Finite Convergence and Performance Evaluation of Adaptive Tabu Search

Deacha Puangdownreong[†], Thanatchai Kulworawanichpong, and Sarawut Sujitjorn

School of Electrical Engineering, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand
[†] Seconded from Department of Electrical Engineering, Faculty of Engineering, South-East Asia University, Bangkok, 10160, Thailand

Abstract. The naïve tabu search (NTS) has been enhanced with two adaptive mechanisms namely back-tracking and adaptive search radius. The proposed search is called adaptive tabu search (ATS). The paper provides convergence and performance analyses of the ATS.

1 Introduction

The tabu search (TS) method was proposed in 1986 by Glover to solve combinatorial optimization problems [1]. Two principles of the TS method are the neighborhood search approach and the tabu list (TL), respectively. The method is often applied in the simplest form referred to as naïve tabu search (NTS) that is usually trapped by local solutions. The method has found a variety of applications such as [2-7] although a dead-lock by a local solution can occur.

We propose an enhanced version of the NTS that composes of two mechanisms: namely back-tracking, and adaptive search radius mechanisms. Moreover, the method possesses a random movement of solution findings in the preset neighborhood. These additional features have made the method more efficient and powerful. The search method has been named the adaptive tabu search (ATS) and successfully applied to identify linear and nonlinear system models [8].

The convergence analysis of the conventional TS method has been proved [9,10]. The proofs were based on the deterministic recency and frequency approaches. In this paper, a new proof is provided for the ATS method to ensure its convergence. In addition, the performance evaluation was conducted through many-thousand search trials on three nonlinear mathematical functions. These are the unsymmetrical trigonometric sum, one of Bohachevsky's functions, and the circle function. This paper reports the finite convergence analysis, and the performance evaluation of the ATS.

2 Convergence of the ATS

2.1 Definitions

Definition 1. Let Ω be a finite search space having n members ($n < \infty$).

Definition 2. Let the finite search space Ω have k strictly local minima and be divided into k regions denoted by Λ_i (i = 1,2,…,k). Each region having a total of w members contains only one local minimum, which must not be located at the boundary.

Definition 3. Let Ψ be a randomly-created finite sub-space of Ω, $\Psi \subset \Omega$, having m members ($m < n$).

Definition 4. Let a finite sequence $S = \{x_{0,i}\}$, i = 1,2,…,p, be a collection of solution movements, x_0, consisting of p solutions to reach the global minimum ($k < p$).

Definition 5. Let $Time(x)$ be a time consumed to visit a solution x in the search space Ω and it assumes to be constant for visiting any $x \in \Omega$. That is $Time(x_i) = Time(x) > 0$ for i = 1,2,…,n.

Definition 6. Let *Iteration* be a cumulative number of iterations indicating how many solutions in Ψ were already visited. *Iteration* is initialized at the start of a new sub-space exploration. After exploring all generated solutions in Ψ, the updated *Iteration* is equal to m and the time is $m \cdot Time(x)$.

Definition 7. Let *Count* be a cumulative search round of sub-space explorations indicating how many sub-spaces in Ω were already explored entirely. *Count* is initialized only once at the beginning. *Count* is updated when all solutions in any Ψ have been visited. After an entire exploration, *Count* is equal to p and the overall time consumed is $p \cdot m \cdot Time(x)$.

Definition 8. Let BT denote the back-tracking mechanism to allow the use of any previously visited local minimum recorded in the TL for generating a new starting point rather than the one just obtained.

Definition 9. Let AR denote the adaptive search radius mechanism that reduces the accessing time to a local minimum. Given that $\rho = \mu.r$ is the adaptive radius where r is a nominal radius and an arbitrary constant while $0 < \mu \leq 1$. The radius is used to define a neighborhood around a current solution.

2.2 ATS Algorithms

Step 1) Initialise the Tabu List (TL= $\varnothing$), *Iteration* = 0 and *Count* = 0.

Step 2) Randomly select an initial solution $x_{0,Count}$ from the search space Ω and assign it as an initial global minimum x^*. The time used for visiting the initial solution is $Time(x)$.

