# Adaptive Tabu Search and Management Agent

**Sarawut Sujitjorn**[1], **Jukkrit Kluabwang**[2],
**Deacha Puangdownreong**[3], and **Nuapett Sarasiri**[4], Non-members

**ABSTRACT**

This paper elaborates the details of the adaptive tabu search (ATS) and the management agent (MA). It starts with briefing about local search, and evolutionary algorithm. The generic tabu search is also discussed in brief. A detailed explanation of the ATS with its recommendations for use is given. The MA that is general enough to be used with any kinds of search methods is elaborated. The paper discusses the way the MA organizes its search units, and presents the search performance assessment. The assessment employs symmetrical, and asymmetrical problems based on 3D surface optimization. Applications on control design optimization of a scaled vehicle are presented.

**Keywords**: local search, evolutionary algorithm, adaptive tabu search, management agent.

## 1. INTRODUCTION

Local search has been known and successfully applied for many years. Search methods belonging to this family include simulated annealing [1], threshold accepting [2], tabu search [3], genetic algorithms [4], memetic algorithms [5], scatter search [6], etc. These search algorithms are sometimes referred to as "population based" or "evolutionary" algorithms because they share similarities in that (i) they are stochastic iterative procedures, (ii) they create a pool of solutions and select those possible solutions through "selection process", and (iii) they maintain their existence and evolve through "reproduction" and "replacement" processes [7, 8]. Generally, evolutionary algorithm (EA) consists of three main steps as follows:

Step0: Initialization
Initialize solution pool and search parameters.

Step1: Iteration
Determine solution quality, selection, reproduction, replacement, and update counter.

Step2: Termination
If stop criterion is met, stop search and deliver solution, otherwise go to Step1.

Despite the resemblance of the algorithms, tabu search (TS) can be regarded as an intelligent search method because it utilizes adaptive memory, and opens for users designed search criterion or sub-algorithms. TS has been applied successfully for many optimization methods; among those are power systems [9, 10], transportation [11], flow shop [12], food processing [13], etc. However, the simplistic TS cannot completely handle deadlock problems caused by local minima. Some researchers have proposed modifications to the generic TS to overcome such shortcomings. These include adaptive TS (ATS) [14], probabilistic TS [15], and reactive TS [16]. An initial performance investigation among the genetic algorithm, TS, and ATS was reported in 2002 by applying a surface optimization problem [17]. This previous work performed searches on the Bohachevskys surface in two categories utilizing random, and deterministic initial solutions. Termination criterion was set as 1.5×10-8, 15×10-101, and 0.0, respectively. The ATS [14] outperformed the GA, and the TS in all cases. Till now, the authors of this paper and their coworkers have been applying the ATS for various engineering and optimization problems. These include signal and system identification [18-20], control engineering [21-23], signal processing [24], etc. Some advanced works have been further developed to enhance the performance of the ATS. This is the subject of presentation of this paper.

This paper is organized into 5 sections. Following the introduction, section 2 explains the ATS algorithms with recommendations for use. Parallelization in an upper level for the ATS has been developed, and named management agent (MA). Section 3 elaborates this. Control design applications are presented in section 4 to demonstrate the powerfulness of the MA and the ATS to handle a multi-parameter search situation. Section 5 provides conclusion.

## 2. ADAPTIVE TABU SEARCH (ATS)

ATS is a meta-heuristic method that has been modified from the generic TS. The ATS extensively uses adaptive memory, adaptive neighborhood

Manuscript received on December 15, 2009 ; revised on , .
[1,2,4] The authors are with the School of Electrical Engineering, Suranaree University of Technology, Nakhon Ratchasima, Thailand., E-mail: sarawut@sut.ac.th
[3] The author is with the School of Electrical Engineering, South-East Asia University, Bangkok, Thailand., E-mail:
[1] Corresponding author: sarawut@sut.ac.th

