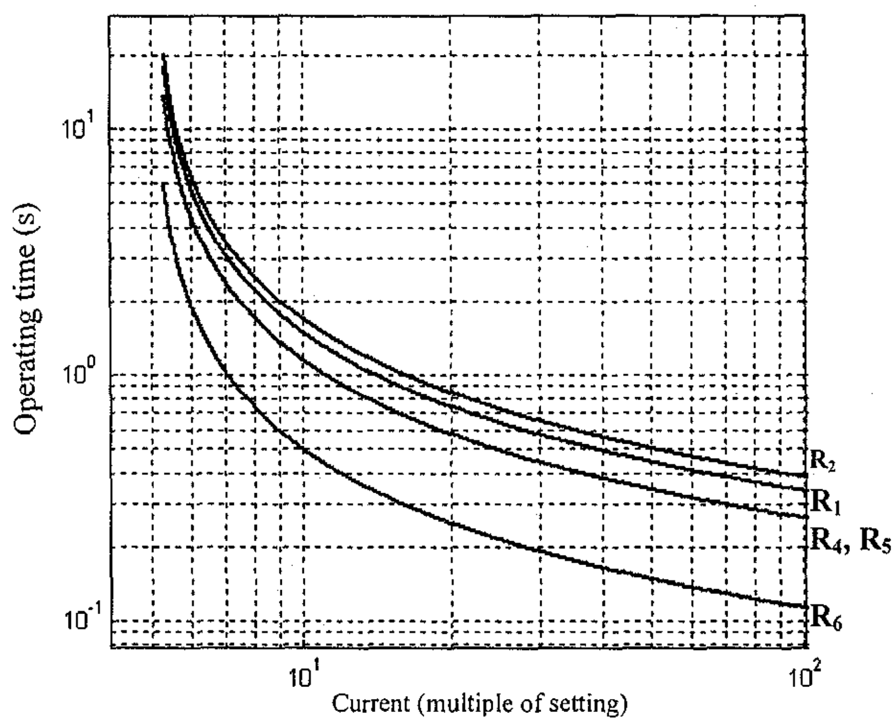
ตารางที่ 9.6 ผลเฉลยการปรับตั้งรีเลย์ดิจิตอลโดยใช้การค้นหาตาบูเชิงปรับตัว

Relay number	TDS	CT's ratio	Operating time (s)	
			Fault at bus 2	Fault at bus 5
1	0.15	150/5	0.71	0.93
2	0.17	100/5	1.27	0.95
4	0.12	50/5	0.86	0.52
5	0.12	50/5	0.86	0.52
6	0.05	50/5	not operate	0.12

หมายเหตุ: relay 3 ไม่เกี่ยวข้องกับกรณีการลัดวงจรทั้งสองที่นำมาพิจารณา



รูปที่ 9.25 กราฟการลู่เข้าของผลเฉลยโดยใช้การค้นหาตาบูเชิงปรับตัว



รูปที่ 9.26 ผลการจัดลำดับความสัมพันธ์ของรีเลย์ป้องกัน

9.7 สรุป

บทนี้ นำเสนอรูปแบบการสร้างปัญหาค่าเหมาะที่สุดสำหรับการออกแบบในงานทางด้านวิศวกรรมไฟฟ้า เน้นไปที่การประยุกต์กับงานทางด้านการวิเคราะห์การทำงานและการวางแผนระบบไฟฟ้ากำลัง เพื่อให้เข้าใจถึงขั้นตอน การกำหนดตัวแปร ฟังก์ชันวัตถุประสงค์ การเลือกเงื่อนไขบังคับ ตลอดจนการเลือกใช้เทคนิคการแก้ปัญหา จะสังเกตได้อย่างชัดเจนว่า งานวิจัยทางด้านระบบไฟฟ้ากำลังนี้มีความก้าวหน้าไปมาก การประยุกต์เทคนิคการหาค่าเหมาะที่สุดและเทคนิคชาญฉลาดเป็นเรื่องปกติสำหรับงานวิจัยทางด้านนี้ หัวข้อย่อยที่ปรากฏในเอกสารผู้เขียนได้

รวบรวมตัวอย่างที่ไม่ยากเกินไป และแทรกการเชื่อมโยงไปสู่งานวิจัยที่ใช้เทคนิคการหาค่าเหมาะที่สุดโดยอ้างอิงจากงานวิจัยของผู้เขียนเป็นสำคัญ ผู้เขียนหวังเป็นอย่างยิ่งว่า นักศึกษา นักวิจัยและผู้สนใจจะได้นำแนวความคิดของการประยุกต์เทคนิคการหาค่าเหมาะที่สุด ไปใช้ในการแก้ปัญหาในงานวิจัยทางวิศวกรรมได้

9.8 เอกสารอ้างอิง

[1] P.K. NAG, **Power plant engineering**, McGraw-Hill, 2002

[2] H. SAADAT, **Power system analysis**, McGraw-Hill, 2004

[3] ธนัดชัย กุลวรวานิชพงษ์, *การจ่ายโหลดอย่างประหยัดของโรงไฟฟ้าพลังความร้อนโดยใช้การโปรแกรมวิวัฒนาการ*, **วารสารเทคโนโลยีสุรนารี**, ปีที่ 9, ฉบับที่ 3, กรกฎาคม – กันยายน, 2545

[4] D.C. Walters & G.B. Sheble, *Genetic algorithm solution of economic dispatch with valve-point loading*, **IEEE Trans. On Power Systems**, Vol. 8, No. 3, 1993

[5] H. Yang, P. Yang & C. Huang, *Evolutionary programming based economic dispacth for units with non-smooth fuel cost functions*, **IEEE Trans. on Power Systems**, Vol. 11, No. 9, 1996

[6] ธนัดชัย กุลวรวานิชพงษ์, **การทำงานที่เหมาะสมในระบบไฟฟ้ากำลังโดยใช้การตัดสินใจแบบฟัซซี**, วิทยานิพนธ์มหาบัณฑิต, จุฬาลงกรณ์มหาวิทยาลัย, พ.ศ. 2543

[7] L. Grigsby, **Electric power engineering handbook**, CRC Press, Boca Raton, 2001

[8] William H. Kersting, **Distribution system modeling and analysis**, CRC Press, Boca Raton, 2002

[9] ทศพล รัตน์นิยมชัย, ธนัดชัย กุลวรวานิชพงษ์, *กำลังงานสูญเสียน้อยที่สุดในระบบจำหน่าย 22-kV ด้วยการควบคุมตัวฟื้นฟูแรงดันพลวัต*, **การประชุมวิชาการวิศวกรรมไฟฟ้าแห่งประเทศไทย ครั้งที่ 28 (EECON28)**, ภูเก็ต, 20 – 21 ตุลาคม 2548

[10] U. Thongkrajay, N. Poolsawat, T. Ratniyomchai, T. Kulworawanichpong, "Unbalanced Three-phase Distribution Power Flow Using Alternative Newton-Raphson Method", **The WSEAS Transactions on Circuits and Systems**, Issue 3, Vol 5, pp. 403-410

[11] S. Srithorn, K. Khojulklang & T. Kulowrawanichpong, *Capacitor switching control using a decision table for a 115-kV power transmission system in Thailand*, **Lecture Notes in Artificial Intelligences LNAI 3215 Part I**, pp. 1262 - 1268, 2004

[12] T. Kulworawanichpong & C.J. Goodman, *Optimal area control of AC railway systems via PWM traction drives*, **IEE Proceedings Part B: Electric Power Applications**, Vol. 152, No. 1, pp. 33 – 40, January 2005

[13] A.J. Wood & B.F. Wollenberg, **Power generation, operation, and control**, John Wiley & Sons, New York, 1996

[14] T. Kulowrawanichpong & S. Sujitjorn, *Optimal power flow using Tabu search*, **IEEE Power Engineering Review**, Power Engineering Letter, pp. 37- 40, July 2002

[15] T. Ratniyomchai, T. Kulworawanichpong, *Modeling of a DVR for Newton-Raphson Power Flows*, **The WSEAS Transactions on Circuits and Systems**, Issue 3, Vol 5, pp. 913-918

[16] T. Kulworawanichpong, T. Ratniyomchai & B. Borriboon, *An alternative Newton-Raphson power flow method based on current-balanced equations*, **IASTED International Conference on Modelling, Identification and Control (MIC2005)**, Innsbruck, Austria, 16 – 18 February 2005

[17] X. Wang & J.R. McDonald, **Modern power system planning**, McGraw-Hill, New York, 1994

[18] H.L. Willis, **Spatial electric load forecasting**, Marcel Dekker, 1996

[19] สราวุฒิ สุจิตจร และ ธนัดชัย กุลวรวานิชพงษ์, *การประยุกต์ใช้ตรรกศาสตร์ฟัซซี่สำหรับการพยากรณ์ความต้องการไฟฟ้าระยะสั้น*, **การประชุมวิชาการวิศวกรรมไฟฟ้าแห่งประเทศไทย ครั้งที่ 23 (EECON23)**, มหาวิทยาลัยเชียงใหม่, เชียงใหม่, ธันวาคม 2543

[20] S.H. Horowitz, A.G. Phadke, **Power System Relaying**, RSP

[21] P.M. Anderson, **Power System Protection**, McGraw-Hill

[22] J.L. Blackburn, ***Protective Relaying: Principles and Applications***, Marcel Dekker

[23] J.M. Gers & E.J. Holmes, ***Protection of Electricity Distribution Networks***, IEE Press

[24] T. Kulworawanichpong, K-N. Areerak & S. Sujitjorn, *Moving towards a new era of intelligent protection through digital relaying in power systems*, **Lecture Notes in Artificial Intelligences LNAI 3215 Part I**, pp. 1255 – 1261, 2004

บรรณานุกรม

1. S.G. Nash & A. Sofer, **Linear and nonlinear programming**, McGraw-Hill,1996
2. S.S. Rao, **Engineering optimization: Theory and practices**, Wiley-Interscience, 1996
3. G.V. Reklaitis, A. Ravindran & K.M. Ragsdell, **Engineering optimization: methods and applications**, Wiley-Interscience, 1983
4. M.S. Bazaraa, H.D. Sherali & C.M. Shetty, **Nonlinear programming: Theory and algorithms**, John Wiley & Sons, 1993
5. J.J. McKeown, D. Meegan & D. Dprevak, **An introduction to unconstrained optimization**, Adam Hilger, 1990
6. E.K.P. Chong & S.H. Zak, **An introduction to optimization**, Wiley-Interscience, 2001
7. P. Venkataraman, **Applied optimization with MATLAB™ Programming**, Wiley-Interscience, 2002
8. J.A. Momoh, **Electric power system applications of optimization**, Marcel Dekker, 2001
9. L. Fortuna, G, Rizzotto, M. Lavorgna, G. Nunnari, M.G. Xibilia & R. Caponetto, **Soft computing: New trends and applications**, Springer Verlag, 2001
10. D.T. Pham & D. Karaboga, **Intelligent optimisation techniques**, Springer Verlag, 2000
11. S. Sujitjorn, T. Kulworawanichpong, D. Puangdownreong & K-L Areerak, **Integrated Intelligent Systems for Engineering Design**, IOS Press, The Netherlands, 2006
12. D.B. Fogel, **Evolutionary computation**, IEEE Press, 1995
13. L. Davis, **Genetic algorithms and simulated annealing**, Pitman, 1987

ผนวก ก. พีชคณิตเชิงเส้น

ก.1 สิ่งที่ควรรู้ เกี่ยวกับเซต

- ทฤษฎีเซตเบื้องต้น: การกระทำเกี่ยวกับเซต
- การนำเนินการไบนารีบนเซตจำนวนจริง
  - การบวก (Addition): "*" หรือ "·"
  - การคูณ (Multiplication): "+"
- สัญลักษณ์ที่ควรรู้
  - $\in, \notin, \cup, \cap, \subset, \not\subset, \exists, \forall$
  - $\wedge, \vee, \Rightarrow, \Leftrightarrow$
- การพิสูจน์ทางคณิตศาสตร์ (Mathematical Proof)

- <u>Direct Proof</u>

การพิสูจน์ทฤษฎีบททางคณิตศาสตร์ต้องใช้การตีความทางตรรกศาสตร์เข้าช่วย รูปแบบของทฤษฎีบทต่าง ๆ มักถูกเขียนด้วยประโยคทางตรรกศาสตร์อย่างง่าย ๆ ประกอบด้วยนิพจน์ที่เชื่อมกันด้วยตัวเชื่อมต่อไปนี้ (และ "$\wedge$", หรือ "$\vee$", ถ้า...แล้ว... "$\Rightarrow$", ก็ต่อเมื่อ "$\Leftrightarrow$")

ประโยคที่นิยมใช้ในทางคณิตศาสตร์ คือ ประโยคที่เชื่อมด้วยตัวเชื่อม $\Rightarrow$ และ $\Leftrightarrow$ การพิสูจน์ประโยคที่เชื่อมด้วย $\Rightarrow$ ตัวอย่างเช่น $P \Rightarrow Q$ ประโยครูปแบบนี้จะเป็นเท็จกรณีเดียว คือ

$P = $ "T" $\wedge$ $Q =$ "F"

การพิสูจน์ประโยคที่เชื่อมด้วย $\Leftrightarrow$: ตัวอย่างเช่น $P \Leftrightarrow Q$ ประโยคในรูปแบบนี้ประกอบด้วย $P \Rightarrow Q \wedge P \Leftarrow Q$

ตัวอย่างที่ ก.1 จงพิสูจน์ว่า $(A \cup B) \cap B' = A \Leftrightarrow A \cap B = \phi$

<u>**วิธีทำ**</u> i) พิสูจน์ว่า $P \Rightarrow Q$ เป็นจริง

$$
\begin{aligned}
(A \cup B) \cap B' = A &\Rightarrow (A \cap B') \cup (B \cap B') \\
&\Rightarrow (A \cap B') \cup \phi \\
&\Rightarrow A \cap B' = A
\end{aligned}
$$

$$
\begin{aligned}
\therefore A \cap B &= A \cap (U - B') = (A \cap U) - (A \cap B') \\
&= A - A = \phi \quad : (P \Rightarrow Q) \textit{ is true } \#\#\#
\end{aligned}
$$

$$
\begin{aligned}
A \cap B = \phi \quad is \ "T" \qquad &\Rightarrow (A \cup B) \cap B' = A \\
&\Rightarrow (A \cap B') \cup (B \cap B') = A \\
&\Rightarrow A \cap B' = A \cap (U - B) = A \\
&\Rightarrow (A \cap U) - (A \cap B) = A \\
&\Rightarrow A - (A \cap B) = A
\end{aligned}
$$

ดังนั้น $A \cap B = \phi$: $(P \Leftarrow Q)$ is true ###

$$(A \cup B) \cap B' = A \Leftrightarrow A \cap B = \phi$$ เป็นจริง

- **Proof by contradiction**

ใช้ในการพิสูจน์นิพจน์ใด ๆ ว่าเป็นจริงหรือเท็จ โดยอาศัยการเชื่อมประโยคด้วย $\Rightarrow$ "*การทดลองที่สอดคล้องกับทฤษฎีไม่ว่าจะมากมายเพียงใดก็ไม่สามารถพิสูจน์ได้อย่างชัดเจนว่าทฤษฎีนั้นเป็นจริง แต่การทดลองที่ไม่สอดคล้องกับทฤษฎีเพียงการทดลองเดียว สามารถพิสูจน์ได้อย่างชัดเจนว่า ทฤษฎีนั้นเป็นเท็จ*"

1. สมมติให้ P เป็นจริง
2. ถ้าพิสูจน์ได้ว่า Q เป็นเท็จ ดังนั้น P เป็นเท็จ
3. แต่ถ้าไม่ใช่ตามข้อ 2. ดังนั้น P เป็นจริง

ตัวอย่างที่ ก.2 จงพิสูจน์ว่า $\sqrt{2}$ ไม่ใช่จำนวนตรรกยะ

Sol^n ถ้า $x = \sqrt{2}$ เป็นจำนวนตรรกยะ แล้ว $x = p/q$; $p,q \in I$ โดยที่ p และ q ต้องไม่มีตัวหารร่วม แต่ $x^2 = 2 = p^2/q^2$ หรือ $p^2 = 2q^2$ จะพบว่า p และ q ต้องเป็นจำนวนคู่ ดังนั้น p และ q มีตัวหารร่วม คือ 2 *เพราะฉะนั้น* $\sqrt{2}$ *ไม่ใช่จำนวนตรรกยะ* #####

ก. 2 สนามสเกลาร์

Field: F = (S,+,*) เป็นระบบนามธรรมที่มีสมาชิกเป็นสเกลาร์จำนวนตั้งแต่สองตัวขึ้นไป ประกอบด้วยเซตที่ไม่ใช่เซตว่าง F นิยามภายใต้การดำเนินการทางการบวก (addition: +) และ การคูณ(multiplication: *) และมีคุณสมบัติดังนี้

- **Addition** {S,+}

  A1: คุณสมบัติปิดการบวก (closure under addition)

  $a,b \in F \quad \Rightarrow (a + b) = (b + a) \in F$

  A2: คุณสมบัติการจัดกลุ่ม (associative properties)

  $a,b,c \in F \quad \Rightarrow a + (b + c) = (a + b) + c$

  A3: คุณสมบัติการมีเอกลักษณ์การบวก (identity element)

  $0, a \in F \quad \Rightarrow a + 0 = 0 + a = a$

  A4: คุณสมบัติการมีอินเวอร์สการบวก (Inverse element)

  $a, -a \in F \quad \Rightarrow a + (-a) = (-a) + a = 0$

- **Multiplication** {S,*}

  M1: คุณสมบัติปิดการคูณ (closure under multiplication)

