ScienceAsia 30 (2004): 183-190

Convergence Analysis of Adaptive Tabu Search

Deacha Puangdownreong, Sarawut Sujitjorn* and Thanatchai Kulworawanichpong

[a] School of Electrical Engineering, Suranaree University of Technology, Nakhon Ratchasima, 30000, Thailand.

* Corresponding author, E-mail: sarawut@ccs.sut.ac.th.

Received 28 Mar 2003
Accepted 30 Jan 2004

ABSTRACT: This paper presents a convergence proof of the adaptive tabu search (ATS) algorithms. The proof consists of two parts, i.e. convergence proof of all interested solutions in a finite search space, and that of searching processes of the ATS algorithms to the global minimum. With the proposed definitions and theorems, the proofs show that the ATS algorithms based on a random process have finite convergence. The searching process also converges to the (near) global minimum rapidly. Two applications, the global minimum finding of Bohachevsky's function and the identification of the nonlinear pendulum model, serve to illustrate the effectiveness of the ATS algorithms.

KEYWORDS: adaptive tabu search, back-tracking, adaptive search radius, convergence analysis
.

INTRODUCTION

Glover[1] proposed the tabu search (TS) method in 1986 to solve combinatorial optimization problems. The principles of the TS method are the neighborhood search approach and the tabu list (TL). Although tabu search has been around for many years, and its fundamental principles are well-elaborated, it is still often implemented in a very simplistic form that disregards all but the most rudimentary features of the method. We refer to such a version by the name "Naïve Tabu Search" (NTS). Successful applications of the method are found in many areas such as flow shop,[2] finance,[3] food processing,[4] power systems,[5] transportation,[6] etc. However, it was shown that the NTS method, of the type sometimes applied in the literature, could not completely escape a local minimum lock.[7] The NTS method was enhanced by two mechanisms: namely back-tracking, and adaptive search radius mechanisms,[7] in order to avoid such a lock. They can be regarded generally as the intensification strategies[8,9] of (1) re-starting from high quality prior solutions and (2) progressive neighborhood contraction, respectively. Moreover, the back-tracking concept is akin to the approach used by Nowicki and Smutnicki[2] and the search subspace reduction has been applied to solve some control problems in finance.[3] The NTS method presented by Areerak[7] was also modified to possess a random movement of solution findings in the preset neighborhood. These additional features have made the modified TS more efficient. This modified version of the NTS has been named the adaptive tabu search (ATS) and successfully applied to linear and nonlinear system models.[10]

The convergence analysis of the conventional TS method has been proved.[11,12] The proofs were based on the deterministic recency and frequency approaches. In this paper, new proofs are provided for the ATS method to ensure its finite convergence. Particularly, our analysis applies to a form of TS that uses the two fundamental intensification strategies mentioned earlier. The proofs are divided into two main parts. Firstly, finite convergence of all solutions of interest in a finite search space is proceeded. Secondly, verification of global convergence for the searching process of the ATS algorithms is accomplished. Both of them are based on pure logic and heuristics. This paper also provides two applications of the ATS method to confirm its performance in the last section.

FINITE CONVERGENCE OF SOLUTIONS

Finite convergence of solutions is the convergent proof of entirely interested solutions in finite search space.

Definition 1: Let Ω be a finite search space having the finite members, n, which are entirely interested solutions x_i , $i = 1,\ldots,n$, where n is the finite positive integer and $n < \infty$.

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

As defined above, all regions are mutually exclusive. So, two key properties are shown as follows: $\bigcup_{i=1}^{n} \Lambda_i = \Omega$ and $\bigcap_{i=1}^{n} \Lambda_i = \varnothing$.

Definition 3: Let Ψ be sub search spaces in Ω, $\Psi \subset \Omega$, containing the finite members, m, which are entirely interested solutions x_i, $i = 1,\ldots,m$, in each sub search space, where m is also the finite positive integer, $m < n$, and m is constant.