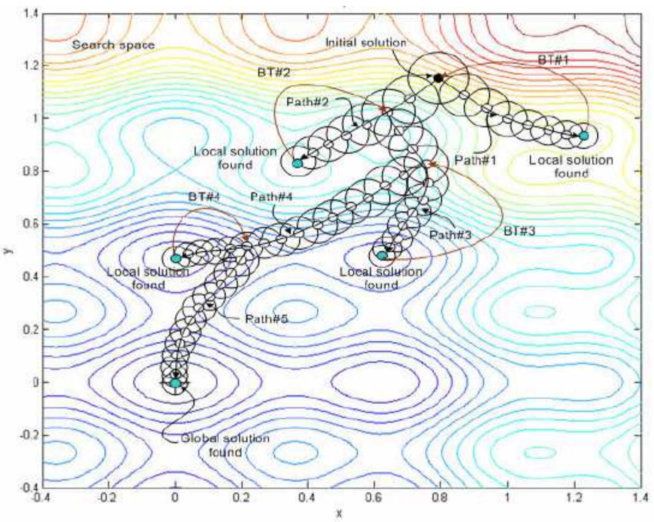



***Fig.1:*** *ATS search moves [25].*

area, diversification, and intensification strategies. Two main mechanisms incorporated into the TS are namely "backtracking" (BT), and "adaptive search radius" (AR). The generic TS uses an adaptive memory so called tabu list (TL) to store previous elite solutions obtained from search moves. Such solutions are prohibited from use in common cases. In contrast, those solutions are particularly helpful when kicking the search off a deadlock caused by some local optima is needed. The BT mechanism helps the search to override the TL such that a new search direction can be formed. The general term for this type of mechanisms is diversification strategy. Regarded as an intensification strategy, the AR helps to focus the search on an extremely high-quality solution with a high chance to be the global solution for the problem. Focusing the search can be done by reducing

Fig. 1. illustrates some search moves of the ATS on a plateau containing many local optima. The ATS may encounter these local solutions many times. Each movement towards a local solution, the AR is invoked to track a possible high-quality solution. This can be noticed from the illustration that the radii of the circular search areas are decreased gradually. Once the ATS realizes upon examining the termination criterion that the current local solution cannot be accepted as the global solution, the BT is invoked. By the BT, one of the previous elite solutions in the TL is selected for being an initial solution for a new search. This is represented by the back-jumping arrows in the picture. A new search direction is then formed on the basis of this BT selected initial solution, and randomly generated possible solutions in the neighborhood around it. Noticeably, the BT and the AR alternately operate until the ATS hits the global solution. This search regime has been termed "strategic oscillation" [3]. While the flow diagram in Fig. 2. summarizes the ATS, its step-by-step algorithms are prescribed as follows:

Step0: Initialize search space, search parameters, count and count max.

Step1: Randomly (or heuristically) select an initial solution $S_0$ from the search space. Set $S_0$ as a current solution.

Step2: Randomly generate a neighborhood (with its initial radius $R$) of $N$ solutions around $S_0$. Set N solutions as the members of the set $X$.

Step3: Evaluate the cost value (J) of each member in $X$ according to a preset objective function. Define $S_1$ as the member with the minimum cost, $J_1$.

Step4: If $J_1 \geq J_0$, store $S_1$ in the TL and $S_0 = S_0$. If $J_1 < J_0$, store $S_0$ in the TL and $S_0 = S_1$.

Step5: Activate the BT when the solution deadlock occurs.

Step6: Exit with the global solution if the termination criterion is met.

Step7: Activate the AR when the current solution $S_0$ is relatively close to a local minimum.

Step8: Update iteration counter (count), and go to Step 2.