  $a,b \in F \quad \Rightarrow (a * b) = (b * a) \in F$

  M2: คุณสมบัติการจัดกลุ่ม (associative properties)

  $a,b,c \in F \quad \Rightarrow a * (b * c) = (a * b) * c$

  M3: คุณสมบัติการมีเอกลักษณ์การคูณ (identity element)

  $1, a \in F \quad \Rightarrow a * 1 = 1 * a = a$

M4: <u>คุณสมบัติการมีอินเวอร์สการคูณ</u> (inverse element)

$$a, a^{-1} \in F \qquad \Rightarrow a * a^{-1} = a^{-1} * a = 1$$

- **คุณสมบัติการแจกแจง**(Distributive properties)

  กำหนดให้ $a, b, c \in F$

$$(a + b) * c = (a * b) + (b * c)$$
$$a * (b + c) = (a * b) + (a * c)$$

ตัวอย่างของสนามสเกลาร์ที่ควรรู้

- เซตของจำนวนจริง $\Re$ ที่นิยามภายใต้การบวกและคูณปกติ
- เซตของจำนวนเชิงซ้อน $\mathbb{C}$ ที่นิยามภายใต้การบวกและคูณปกติ
- เซตของฟังก์ชันพหุนามใด ๆ
- เซตของเมตริกซ์ที่มีคุณสมบัติดังนี้

$$\begin{bmatrix} x & -y \\ y & x \end{bmatrix} \text{ เมื่อ } x, y \in \Re$$

ก.3 ปริภูมิเวกเตอร์เชิงเส้น

กำหนดปริภูมิเวกเตอร์เชิงเส้น (linear vector space) V นิยามเหนือสนามสเกลาร์ F ภายใต้การดำเนินการทางการบวก (vector addition) และการคูณด้วยสเกลาร์ (scalar multiplication) ปริภูมิเวกเตอร์เชิงเส้นต้องมีคุณสมบัติดังนี้

- <u>Axioms for vector addition</u>

  A1: Closure under addition

  $\bar{u}, \bar{v} \in V \quad \Rightarrow \bar{u} + \bar{v} \in V$

  A2: Commutative law

  $\bar{u}, \bar{v} \in V \quad \Rightarrow \bar{u} + \bar{v} = \bar{v} + \bar{u}$

  A3: Associative law

  $\bar{u}, \bar{v}, \bar{w} \in V \quad \Rightarrow \bar{u} + (\bar{v} + \bar{w}) = (\bar{u} + \bar{v}) + \bar{w}$

  A4: Zero vector

  $\bar{u}, \bar{0} \in V \quad \Rightarrow \bar{0} + \bar{u} = \bar{u} + \bar{0} = \bar{u}$

  A5: Negative of a vector

  $\bar{u}, -\bar{u} \in V \quad \Rightarrow \bar{u} + (-\bar{u}) = (-\bar{u}) + \bar{u} = \bar{0}$

- <u>Axioms for scalar multiplication</u>

  M1: Closure under scalar multiplication

  $k \in F; \bar{u} \in V \quad \Rightarrow k\bar{u} \in V$

  M2: Associative law

  $k_1, k_2 \in F; \bar{u} \in V \quad \Rightarrow k_1(k_2\bar{u}) = (k_1k_2)\bar{u}$

  M3: Distributive law

  $k_1, k_2 \in F; \bar{u}, \bar{v} \in V \quad \Rightarrow k_1(\bar{u} + \bar{v}) = k_1\bar{u} + k_1\bar{v}$

  $\Rightarrow (k_1 + k_2)\bar{u} = k_1\bar{u} + k_2\bar{u}$

ตัวอย่างที่ ก.3 กำหนดเซต V มีสมาชิกเป็นจำนวนจริงบวกและมีนิยามการบวกและการคูณเวกเตอร์ด้วยสเกลาร์ ดังนี้ $\bar{x}+\bar{y}=x\cdot y$ และ $k\bar{x}=x^k$ ตามลำดับ

วิธีทำ

A1: (closure)

$$\bar{x}=x,\bar{y}=y\in V\wedge(x,y>0) \quad \Rightarrow \bar{x}+\bar{y}=(x\cdot y)>0\in V$$

A2: (commutative law)

$$\bar{x}+\bar{y}=x\cdot y=y\cdot x=\bar{y}+\bar{x}$$

A3: (associative law)

$$(\bar{x}+\bar{y})+\bar{z}=\overline{(x\cdot y)}+z=(x\cdot y)\cdot z$$

$$=x\cdot(y\cdot z)=\bar{x}+\overline{(y\cdot z)}=\bar{x}+(\bar{y}+\bar{z})$$

A4: (zero vector)

$$\bar{0}+\bar{u}=\bar{u}+\bar{0}=\bar{u}$$

$$\bar{1}+\bar{x}=1\cdot x=x=\bar{x} \qquad \therefore\ 1 \text{ เป็นเวกเตอร์ศูนย์}$$

A5: (negative of a vector)

$$\bar{u}+(-\bar{u})=\bar{0}\ (\text{zero vector})$$

$$\bar{x}+\overline{\left(\frac{1}{x}\right)}=x\cdot\frac{1}{x}=1=\bar{1}\ (\text{zero vector})$$

M1: (closure) กำหนดให้ k เป็นสเกลาร์ใด ๆ และ $x>0$

$$k\bar{x}=x^k \quad \Rightarrow x^k>0$$

M2: (associative law) กำหนดให้ k_1, k_2 เป็นสเกลาร์ใด ๆ

$$k_1(k_2x)=k_1\overline{(x^{k_2})}=(x^{k_2})^{k_1}=x^{k_2k_1}=x^{k_1k_2}=(k_1k_2)\bar{x}$$

M3: (distributive law)

$$(k_1+k_2)\bar{x}=x^{(k_1+k_2)}=x^{k_1}x^{k_2}=\overline{(x^{k_1})}+\overline{(x^{k_2})}=k_1\bar{x}+k_2\bar{x}$$

$$k(\bar{x}+\bar{y})=(\bar{x}+\bar{y})^k=(x\cdot y)^k=x^k\cdot y^k=\overline{(x^k)}+\overline{(y^k)}=k\bar{x}+k\bar{y}$$

ปริภูมิเวกเตอร์ที่ควรรู้

- เซตของจำนวนจริง $\Re,\Re^2,\Re^3,\ldots,\Re^n,\ldots$
- เซตของจำนวนเชิงซ้อน $\mathbb{C},\mathbb{C}^2,\mathbb{C}^3,\ldots,\mathbb{C}^n,\ldots$
- เซตของพหุนาม $P_n(x)$ ที่มีดีกรี $\le$ n ใด ๆ

ก.4 ปริภูมิย่อยและผลรวมเชิงเส้น

ถ้าเซตย่อย W ของปริภูมิเวกเตอร์ V มีคุณสมบัติเป็นปริภูมิเวกเตอร์ด้วยตัวของมันเองภายใต้เงื่อนไขการบวกเวกเตอร์และการคูณด้วยสเกลาร์ที่นิยามบน V แล้ว จะเรียก W ว่าเป็นปริภูมิย่อย (subspace) ของ V

การพิสูจน์ว่า W เป็นปริภูมิย่อยหรือไม่นั้นไม่จำเป็นที่จะต้องพิสูจน์ตาม A1 – A5 และ M1 – M3 เพียงแค่พิสูจน์ว่า W มีคุณสมบัติภายใต้การบวกและการคูณด้วยสเกลาร์ก็เพียงพอ

ถ้า $W\subset V$ และ $W\neq\phi$ แล้ว W จะเป็นปริภูมิย่อยก็ต่อเมื่อ

W1: $\vec{u},\vec{v}\in W \quad \Rightarrow \vec{u}+\vec{v}\in W$

W2: $k\in F,\vec{u}\in W \quad \Rightarrow k\vec{u}\in W$

ตัวอย่างที่ ก.4 จงพิสูจน์ว่าเซตของพหุนามขั้น n $\left\{P_n^{\perp}(x)\right\}$ ที่สอดคล้องกับเงื่อนไข $\frac{d^2P_n(x)}{dx^2}=0$ เป็นปริภูมิย่อยของ $P_n(x)$ หรือไม่

วิธีทำ

W1: $\vec{p}_1,\vec{p}_2,\vec{p}_3\in P_n^{\perp}(x) \ \wedge \frac{d^2P_1(x)}{dx^2}=0,\frac{d^2P_2(x)}{dx^2}=0$

$$\vec{p}_3=\vec{p}_1+\vec{p}_2 \quad \Rightarrow \frac{d^2P_3(x)}{dx^2}=\frac{d^2(P_1(x)+P_2(x))}{dx^2}=\frac{d^2P_1(x)}{dx^2}+\frac{d^2P_2(x)}{dx^2}=0$$

$\therefore \vec{p}_1+\vec{p}_2\in P_n^{\perp}(x)$ ######

W2: $k\in F,\vec{p}\in P_n^{\perp}(x) \ \wedge \frac{d^2P(x)}{dx^2}=0$

$$\frac{d^2(k\vec{p})}{dx^2}=\frac{d^2\{kP(x)\}}{dx^2}=k\frac{d^2P(x)}{dx^2}=0 \qquad \therefore k\vec{p}\in P_n^{\perp}(x)$$ ######

นิยาม กำหนดเซตของเวกเตอร์จำนวน k ตัวใด ๆ ในปริภูมิเวกเตอร์ V ดังนี้ $\{\vec{v}_1,\vec{v}_2,\ldots,\vec{v}_k\}\subset v$ และเซตของสเกลาร์ k ตัว โดยที่ $\{a_1,a_2,\ldots,a_k\}\subset F$ จะเรียกเวกเตอร์ $\vec{w}\in V$ ว่าเป็นผลบวกเชิงเส้น (Linear combination) ของเวกเตอร์ $\{\vec{v}_1,\vec{v}_2,\ldots,\vec{v}_k\}$ ก็ต่อเมื่อ

$$\vec{w}=a_1\vec{v}_1+a_2\vec{v}_2+\ldots+a_k\vec{v}_k$$

ตัวอย่าง ก.5 จงหาว่า $\vec{u}=(1,-1,1)$ เป็นผลบวกเชิงเส้นของ $\vec{u}_1=(2,1,-1)$, $\vec{u}_2=(1,-7,5)$, และ $\vec{u}_3=(0,0,1)$ หรือไม่

วิธีทำ ให้ $\vec{u}=a_1\vec{u}_1+a_2\vec{u}_2+a_3\vec{u}_3$ ถ้าสามารถหาค่า a_1, a_2, a_3 ได้ แสดงว่า $\vec{u}$ เป็นผลบวกเชิงเส้นของ $\{\vec{u}_1,\vec{u}_2,\vec{u}_3\}$ จริง

$$(1,-1,1)=a_1(2,1,-1)+a_2(1,-7,5)+a_3(0,0,1)$$
$$=(2a_1+a_2,a_1-7a_2,-a_1+5a_2+a_3)$$

จะได้สมการเชิงเส้นดังนี้

$$\left.\begin{aligned}2a_1+a_2+0a_3&=1\\a_1-7a_2+0a_3&=-1\\-a_1+5a_2+a_3&=1\end{aligned}\right\}\therefore a_1=0.4,a_2=0.2,a_3=0.4$$

สรุปได้ว่า $\vec{u}$ เป็นผลบวกเชิงเส้นของ $\{\vec{u}_1,\vec{u}_2,\vec{u}_3\}$ จริงในรูป

$$\vec{u}=0.4\vec{u}_1+0.2\vec{u}_2+0.4\vec{u}_3$$ ######

ก.5 อิสระเชิงเส้น ไม่อิสระเชิงเส้น แผ่ทั่วถึง และมูลฐานของปริภูมิเวกเตอร์

กำหนดเซต $\{\vec{u}_1,\vec{u}_2,\ldots,\vec{u}_n\}\subset V$ จะเรียกเซตนี้ว่า มีความเป็นอิสระเชิงเส้น (linearly independent) ถ้ามีค่าคงตัวที่สอดคล้องกับผลรวมเชิงเส้น

$$a_1\vec{u}_1+a_2\vec{u}_2+\ldots+a_n\vec{u}_n=\vec{0}$$ โดยที่ $\forall a_i=0$ เท่านั้น

ถ้า $\exists a_i \neq 0$ แล้ว จะเรียกว่า ไม่เป็นอิสระเชิงเส้น (linearly dependent)

ตัวอย่างที่ ก.6 จงแสดงให้เห็นว่า (4,-8) และ (-6,12) ใน $\Re^2$เป็นอิสระเชิงเส้นต่อกันหรือไม่

วิธีทำ เขียนเวกเตอร์ดังกล่าวในรูปผลรวมเชิงเส้นดังนี้

$$a_1(4,-8) + a_2(-6,12) = (0,0)$$

$$\therefore \left.\begin{aligned} 4a_1 - 6a_2 &= 0 \\ -8a_1 + 12a_2 &= 0 \end{aligned}\right\} a_1 = \frac{3}{2}a_2$$

$\exists a_i \neq 0$ ดังนั้น เวกเตอร์ดังกล่าวไม่เป็นอิสระเชิงเส้นต่อกัน ###

__ตัวอย่างที่ ก.7__ จงแสดงให้เห็นว่า 1, (x+1), $(x+1)^2$ ในปริภูมิ $P_2(x)$ เป็นอิสระเชิงเส้นต่อกันหรือไม่

วิธีทำ เขียนเวกเตอร์ดังกล่าวในรูปผลรวมเชิงเส้นดังนี้

$$a_1(1) + a_2(x+1) + a_3(x+1)^2 = 0$$

$$a_1(1) + a_2(x+1) + a_3(x^2+2x+1) = 0$$

$$a_3x^2 + (a_2+2a_3)x+(a_1+a_2+a_3) = 0$$

จะได้ว่า $a_3 = 0$

$a_2+2a_3 = 0$

$a_1+a_2+a_3 = 0$

$a_1, a_2, a_3 = 0$ (เป็นอิสระเชิงเส้นต่อกัน) ######

เซตของเวกเตอร์ $\{\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_n\} \subset V$ จะเรียกว่า แผ่โดยทั่วถึง (span) V ก็ต่อเมื่อเวกเตอร์ $\bar{v} \in V$ ใด ๆ สามารถเขียนในรูปผลรวมเชิงเส้นของเซต $\{\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_n\}$ ได้

ถ้า $\{\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_n\} \subset V$ เป็นอิสระเชิงเส้นต่อกัน และแผ่ทั่วถึง V (span V) แล้ว จะเรียกเซตนี้ว่าเป็นมูลฐาน (basis) ชุดหนึ่งของปริภูมิ V

มิติของปริภูมิเวกเตอร์มีค่าเท่ากับจำนวนของเวกเตอร์ที่เป็นมูลฐานของปริภูมิ

ตัวอย่างที่ ก.8 จงพิสูจน์ว่า {(1,0,0),(1,1,0),(1,1,1)} เป็นมูลฐานชุดหนึ่งของปริภูมิ $\Re^3$

วิธีทำ

i) __*พิสูจน์ความเป็นอิสระเชิงเส้น*__ ***(Linearly independent)***

$$a_1(1,0,0) + a_2(1,1,0) + a_3(1,1,1) = 0$$

จะได้ $a_1 + a_2 + a_3 = 0$

$a_2 + a_3 = 0$ จะได้ว่า $a_1, a_2, a_3 = 0$

$a_3 = 0$ (__*อิสระเชิงเส้น*__)

ii) __*พิสูจน์ span*__

$$\bar{v} = (x, y, z) \in \Re^3 \quad \Rightarrow \bar{v} = a_1(1,0,0) + a_2(1,1,0) + a_3(1,1,1)$$

$$(x,y,z) = (a_1+ a_2+ a_3, a_2+ a_3, a_3)$$

พบว่า เมื่อ $x, y, z \in \Re$ จะสามารถหาค่า a_1, a_2, a_3 ได้เสมอ

โดยที่ $a_1 = z; a_2 = y - z; a_3 = x - y$

$\therefore$ {(1,0,0),(1,1,0),(1,1,1)} เป็นมูลฐานชุดหนึ่งของปริภูมิ $\Re^3$ ###

กำหนด $\{\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_n\}, \{\bar{f}_1, \bar{f}_2, \ldots, \bar{f}_n\}$ เป็นมูลฐานของปริภูมิเวกเตอร์ V ใด ๆ และถ้า $\bar{v}_{\bar{e}} \in V$ ที่เขียนขึ้นโดยใช้มูลฐาน $\{e_i\}$ แล้ว สามารถที่จะเปลี่ยนรูปการเขียน $\bar{v}_{\bar{e}} \rightarrow \bar{v}_{\bar{f}} \in V$ โดยใช้มูลฐาน $\{f_i\}$ ได้ดังนี้

เขียน $\bar{e}_i$ ในรูปผลรวมเชิงเส้นของ $\{f_i\}$ จะได้ เมตริกซ์การเปลี่ยนมูลฐาน (chang of basis matrix : C) ดังนี้

$$\left.\begin{array}{l} \bar{e}_1 = c_{11}\bar{f}1 + c_{12}\bar{f}_2 + \ldots + c_{1n}\bar{f}_n \\ \bar{e}_2 = c_{21}\bar{f}_1 + c_{22}\bar{f}_2 + \ldots + c_{2n}\bar{f}_n \\ \vdots \quad\quad \vdots \quad\quad \vdots \quad\quad \ddots \quad\quad \vdots \\ \bar{e}_n = c_{n1}\bar{f}_1 + c_{n2}\bar{f}_2 + \ldots + c_{nn}\bar{f}_n \end{array}\right\} \Rightarrow C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}$$