The mechanism of creation of Ψ is defined as follows. In the search process for the optimum solution, the current solution is defined as x_0. The Ψ will be uniformly randomly created around x_0 by $\|\tilde{x} - x\| \le \rho$, in which $\tilde{x}$ is any random solution and r is a finite search radius.

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

Regarding definition 4, based on the sequence S, a collection of sub-spaces, $\theta = \{\Psi_i\}, i = 1,2,\ldots,p$, is thus formed. The search space is not necessary to be entirely explored, i.e. $\bigcup_{i=1}^{p} \Psi_i \subset \Omega$. Construction of the next generated sub-space Ψ_{i+1} according to the ATS mechanism performs by assigning the best solution of the previously generated sub-space Ψ_i as the centre point $x_{0,i+1}$. This random approach gives $\Psi_t \cap \Psi_{t+1} \neq \varnothing$, where $\varnothing$ is the empty set, because at least the centre point $x_{0,i+1}$ is in both Ψ_i and Ψ_{i+1}.

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

Regarding definition 5, by the solution-visited time defined above, visiting m randomly generated solutions of Ψ spends $m \times Time(x)$. So, to entirely explore all generated sub-spaces, the overall time consumed is $p \times m \times Time(x)$. Note that by the proposed method, the overall time to explore all generated sub-spaces must be less than the overall time to explore the entire search space to ensure the fast and robust convergence. That is, $\sum_{i=1}^{n} w_i \cdot Time(x) > p \cdot m \cdot Time(x) > m \cdot Time(x) > Time(x) > 0$.

Definition 6: Let *Interation* be a cumulative number of iterations to count how many solutions in Y were already visited. *Interation* is set to be zero before starting a new sub-space exploration. Once any solution in the sub-space was visited, thus *Interation*=*Interation*+1. After all generated solutions in Ψ have been explored, *Interation* is now equal to m and the time is now $m \times Time(x)$.

Definition 7: Let *Count* be a cumulative search round of sub-space explorations to count how many sub-spaces in Ω were already explored entirely. *Count* is set to zero only once at the beginning. When all solutions in any Ψ have been visited, thus *Count*=*Count*+1. After all generated subspaces in have been explored entirely, *Count* is equal to p and the overall time consumed is $p \times m \times Time(x)$.

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. This scheme is added to enhance the ability to escape a local minimum entrapment.

Definition 9: Let AR denote the adaptive search radius mechanism to give an alternative sub-space which is able to reduce the time to access a local minimum. Given that r = $\mu \times r$ is the adaptive radius where r is a nominal radius and an arbitrary constant while $0 < \mu \le 1$. The radius is used to define a neighborhood around a current solution.

The formulation of the ATS method is based on the nine definitions and the corresponding ATS algorithms are as follows.

Step 1) Initialise the Tabu List (TL=$\varnothing$), *Interation* = 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) Set *Count* =*Count* +1, then create a sub-space Ψ_{Count}. Evaluate the objective function of $\forall \in \Psi_{Count}$. *Iteration* is updated when a single x is examined (*Iteration = Iteration + 1*). A solution that gives the minimum objective function among them is given as x'. When the exploration of the subspace is finished (*Iteration* = m), the cumulative time consumed is $m \times Time(x)$.

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. $x^* = x_{0,Coun}$ if $x_{0,Coun} < x^*$.

Step 6) Check the termination criteria (TC) and the aspiration criteria (AC), respectively.

-Go to step 7 if TC is satisfied, otherwise repeat step 3.

-Activate the AR mechanism if necessary 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. The last updated x^* is the global minimum found.

As can be seen, only a number of solutions in Ω would be randomly visited and it is sufficient to locate the global minimum by *Count*=p and $p \times m \times Time(x)$ of the overall time consumed.

Verification of the global convergence

Verification of the global convergence is the proof of convergence of the ATS algorithms to the global optimum solution. This proof is based on three principles as follows.