Step 3) Update *Count* by 1, then create a sub-space Ψ_{Count}. Evaluate the objective function of $\forall x \in \Psi_{Count}$. Update *Iteration* by 1 when an x is examined. A solution with the minimum objective function is x'. When the exploration of the subspace is finished (*Iteration* = m), the cumulative time consumed is $m \cdot Time(x)$.

Step 4) If $x' < x_{0,Count}$, keep $x_{0,Count}$ in the TL and set $x_{0,Count} = x'$. Otherwise put x' in the TL instead.

Step 5) Update the global minimum. $x^* = x_{0,Count}$ if $x_{0,Count} < x^*$.

Step 6) Evaluate the termination criteria (TC) and the aspiration criteria (AC).
- Go to step 7 if TC is satisfied, otherwise repeat step 3.
- Activate the AR mechanism to speed up the searching process.
- Activate the BT mechanism if a local minimum trap occurs. Reset *Iteration* and repeat step 3.

Step 7) Terminate the search process. Accept the last updated x^* as the global solution.

Only a few numbers of solutions in Ω would be randomly visited and it is sufficient to locate the global minimum by $Count = p$ and $p \cdot m \cdot Time(x)$ of the overall time consumed.

2.3 Proof of Global Convergence

Theorem A. If a total number of members, m, in a sub-space Ψ is large enough to give good representatives of a neighborhood, a local minimum nearby can be found by generating a sequence of some successive sub-spaces.

Proof. Let $\hat{x}$ be a strictly local minimum in a considered region, $\Lambda(x_0)$, of x_0. That is $f(\hat{x}) < f(x)$ for $\forall x \in \Lambda(x_0)$ and also for $\forall x \in N_\rho(x_0)$. This implies that both $N_\rho(x_0)$ and sets of solutions nearby lie on the same region, $\Lambda(x_0)$. In a similar manner as the Hill-climbing algorithm, updating a current solution leads descent direction to reach a nearby local minimum.

Given an initial solution $x_{t=0}$ in a finite sub-space $\Psi_t \subset \Omega$. To generate a sequence of x_{t+1}, the descent property must be held to guarantee that a next move leads to a local minimum. At any current solution, there are only two possible outcomes that are either i) the solution is improved, $f(x_{t+1}) < f(x_t)$, or ii) the solution is not improved, $f(x_{t+1}) \geq f(x_t)$. In the ATS method, given that the neighborhood, $N_\rho(x_t)$, of the current solution x_t is created and has a total of N members. The sub-space $\Psi_{t+1} \subset N_\rho(x_t)$ is then randomly generated with m finite members where $m < N$, and m is constant. This process is based on the assumption that not all members in the neighborhood give better cost than x_t does, but only u members of $N_\rho(x_t)$ satisfy $f(x) < f(x_t)$ where $x \in \Psi_{t+1}$. The probability to improve the solution $P = P\left(f(x) < f(x_t)\right)$ is given as follows.

Case 1: $(m > N - u)$

$P = 1$, in this case, at least one of m satisfies the condition.

Case 2: $(m \leq N - u)$

In this case, there are $\binom{N}{m} = \frac{N!}{(N-m)!m!}$ of the possible combination for randomly selecting m members out of N. In addition, $\binom{N-u}{m} = \frac{(N-u)!}{(N-u-m)!m!}$ is a total of ways that the solution is not improved. Thus, the probability of the sampling, which cannot improve the current solution, is shown as follows.

$$P = \frac{(N-u)!(N-m)!}{N!(N-u-m)!} \tag{1}$$

When m and N are both fixed, Eq. (1) depends on u only. u is large when the current solution is close to the local minimum. This search process updates the current solution with the best member in each iteration. Therefore, the solution will move towards the local minimum when the time increases. That is $\lim_{t\to\infty} u(t) = 0$. From Eq. (1), the probability of the event that the solution cannot be improved anymore (local minimum found) is expressed below.

$$\lim_{t\to\infty} P(t) = \lim_{t\to\infty} \frac{\big(N-u(t)\big)!(N-m)!}{N!\big(N-u(t)-m\big)!} = 1 \tag{2}$$

When the process is repeatedly performed with a considerable amount of time, the probability of finding the local eventually global minimum is close to unity.