จะได้ความสัมพันธ์ในการเปลี่ยนมูลฐานดังนี้

$$\bar{v}_{\bar{f}} = C\bar{v}_{\bar{e}}$$

ตัวอย่างที่ ก.9 พิจารณามูลฐานปกติ $\{e_i\}$ = {(1,0),(0,1)} ของปริภูมิ $\Re^2$ และกำหนดให้ $\{f_i\}$ = {(1,1),(-1,0)} เป็นมูลฐานอีกชุดหนึ่งของปริภูมินี้ด้วย ถ้ากำหนดให้เวกเตอร์ $\bar{v}_{\bar{e}} = (2,2)$ เป็นเวกเตอร์ในปริภูมิที่เขียนขึ้นภายใต้มูลฐาน $\{e_i\}$ แล้ว จงหา $\bar{v}_{\bar{f}}$ ที่เขียนภายใต้มูลฐาน $\{f_i\}$

วิธีทำ สำหรับ $\bar{e}_1 = (1,0) = c_{11}(1,1) + c_{12}(-1,0)$

แก้สมการจะได้ $c_{11} = 0; c_{12} = -1;$

สำหรับ $\bar{e}_2 = (0,1) = c_{21}(1,1) + c_{22}(-1,0)$

แก้สมการจะได้ $c_{21} = 1; c_{22} = 1;$

จะได้ $$C = \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}$$

ดังนั้น $$\bar{v}_{\bar{f}} = C\bar{v}_{\bar{e}} = \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} -2 \\ 4 \end{bmatrix} = (-2,4)$$ ######

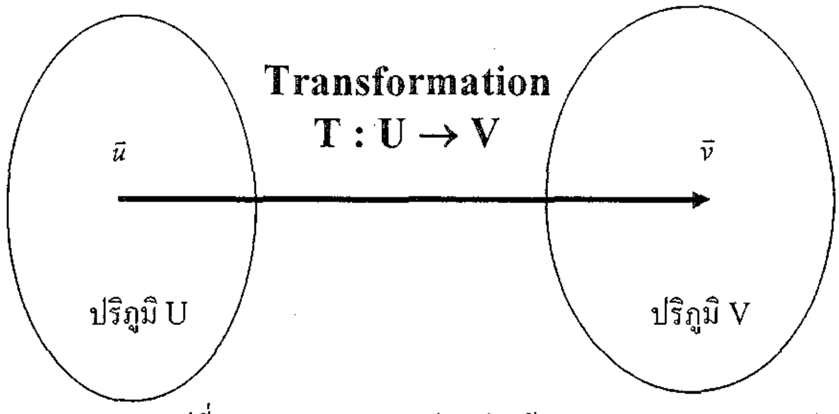
ก.5 การแปลงเชิงเส้น

ฟังก์ชัน (function) การแปลง (transformation) หรือ แมปปิง (mapping) คือ การส่งผ่านค่าของเวกเตอร์จากปริภูมิหนึ่งไปยังอีกปริภูมิหนึ่ง

การแปลง $T : U \rightarrow V$ จะเรียกว่าเป็นการแปลงเชิงเส้นก็ต่อเมื่อการแปลง T มีคุณสมบัติดังนี้

ถ้ากำหนด $\vec{u}, \vec{v} \in U$ และ $k \in F$ แล้ว

$T(\bar{u}+\bar{v}) = T(\bar{u}) + T(\bar{v})$
$T(k\bar{u}) = kT(\bar{u})$



รูปที่ ก.1 แผนภาพการแปลงเชิงเส้น

ตัวอย่างที่ ก.10 จงพิสูจน์ว่าการแปลงต่อไปนี้เป็นการแปลงเชิงเส้นหรือไม่

ก) $T : \Re^2 \to \Re^3$ โดยที่ T(x,y) = (x,2y,x-y)

➢ กำหนด $\bar{u}, \bar{v} \in \Re^2$; $\bar{u} = (x_1, y_1)$, $\bar{v} = (x_2, y_2)$; $k \in \Re$

$$T(\bar{u}+\bar{v}) = T(x_1 + x_2, y_1 + y_2) = (x_1 + x_2, 2(y_1 + y_2), (x_1 + x_2) - (y_1 + y_2))$$
$$= (x_1 + x_2, 2y_1 + 2y_2, (x_1 - y_1) + (x_2 - y_2))$$
$$= (x_1, 2y_1, x_1 - y_1) + (x_2, 2y_2, x_2 - y_2) = T(\bar{u}) + T(\bar{v}) \qquad \#(1)$$
$$T(k\bar{u}) = T(kx_1, ky_1) = (kx_1, 2(ky_1), kx_1 - ky_1) = (k(x_1), k(2y_1), k(x_1 - y_1))$$
$$= k(x_1, 2y_1, x_1 - y_1) = kT(\bar{u}) \qquad \#(2)$$

$\therefore$ T(x,y) = (x,2y,x-y) เป็นการแปลงเชิงเส้น ######

ข) $T : C \to C$ โดยที่ T(z) = 10 – 3z

➢ กำหนด $\bar{z}_1 = (a_1, b_1), \bar{z}_2 = (a_2, b_2) \in C$; $a_1, a_2, b_1, b_2 \in \Re$

* $T(\bar{z})$ = T(a,b) = 10 – 3z = 10 – 3(a,b) = (10 – 3a,-3b)

$$T(\bar{z}_1 + \bar{z}_2) = T(a_1 + a_2, b_1 + b_2) = (10 - 3(a_1 + a_2), -3(b_1 + b_2))$$
$$T(\bar{z}_1) + T(\bar{z}_2) = T(a_1, b_1) + T(a_2, b_2) = (10 - 3a_1, -3b_1) + (10 - 3a_2, -3b_2)$$
$$= (20 - 3(a_1 + a_2), -3(b_1 + b_2))$$

จะพบว่า $T(\bar{z}_1 + \bar{z}_2) \neq T(\bar{z}_1) + T(\bar{z}_2)$

$\therefore T(\bar{z})$ = 10 – 3z ไม่ใช่การแปลงเชิงเส้น #######

ภาพ (image) คือ ค่าการแปลง T ของเวกเตอร์ $\bar{u}$ ใด ๆ จะเรียกว่าเป็นภาพของเวกเตอร์ $\bar{u}$ ภายใต้การแปลง T

เคอร์เนล (kernel) กำหนดการแปลงเชิงเส้น $T : U \to V$ เคอร์เนล คือ เซตของเวกเตอร์ $\bar{u}$ ที่ทำให้ $T(\bar{u}) = \bar{0}$ (ภาพของ $\bar{u}$ เป็นเวกเตอร์ศูนย์)

$$\mathrm{Ker}\,T = \{\bar{u} : \bar{u} \in U \ \text{โดยที่} \ T(\bar{u}) = 0\}$$

ตัวอย่างที่ ก.11 กำหนดการแปลงเชิงเส้น $T : \Re^3 \to \Re^3$ ดังนี้

T(x,y,z) = (-z,x,x+z)

ก) จงหาภาพของ (-3,10,5) ภายใต้การแปลง T

➢ T(-3,10,5) = (-5,-3,2) #####

ข) จงหาเคอร์เนลของการแปลง T

➢ $T(x,y,z) = (-z,x,x+z) = \bar{0} = (0,0,0)$

จะได้ $z = 0$, $x = 0$ และ $y \in \Re$

ดังนั้น $\text{Ker } T = \{ (0,t,0)$ เมื่อ $t \in \Re\}$

ตัวอย่างที่ ก.12 กำหนดเมตริกซ์ $A = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 1 & 2 \end{bmatrix}$ เป็นการแปลงเชิงเส้นจากปริภูมิ $\Re^3$ ไปยัง $\Re^2$

ก) จงหาภาพของ $\begin{bmatrix} 5 \\ 2 \\ -1 \end{bmatrix}$ ภายใต้การแปลงนี้

➢ $T(\bar{u}) = A\bar{u} = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 5-4-1 \\ 0+2-2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ #######

ข) จงหาเคอร์เนลของการแปลงนี้

➢ $T(\bar{u}) = A\bar{u} = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} a-2b+c \\ b+2c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

จะได้ว่า $b = -2c$

$a = 2b - c = -4c - c = -5c$

$\therefore \text{Ker } T = \{(-5c,-2c,c)$ โดยที่ $c \in \Re\}$

กำหนดการแปลงเชิงเส้น $T: U \to V$ กำหนดรูปแบบการถ่ายโอนเวกเตอร์ในรูปเมตริกซ์ $\bar{v}_i = A\bar{u}_i$ และถ้าให้ $\{\bar{e}_1,\bar{e}_2,...,\bar{e}_n\}$, $\{\bar{f}_1,\bar{f}_2,...,\bar{f}_m\}$ เป็นมูลฐานชุดหนึ่งของปริภูมิ U และ V ตามลำดับแล้ว จะเรียกเมตริกซ์ A ว่าเป็น matrix representation ของการแปลง T ที่มีสัมประสิทธิ์เป็นสเกลาร์ในสนาม F โดย A สามารถนำเสนอได้ดังนี้

$$\left.\begin{array}{l} T(\bar{f}_1) = a_{11}\bar{e}_1 + a_{12}\bar{e}_2 + ... + a_{1n}\bar{e}_n \\ T(\bar{f}_2) = a_{21}\bar{e}_1 + a_{22}\bar{e}_2 + ... + a_{2n}\bar{e}_n \\ \quad\vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad \ddots \qquad \vdots \\ T(\bar{f}_m) = a_{m1}\bar{e}_1 + a_{m2}\bar{e}_2 + ... + a_{mn}\bar{e}_n \end{array}\right\} \therefore A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}^T$$

ตัวอย่างที่ ก.13 กำหนดการแปลงเชิงเส้น $F : \Re^3 \to \Re^2$ โดยที่ F(x,y,z) = (2x-4y+9z,5x+3y-2z) จงหารูปแบบเมตริกซ์ของการแปลงนี้เทียบกับมูลฐานต่อไปนี้

ก. มูลฐานปกติของปริภูมิ $\Re^3$ และ $\Re^2$ ตามลำดับ

➢ $F(1,0,0) = a_{11}(1,0) + a_{12}(0,1) = (2,5)$; $a_{11} = 2$, $a_{12} = 5$

$F(0,1,0) = a_{21}(1,0) + a_{22}(0,1) = (-4,3); a_{21} = -4$, $a_{22} = 3$
$F(0,0,1) = a_{31}(1,0) + a_{32}(0,1) = (9,-2);$ $a_{31} = 9$, $a_{32} = -2$

$$\therefore A = \begin{bmatrix} 2 & 5 \\ -4 & 3 \\ 9 & 2 \end{bmatrix}^T = \begin{bmatrix} 2 & -4 & 9 \\ 5 & 3 & 2 \end{bmatrix} \quad \text{######}$$

ข. {(0,0,1),(1,0,1),(0,1,1)} และ {(0,1),(-1,1)}

➢ $F(0,0,1) = a_{11}(0,1) + a_{12}(-1,1) = (9,-2);$ $a_{11} = 11$, $a_{12} = -9$
$F(1,0,1) = a_{21}(0,1) + a_{22}(-1,1) = (11,3);$ $a_{21} = 14$, $a_{22} = -11$
$F(0,1,1) = a_{31}(0,1) + a_{32}(-1,1) = (5,1); a_{31} = 6$, $a_{32} = -5$

$$\therefore A = \begin{bmatrix} 11 & -9 \\ 14 & -11 \\ 6 & -5 \end{bmatrix}^T = \begin{bmatrix} 11 & 14 & 6 \\ -9 & -11 & -5 \end{bmatrix} \quad \text{######}$$

การแปลงเชิงเส้น T: U → V จะเป็น<u>*การแปลงเอกฐาน*</u> (singular transformation) ก็ต่อเมื่อ มีเวกเตอร์ $\bar{u}$ บางตัวที่ไม่ใช่เวกเตอร์ศูนย์ใน U ซึ่งมีภาพใน V เป็น $\bar{0}$ แต่ถ้า Ker T ={$\bar{0}$} แล้ว จะเรียกว่า<u>*การแปลงไม่เอกฐาน*</u>

การแปลงเอกลักษณ์ I: U → U โดยที่ $I(\bar{u}) = \bar{u}$

ถ้ากำหนดการแปลงเชิงเส้น T: U → V แล้ว การแปลงผกผันของการแปลง T นิยามโดย T^{-1} : V → U และการแปลงนี้จะมีอยู่ก็ต่อเมื่อ การแปลง T เป็นการแปลงแบบไม่เอกฐาน

ตัวอย่างที่ ก.14 กำหนดการแปลงเชิงเส้น $F : \Re^3 \to \Re^2$ และ $G : \Re^2 \to \Re^2$

โดยที่ F(x,y,z) = (x+y-z,0) และ G(x,y) = (2x+y,-y)

ก. การแปลง F และ G เป็นการแปลงแบบเอกฐานหรือไม่

➢ หา Ker F และ Ker G

Ker F : (x+y-z,0) = (0,0) ∴Ker F ={(x,y,z) : x+y-z = 0}
ดังนั้น F เป็นการแปลงเอกฐาน ######
Ker G : (2x+y,-y) = (0,0) ∴Ker G = {(0,0)}
ดังนั้น G เป็นการแปลงไม่เอกฐาน ######

ข. จงหาการแปลงผกผันของ F และ G

➢ F เป็นการแปลงเอกฐานดังนั้นไม่มีการแปลงผกผัน

$G^{-1}(x,y) = (a,b) : G(a,b) = (x,y) = (2a+b,-b)$

จะได้ $2a + b = x$

$-b = y$

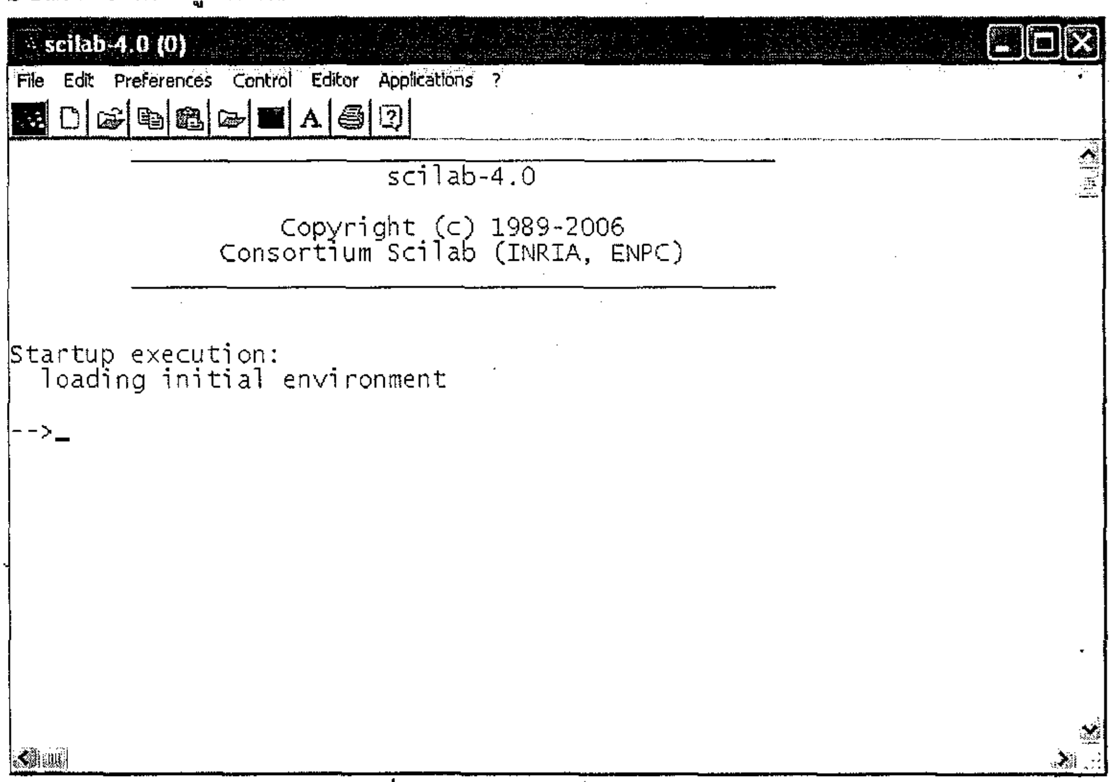
แก้สมการได้ $b = -y$ และ $a = (x - b)/2 = 0.5x+0.5y$

ดังนั้น $G^{-1}(x,y) = (-y,0.5x+0.5y)$ ######

ผนวก ข. การใช้งาน SCILAB เบื้องต้น

ข.1 เกริ่นนำ

โปรแกรม SCILAB™ เป็นโปรแกรมคำนวณเชิงตัวเลขภายใต้สภาวะแวดล้อมเชิงกราฟิก (numerical programming and graphic environment) เช่นเดียวกับ MATLAB™ เดิมมีชื่อว่า Basile เริ่มให้ดาวน์โหลดใช้ฟรีในปี ค. ศ. 1994 พัฒนาขึ้นโดย INRIA (Institut Nationale de Recherche en Informatique et en Automatique หรือ National Institute for Informatics and Automation Research) โดยเปลี่ยนมาใช้ชื่อ SCILAB ในปี ค. ศ. 2004 โปรแกรม SCILAB เป็นซอฟท์แวร์โอเพนซอร์ส (open source) สามารถดาวน์โหลดซอฟท์แวร์แพคเกจและซอร์สโคดของซอฟท์แวร์ได้ฟรีที่ http://www.scilab.org/ ตัวซอฟท์แวร์รันได้ทั้งบนแพลตฟอร์ม Windows Unix และ Linux เป็นซอฟท์แวร์แพ็คเกจขนาดใหญ่ ประกอบด้วยไฟล์ประมาณ 13,000 ไฟล์ สำหรับเวอร์ชัน 3.0 พิจารณารูปหน้าต่างคำสั่ง (command window) ของ SCILAB ได้ในรูปที่ ข.1