1. *Guarantee the local minimum:* this proves the convergence of the search process to a local optimum solution in each sub search space Ψ.

2. *Guarantee the speed of search process:* This proof shows that the search process having the AR mechanism can converge to a local optimum solution with shorter time by decreasing the search radius r.

3. *Guarantee the global minimum:* this proof shows that the use of the BT mechanism makes it possible for the ATS method to find the multiple local minima. When the process is finished, one of the many local optimum solutions is the (near) global one.

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 a few successive sub-spaces.

Proof: A sub-space Ψ is, uniformly and randomly, generated around an initial x_0 solution with a certain radius ρ. The set of all solutions x, which are contained in the ball $\|x-x_0\|\le\rho$, is denoted by $N_\rho(x_0)$ and called "Neighborhood" of x_0. It is clear that $\Psi\subset N_\rho(x_0)$ and m must be less than a total number of neighborhood members.

Let $\check{x}$ be a strictly local minimum in a considered region, $\Lambda(x_0)$, of x_0. That is $f(\check{x})<f(x)$ 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, $\Lambda(x_0)$.

As applying the similar concept of the local search or the so-called Hill-climbing algorithm, updating a current solution leads the descent direction to a nearby local minimum. Since any Ψ_i is formed and the best solution is already obtained, that gives $f(x'_i)\le f(x'_j)$ for $i>j$. At any i^{th} search round, a distance between x'_i and the local minimum $\check{x}$ is defined by a positive number $\xi_i=\|x'_i-\check{x}\|$ and the associated error of the objective functions is given by $g_i=\|f(x'_i)-f(\check{x})\|$. If the search round i^{th} is relatively large, say M, the error and solution found thus far are bounded. This yields $g_i=\|f(x'_i)-f(\check{x})\|<\varepsilon$ for $\xi_i=\|x'_i-\check{x}\|<\delta$ where ε and δ are relatively small positive numbers being the maximum error allowances of $f(x'_i)$ and x'_i, respectively. To summarise the proof, the following steps are presented.

i) Define the descent property of the algorithm, $f(x'_i)\le f(x'_j)$ for $i>j>0$.

ii) That is for $0\le\|f(x'_{i+1})-f(\check{x})\|\le\|f(x'_i)-f(\check{x})\|$ for $\forall i<0$.

iii) The ATS algorithms have finite convergence property when the following statement is satisfied. $\lim_{i\to M}\xi_i<\delta\rightarrow\lim_{i\to M}g_i<\varepsilon$, where $1<M<n<\infty$.

iv) If there exists at least one positive integer M which satisfies (iii), the process is said to have the finite convergence to a local minimum with $M\times Time(x)$ of the time consumed.

Theorem B: The AR mechanism can accelerate the search process to succeed finding the minimum. By appropriately adjusting the search radius ρ, the process can rapidly converge to the local minimum in a considered region, Λ.

Proof: The AR mechanism is used for increasing the speed of the search process. It can make the process converge to a local minimum faster than the conventional Tabu search without this scheme.

Let the sub-space Ψ have a search radius $\rho=\mu\times r$. At the beginning of the search process, initiate the search radius $\rho=r(\mu=1.0)$. From theorem A, when an updated solution is very close to $\check{x}$, say $\xi<\rho$ or $\check{x}\in N_\rho(x'_i)$, the probability of getting $\check{x}$ (one of N) is $\frac{m\times m!}{N(N-1)\cdots(N-m+1)}$ where N is a total number of $N_\rho(x'_i)$ members and $m<N$ due to the previous clarification in theorem A. Clearly, the larger radius leads (larger N) to the smaller probability. By this simple logic, the radius is thus reduced by a factor of μ ($0<\mu<1$) to yield a better chance to reach $\check{x}$ faster. After successfully obtaining $\check{x}$, the radius must be reset to the default value of r to restart a new search round again by using a different initial solution. Briefly, this proof is summarised below.

i) Let $\rho=\mu\cdot r,\ 0<\mu\le1.0\wedge r\in\Re^+$, where $\Re^+$ is positive real numbers.