The ATS needs a criterion to identify that it is moving towards a local optimum such that the AR is invoked at an appropriate instant. The users of the algorithms may implement probability measure, possibility measure, distance measure, etc. One simple approach is to use a threshold of the cost value. As an example, let $\varepsilon$= cost of the current solution, $\Delta_i$ = preset cost for $i = 1, 2, \ldots, n$, $R$ = search radius, and $\delta_i$= reduction factor for $i = 1, 2, \ldots, n$. The AR can be simply implemented as "if ($\varepsilon \leq \Delta_i$) then ($R\delta_i \cdot R$)". By doing so, the neighborhood area is gradually decreased. This approach eventually leads

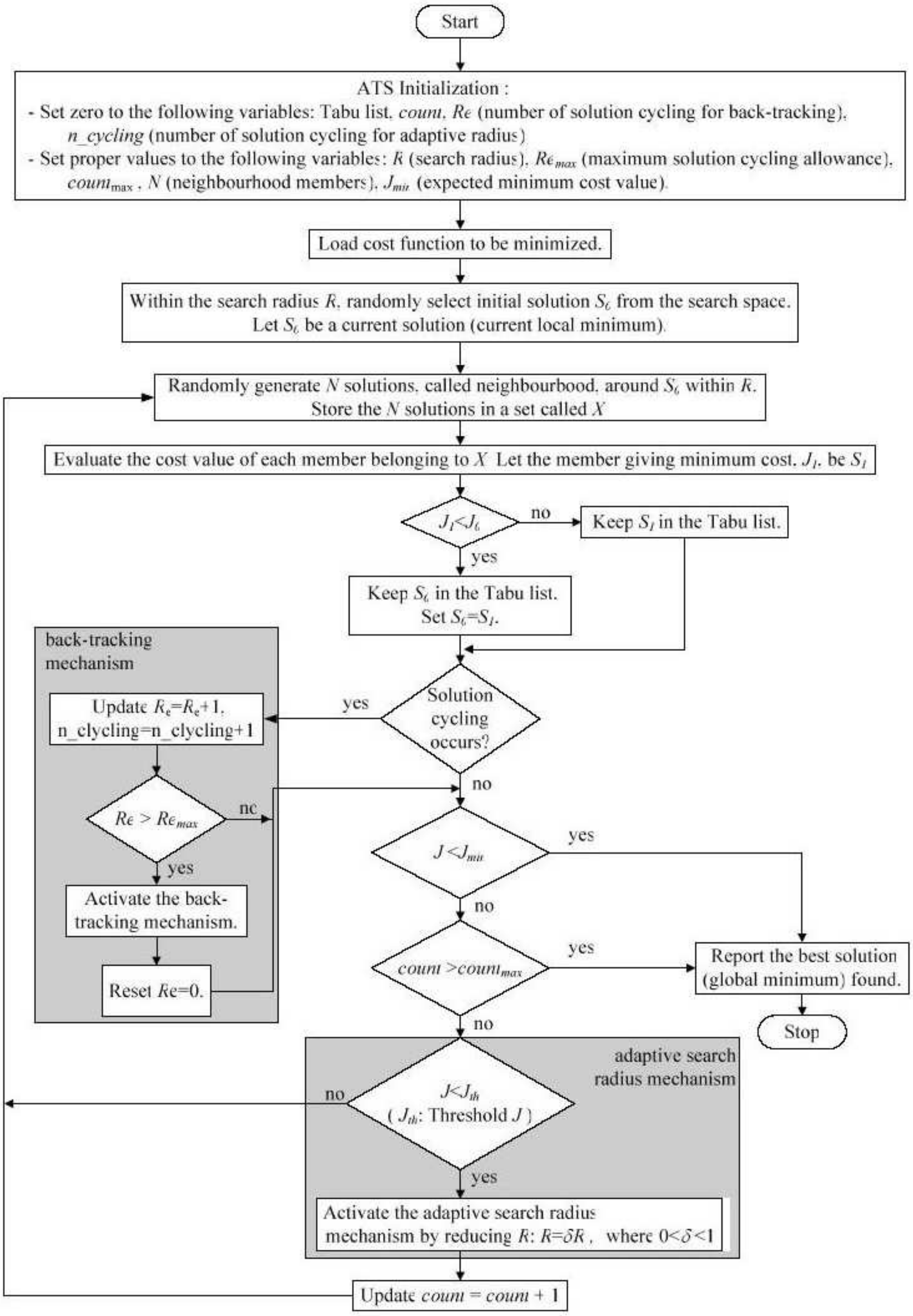



***Fig. 2:*** *ATS flow diagram [25].*

to the global solution of high quality within a finite time. For proof, see [14]. To invoke the BT at the right instant, the ATS needs a criterion to identify a deadlock. One simple technique is to use a frequency counter for the case that solutions of the same quality repeat themselves, and the algorithms cannot improve the situation. Under this circumstance, the BT looks up the TL, and selects one solution being stored to use as a new initial solution for the generation of the next search move. At the moment, the search is successfully kicked off the deadlock valley, and a new search direction begins. By doing so repeatedly, diversification of the search is guaranteed; the ATS can always escape from the deadlock caused by local optima; eventually the global solution can be tracked down. For proof, see [14].

There are some search parameters to be initialized. These include search radius, $N$ members of a neighborhood, search radius reduction factor, deadlock counter, $k^{th}$ backward jump for the BT to select a previous solution from the TL, and maximum iteration counter. To start with the ATS, the following matters are recommended for an efficient search: (a) use a guided initial solution, (b) the initial search radius be 7-15% of the radius of the entire search space, (c) use N between 30-40, (d) allow 5-15 times of repetition of current solution before invoking the BT, and use $k^{th}$ backward jump close to this figure, (e) use between 20-25%, and (f) conduct some initial searches to figure out a suitable termination criterion and any difficulties before attempting a real search for the problem of interest.