รูปที่ ข.1 หน้าต่างคำสั่งของ SCILAB

ข.2 ข้อกำหนดพื้นฐาน

ข.2.1 <u>หลักการตั้งชื่อตัวแปร</u>

- ชื่อตัวแปรจะต้องขึ้นต้นด้วยตัวอักษรเท่านั้น

- ชื่อตัวแปรจะยาวเท่าไรก็ได้ จะบรรจุด้วยตัวอักษร, ตัวเลข หรือ อักษรพิเศษ เช่น underscore "_" dollar sign "$" เป็นต้น
- การตั้งชื่อตัวแปรจะคำนึงถึง ตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่ด้วย จะถือว่า AB, Ab, aB, และ ab เป็นตัวแปรคนละตัวกัน
- โปรแกรม SCILAB จะแยกตัวแปรสองตัวออกจากกัน โดยพิจารณาอักขระของชื่อเพียง 24 ตัวแรกเท่านั้น นั่นคือ การตั้งชื่อยาวเกิน 24 ตัวอักษรจะไม่มีความหมายใด ๆ

ข.2.2 <u>การกำหนดค่าตัวแปร</u>

- ถ้าค่าของตัวแปรเป็นสเกลาร์ใด ๆ สามารถที่จะกำหนดค่าตัวแปรได้โดยตรง เช่น A=3
- ถ้าค่าของตัวแปรเป็นเวกเตอร์ ต้องกำหนดค่าภายในเครื่องหมาย [ ]

  โดยที่การกำหนดค่าในตำแหน่งแถวนอน อาจแยกสมาชิกออกจากกันด้วยช่องว่างหรือเครื่องหมาย comma "," เช่น A=[1 2 3] หรือ A=[1,2,3]

  การกำหนดค่าให้กับสมาชิกของแถว อาจแยกสมาชิกในแต่ละแถวออกจากกันด้วย semicolon ";" หรือใช้การขึ้นบรรทัดใหม่ เช่น A=[1 2;3 4] หรือ

```
A = [1 2
     3 4]
```

  เมื่อมีการกำหนดค่าให้กับตัวแปรใด ๆ SCILAB จะตอบกลับมาเพื่อรับทราบการกำหนดค่าดังนี้

```
-->A = [1 2;3 4]
 A  =
   1.   2.
   3.   4.
```

ถ้าไม่ต้องการให้ MATLAB ตอบค่ากลับมาจะต้องใส่เครื่องหมาย ; ต่อท้ายคำสั่ง เช่น

```
-->A = [1 2;3 4];
```

สามารถอ้างถึงสมาชิกในเมตริกซ์ได้ เช่น อาจกำหนดค่า A(1,2)=5 แล้ว MATLAB จะตอบกลับมาดังนี้

```
-->A(1,2) = 5
 A  =
   1.   5.
   3.   4.
```

SCILAB ยอมให้มีการกำหนดค่าเมตริกซ์ใหม่โดยอ้างอิงเมตริกซ์เดิมที่มีอยู่ได้ดังนี้

```
-->A = [A;5 6]
 A  =
   1.   5.
   3.   4.
   5.   6.
```

ข.2.3 ค่าพิเศษของ SCILAB

%pi = ค่าพาย (3.14159…)

%i แทนค่าของ $\sqrt{-1}$

%inf แทนค่า infinity (∞)

%nan แทน Not-a-Number คือค่าที่ไม่สามารถนิยามได้ทางคณิตศาสตร์ เช่น 0/0

ข.2.4 การดำเนินการเมตริกซ์

บวก, ลบ, คูณ และหาร ใช้ +, -, *, / ตามลำดับ

เมตริกซ์ผกผันของเมตริกซ์ A ใช้ฟังก์ชัน inv(A)

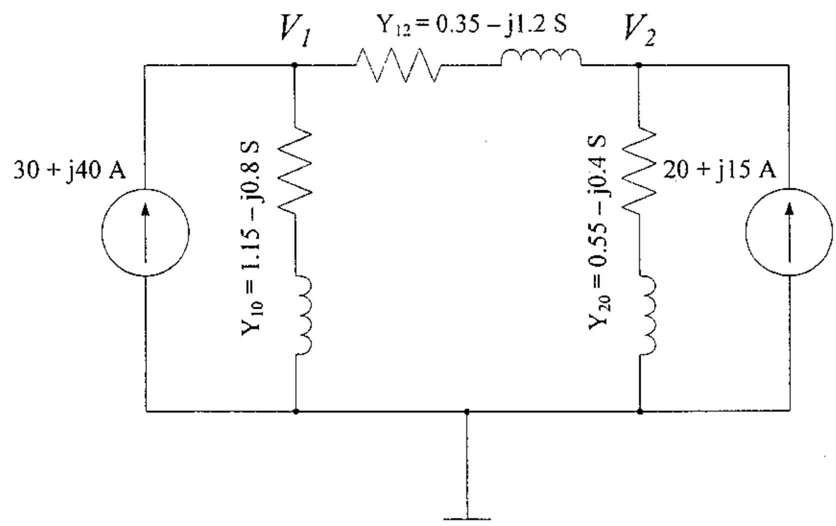
ทรานสโพส (transpose) ของเมตริกซ์ A ใช้สัญลักษณ์ A'

ยกกำลังใช้สัญลักษณ์ ^ เช่น A^2 (A ยกกำลังสอง)

ดีเทอร์มิแนนท์ (determinant) ของเมตริกซ์ A ใช้ฟังก์ชัน det(A)

ข.3 การแก้ปัญหาโดยใช้การโปรแกรม SCILAB

โปรแกรม SCILAB มีพื้นฐานมาจากการโปรแกรมภาษา C และ FORTRAN ดังนั้น คำสั่งภาษา C และ FORTRAN สามารถนำมาใช้ในการโปรแกรมด้วย SCILAB ได้ ในที่นี้จะนำเสนอตัวอย่างการใช้โปรแกรม SCILAB แก้ปัญหาดังนี้ จากรูปที่ ข.2



รูปที่ ข.2 วงจรตัวอย่าง

โดยใช้ KCLสามารถสร้างสมการโนดได้ดังนี้

$$\begin{bmatrix} 1.5-j2.0 & -0.35+j1.2 \\ -0.35+j1.2 & 0.9-j1.6 \end{bmatrix}\begin{bmatrix} V_1 \\ V_2 \end{bmatrix}=\begin{bmatrix} 30+j40 \\ 20+j15 \end{bmatrix}$$

การใช้โปรแกรม MATLAB ในการแก้ปัญหาสามารถทำได้ดังนี้

```
--> I = [30+40*%i;  20+15*%i];
--> Y = [1.5–2*%i   –0.35 +1.2*%i;  -0.35+1.2*%i     0.9-1.6*%i];
```

```
--> disp('The solution is'); V = inv(Y)*I
The solution is
V =
   3.5902047 + 35.092793i
   6.015537 + 36.221212i
```

ข.4 กราฟฟิกเบื้องต้น

โปรแกรม SCILAB สามารถสร้างกราฟที่มีความละเอียดสูง โปรแกรมนี้สามารถสร้างกราฟได้หลายประเภท เช่น 2-D, 3-D, linear, semilog, log, polar, bar chart และ contour คำสั่งที่ได้สร้างกราฟมีหลายคำสั่ง เช่น **plot, plot2d, bar, contour, mesh** แต่ในที่นี้จะแนะนำการใช้งานในบางคำสั่งเท่านั้น คำสั่งอื่นนอกเหนือจากนี้สามารถศึกษาได้จาก Help ของโปรแกรมที่ต้องการศึกษาได้ เช่น ถ้าต้องการทราบการใช้งานพื้นฐานของฟังก์ชัน **barh** ให้ใช้คำสั่ง help barh ฟังก์ชัน **plot** ให้ใช้คำสั่ง help plot

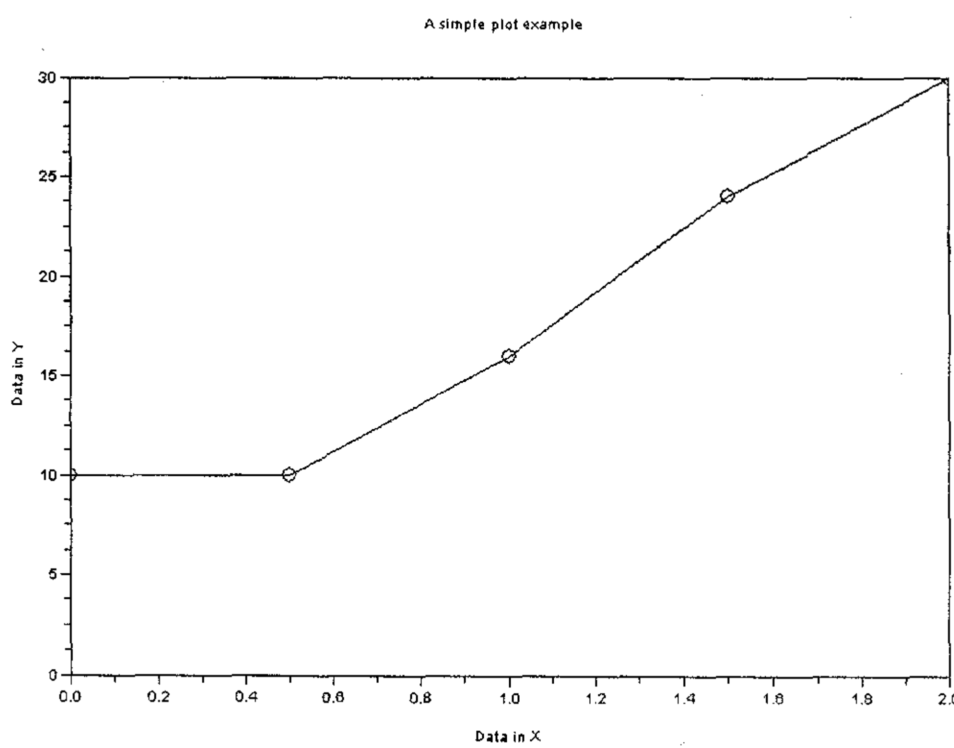
กำหนดข้อมูลดังนี้

X	0	0.5	1.0	1.5	2.0
Y	10	10	16	24	30

ต้องการสร้างกราฟจากข้อมูลในตาราง สามารถทำได้ดังนี้

```
--> X = [0  0.5  1.0  1.5  2.0];
--> Y = [10  10  16  24  30];
--> plot2d(X,Y,style=[-9]);
--> plot2d(X,Y,rect=[0 0 2 30]);
--> xtitle('A simple plot example','Data in X','Data in Y');
```

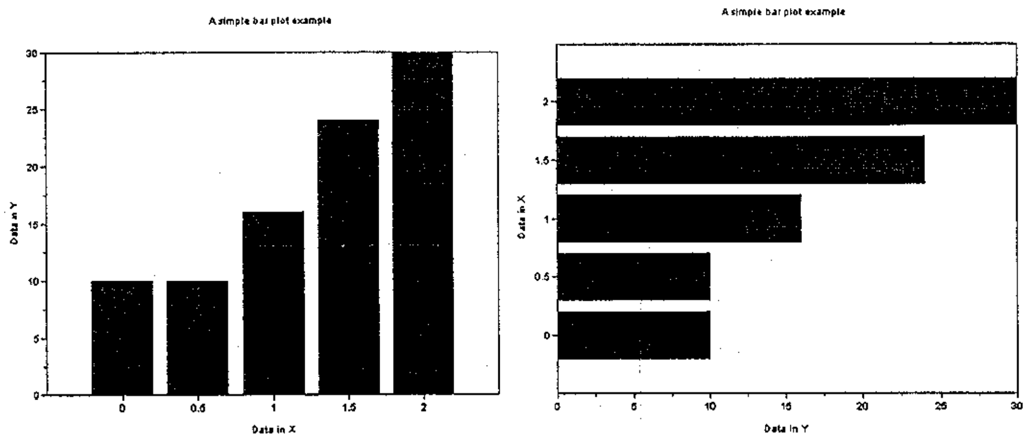
จะได้ผลลัพธ์ดังกราฟในรูปที่ ข.3



รูปที่ ข.3 ตัวอย่างกราฟอย่างง่าย

ถ้าต้องการสร้างกราฟแท่งจะได้ว่า

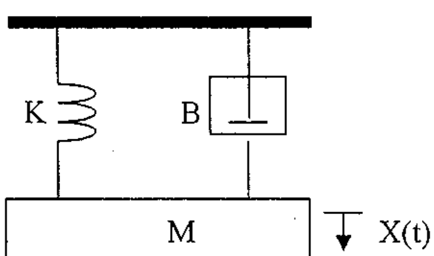
```
--> X = [0  0.5  1.0  1.5  2.0];
--> Y = [10  10  16  24  30];
--> bar(X,Y);
--> xtitle('A simple bar plot example','Data in X','Data in Y');
--> scf(1);
--> barh(X,Y);
--> xtitle('A simple bar plot example','Data in X','Data in Y');
```



รูปที่ ข.4 ตัวอย่างกราฟแท่งอย่างง่าย

ข.5 ผลเฉลยของระบบสมการเชิงอนุพันธ์

การหาผลเฉลยของสมการเชิงอนุพันธ์โดยใช้ SCILAB ทำได้โดยใช้ฟังก์ชันที่ชื่อว่า ode ย่อมาจาก (ordinary differential equation) พิจารณาจากตัวอย่างต่อไปนี้



รูปที่ ข.5 ระบบการเคลื่อนที่ทางกล

โดยการประยุกต์กฎการเคลื่อนที่ของนิวตัน เขียนสมการเชิงอนุพันธ์ที่ใช้อธิบายคุณสมบัติของระบบได้ดังนี้

$$M\frac{d^2X}{dt^2} + B\frac{dX}{dt} + KX = F(t)$$

ถ้ากำหนดตัวแปรสถานะ x_1 และ x_2 นิยามดังนี้

$$x_1 = X\,;\;\; x_2 = \frac{dX}{dt}$$

จะได้สมการสถานะดังนี้

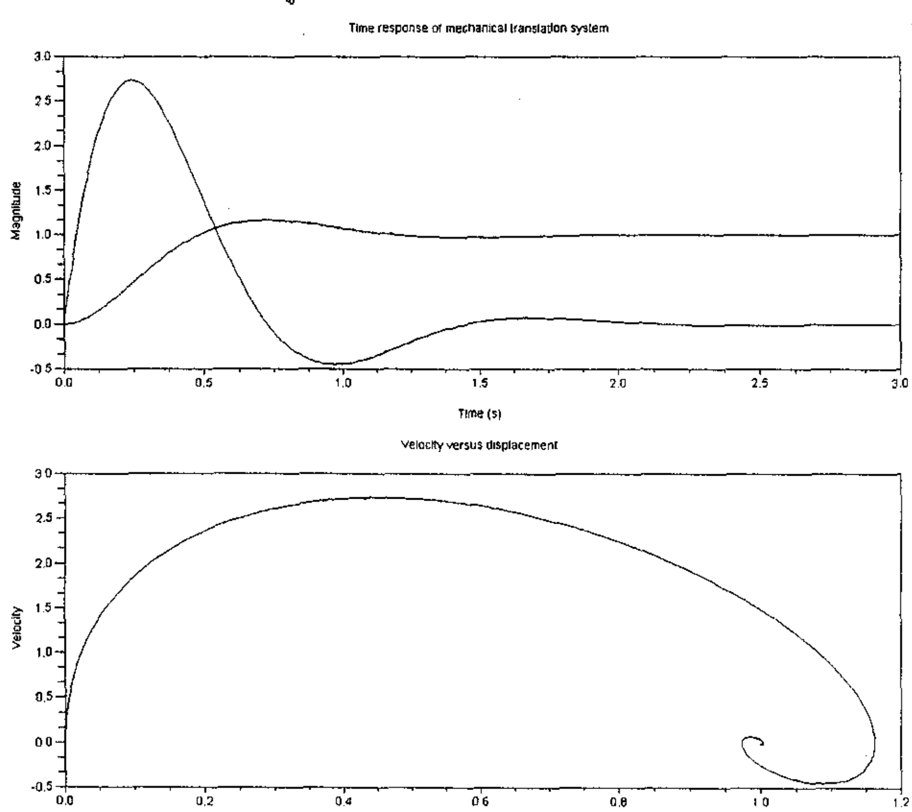
$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = \frac{1}{M}[F(t) - Bx_2 - Kx_1]$$

กำหนดให้ M = 1 kg; B = 5 N/m/s; K = 25 N/m และ F(t) = 25 ในการจำลองผลโดยใช้ ฟังก์ชัน ode ก่อนที่จะใช้งานต้องกำหนดสมการสถานะของระบบ ซึ่งทำได้ดังนี้

```
-->F=25;
-->M=1;B=5;K=25;
-->function xdot=mechsys(t,x)
-->xdot = [x(2);1/M*(F-B*x(2)-K*x(1))];
-->endfunction
-->x0 = [0;0];
-->t0 = 0;
-->t = [0:0.01:3];
-->x = ode(x0,t0,t,mechsys);
-->d=x(1,:);v=x(2,:);
-->subplot(2,1,1),plot2d(t,d);
-->subplot(2,1,1),plot2d(t,v);
-->xtitle('Time response of mechanical translation system','Time (s)','Magnitude');
-->subplot(2,1,2),plot2d(d,v);
-->xtitle('Velocity versus displacement','Displacement','Velocity');
```