ii) $\xi<\rho$ that is $\hat{x}\in N_\rho(x'_i)\rightarrow P_\rho(\hat{x})=\dfrac{m\times m!}{N(N-1)\cdots(N-m+1)}$

iii) Let $\xi<\rho'<\rho\wedge(N>N'\in I^+)$. Thus,

$$\left(P_{\rho'}(\hat{x})=\frac{m\times m!}{N'(N'-1)\cdots(N'-m+1)}\right)>\left(P_\rho(\hat{x})=\frac{m\times m!}{N(N-1)\cdots(N-m+1)}\right)$$

where ρ' is a new decreased radius, N is the total number of $N_{\rho'}(x'_i)$ members, N' is the total number of $N_{\rho'}(x'_i)$ members and $m<N'<N$.

iv) With decreasing radius as shown in iii), the speed of the search process is improved due to increasing the local-minimum-found probability. If a total number of search rounds to locate a local minimum is M (M is a positive integer) without the adaptive radius, the time consumed is thus $M\times Time(x)$. Once the AR mechanism is applied, the total search round is reduced by a factor of α where $0<\alpha<1$, therefore the time consumed is $\alpha\times M\times Time(x)<M\times Time(x)$.

The AR mechanism is activated when a current solution is sufficiently close to a local minimum. With a sequence of p sub-spaces, only k local minima can be found. This implies that the AR mechanism is activated only k times throughout the searching process. Thus, the overall searching time when the AR mechanism is included is

$$\sum_{j=1}^{k} \alpha_j \times M_j \times Time(x) < \sum_{j=1}^{k} M_j \times Time(x) = p \times m \times Time(x) < \sum_{q=1}^{n} w_q \times Time(x)$$

In addition, local convergence of the ATS method can be proved by using probabilistic modeling.

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, $f(x_t) \geq f(x_{t+1})$. 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 where $x \in \Psi_{t+1}$. The probability to improve the solution P=($f(x) < f(x_t)$) 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.

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

As can be seen, when the process is repeatedly performed with a considerable amount of time, the probability of finding the local minimum is close to unity.

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

Proof: During the search process, in many situations, the algorithm sometimes fails to improve the current solution x_0. The process will then use x_0 as the initial solution for the next search round. This may lead an endless loop to produce an entrapment of solutions. However, based on a random process, the next search round may offer a new search direction that leads the movement relatively close to the boundary of $\Lambda(x_0)$ and a region nearby. Because $N_\rho(x_0)$ is defined around x_0 with a certain radius ρ, it may overlay with more than one region. That is $N_\rho(x_0)$-$\Lambda(x_0) \not\subset \Lambda(x_0)$. This property makes the search process able to escape, effectively, from the entrapment of the already visited local minimum.

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 that increase the possibility to run away from the already visited local minimum. This noticeably makes a sequence of solutions found jump from one region to another.

Given n_{re} be a counter for a solution cycling. Note that "solution cycling" means the searching process 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 increased every time a new final solution of any search round being equal to the one previously visited and already stored in the list. Then, 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 searching whether it can eventually escape from the solution lock or not. Otherwise ($n_{re} > n_{re\_Max}$), performing the BT process. Once $n_{re} > n_{re\_Max}$, one of the solutions recorded in the TL is selected to be a new initial solution for creating the next sub-space Ψ. This condition, $n_{re} > n_{re\_Max}$, is a kind of Aspiration Criteria. In the ATS method, the BT mechanism will select a solution $x_h \subset$TL in such a way that $x_h = \max_{x_i \in TL} \|x_i - x_o\|$ and it notes that the condition $f(x_0) > f(x_h)$ must hold. After selecting an appropriate solution in the TL, set $x_0 = x_h$ as a new initial solution for the next search round. Therefore,

i) If the local minimum 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,

$$\text{length(TL)} >> 1 \rightarrow \exists x \in \text{TL} \wedge \|x - x_B\| < \gamma$$

where is a boundary point and 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 $\Lambda(x_0)$ with high probability.