## 3. MANAGEMENT AGENT (MA)

Upper level parallel algorithms have been developed for search methods in general. The algorithms are referred to as "management agent" (MA). The MA provides an organizing strategy to a particular search method such that many search units can operate in either parallel or sequential manner according to the computing hardware. The search method employed as search units is not limited to the ATS. One advantage of using the MA is that there is no need for any modifications to be made to the search units. The MA consists of three mechanisms namely partitioning, sequencing, and discarding mechanisms (PM, SM, DM), respectively.

The PM serves to divide an entire search space into several to many sub-search-spaces, bearing in mind that too many sub-search-spaces would result in a slow search process. Fig. 3. shows an example of a symmetrical partitioning made to the Bohachevsky's surface. One may employ asymmetrical partitioning.

The SM is a time-sharing tactic to organize the search units to run one-by-one on a single iteration approach providing the hardware has a single CPU. The process is interrupted by the DM for every m times repetition for some information transfer between the two mechanisms. With a multiple CPU platform, the SM serves to distribute the search units among the available CPUs.

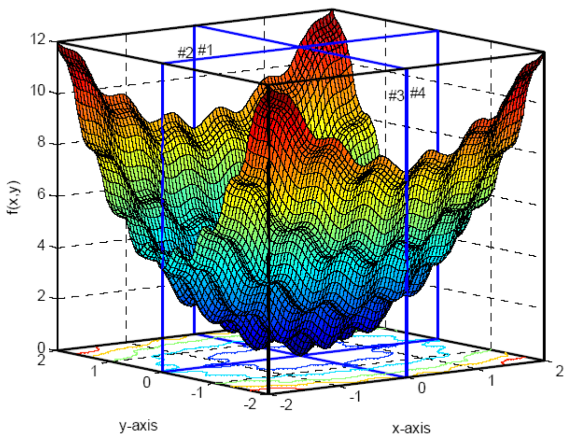



***Fig.3:*** *Partitioning of the Bohachevskys surface.*

The DM serves to discard some unlikely to be successful search paths. Upon its interruption issued to the SM, a bunch of information concerning the actual values of the current solutions with their quality indices is supplied to the DM. Using such information decisively, the DM appropriately discards some certain search paths. The discarding process occurs repeatedly until only one search path is left to continue searching, and eventually hit the global solution. One may employ probability measure, possibility measure, distance measure, etc., as a discarding criterion.