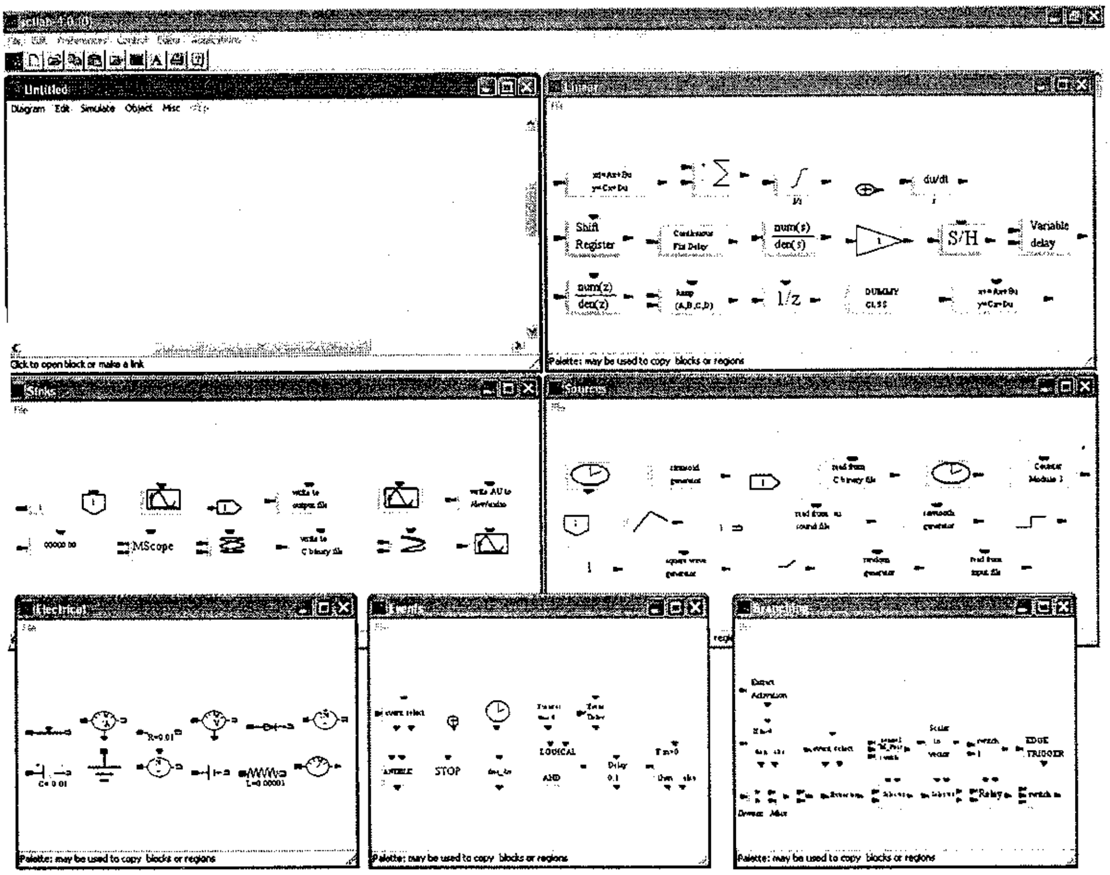
จะได้ผลลัพท์ดังแสดงในรูปที่ 6



รูปที่ ข.6 ผลเฉลยจากระบบสมการเชิงอนุพันธ์โดยใช้ฟังก์ชัน ode ของ SCILAB

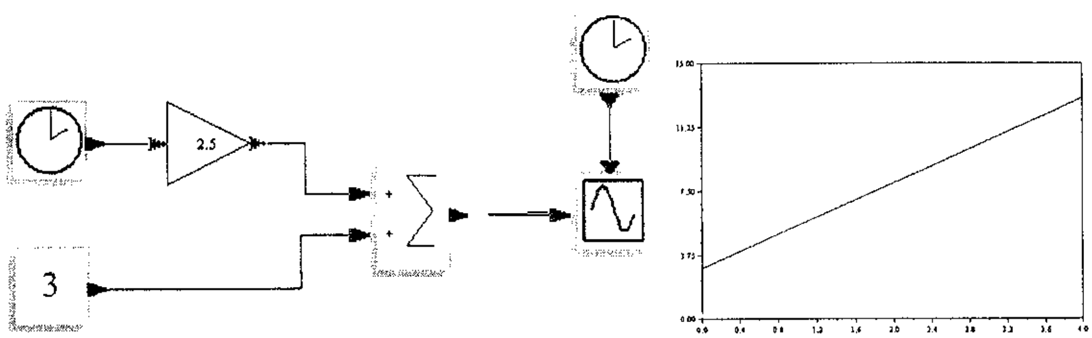
ข.7 การจำลองผลเชิงแผนภาพโดยใช้ SCICOS

การใช้งาน SCICOS จะช่วยแก้ปัญหาได้อย่างรวดเร็วและสะดวกเนื่องจากมีรูปแบบเป็น GUI (Graphic User Interface) การใช้งาน SCICOS ทำได้โดยการเรียกใช้คำสั่ง scicos จะปรากฏหน้าต่างทำงานของ SCICOS ซึ่งจะมีอุปกรณ์ต่าง ๆ ให้เลือกใช้ เช่น เครื่องกำเนิดสัญญาณรูปแบบต่าง ๆ ภาคแสดงผลแบบต่าง ๆ หรือฟังก์ชันถ่ายโอน เป็นต้น ดังรูปที่ ข.7



รูปที่ ข.7 หน้าต่างทำงานและอุปกรณ์จากคลังอุปกรณ์ของ SCICOS

SCICOS สามารถเรียนรู้ได้ง่าย ดังนั้นในที่นี้จะแสดงตัวอย่างการใช้ SCICOS ในการแก้ปัญหาเท่านั้น ถ้าต้องการใช้ SCICOS แก้สมการเพื่อหาค่า Y = 2.5X + 3 ซึ่งเป็นสมการเชิงเส้นสามารถสร้างแบบจำลองโดยใช้ SCICOS ได้ดังนี้



รูปที่ ข.8 การแก้สมการอย่างง่ายด้วย SCICOS

พิจารณาตัวอย่างของสมการเชิงอนุพันธ์ต่อไปนี้

$$2\frac{d^3y}{dt^3}+4\frac{d^2y}{dt^2}+8\frac{dy}{dt}+10y=10u(t)$$

เลือกตัวแปรสถานะดังนี้

$$x_1 = y; x_2 = \dot{y}; x_3 = \ddot{y}$$

เขียนสมการสถานะได้ดังนี้

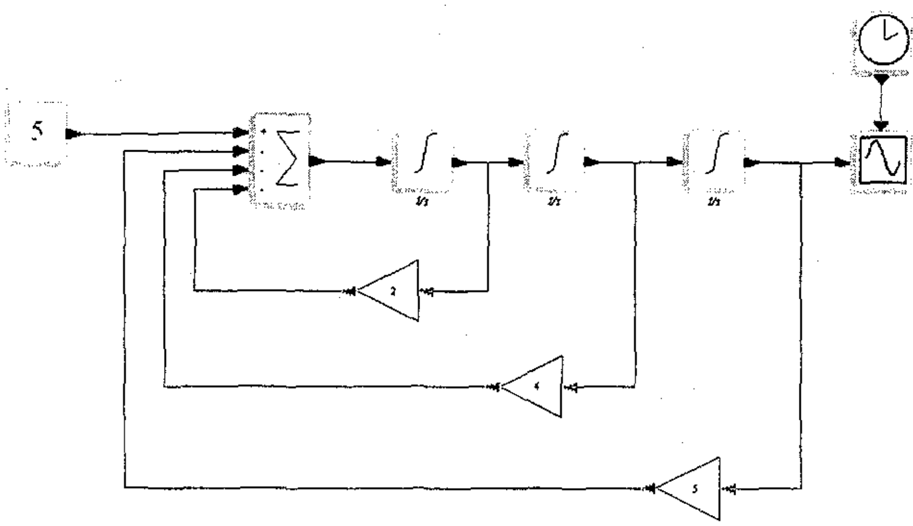
$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

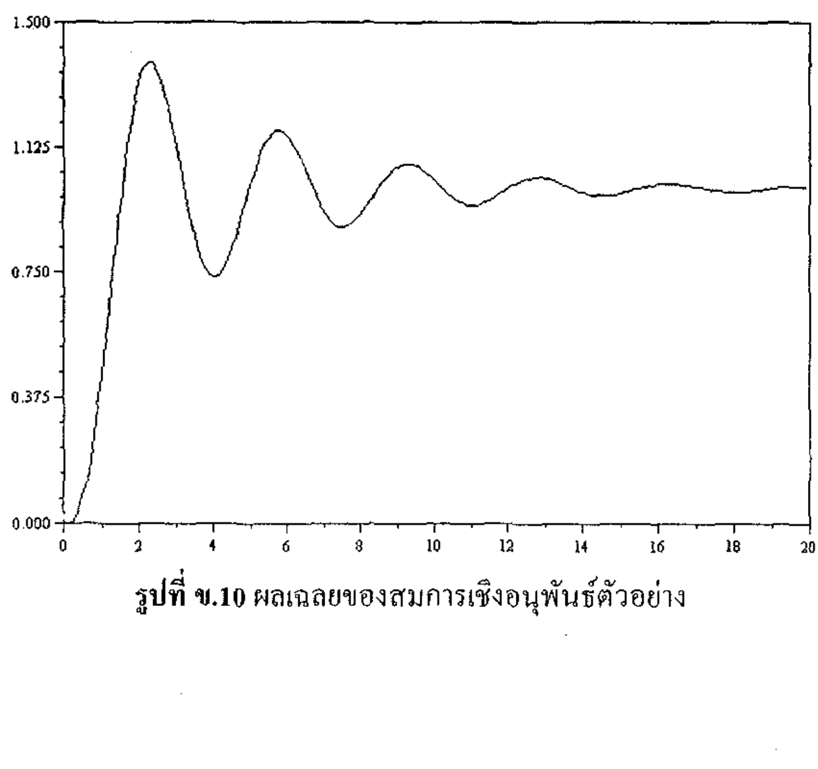
$$\dot{x}_3 = -5x_1 - 4x_2 - 2x_3 + 5u(t)$$

และสมการเอาต์พุต y = [1 0 0] x

สร้างแบบจำลองของระบบโดยใช้ SCICOS จะได้ดังรูปที่ ข.9 และผลลัพธ์จะได้ดังรูปที่ ข.10 ดังนี้



รูปที่ ข.9 การแก้สมการเชิงอนุพันธ์ด้วย SCICOS

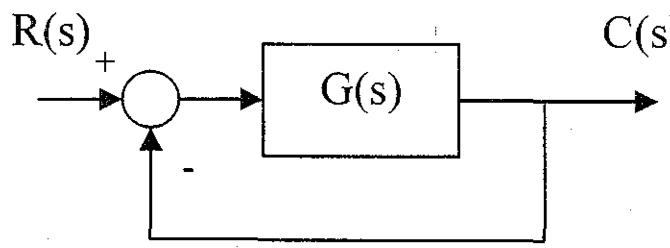


รูปที่ ข.10 ผลเฉลยของสมการเชิงอนุพันธ์ตัวอย่าง

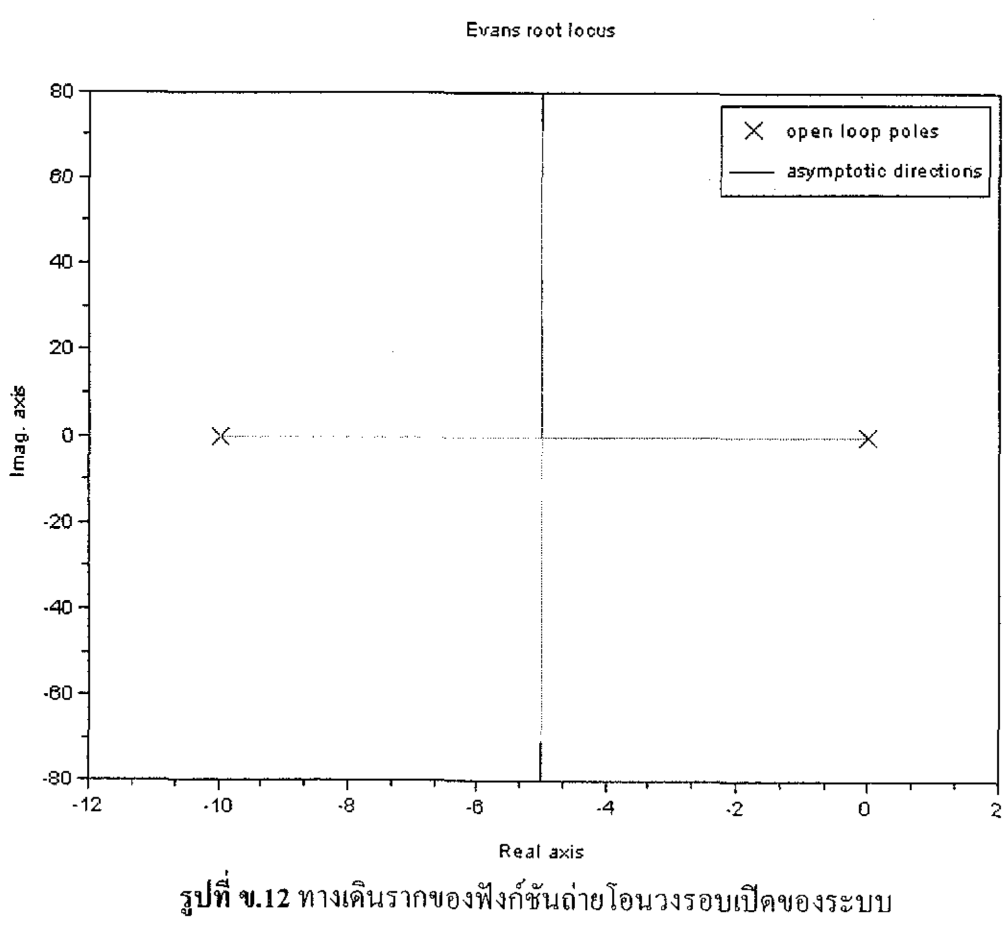
ข.8 การใช้งานกล่องเครื่องมือระบบควบคุม (SCILAB's Control TOOLBOX)

กำหนดระบบควบคุมดังรูปที่ ข.11 ถ้าระบบมีฟังก์ชันถ่ายโอน $G(s)=\dfrac{1}{s(s+10)}$ สามารถวิเคราะห์สมรรถนะของระบบในโดเมนเวลาและโดเมนความถี่ได้โดยใช้ชุดคำสั่ง SCILAB ดังนี้

```
--> s = poly(0,'s');
--> G = 1/(s*(s+10))          // ฟังก์ชันถ่ายโอนวงรอบเปิดของระบบ
 G  =
      1
   -------
        2
   10s + s
--> evans(G)                  // คำสั่งวาดทางเดินของราก (root locus)
```

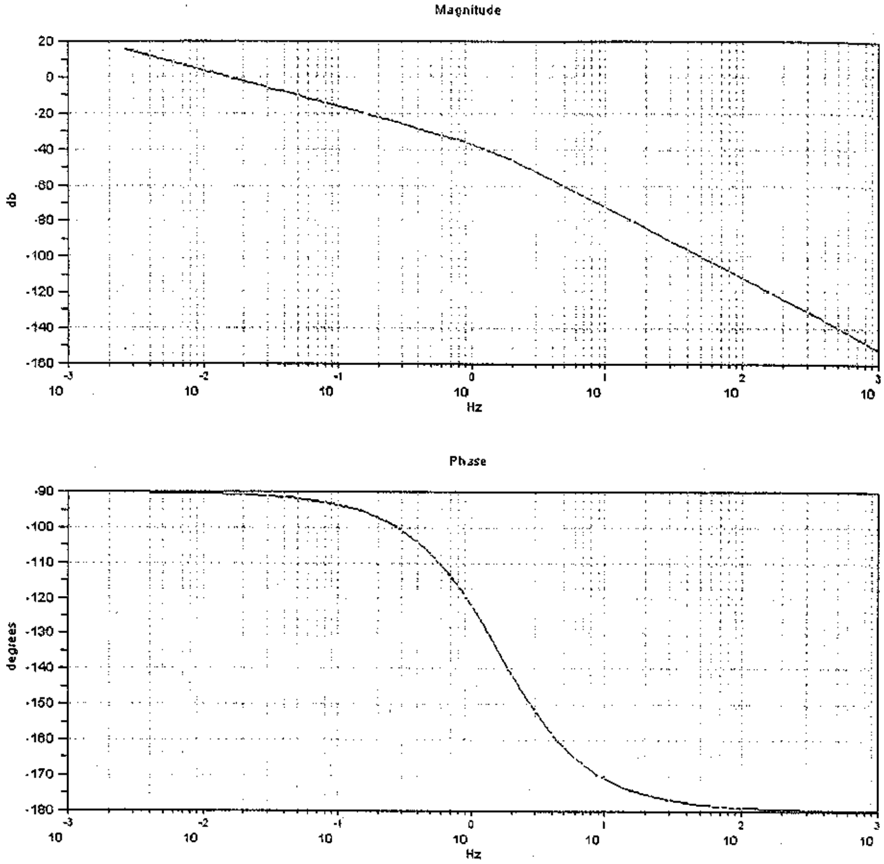


รูปที่ ข.11 แผนภาพระบบควบคุมตัวอย่าง



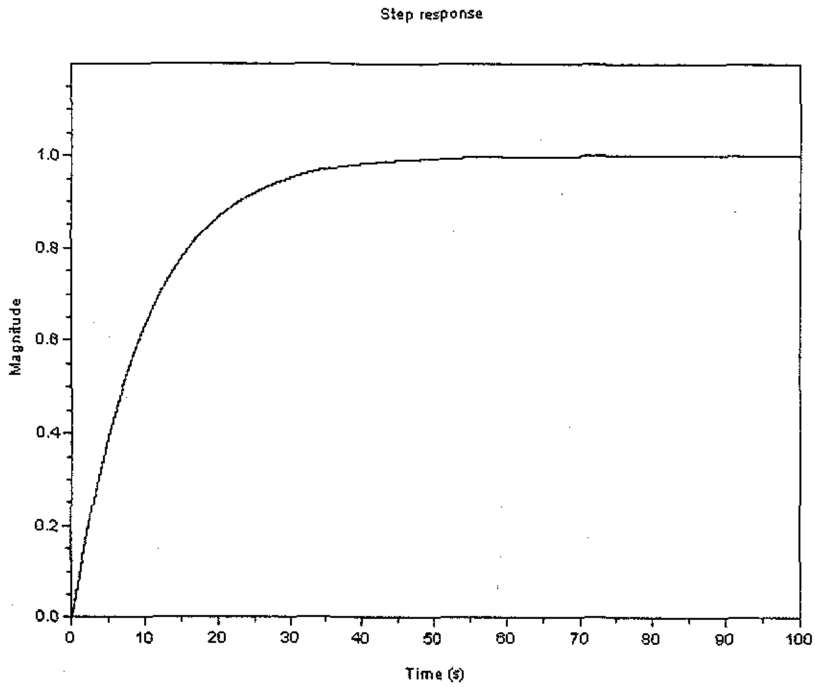
รูปที่ ข.12 ทางเดินรากของฟังก์ชันถ่ายโอนวงรอบเปิดของระบบ

```
-->syst = syslin('c',G);
-->bode(syst)
```