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

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 and with all different local minima being found within finite searching time $p\times m\times Time(x)<\sum_{i=1}^{n}w_i\times Time(x)$, one of them is the global minimum.

Applications

This section presents two applications in the domains of function minimization and identification. The proposed ATS method has been applied to achieve the satisfactory solutions of the problems. During the searching process, sequences of solutions and errors are monitored numerically. The two problems are stated as follows.

Minimization of the Bohachevsky's Surface

Eq. (3) describes the Bohachevsky's function[13] having multiple local minima as depicted in Fig 1. This function possesses the global minimum being on *x=0, y=0* making $f(0,0)=0(1\times10^{-5}$ approximating zero used as the termination criterion). According to the definition 2 and the proof of theorem A, the sub regions of the Bohachevsky's surface satisfy a convex property (see the appendix).

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

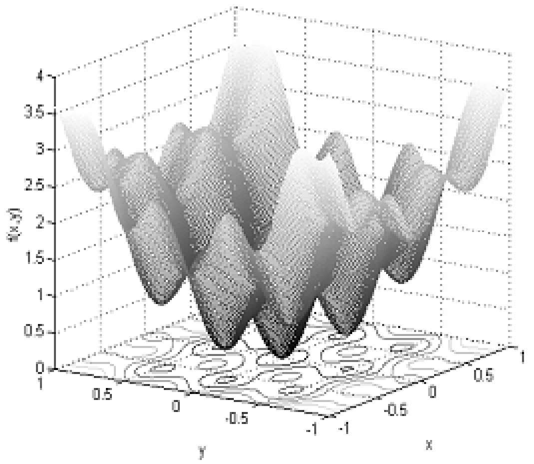

Fig1. Selected Bohachevsky's function.

The ATS was run on a Pentium 4, 1.6 GHz, 256 Mbytes RAM, 40 Gbytes HD, with MATLAB™ codes. The tests were conducted with 10,000 trials and each of which started with a random solution. Each trial stopped when either of the following stop criteria was met: i) the maximum search round of 10,000, or ii) the cost value $\leq 1\times10^{-5}$. The appropriate values of the ATS' search parameters were set as follows: ρ = 0.2 (ρ = 0.2, μ = 1.0), 10% of search space, m = 30, = 5, k^{th} = -5 (the k^{th} is a backward solution selected by the BT mechanism), and $\mu_{new}=0.2\mu_{old}$ in the AR mechanism employing the following statements: if [$\varepsilon_1\leq1.0\times10^{-1}$], then [$\rho$ = 0.04], if [$\varepsilon_2\leq1.0\times10^{-2}$], then [$\rho$ = 8.0≤10^{-3}], and if [$\varepsilon_3\leq1.0\times10^{-3}$], then [$\rho=1.6\times10^{-3}$], where ε_i are the cost values.

The ATS could track down the true minimum for all trials before the maximum search round was hit. It consumed 0.11 seconds of average search time, 26.48 average search rounds, and provided the average solution of 4.25×10^{-6}.

Identification of a Non-linear Pendulum Model

The diagram in Fig 2 represents a pendulum on a cart system. The pendulum swings freely when the cart moves according to the applied force, *f*. This force is transmitted from the motor through the belt. The angle f indicates the oscillation of the pendulum. For some large angles of oscillation, the equations of motion describing the pendulum and the cart dynamics can be expressed by Eqs. (4) and (5)[14]. Note that, in Eqs. (4) and (5), *y* = cart position, *M* = cart mass (unknown), *m* = pendulum mass = 0.251 kg., *l* = length of pendulum rod = 0.4 m, and g = acceleration due to the gravity = 9.81 m/sec^2. The rod is assumed to be weightless. The term *f* in both equations represents an external force applied to the cart. The force is transmitted from the motor axle through a flexible belt. Therefore, *f* is non-uniform. In order to model the system accurately, a seventh order polynomial representation of the force *f* in terms of the motor input *u* is assumed.