Theorem B. The BT mechanism leads the search process to obtain multiple local minima. Among them, one is the global minimum.

Proof. As previously mentioned, the random search process might fail to escape from a trap due to ineffectiveness of the algorithms. The use of some solution stored in the TL as an initial solution for the next search round enables various search directions. It increases possibility to run away from the already visited local minimum. Given n_{re} be a counter for a solution cycling. "Solution cycling" means that the search cannot escape the entrapment of the just visited local minimum, so the movement of solutions will return to the just visited local minimum at the end of the next search round. The counter is updated every time a new final solution of any search round being equal to the one previously visited and already stored in the list. Let n_{re_Max} be the maximum number allowance of the solution cycling. Therefore, the BT mechanism is activated by the following condition. If $n_{re} < n_{re\_Max}$, then continue the search whether it can eventually escape from the solution lock, otherwise, performing the BT process. Once $n_{re} \geq n_{re\_Max}$, one of the solutions recorded in the TL is selected to be a new initial solution for creating the next sub-space Ψ. $n_{re} \geq n_{re\_Max}$, is an Aspiration Criteria. The BT mechanism will select a solution $x_h \in$ TL in such a way that $x_h = \max_{x_i \in TL} \|x_i - x_o\|$ and the condition $f(x_0) < f(x_h)$ must hold. After selecting the solution, set $x_0 = x_h$ as a new initial solution for the next search round. Therefore,

i) If the local minimum $\hat{x}$ is obtained already and the length of TL is sizeable, there exists at least one solution that is relatively close to the boundary of $\Lambda(x_0)$. Therefore, length(TL) $>> 1 \rightarrow \exists x \in$ TL $\wedge \|x - x_B\| < \gamma$, where x_B is a boundary point and γ is the maximum allowance.

ii) During the search process if a current x_0 is relatively close to boundary of $\Lambda(x_0)$ as stated in (i), together with a certain radius ρ that is relatively large enough to be able to reach some solutions outside $\Lambda(x_0)$, the best solution of a current Ψ can be located outside $\Lambda(x_0)$.

- $\left(\Psi(x_0)-\Lambda(x_0)\right)\subset\left(N_\rho(x_0)-\Lambda(x_0)\right)\not\subset\Lambda(x_0)\rightarrow\exists x\notin\Lambda(x_0)$

iii) With proceeding a new search from a solution found outside $\Lambda(x_0)$ according to (ii), this restarts a new descent process to reach another local minimum of a new region nearby. By repeating the procedures with all k different local minima being found within a finite search time $p\cdot m\cdot Time(x)<\sum_{i=1}^{n}w_i\cdot Time(x)$, and with the cost-value termination criterion being completely satisfied, one of the local minima is the global minimum.

3 Performance Evaluation

The ATS was coded in MATLAB™ for running on a Pentium 4, 1.6 GHz, 256 Mbytes RAM, 40 Gbytes HD. The search was conducted against three following functions to find the global minimum. Firstly, the unsymmetrical trigonometric sum function (TSF) is expressed by Eq.(3). The global minimum is on x = –0.26 making $f(x) = 4.56\times10^{-5}$ and is used as the termination criterion. Secondly, the Bohachevsky's function (BF) [11], Eq.(4), and thirdly, the circle function (CF) [12], Eq.(5), are used. Both functions have the global minimum at $x = y = 0$ with $f(0,0) = 0$. We use 1×10^{-5} to approximate zero and it is set as the termination criterion for the last two cases.

$$f(x)=\left(\sin(x)+2.5\sin(2x)+1.5\sin(4x)+2\sin(8x)\right)+x^2+4.4716 \tag{3}$$

$$f(x,y)=x^2+2y^2-0.3\cos(3\pi x)-0.4\cos(4\pi y)+0.7\,,\ x=y\in[-1.0\,,1.0] \tag{4}$$

$$f(x,y)=\left(x^2+y^2\right)^{1/4}\left(\sin^2\left(50\left(x^2+y^2\right)^{1/10}\right)+0.1\right),\ x=y\in[-0.5\,,0.5] \tag{5}$$