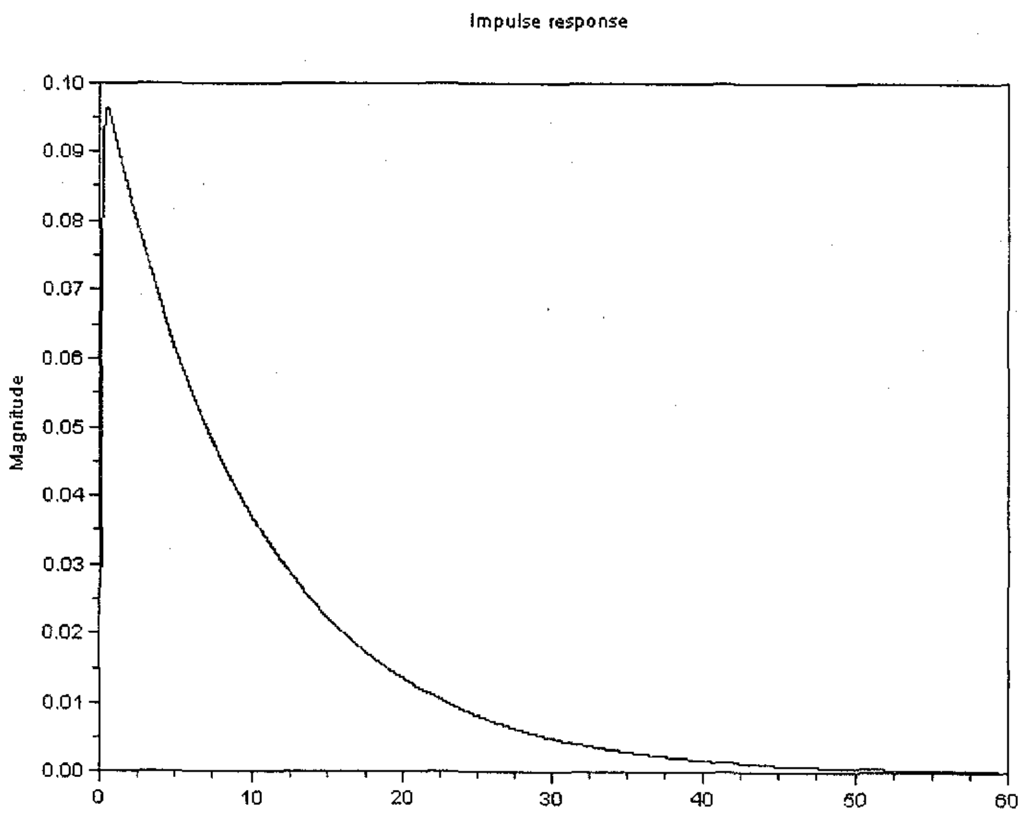


รูปที่ ข.13 แผนภาพโบดของฟังก์ชันถ่ายโอนวงรอบเปิดของระบบ

```
--> CTF = G/(1+G)
 CTF  =
        1
   -----------
               2
   1 + 10s + s
--> t = 0:1e-2:100;
--> y1 = csim('step',t,CTF);
--> plot2d(t,y1,rect=[0 0 100 1.2])
--> xtitle('Step response','Time (s)','Magnitude');
--> scf(1);
-->y2 = csim('impulse',t,CTF);
--> plot2d(t,y2,rect=[0 0 60 0.1])
--> xtitle('Impulse response','Time (s)','Magnitude');
```



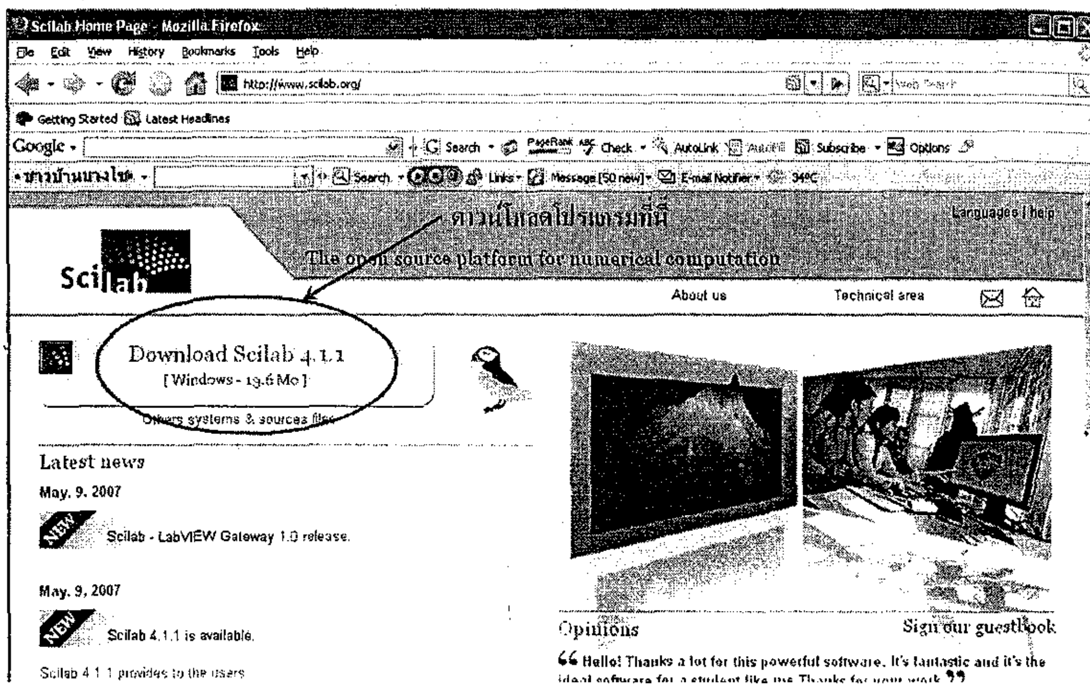
รูปที่ ข.14 กราฟการตอบสนองต่ออินพุตแบบขั้นบันไดหนึ่งหน่วย



รูปที่ ข.15 กราฟการตอบสนองต่ออินพุตแบบอิมพัลส์หนึ่งหน่วย

ข.9 การดาวน์โหลดซอฟท์แวร์ SCILAB และการติดตั้ง

ซอฟท์แวร์นี้ดาวน์โหลดได้ฟรีที่ http://www.scilab.org ให้ดำเนินการติดตั้งตามคำสั่งที่ปรากฏในระหว่างขั้นตอนการติดตั้งโปรแกรม



รูปที่ ข.16 หน้าเว็บของ www.scilab.org

ผนวก ค. Source Codes

ค.1 เกริ่นนำ

ส่วนนี้เป็นการนำเสนอซอร์สโคดของโปรแกรมที่ผู้แต่งได้พัฒนาขึ้น ประกอบการคำนวณการหาค่าเหมาะที่สุดทั้งระเบียบวิธีทางคณิตศาสตร์และระเบียบวิธีชาญฉลาด ชุดคำสั่งที่ปรากฏนี้อาจจะให้ผมการรันโปรแกรมที่แตกต่างไปจากในเอกสารบ้างเล็กน้อย เนื่องจากในระหว่างการสร้างภาพประกอบตัวอย่างนั้น ผู้แต่งได้ดัดแปลงแก้ไขชุดคำสั่งการวาดกราฟให้อ่านรูปกราฟได้ง่าย อย่างไรก็ตาม การปรับแต่งเกิดขึ้นเฉพาะส่วนการแสดงผล รายละเอียดการคำนวณไม่มีการแก้ไขแต่อย่างใด ถึงกระนั้น โปรแกรมที่ได้พัฒนาขึ้นนี้ย่อมไม่มีความสมบูรณ์โดยปราศจากความผิดพลาด หากนักศึกษา นักวิจัย หรือผู้สนใจท่านใดตรวจพบข้อผิดพลาดหรือบักของโปรแกรม ผู้แต่งยินดีตรวจสอบเพื่อแก้ไขข้อผิดพลาดดังกล่าวให้สมบูรณ์ยิ่งขึ้น โดยส่งข้อเสนอแนะมาที่

thanatchai@gmail.com

ด้วยเจตนารมณ์ของกลุ่มผู้พัฒนาซอฟท์แวร์ SCILAB เพื่อให้เป็นโอเพนซอร์ส (open source) โดยไม่เสียค่าใช้จ่ายใด ๆ ในการใช้โปรแกรม ชุดคำสั่งของผู้เขียนที่ปรากฏ ณ ที่นี้ หรือที่ปรากฏผ่านแหล่งข้อมูลอื่น ๆ ผู้เขียนยินดีให้ผู้สนใจนำไปศึกษาเพื่อใช้งานโดยตรง หรือดัดแปลงปรับปรุงโปรแกรมให้มีสมรรถนะที่ดียิ่งขึ้น โดยการดัดแปลงแก้ไขหรือการทำซ้ำดังกล่าวต้องไม่มีวัตถุประสงค์ทางการค้าแอบแฝงอยู่

เอกสารฉบับนี้ ผู้เขียนได้พัฒนาชุดคำสั่งโดยแบ่งออกเป็นกล่องเครื่องมือ (TOOLBOX) รวมทั้งสิ้น 7 กล่องเครื่องมือ ดังนี้

ค.2 กล่องเครื่องมือ univarate

ใช้สำหรับการหาค่าเหมาะที่สุดของตัวแปรหนึ่งมิติ

<u>โปรแกรม SCILAB ในกล่องเครื่องมือ univariate</u>

```
========================
          gridsearch.sci
========================
function [xmin,fmin] = gridsearch(func,interval,
                        Npnt,tolerance,MaxIter)
Interval_Err = interval(2) - interval(1);
fpnt(1) = func(interval(1));
fpnt(Npnt) = func(interval(2));
count = 0;
xmin = (interval(1) + interval(2))/2;
fmin = func(xmin);
disp('Iter  XL   XU    Xmin    Fmin    Err');
disp([count interval xmin fmin Interval_Err]);
while Interval_Err > tolerance
```

```
  dx = Interval_Err/(Npnt-1);
  for k = 2:Npnt-1
     fpnt(k) = func(interval(1) + dx*(k-1));
  end
  [fmin,idmin] = min(fpnt);
  xmin = interval(1) + dx*(idmin-1);
  interval(1) = interval(1) + dx*(idmin-2);
  interval(2) = interval(1) + 2*dx;
  fmin = func(xmin);
  Interval_Err = interval(2) - interval(1);
  count = count + 1;
  disp([count interval xmin fmin Interval_Err]);
  if count > MaxIter
     break;
```

```
    end
end
xmin = (interval(1) + interval(2))/2;
endfunction
```

```
========================
          fibonacci.sci
========================
function [xmin,fmin] = fibonacci(fobj,xinterval,vtol,ftol)
k=0;
f0=%inf;
ferr=%inf;
delta=%inf;
F(1)=1;F(2)=1;
while delta>vtol
   if k>1 then
           F(k+1)=F(k)+F(k-1);
   end
           delta=(xinterval(2)-xinterval(1))/F(k+1);
   k=k+1;
end
xl=xinterval(1);
xu=xinterval(2);
disp('Iter  Xmin  Fmin   XU-XL  Ferr');
while (k>2)
   L1 = xu - xl;
   L2 = F(k-2)/F(k)*L1;
   if L2<0.5*L1 then
      x01 = xl + L2;
      x02 = xu - L2;
   else
      x01 = xu - L2;
      x02 = xl + L2;
   end
   if fobj(x01)<fobj(x02) then
      xu = x02;
   elseif fobj(x01)>fobj(x02) then
      xl = x01;
   else
      xl = x01;
      xu = x02;
   end
   xmin = (xl + xu)/2;
   fmin = fobj(xmin);
   ferr=abs((f0-fmin));
   f0=fmin;
```

```
    disp([k xl xu xmin fmin xu-xl ferr]);
    k=k-1;
end
endfunction
```

```
========================
          quadratic.sci
========================
function [xmin,fmin] = quadratic(fobj,xinterval,vtol,opt)
k=0;
a=xinterval(1);
b=xinterval(2);
while (b-a>vtol)
   k=k+1;
   c=(a+b)/2;
   fa=fobj(a);
   fb=fobj(b);
   fc=fobj(c);
   xs=0.5*((b^2-c^2)*fa+(c^2-a^2)*fb+(a^2-
b^2)*fc)/((b-c)*fa+(c-a)*fb+(a-b)*fc);
   fs=fobj(xs);
   if opt==1 then
      disp([k  a  b  xs  fs  b-a]);
   end
   if (xs<c)&(fs<fc) then
      a=a;
      b=c;
   elseif (xs>c)&(fs>fc) then
      a=a;
      b=xs;
   elseif (xs<c)&(fs>fc) then
      a=xs;
      b=b;
   elseif (xs>c)&(fs<fc) then
      a=c;
      b=b;
   end
end
xmin=c;fmin=fc;
endfunction
```

```
========================
          cubic.sci
========================
function [xmin,fmin]=cubicsearch(fobj,fgrad,xinterval,
                                   vtol,gtol,opt)
```

```
k=0;
gmin=%inf; f0=%inf;
xl=xinterval(1);
xu=xinterval(2);
fa=fobj(xl);
fb=fobj(xu);
ga=fgrad(xl);
gb=fgrad(xu);
while (xu-xl>vtol)&(gmin>gtol)
   k=k+1;
   W=3*(fa-fb)/(xu-xl)+ga+gb;
   V=sqrt(W*W-ga*gb);
   U=1-(gb+V-W)/(gb-ga+2*V);
   xmin=xl+(xu-xl)*U;
   fmin=fobj(xmin);
   gmin=fgrad(xmin);
```

```
   if ((ga<0)&((gmin>0)|(fmin>fa))) then
           xu=xmin;
      fb=fmin;
      gb=gmin;
   else
         xl=xmin;
      fa=fmin;
      ga=gmin;
   end
   if (opt==1) then
      disp([k xl xu xmin fmin xu-xl gmin]);
   end
end
endfunction
```