$$\ddot{\phi}=\frac{f\cos(\phi)+0.5ml\sin(2\phi)(\dot{\phi})^2+(M+m)g\sin(\phi)}{l\left[m\cos^2(\phi)-(M+m)\right]} \tag{4}$$

$$\ddot{y}=\frac{f+0.5mg\sin(2\phi)+ml\sin(\phi)(\dot{\phi})^2}{\left[(M+m)-m\cos^2(\phi)\right]} \tag{5}$$

To obtain coefficients of the force expression and the mass *M*, the ATS method is employed. The stop criteria for the search are the cost $J\leq1.32$, where J is the sum-squared error of the estimated data compared to the measured data, and the maximum search round of 10,000 rounds. The sum-squared error satisfies a convex property (see the appendix).

The ATS was run on a Pentium 4, 1.6 GHz, 256 Mbytes RAM, 40 Gbytes HD, with C codes. The tests were conducted with 10,000 trials with random initial

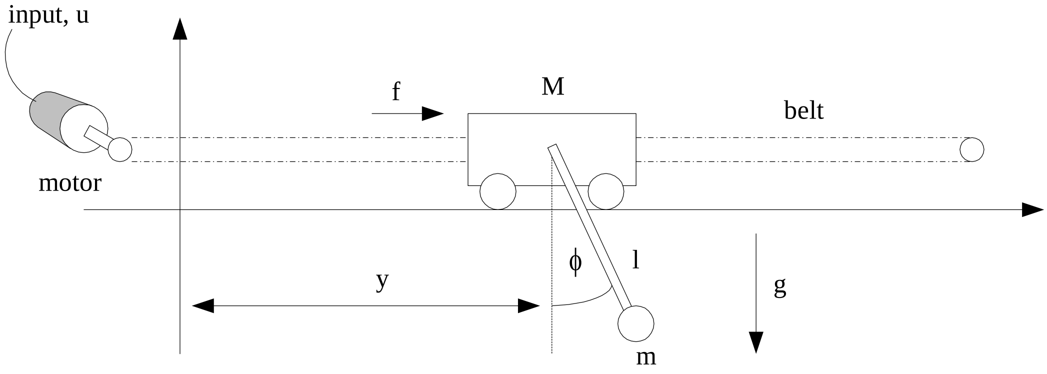

Fig 2. The pendulum on a cart system.

solutions. The method provided the solutions with average J = 1.3188, average search rounds of 841.55, and consumed 83.72 seconds of average search time. The obtained coefficients of the force expression and the mass, M, are shown in Eq. (6) and (7), respectively.

$$f = 4.1152u^7 + 6.7594u^6 - 1.8981u^5 - 5.3380u^4 - 4.9895u^3 - 2.3534u^2 + 2.4902u - 0.1385 \quad (6)$$

$$M = 0.9005 \text{ kg} \quad (7)$$

Fig 3 illustrates the plots of the measured data and the calculated data based on the model. The results show that the model provides a satisfactory approximation of the pendulum's oscillation.

Conclusions

We have illustrated the proof of a finite convergence property of the ATS algorithms. The back-tracking and adaptive radius mechanisms significantly enhance the global optimum finding, and the fast convergence of the search. The effectiveness of the ATS method has been well demonstrated by two examples: (i) minimum finding on the Bohachevsky's surface, and (ii) identification of the nonlinear pendulum model.