*Fig. 4.* illustrates the flow diagram of the MA in which the ATS is used as search units to be executed on a single CPU platform. In stead of a sequential execution, the ATS search units can be distributed to run parallelly on a multi-core platform by the SM. The steps of the MA algorithms are summarized as follows:

Step0: Identify search spaces, define number of search units, initializing search-unit algorithms (ATS).
Step1: Invoke PM, situate ATSs to sub-search-spaces, and remove partitions.
Step2: Invoke SM and execute $ATS\#1, ATS\#2, \ldots, ATS\#n$ for $n = n - k, k \leq n - 1, n_{min} = 1$.
Step3: Exit with the global solution if the termination criterion is met.
Step4: Invoke DM.
Step5: Update counters. Go to Step 2.

It can be noticed that during Step 2 the ATSs as search-unit algorithms work independently without any modifications or intrusion made by the MA. Hence, the convergent property of the ATSs is always preserved.

Performance of the MA having the ATSs as its search units, from now on will be called shortly as MA(ATS), has been assessed using some surface optimization problems. The following well-known surfaces containing one global and many local optima are employed: Bohachevsky's, Rastrigin's and Shekel's foxholes functions (BF, RF, SF), respectively. These are depicted in Fig. 5. Their algebraic expressions are given in equations (1)-(3), respectively.

$$f(x,y) = x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(0.4\pi y) + 0.7 \quad (1)$$

$$f(x,y) = x^2 + 2y^2 - 10\cos(2\pi x) - 10\cos(2\pi y) + 20 \quad (2)$$

$$f(x_1,x_2) = \left| \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6} \right|^{-1} \quad (3)$$

when

$$a_{ij} = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 \ldots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 \ldots & 32 & 32 & 32 \end{pmatrix}$$