ค.3 กล่องเครื่องมือ unconst

ใช้สำหรับการหาค่าเหมาะที่สุดหลายตัวแปรแบบไม่มีเงื่อนไขบังคับ

<u>โปรแกรม SCILAB ในกล่องเครื่องมือ unconst</u>

```
=========================
          randwalk.sci
=========================
function [xmin,fmin,k]=randwalk(fobj,x0,N,lambda,
                              max_iter,err_func)
n=length(x0);
xk=x0;
fk=fobj(x0);
fmin=fk;
xmin=xk;
ferr=%inf;
k=0;
disp([k lambda x0' fmin ferr]);
for k=1:max_iter
   for l=1:N
      pl=-1+2*rand(n,1);
      xii=xk+lambda*pl;
      fii=fobj(xii);
      if fii<fmin then
         fmin=fii;
         xmin=xii;
      end
   end
```

```
   ferr=abs(fk-fmin);
   fk=fmin;
   xk=xmin;
   disp([k lambda xmin' fmin ferr]);
   lambda=lambda/2;
   if ferr<err_func then
      break;
   end
end
endfunction
```

```
=========================
          univar.sci
=========================
function [xmin,fmin,k]=univar(fobj,x0,epsilon,ftol,opt)
n=length(x0);
f0=fobj(x0);
k=0;
ferr=%inf;
if opt==1 then
   disp([k x0' f0 ferr])
end
while (ferr>ftol)
   k=k+1;
```

```
  m=modulo(k,n);
  p=zeros(n,1);
  p(m+1,1)=1.0;
  fp=fobj(x0+epsilon*p);
  fn=fobj(x0-epsilon*p);
  if fp<f0 then
     p=p;
  elseif fn<f0 then
     p=-p;
  else
     break;
  end
  lambda=lsearch(fobj,x0,p);
  xmin=x0+lambda*p;
  fmin=fobj(xmin)
  ferr=abs(fmin-f0)
  if opt==1 then
     disp([k xmin' fmin ferr])
  elseif opt==2
     plot2d([x0(1) xmin(1)],[x0(2) xmin(2)],style=[-9]);
     plot2d([x0(1) xmin(1)],[x0(2) xmin(2)]);
     xtitle('Convergence curve','Iteration','Value');
  end
  f0=fmin;
  x0=xmin;
end
endfunction
```

```
=========================
          lsearch.sci
=========================
function x=lsearch(fobj,x0,p)
global x_init s_dir fobj_g
f0=fobj(x0);
f1=fobj(x0+p);
L0=0;
L1=1;
if f0>f1 then
  k=1;
  while f0>f1
     L1=L1+2*k*L1;
     f1=fobj(x0+L1*p);
     k=k+1;
  end
end
x_init=x0;
```

```
s_dir=p;
fobj_g=fobj;
x=cubicsearch1(flsearch,[L0 L1],1e-4,1e-4,0);
endfunction
=========================
          flsearch.sci
=========================
function f=flsearch(z)
global x_init s_dir fobj_g
x=x_init+z*s_dir;
f=fobj_g(x);
endfunction
```

```
=========================
          conjgrad.sci
=========================
function [x,f,k]=conjgrad(fobj,fgrad,x0,ftol,vtol,opt)
k=0;
g=fgrad(x0);
s_dir0=-g;
s_dir=s_dir0;
lambda=lsearch(fobj,x0,s_dir);
x=x0-lambda*s_dir;
f=fobj(x);
if opt==1 then
   disp([k x' f norm(g,%inf)]);
end
k=k+1;
x0=x;
x_init=x;
s_dir0=s_dir;
g0=g;
while norm(g,%inf)>ftol
   beta1=g'*g/(g0'*g0);
   s_dir=-g+beta1*s_dir0;
   lambda=lsearch(fobj,x0,s_dir);
   x=x0+lambda*s_dir;
   f=fobj(x);
   g=fgrad(x);
   if opt==1 then
       disp([k x' f norm(g,%inf)]);
   end
   k=k+1;
   x0=x;
   s_sdir0=s_dir;
   g0=g;
```

```
end
endfunction
```

```
========================
        steepest.sci
========================
function [xmin,fmin,k]=steepest(fobj,fgrad,x0,ftol,
                                 vtol,opt)
f0=fobj(x0);
g0=fgrad(x0);
ferr=norm(g0);
k=0;
if opt==1 then
   disp([k x0' f0 ferr]);
end
while (ferr>ftol)
   k=k+1;
   p=-g0;
   lambda=lsearch(fobj,x0,p);
   xmin=x0+lambda*p;
   fmin=fobj(xmin);
   gmin=fgrad(xmin);
   ferr=norm(gmin);
   if opt==1 then
      disp([k xmin' fmin ferr]);
   end
   x0=xmin;f0=fmin;g0=gmin;
end
endfunction
```

```
========================
        dfp.sci
========================
function [x,f,k]=dfp(fobj,fgrad,x0,B0,ftol,vtol,opt)
k=0;
g0=fgrad(fobj,x0);
s_dir=-B0*g0;
if opt==1 then
   disp([k x0' fobj(x0) norm(g0,%inf)]);
end
lambda=lsearch(fobj,x0,s_dir);
x=x0+lambda*s_dir;
f=fobj(x);
g=fgrad(fobj,x);
k=k+1;
if opt==1 then
   disp([k x' f norm(g,%inf)]);
end
s=x-x0;
y=g-g0;
k=k+1;
x0=x;
x_init=x;
g0=g;
while norm(g,%inf)>ftol
   B=B0-(B0*y*y'*B0)/(y'*B0*y)+(s*s')/(y'*s);
   s_dir=-B*g;
   lambda=lsearch(fobj,x0,s_dir);
   x=x0+lambda*s_dir;
   f=fobj(x);
   g=fgrad(fobj,x);
   if opt==1 then
      disp([k x' f norm(g,%inf)]);
   end
   s=x-x0;
   y=g-g0;
   k=k+1;
   x0=x;
   g0=g;
   B0=B;
end
endfunction
```

```
========================
        bfgs.sci
========================
function [x,f,k]=bfgs(fobj,fgrad,x0,B0,ftol,vtol,opt)
k=0;
g0=fgrad(x0);
if opt==1 then
   disp([k x0' fobj(x0) norm(g0,%inf)]);
end
s_dir=-linsolve(B0,g0);
lambda=lsearch(fobj,x0,s_dir);
x=x0+lambda*s_dir;
f=fobj(x);
g=fgrad(x);
k=k+1;
if opt==1 then
   disp([k x' f norm(g,%inf)]);
end
s=x-x0;
```

```
y=g-g0;
k=k+1;
x0=x; x_init=x;
g0=g;
while norm(g,%inf)>ftol
   B=B0-(B0*s)*(B0*s)'/(s'*B0*s)+(y*y')/(y'*s);
   s_dir=-linsolve(B,g);
   lambda=lsearch(fobj,x0,s_dir);
   x=x0+lambda*s_dir;
   f=fobj(x);
   g=fgrad(x);
   if opt==1 then
      disp([k x' f norm(g,%inf)]);
   end
   s=x-x0;
   y=g-g0;
   k=k+1;
   x0=x;
   g0=g;
   B0=B;
end
endfunction
```

ค.4 กล่องเครื่องมือ constrained

ใช้สำหรับการหาค่าเหมาะที่สุดหลายตัวแปรแบบมีเงื่อนไขบังคับ

<u>โปรแกรม SCILAB ในกล่องเครื่องมือ constrained</u>

```
=========================
          redgrad.sci
=========================
function [x1,f1,g1,k] = redgrad(fobj,A,b,ferr_max,
                                  gerr_max,opt)
k=0;f1=[];g1=[];
m=size(A,1);
n=size(A,2);
B=A(1:m,1:m);
N=A(1:m,m+1:n);
x0=[linsolve(B,-b);zeros(n-m,1)];
Z=[linsolve(B,N);eye(n-m)];

f0=fobj(x0);
g0=fgradnum1(fobj,x0);
redgradient=Z'*g0;
if opt==1 then
   disp([k x0' f0 norm(redgradient,%inf)]);
end
while norm(redgradient,%inf)>gerr_max
   k=k+1;
   p0=-Z*redgradient;
   alpha=lsearch(fobj,x0,p0);
   x1=x0+alpha*p0;
   f1=fobj(x1);
   g1=fgradnum1(fobj,x1);

   redgradient=Z'*g1;
   if opt==1 then
      disp([k x1' f1 norm(redgradient,%inf)]);
   end
   x0=x1;g0=g1;f0=f1;
end
x1=x0;f1=f0;g1=g0;
endfunction
```

ค.5 กล่องเครื่องมือ SGA

ใช้สำหรับการหาค่าเหมาะที่สุดด้วยจีนเนติกอัลกอริทึม

โปรแกรม SCILAB ในกล่องเครื่องมือ SGA

```
========================
        bin2re.sci
========================
function p = bin2re(ps,xlimit)
N = size(xlimit,1);
k = 0;
for t=1:N
  M = xlimit(t,3)+k;
  g = ps(1,k+1:M);
  sumg = 0;
  for u=1:xlimit(t,3)
    if g(1,u)==1 then
      sumg = sumg+2^(xlimit(t,3)-u);
    end
  end
  p(1,t) = xlimit(t,1) + sumg*(xlimit(t,2)-
xlimit(t,1))/(2^(xlimit(t,3))-1);
  k = k + xlimit(t,3);
end
endfunction
```

```
========================
        re2bin.sci
========================
function xs = re2bin(x,xlimit)
N = round((x-xlimit(1))/(xlimit(2)-xlimit(1))*2^xlimit(3));
M = N;
k = xlimit(3);
for k=1:xlimit(3)
  xs(1,xlimit(3)-k+1) = modulo(M,2);
  M = fix(M/2);
end
endfunction
```

```
========================
        fga.sci
========================
function f=fga(fobj,p,xlimit)
for t=1:size(p,1)
  x = bin2re(p(t,:),xlimit);
  f(t,1) = fobj(x);
```

```
end
endfunction
```

```
========================
        mutation.sci
========================
function [ps,fs] = mutation(pz,fz,fobj,mprop,xlimit)
global pbest fpbest
N = size(pz,1);
Nmutat = round(mprop*N);
for k=1:Nmutat
   u1 = round(rand1(1,N));
   ub = round(rand1(1,size(pz,2))); // single-bit
mutation
   if pz(u1,ub)==0 then
      pz(u1,ub) = 1;
   else
      pz(u1,ub) = 0;
   end
   fz(u1,1) = fga(fobj,pz(u1,:),xlimit);
   if fz(u1,1)<fpbest then
      fpbest = fz(u1,1);
      pbest = pz(u1,:);
   end
end
ps = pz;
fs = fz;
endfunction
```

```
========================
        pop_ga.sci
========================
function ps = pop_ga(Ne,Np,xlimit)
p = randp(Ne,Np,xlimit(:,(1:2)));
ps = [];
for t=1:Np
  xss = [];
  for h=1:Ne
    xs = re2bin(p(t,h),xlimit(h,:));
    xss = [xss,xs];
  end
```

```
y=g-g0;
k=k+1;
x0=x; x_init=x;
g0=g;
while norm(g,%inf)>ftol
    B=B0-(B0*s)*(B0*s)'/(s'*B0*s)+(y*y')/(y'*s);
    s_dir=-linsolve(B,g);
    lambda=lsearch(fobj,x0,s_dir);
    x=x0+lambda*s_dir;
    f=fobj(x);
    g=fgrad(x);
    if opt==1 then
      disp([k x' f norm(g,%inf)]);
    end
    s=x-x0;
    y=g-g0;
    k=k+1;
    x0=x;
    g0=g;
    B0=B;
end
endfunction
```

ค.4 กล่องเครื่องมือ constrained

ใช้สำหรับการหาค่าเหมาะที่สุดหลายตัวแปรแบบมีเงื่อนไขบังคับ

<u>โปรแกรม SCILAB ในกล่องเครื่องมือ constrained</u>

```
========================
        redgrad.sci
========================
function [x1,f1,g1,k] = redgrad(fobj,A,b,ferr_max,
                              gerr_max,opt)
k=0;f1=[];g1=[];
m=size(A,1);
n=size(A,2);
B=A(1:m,1:m);
N=A(1:m,m+1:n);
x0=[linsolve(B,-b);zeros(n-m,1)];
Z=[linsolve(B,N);eye(n-m)];

f0=fobj(x0);
g0=fgradnum1(fobj,x0);
redgradient=Z'*g0;
if opt==1 then
   disp([k x0' f0 norm(redgradient,%inf)]);
end
while norm(redgradient,%inf)>gerr_max
   k=k+1;
   p0=-Z*redgradient;
   alpha=lsearch(fobj,x0,p0);
   x1=x0+alpha*p0;
   f1=fobj(x1);
   g1=fgradnum1(fobj,x1);

   redgradient=Z'*g1;
   if opt==1 then
      disp([k x1' f1 norm(redgradient,%inf)]);
   end
   x0=x1;g0=g1;f0=f1;
end
x1=x0;f1=f0;g1=g0;
endfunction
```

ค.5 กล่องเครื่องมือ SGA

ใช้สำหรับการหาค่าเหมาะที่สุดด้วยจีนเนติกอัลกอริทึม

โปรแกรม SCILAB ในกล่องเครื่องมือ SGA

```
========================
        bin2re.sci
========================
function p = bin2re(ps,xlimit)
N = size(xlimit,1);
k = 0;
for t=1:N
  M = xlimit(t,3)+k;
  g = ps(1,k+1:M);
  sumg = 0;
  for u=1:xlimit(t,3)
    if g(1,u)==1 then
      sumg = sumg+2^(xlimit(t,3)-u);
    end
  end
  p(1,t) = xlimit(t,1) + sumg*(xlimit(t,2)-
xlimit(t,1))/(2^(xlimit(t,3))-1);
  k = k + xlimit(t,3);
end
endfunction
```

```
========================
        re2bin.sci
========================
function xs = re2bin(x,xlimit)
N = round((x-xlimit(1))/(xlimit(2)-xlimit(1))*2^xlimit(3));
M = N;
k = xlimit(3);
for k=1:xlimit(3)
  xs(1,xlimit(3)-k+1) = modulo(M,2);
  M = fix(M/2);
end
endfunction
```

```
========================
        fga.sci
========================
function f=fga(fobj,p,xlimit)
for t=1:size(p,1)
  x = bin2re(p(t,:),xlimit);
  f(t,1) = fobj(x);
end
endfunction
```

```
========================
        mutation.sci
========================
function [ps,fs] = mutation(pz,fz,fobj,mprop,xlimit)
global pbest fpbest
N = size(pz,1);
Nmutat = round(mprop*N);
for k=1:Nmutat
  u1 = round(rand1(1,N));
  ub = round(rand1(1,size(pz,2))); // single-bit
mutation
  if pz(u1,ub)==0 then
    pz(u1,ub) = 1;
  else
    pz(u1,ub) = 0;
  end
  fz(u1,1) = fga(fobj,pz(u1,:),xlimit);
  if fz(u1,1)<fpbest then
    fpbest = fz(u1,1);
    pbest = pz(u1,:);
  end
end
ps = pz;
fs = fz;
endfunction
```

```
========================
        pop_ga.sci
========================
function ps = pop_ga(Ne,Np,xlimit)
p = randp(Ne,Np,xlimit(:,(1:2)));
ps = [];
for t=1:Np
  xss = [];
  for h=1:Ne
    xs = re2bin(p(t,h),xlimit(h,:));
    xss = [xss,xs];
  end
```

```
  ps = [ps;xss];
end
endfunction
```

```
========================
         rand1.sci
========================
function x = rand1(a,b)
z = rand();
x = a + z*(b-a);
endfunction
```

```
========================
         delrow.sci
========================
function M = delrow(A,r)
s_A = size(A,1);
if (r<=0)|(r>s_A) then
  error('The selected row cannot be deleted or the
provided row''s index is invalid!');
end
M = [A(1:r-1,:);A(r+1:s_A,:)];
endfunction
```

```
========================
         randp.sci
========================
function p = randp(Ne,Np,p_limit)
for t=1:Np
   for u=1:Ne
      A = rand1(p_limit(u,1),p_limit(u,2));
      p(t,u) = A;
      if p(t,u)>p_limit(u,2) then
         p(t,u) = p_limit(u,2);
      elseif p(t,u)<p_limit(u,1) then
         p(t,u) = p_limit(u,1);
      end
   end
end
endfunction
```

```
========================
         randp.sci
========================
function [pg,xg,fg] = sga(fobj,Ne,Np,xlimit,
         xprop,mprop,opt)
// sga function is a simple genetic algorithm version
         1.0 coded in SCILAB
// Only crossover and mutation operator are provided
         with their probability
// fitness function is evaluated by using the objective
         function directly
// fobj     Objective function
// Ne       Number of variables
// Np       Number of population
// xlimit   limits of variables xlimit = [x1min x1max
              x1_bit_res;x2min x2max x2_bit_res;...];
// xprop    crossover probability
// mprop    mutation probability
// opt(1)   Maximum number of generation counted
// opt(2)   Maximum number of generation stalled
// opt(3)   display detail during calculation
// Developed by
//          Dr Thanatchai KULWORAWANICHPONG
//          Power & Energy Research Unit
//          School of Electrical Engineering
//          Institute of Engineering
//          Suranaree University of Technology
//          Nakhon Ratchasima, THAILAND
// Last Updated
//          28 July 2007
global pbest fpbest
ps = pop_ga(Ne,Np,xlimit);
fs = fga(fobj,ps,xlimit);
[fg,kg] = min(fs);
pbest = ps(kg,:);
fpbest = fg;
if opt(3)==1 then
   disp('Initial population');
   disp([ps msprintf('%f',fs)])
   disp(ps)
   disp(fs)
end
count=0;
subplot(2,1,1),plot2d(0,fpbest,style=[-9],logflag='nl')
xset('font',2,3);
xtitle('Convergence curve (best solution)','Generation
in progress','Objective');
subplot(2,1,2),barh([count 0],0.1);
subplot(2,1,2),xset('font',2,3);
xstring(1,2,2,'Generation Counted');
xstring(1,1,2,'Generation Stalled');
for z=1:opt(1)
```

```
    [px,fx] = xcross(ps,fs,fobj,xprop,xlimit);
    if opt(3)==1 then
       disp(' ');
disp('================================');
       disp(['Generation # ' msprintf('%i',z)]);
disp('================================');
       disp('1. After Crossover');
       disp([px fx]);
       disp(px);
    end
    ps = [];fs = [];
    [ps,fs] = mutation(px,fx,fobj,mprop,xlimit);
    if opt(3)==1 then
       disp('2. After Mutation');
       disp([ps fs]);
       disp(ps);
    end
    if fpbest==fg then
       count = count+1;
    else
       count = 0;
    end
    if opt(3)==1 then
       disp('*** The best solution of this generation
***');
       disp(pbest)
       disp(fpbest)
    end
    subplot(2,1,1),plot2d(z,fpbest,style=[-9],logflag='nl')
    subplot(2,1,1),plot2d([z-1 z],[fg fpbest],logflag='nl')
    subplot(2,1,2),barh([count z],0.1);
    fg = fpbest;
    pg = pbest;
    xg = bin2re(pg,xlimit);
    if count>=opt(2) then
       break;
    end
    px = [];fx =[];
end
endfunction
```

```
========================
         xcross.sci
========================
function [p,f] = xcross(pz,fz,fobj,xprop,xlimit)
global pbest fpbest
N = size(pz,1);
Nx = round(0.5*xprop*N);
k=1;
for t=1:Nx
   u1 = round(rand1(1,2*Nx-k+1));
   k = k+1;
   p1(t,:) = pz(u1,:);
   f1(t,1) = fz(u1,1);
   pz = delrow(pz,u1);
   fz = delrow(fz,u1);
   u1 = round(rand1(1,2*Nx-k+1));
   k = k+1;
   p2(t,:) = pz(u1,:);
   f2(t,1) = fz(u1,1);
   pz = delrow(pz,u1);
   fz = delrow(fz,u1);
end
for t=1:Nx
   u=round(rand1(1,size(pz,2)));
   p3(t,:)=[p1(t,1:u) p2(t,u+1:size(pz,2))];
   f3(t,1)=fga(fobj,p3(t,:),xlimit);
   p4(t,:)=[p2(t,1:u) p1(t,u+1:size(pz,2))];
   f4(t,1)=fga(fobj,p4(t,:),xlimit);
end
p = [p3;p4];
fp = [f3;f4];
[y,indx] = sort(fp);
py=p(indx,:);
p = [pz;p(indx,:)];
f = [fz;y];
if f(length(f))<fpbest then
    fpbest = f(length(f));
    pbest = p(length(f),:);
end
endfunction
```