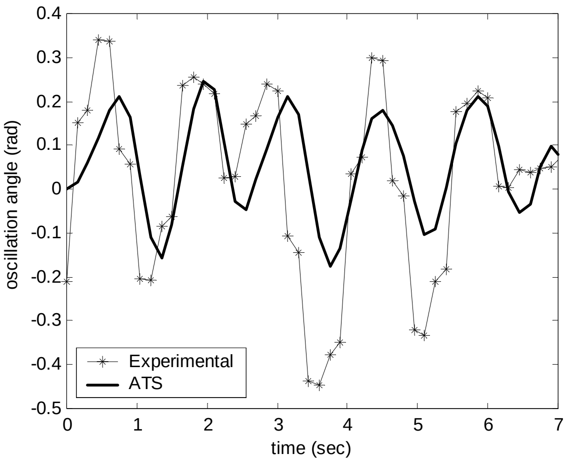

Fig 3. The model plotted against the measured angle.

Acknowledgement

The authors' thanks are due to the reviewers for their constructive comments.

Nomenclature

AC	Aspiration criteria
AR	Adaptive search radius mechanism
ATS	Adaptive Tabu search
BT	Back-tracking mechanism
Count	Search round
f	Function, force (N)
g	Gravity (m/sec^2)
g	Error of the objective function
Iteration	Iteration
J	Preset cost used to be one of the TCs
k	Finite number of the local minima
k^{th}	Backward solution selected by BT
l	length of pendulum rod (m)
m	Finite number of solutions in Ψ, pendulum mass (kg)
M	Cart mass (kg), any positive integer
n_{re}	Number of repetition of the solution
n_{re_Max}	Allowable number of repetition of the solution
N	Total number of $N_\rho(x'_i)$ members
$N_\rho(x_0)$	Neighborhood of x_0 with radius ρ
$N_\rho(x'_i)$	Neighborhood of x'_i with radius ρ
p	Number of collection of solution movements to reach the global minimum
P	Probability
r	Common radius
$S\{x_{0,i}\}$	Finite sequence, consequence of the movement of

	x_0
TC	Termination criteria
Time(x)	Time that any solution x was visited
TL	Tabu list
TS	Conventional Tabu search
u	Some members of neighborhood , motor input (volt)
w	Total number of members in each sub region
x	Any solution
x_0	Current or initial solution
$x_{0,Count}$	Current or initial solution of the Count
$x_{0,i+1}$	Solution used for creating Ψ_{i+1}
x_B	Boundary point
x_h	Solutions recorded in the TL
x^*	Current (near) global optimum solution
$\tilde{x}$	Any random solution
$\hat{x}$	Local minimum
x'	Solution with minimum cost in current Ψ
y	Cart position (m)

Greek symbols

Λ	Sub region
$\Lambda(x_0)$	Sub region consisting of x_0
μ	Decreasing factor
ξ_i	Distance between and
$\varnothing$	Empty set
ρ	Search radius
Ω	Finite search space
Ψ	Sub search space in W
Ψ_{Count}	Sub search space created in the Count
γ	Maximum allowance
δ	Tolerant distance between the solution found and the local minimum
ε	Tolerant error of the objective function
ϕ	Oscillation angle (radian)
θ	Sequence of collection of sub-spaces