We report only the performance of the ATS because the NTS is completely unable to locate the global minimum. Five parameters are considered to influence the search performance of the ATS. They are: i) the initial search radius (R), ii) the number of neighborhood members (n), iii) the number of repetitions of solution cycling before back-tracking (n_re_max), iv) the k^{th} backward solution selected by the back-tracking mechanism (k^{th} backward selection), and v) the percentage of search radius reduction compared to the radius before adaptation.

The first four tests begin with tuning four parameters (R, n, n_re_max and k^{th} backward selection), where the search radius is non-adaptive. Each parameter setting is carried out with the maximum of 1,000 trials. It starts with a random initial solution generated by MATLAB. It stops when either of the following termination criteria is met: i) the maximum search round of 10,000, or ii) the cost function ≤ ε (a very small number to approximate zero).

The setting from the first four tests that gives the best result is applied to conduct the fifth test, in which the search radius is adaptive. The adaptive radius scheme is set to have three steps of reduction as: i) if [cost function < 10^{-1}] then [$R^{(new)} = R^{(old)}/DF$]; ii) if [cost function < 10^{-2}] then [$R^{(new)} = R^{(old)}/DF$]; and iii) if [cost function < 10^{-3}]

then $[R^{(new)} = R^{(old)}/DF]$, where $R^{(old)}$ and $R^{(new)}$ are the search radius before and after adaptation, and DF is the factor of radius reduction and it is assigned by the following values: 10, 15, 20, 25, and 30% of the current radius. The parameters for the fifth test obtained from Tables 1-4 are given as follows: TFS–{R = 2.5%, n = 30, n_re_max = 5, and $k^{th} = -2$}; BF–{R = 10.0%, n = 30, n_re_max = 5, and $k^{th} = -5$}; and CF–{R = 7.5%, n = 40, n_re_max = 5, and $k^{th} = -5$}. The results of this test are summarized in Table 5. It shows that 20-25% reduction of the search radius gives good performance in terms of speed and convergence.

4 Conclusions

The convergence analysis and the performance evaluation of the ATS method have been presented in this paper. The global convergence can be guaranteed. We present the ATS performance in terms of function minimization. The NTS is completely unable to locate the global solution while the ATS can. To apply the ATS successfully, one is recommended to evaluate its performance against a set of interested problems such that its parameters rendering fast search could be identified.

Table 1. Effects of search radius (R)

	Average search round			Average search time (s)			No of successful trials		
R (%)	TSF	BF	CF	TSF	BF	CF	TSF	BF	CF
2.5	459.45	7923.30	9341.4	1.31	48.51	52.43	1000	203	214
5.0	936.41	7156.60	6610.2	2.81	41.18	38.87	1000	296	405
7.5	897.83	3876.60	3742.3	3.28	20.95	20.63	1000	757	885
10.0	924.17	1353.10	4878.8	3.92	6.06	25.40	1000	1000	816
12.5	992.33	2263.70	5955.1	4.78	10.60	33.03	1000	988	700
15.0	933.06	3071.40	6796.2	5.02	14.91	37.60	1000	954	548
20.0	1035.50	4832.40	8038.0	6.78	24.59	47.148	1000	808	363

Table 2. Effects of neighborhood size (n)

	Average search round			Average search time (s)			No of successful trials		
n	TSF	BF	CF	TSF	BF	CF	TSF	BF	CF
10	1473.5	4135.10	7134.4	2.14	10.33	21.82	998	893	526
20	707.15	2203.30	4802.4	1.44	7.72	20.23	1000	987	804
30	459.45	1353.10	3742.3	1.31	6.06	20.63	1000	1000	885
40	357.17	1089.10	3029.3	1.40	6.37	19.37	1000	1000	910
50	287.32	904.08	2928.5	1.33	6.49	25.06	1000	1000	873
60	230.81	802.53	2304.2	1.26	6.70	22.32	1000	1000	908