ค.6 กล่องเครื่องมือ EP

ใช้สำหรับการหาค่าเหมาะที่สุดด้วยกำหนดการวิวัฒนาการ

โปรแกรม SCILAB ในกล่องเครื่องมือ EP

```
========================
        rand1.sci
========================
function x = rand1(a,b)
z = rand();
x = a + z*(b-a);
endfunction
```

```
========================
        randp.sci
========================
function p = randp(Ne,Np,p_limit)
for t=1:Np
   for u=1:Ne
      A = rand1(p_limit(u,1),p_limit(u,2));
      p(t,u) = A;
      if p(t,u)>p_limit(u,2) then
         p(t,u) = p_limit(u,2);
      elseif p(t,u)<p_limit(u,1) then
         p(t,u) = p_limit(u,1);
      end
   end
end
endfunction
```

```
========================
        survive.sci
========================
function Qsv = survive(Q)
Nrow = size(Q,1);
Nrow_half = round(Nrow/2);
Qsv = Q(Nrow_half+1:Nrow,:);
endfunction
```

```
========================
        calc_f.sci
========================
function f = calc_f(fobj,p)
rc = size(p);
for t=1:rc(1)
   f(t,1) = fobj(p(t,:));
end
endfunction
```

```
========================
        offspr.sci
========================
function x = offspr(p,f,xbeta,p_limit,p_range,option)
fmin = min(f); fmax = max(f); fave = mean(f);
fsum = sum(f);
if option==1 then
  fc = fmin;
elseif option==2 then
  fc = fave;
elseif  option==3 then
  fc = fmax;
elseif option==4 then
  fc = fsum;
else
  fc = fsum;
end
rp = size(p);
for t=1:rp(1)
  for u=1:rp(2)
    sigma = xbeta*f(t,1)*p_range(u,1)/fc;
    A = p(t,u) + grand(1,1,'nor',0,sigma);
    if A>p_limit(u,2) then
      A = p_limit(u,2);
    elseif A<p_limit(u,1) then
      A = p_limit(u,1);
    end
    x(t,u) = A;
  end
end
endfunction
```

```
========================
        epmain.sci
========================
function [xg,fg] = epmain(fobj,Ne,Np,xlimit,xbeta,opt)
// Evolution Strategies
// epmain function is a simple evolutionary
```

```
        programming version 1.0 coded in SCILAB
// fitness function is evaluated by using the objective
        function directly
// fobj     Objective function
// Ne       Number of variables
// Np       Number of population
// xlimit   limits of variables xlimit = [x1min x1max
            x1_bit_res;x2min x2max x2_bit_res;...];
// xbeta    scaling factor for creating offspring
// opt(1)   Maximum number of generation counted
// opt(2)   Maximum number of generation stalled
// opt(3)   display detail during calculation
// opt(4)   choice for variance calculation during the
            offspring creation
//       1: Use of the minimum objective function
            of population
//       2: Use of the average objective function
            of population
//       3: Use of the maximum objective function
            of population
//       4: Use of the total sum of the objective
            function of population (default)
// Developed by
//       Dr Thanatchai KULWORAWANICHPONG
//       Power & Energy Research Unit
//       School of Electrical Engineering
//       Institute of Engineering
//       Suranaree University of Technology
//       Nakhon Ratchasima, THAILAND
// Last Updated
//       28 July 2007
global pbest fbpest
p_range = xlimit(:,2) - xlimit(:,1);
count = 0;
p1 = randp(Ne,Np,xlimit); f1 = calc_f(fobj,p1);
if opt(3)==1 then
   disp('Initial population');
   disp([p1 f1])
end
[fg,idmin] = min(f1);
pg = p1(idmin,:);
fpbest = fg; pbest = pg;
f0 = fg;
subplot(2,1,1),plot2d(0,fpbest,style=[-9],logflag='nl')
xset('font',2,3);
xtitle('Convergence curve (best solution)','
          ','Objective');
subplot(2,1,2),bar(0,count);
xtitle(' ','Generation','Generation counted');
for k=1:opt(1)
   p2 = offspr(p1,f1,xbeta,xlimit,p_range,opt(4));
   f2 = calc_f(fobj,p2);
   if opt(3)==1 then
      disp(' ');
disp('================================');
      disp(['Generation # ' msprintf('%i',k)]);
disp('================================');
      disp('1. Offspring creation');
      disp([p2 f2]);
   end
         Pc = [p1 f1;p2 f2];
   if opt(3)==1 then
      disp(' ');
      disp('2. Competing pool');
      disp(Pc);
   end
   [y,indx] = sort(Pc(:,Ne+1));
   Pc_sort = Pc(indx,:);
   Psv = survive(Pc_sort);
   p1 = Psv(:,1:Ne); f1 = Psv(:,Ne+1);
   pg = p1(Np,:); fg = f1(Np,1);
   if fg<fpbest then
      fpbest = fg; pbest = pg;
      count = 0;
   else
      count = count + 1;
   end
   subplot(2,1,1),plot2d(k,fpbest,style=[-9],logflag='nl')
   subplot(2,1,1),plot2d([k-1 k],[f0 fpbest],logflag='nl')
   subplot(2,1,2),bar(k,count);
   f0 = fpbest;
   if opt(3)==1 then
      disp(' ');
      disp('3. Best solution found so far');
      disp([pbest fpbest count]);
   end
   if count>=opt(2) then
      break;
   end
end
xg = pbest; fg = fpbest;
endfunction
```

ค.7 กล่องเครื่องมือ particleswarm

ใช้สำหรับการหาค่าเหมาะที่สุดด้วยฝูงอนุภาค

โปรแกรม SCILAB ในกล่องเครื่องมือ particleswarm

```
========================
          rand1.sci
========================
function x = randx(xlimit)
// Uniformly random generation of an individual
          search vector
n = size(xlimit,1);
for u=1:n
   a =rand();
   x(1,u) = xlimit(u,1) + a*(xlimit(u,2)-xlimit(u,1));
   if x(1,u)>xlimit(u,2) then
      x(1,u) = xlimit(u,2);
   elseif x(1,u)<xlimit(u,1) then
      x(1,u) = xlimit(u,1);
   end
end
endfunction
```

```
========================
          randp.sci
========================
function [gbest,fgbest,k] = psomain(fobj,Ne,Np,xlimit,
                                          Vmax,opt)
// Particle Swarm Optimization Program
// psomain function is a simple particle swarm
     optimization version 1.0 coded in SCILAB
// fobj      Objective function
// Ne        Number of variables
// Np        Number of population
// xlimit   limits of variables xlimit = [x1min x1max
          x1_bit_res;x2min x2max x2_bit_res;...];
// xbeta    scaling factor for creating offspring
// opt(1)   Maximum number of generation counted
// opt(2)   Maximum number of generation stalled
// opt(3)   display detail during calculation
// Developed by
//          Dr Thanatchai KULWORAWANICHPONG
//          Power & Energy Research Unit
//          School of Electrical Engineering
//          Institute of Engineering
//          Suranaree University of Technology
//          Nakhon Ratchasima, THAILAND
// Last Updated
//          28 July 2007
p = [];
for u=1:Np
   pp = randx(xlimit);
   p = [p;pp];
   fp(u,1) = fobj(pp);
end
velocity = zeros(Np,Ne);
lbest = p;
flbest = fp;
[fgbest,idmin] = min(fp);
gbest = p(idmin,:);
fgbest0 = fgbest;
gbest0 = gbest;
count = 0;
if opt(3)==1 then
   disp('Initial population');
   disp([p fp])
end
subplot(2,1,1),plot2d(0,fgbest,style=[-9],logflag='nl')
xset('font',2,3);
xtitle('Convergence curve (best solution)','Generation
in progress','Objective');
subplot(2,1,2),bar(0,count);
xtitle(' ','Generation','Generation counted');
for k=1:opt(1)
   a = rand(2,1,'uniform');
   Gbest = [];
   for u=1:Np
      Gbest = [Gbest;gbest];
   end
   velocity   =   velocity   +   a(1,1)*(lbest-p)   +
a(2,1)*(Gbest-p);
   for u=1:Np
      for v=1:Ne
         if velocity(u,v)>Vmax then
            velocity(u,v)=Vmax;
         elseif velocity(u,v)<-Vmax then
            velocity(u,v)=-Vmax;
```

```
      end
    end
  end
  p1 = p + velocity;
  for u=1:Np
    for v=1:Ne
      if p1(u,v)<xlimit(v,1) then
        p1(u,v)=xlimit(v,1);
      elseif p1(u,v)>xlimit(v,2) then
        p1(u,v)=xlimit(v,2);
      end
    end
  end
  for u=1:Np
    f1(u,1) = fobj(p1(u,:));
    if f1(u,1)<fp(u,1) then
      lbest(u,:) = p1(u,:);
      flbest(u,1) = f1(u,1);
      if f1(u,1)<fgbest then
        gbest = p1(u,:);
        fgbest = f1(u,1);
      end
    end
  end
  if opt(3)==1 then
```

```
      disp(' ');
disp('================================');
      disp(['Generation # ' msprintf('%i',k)]);
disp('================================');
      disp('1. Solution Movement');
      disp([p1 f1]);
    end
    subplot(2,1,1),plot2d(k,fgbest,style=[-9],logflag='nl')
    subplot(2,1,1),plot2d([k-1                k],[fgbest0
fgbest],logflag='nl')
    subplot(2,1,2),bar(k,count);
    if fgbest>=fgbest0 then
      count = count + 1;
    else
      count = 0;
      fgbest0 = fgbest;
      gbest0 = gbest;
    end
    if count>=opt(2) then
      break;
    end
    p = p1;
    f = f1;
end
endfunction
```

ค.8 กล่องเครื่องมือ Tabu Search

ใช้สำหรับการหาค่าเหมาะที่สุดด้วยการค้นหาตาบู

โปรแกรม SCILAB ในกล่องเครื่องมือ

Tabu Search

```
========================
      randp.sci
========================
function p = randp(Ne,Np,p_limit)
vstep = (p_limit(:,2)-p_limit(:,1))/(p_limit(:,3)-1);
for t=1:Np
  for u=1:Ne
    A = rand(1,1,'uniform');
    B = round(A*xlimit(u,3));
    p(t,u) = p_limit(u,1) + B*vstep(u,1);
    if p(t,u)>p_limit(u,2) then
      p(t,u) = p_limit(u,2);
    elseif p(t,u)<p_limit(u,1) then
```

```
      p(t,u) = p_limit(u,1);
    end
  end
end
endfunction
```

```
========================
      tsmain.sci
========================
function [xgbest,fgbest] = tsmain(fobj,x0,xlimit,opt)

// Tabu Search Program
// tsmain function is a simple tabu search version 1.0
      coded in SCILAB
```

```
// fobj     Objective function
// x0       Initial guess solution
// xlimit   Limits of variables xlimit = [x1min x1max
              x1_point;x2min x2max x2_point;...];
// opt(1)   Maximum number of generation counted
// opt(2)   Maximum number of generation stalled
// opt(3)   Maximum number of search radius
              adjustment
// opt(4)   Display detail during calculation
// opt(5)   Option to randomly generate an initial
              guess solution
//          1: enable this option
//        Otherwise use x0 as the initial guess solution
// opt(6)   Option to apply a search radius (percent of
              variable range)
// Developed by
//          Dr Thanatchai KULWORAWANICHPONG
//          Power & Energy Research Unit
//          School of Electrical Engineering
//          Institute of Engineering
//          Suranaree University of Technology
//          Nakhon Ratchasima, THAILAND
// Last Updated
//          28 July 2007
Ne = length(x0);
xstep = (xlimit(:,2)-xlimit(:,1))./(xlimit(:,3)-1);
Xnb_p = zeros(Ne,Ne); Xnb_n = zeros(Ne,Ne);
if opt(5)==1 then
   x0 = randp(Ne,1,xlimit);
end
f0 = fobj(x0); count = 0; SRcount = 0;
xgbest = x0; fgbest = f0;
if opt(4)==1 then
   disp('Initial guess solution');
   disp([x0 f0])
end
subplot(2,1,1),plot2d(0,fgbest,style=[-9],logflag='nl')
xset('font',2,3);
xtitle('Convergence curve (best solution)','Generation
in progress','Objective');
subplot(2,1,2),barh([SRcount count 0],0.1);
subplot(2,1,2),xset('font',2,3);
xstring(1,3.2,'Generation Counted');
xstring(1,2.2,'Generation Stalled');
xstring(1,1.2,'Search Radius Adjusted');
for k=1:opt(1)
   for i=1:Ne
      for j=1:Ne
         a = rand(); xj = (-1)^j;
         Xnb(i,j) = x0(j) +
            round(a*opt(6)*xlimit(j,3))*xj*xstep(j,1);
         Xnb(i+Ne,j) = x0(j) -
            round(a*opt(6)*xlimit(j,3))*xj*xstep(j,1);
      end
      Fnb(i,1) = fobj(Xnb(i,:));
      Fnb(i+Ne,1) = fobj(Xnb(i+Ne,:));
   end
   [FLbest,IDL] = min(Fnb);
   XLbest = Xnb(IDL,:);
   if opt(4)==1 then
      disp(' ');
disp('================================');
      disp(['Generation # ' msprintf('%i',k)]);
disp('================================');
      disp('1. Best neighbor');
      disp([XLbest FLbest]);
   end
   if FLbest<fgbest then
      fgbest = FLbest; xgbest = XLbest; count = 0;
   else
      count = count + 1;
   end
   x0 = XLbest;
   subplot(2,1,1),plot2d(k,fgbest,style=[-9],logflag='nl')
   subplot(2,1,1),plot2d([k-1 k],[f0 fgbest],logflag='nl')
   subplot(2,1,2),barh([SRcount count k],0.1);
   f0 = fgbest;
   if count>opt(2) then
    SRcount = SRcount + 1;
    opt(6) = opt(6)*0.75;
   end
   if SRcount>opt(3) then
      break;
   end
end
endfunction
```

แบบประวัติส่วนตัว

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
111 ถ.มหาวิทยาลัย ต.สุรนารี อ.เมือง จ.นครราชสีมา 30000
โทรศัพท์ 0 4422 4404 โทรสาร 0 4422 4601

thanatch@sut.ac.th

ผู้ช่วยศาสตราจารย์ ดร.ธนัดชัย กุลวรวานิชพงษ์
Asst. Prof. Dr. Thanatchai Kulworawanichpong

การศึกษา/คุณวุฒิ

2540 วิศวกรรมศาสตรบัณฑิต (วิศวกรรมไฟฟ้า) มหาวิทยาลัยเทคโนโลยีสุรนารี
2543 วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า) จุฬาลงกรณ์มหาวิทยาลัย
2546 PhD (Electronic and Electrical Engineering)
The University of Birmingham, United Kingdom

ตำแหน่งปัจจุบัน

1. **ผู้ช่วยศาสตราจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า**
   สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
2. **หัวหน้าหน่วยวิจัยไฟฟ้ากำลังและพลังงาน** สาขาวิชาวิศวกรรมไฟฟ้า
   สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ประวัติการทำงาน

พ.ศ. 2549 - ปัจจุบัน ผู้ช่วยศาสตราจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเทคโนโลยีสุรนารี
พ.ศ. 2543 - 2549 อาจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเทคโนโลยีสุรนารี
พ.ศ. 2540 - 2543 อาจารย์พิเศษระดับปริญญาตรี สาขาวิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเทคโนโลยีสุรนารี

เกียรติประวัติที่ได้รับ

พ.ศ. 2543 - 2546 **นักเรียนทุนรัฐบาลไทย กระทรวงพลังงาน**
ศึกษาต่อระดับปริญญาเอก สหราชอาณาจักร
พ.ศ. 2541 - 2542 **นักเรียนทุนพัฒนาอาจารย์** ทบวงมหาวิทยาลัย
ศึกษาต่อระดับปริญญาโท จุฬาลงกรณ์มหาวิทยาลัย
พ.ศ. 2540 **เกียรตินิยมอันดับ 1 เหรียญทอง**
มหาวิทยาลัยเทคโนโลยีสุรนารี

ผลงานวิชาการ

(1) งานวิจัยตีพิมพ์ บทความวิชาการเผยแพร่ระดับชาติและนานาชาติ 40 เรื่อง
(2) งานแต่งตำรา ได้เขียนตำราทั้งตีพิมพ์ เผยแพร่ และใช้ประกอบการสอน
การบรรยายระดับอุดมศึกษา 8 เล่ม

ผู้ช่วยศาสตราจารย์ ดร.ธนัดชัย กุลวรวานิชพงษ์
Asst. Prof. Dr. Thanatchai Kulworawanichpong