References

1. Glover F (1986) Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* **13**, 533-49.
2. Nowicki E, and Smutnicki C (1996) A fast tabu search algorithm for the flow shop problem. *European Journal of Operational Research* **91**, 160-75.
3. Glover F, Mulvey JM, and Hoyland K (1994) Solving dynamic stochastic control problems in finance using tabu search with variable scaling. *Statistics and Operations Research Technical Report SOR-94-13*, Princeton University, USA.
4. Zhang G, Habenicht W, and SpieB WEL (2003) Improving the structure of deep frozen and chilled food chain with tabu search procedure. *J of Food Engineering* **60**(1), 67-79.
5. Kulworawanichpong T and Sujitjorn S (2002) Optimal power flow using Tabu search. *IEEE Power Engineering Review* **22**(6), 37-40.
6. Cordeau J-F and Laporte G (2003) A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological* **37**(6), 579-94.
7. Areerak K-N (2001) Nonlinear identification and stability test for a two-inertia system with mechanical resonance. *Master Thesis*: Suranaree University of Technology, Thailand.
8. Glover F and Laguna M (1997) *Tabu Search*. Kluwer Academic Publishers. Norwell.
9. Glover F (1994) Tabu search for nonlinear and parametric optimization (with links to genetic algorithms). *Discrete Applied Mathematics* **49**, 231-55.
10. Puangdownreong D, Areerak K-N, Srikaew A, Sujitjorn S, and Totarong P (2002) System Identification via Adaptive Tabu Search. *Proc IEEE Int Conf on Industrial Technology (ICIT'02)* **2**, 915-20.
11. Glover F and Hanafi S (2001) Finite Convergence of Tabu Search. *Proc. MIC'2001-4th Metaheuristics Int Conf*, 333-6.
12. Hanafi S (2000) On the Convergence of Tabu search. *J Heuristics* **7**, 47-58.
13. Bohachevsky IO, Johnson ME, and Stein ML (1986) Generalized simulated annealing for function optimization. *Technometrics* **28**(3), 209-18.
14. Bloch AM, Leonard NE, and Marsden JE (2000) Controlled Lagrangians and the stabilization of mechanical systems I: the first matching theorem. *IEEE Trans Automatic Control* **45**(12), 2253-70.
15. Kreyszig E (1978) *Introductory Functional Analysis with Applications*. John Wiley & Sons.
16. Chong EKP and Zak SH (2001) *An Introduction of Optimization*. Wiley Interscience.

Appendix

Convex properties of the Bohachevsky's function and the sum-squared error (J) are proved in this section. The definition of convex function[15] is as follows: A function $f:\Re^n \to \Re$ is said to be *convex* if its domain $D(f)$ is a convex set and for all u, $v \in D(f)$, $f(\lambda u+(1-\lambda)v) \le \lambda f(u)+(1-\lambda)f(v)$, where 0 ≤ l ≤ 1.

The definition of convex set[16] is as follows: a set Q Ì $\Re^n$ is *convex* if for all u, $v \in$ Q, the line segment between u and v is in Q. Note that Q is convex if and only if $\alpha u+(1-\alpha)v \in \Theta$ for all u, $v \in$ Q and a ∈ (0, 1). Examples of well-known convex set include: the empty set, a set consisting of a single point, a line or line segment, a subspace, a hyperplane, a half-space, and .

Convex Property of the Bohachevsky's Function

From Eq. (3), the Bohachevsky's function can be decomposed into three parts, i.e. 1) $f_1(x)$ =, 2) f_2(y) =, and 3) f_3() = 0.7, where $x = y \in$ [-1.0, 1.0].

$f_1(x)$ consists of two parts, i.e. which is exactly convex, and –0.3cos(3px) which needs to be proved. Expression , = –0.3cos(3px), is convex if and only if ≤ 0. This means that is convex when: . From the convex properties[16], the intersection of any collection of convex sets is convex. So, we can conclude that some regions of $f_1(x)$ are convex.

Like $f_1(x)$, $f_2(y)$ consists of two parts, i.e. which is exactly convex, and –0.4cos(4py) which needs to be proved as well. Expression , = –0.4cos(4py), is convex if and only if ≤ 0. This means that is convex when: . We can conclude that some spaces of $f_2(y)$ are convex. $f_3()$, a line segment, is also convex.

From the intersection property of the convex set, we can conclude that some regions or spaces of the Bohachevsky's function satisfy convex property.

Convex Property of the Sum-Squared Error

The sum-squared error (SSE), J, used in our applications is the SSE of the estimated data compared to the measured data. J can be expressed by , where is the estimated data, and is the measured ones. is obtained from of the models having nine unknown parameters (see Eq. (4)-(7)). Because of the complexity of the models, verification of J in the context of convex property may not be proved in strictly mathematical style. However, we can consider the behavior of the search process. During the search process, the intersections of convex regions of nine unknown parameters occur. This corresponds to the cost value J being minimized gradually. From this behavior, we can intuitively conclude that the SSE satisfies convex property in some ranges of parameters.