Table 3. Effects of n_re_max

n_re_ max	Average search round			Average search time (s)			No of successful trials		
	TSF	BF	CF	TSF	BF	CF	TSF	BF	CF
5	472.52	1310.90	3029.3	1.36	5.91	19.37	1000	1000	910
10	473.65	1322.10	3279.4	1.43	5.95	22.15	1000	1000	871
15	477.45	1353.10	3438.3	1.43	6.06	23.91	1000	1000	858
20	475.71	1518.80	3466.4	1.44	6.61	24.12	1000	998	851
25	485.43	1438.80	3360.0	1.42	6.51	23.37	1000	1000	862

Table 4. Effects of the k^{th} backward selection

k^{th}	Average search round			Average search time (s)			No of successful trials		
	TSF	BF	CF	TSF	BF	CF	TSF	BF	CF
-1	461.60	1498.40	3455.1	1.49	6.33	24.09	1000	998	866
-2	493.99	1488.40	3364.4	1.47	6.68	23.65	1000	998	871
-3	477.19	1478.40	3253.7	1.57	6.90	22.21	1000	998	867
-4	464.33	1586.30	3115.4	1.52	7.31	21.45	1000	998	884
-5	470.31	1462.80	3029.3	1.53	6.21	19.37	1000	999	910

Table 5. Effects of reduced R

Reduced R	Average search round			Average search time (s)			No of successful trials		
	TSF	BF	CF	TSF	BF	CF	TSF	BF	CF
10%	11.07	24.36	1195.4	0.03	0.09	8.70	1000	1000	892
15%	13.36	26.18	1200.9	0.04	0.10	10.52	1000	1000	887
20%	17.85	30.16	600.94	0.05	0.13	4.20	1000	1000	942
25%	22.27	38.41	601.14	0.06	0.16	4.42	1000	1000	940
30%	34.64	64.24	802.92	0.11	0.31	7.15	1000	1000	914

References

1. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers and Operations Research 13 (1986) 533-549
2. Glover, F., Mulvey, J.M., Hoyland, K.: Solving dynamic stochastic control problems in finance using tabu search with variable scaling. Statistics and Operations Research Technical Report SOR-94-13, Princeton University, (1994)
3. Zhang, G., Habenicht, W., SpieB, W.E.L.: Improving the structure of deep frozen and chilled food chain with tabu search procedure. Journal of Food Engineering 6, 1 (2003) 67-79
4. Silva, E.L.D., Areiza, J.M.O., Oliveira, G.C.D., Binato, B.: Transmission network expansion planning under a tabu search approach. IEEE Trans Power Systems 16, 1 (2001) 62-68
5. Kulworawanichpong, T., Sujitjorn, S.: Optimal power flow using Tabu search. IEEE Power Engineering Review 22, 6 (2002) 37-40
6. Lin, B., Miller, D.C.: Application of tabu search to model identification. AIChE Annual Meeting Los Angleles. (2000)

7. Cordeau, J-F., Laporte, G.: A tabu search heuristic for the static multi-vehicle dial-a-ride problem. Transportation Research Part B: Methodological. 37, 6 (2003) 579-594
8. Puangdownreong, D., Areerak, K-N., Srikaew, A., Sujitjorn, S., Totarong, P.: System Identification via Adaptive Tabu Search. Proc. IEEE Int. Conf. on Industrial Technology (ICIT'02) 2 (2002) 915-920
9. Glover, F., Hanafi, S.: Finite Convergence of Tabu Search. Proc. MIC'2001-4th Metaheuristics Int. Conf, (2001) 333-336
10. Hanafi, S.: On the Convergence of Tabu search. J. Heuristics 7 (2000) 47-58
11. Bohachevsky, I.O., Johnson, M.E., Stein, M.L.: Generalized simulated annealing for function optimization. Technometrics 28, 3 (1986) 209-218
12. Miller, S.D., Marchetto, J., Airaghi, S., Koumoutsakos, P.: Optimization based on bacterial chemotaxis. IEEE Trans. Evol. Comput. 6 (2002) 16-29