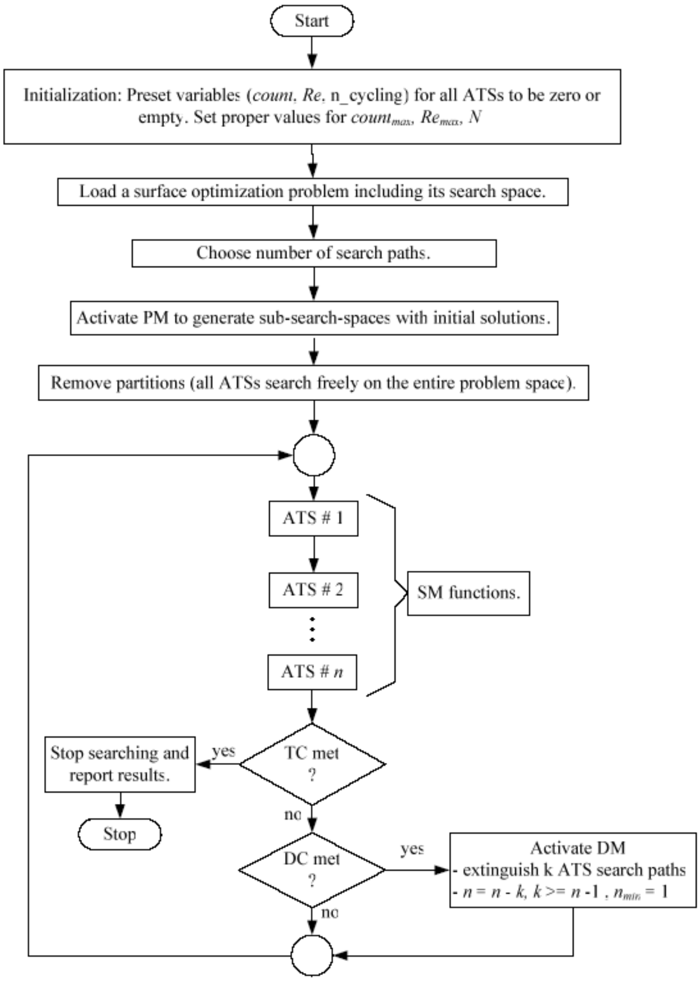



***Fig.4:*** *MA flow diagram.*

The BF and RF have their global minima of zeroes located at (0,0). The global minimum of the SF is 1 located at (-32,-32). Some tests, which are arranged to have these locations of the global minima situated at the centers of the corresponding search spaces, are regarded as symmetrical problems. For asymmetrical problems, such locations are eccentric. Search problems are arranged in this manner in order to study the effects of (a) symmetrical partitioning. Table 1 shows the ATS parameters. The PM of the MA is assigned to divide the entire search space into 2, 4, 8, 16, 32, and 64 sub search- spaces. Correspondingly, the MA(ATS) is termed $MA\#2, MA\#4, \ldots, MA\#64$ understood that $MA\#64$ has 64 paths of the ATS in operation, for instance. Each time the DM is activated, half of the existing search paths are forced to stop according to the discarding criterion. Table 2 summarizes the $i^{th}$ iteration at which the DM is invoked.

The MA(ATS) was coded in MATLABTM, and run on a Pentium 4, 2.4 GHz, 640 Mbytes SD-RAM. Each set of numeric figures disclosed as the results in Table 3 is an average over 50 runs. Based on these results, we can obtain "speed up ratios" expressed as

$$speed\ pu\ ratios = \frac{averagesearchtimebyATS}{averagesearchtimebyMA(ATS)}. \quad (4)$$

The speed up ratios shown in *Fig.6.* indicate that the MA(ATS) is 1.01-3.16 times faster than the ATS under symmetrical problems, and 1.08-3.54 times faster under asymmetrical problems. Hence, the search performance of the MA(ATS) is not affected much by the symmetrical property of the problems. Furthermore, the greatest number of search paths does not always produce the fastest search. Users need to figure out a suitable number of search paths that would produce the fastest search for each particular problem.

## 4. APPLICATIONS

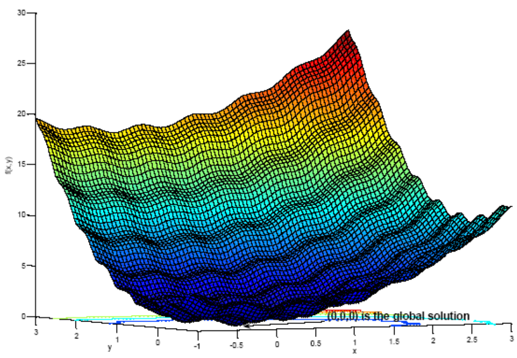
The control problem presented herein requires an optimally seeked controller for the yaw rate control of a scaled vehicle shown in Figs. 7(a) and (b). The polynomial models, G and Gd, represent the vehicle dynamics. The original controllers (Gdc, Gff, and Gfb) are designed using the model reference control

***Table 1:*** *ATS parameters.*

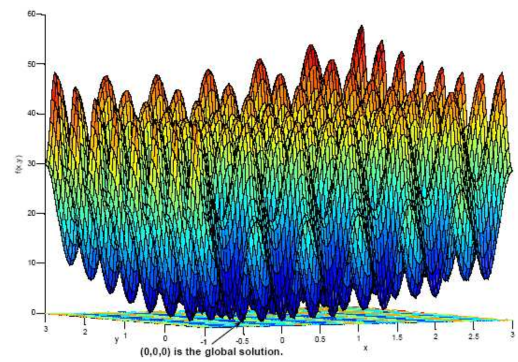
| Test functions | ATS parameters | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | radius ($R$) | no. of neighbours | BT | TC | | AR | | search-spaces |
| | | | $Re_{max}$ | $count_{max}$ | $J_{min}$ | stage I | stage II | |
| BF | 0.2 (5%) | 30 | 5 | 10,000 | 1x10$^{-9}$ | $J$<1x10$^{-1}$, $R$=2x10$^{-3}$ | $J$<1x10$^{-3}$, $R$=2x10$^{-4}$ | Up to PM |
| RF | 0.2 (5%) | 30 | 5 | 10,000 | 1x10$^{-8}$ | | | |
| SF | 0.8 (1%) | 30 | 5 | 10,000 | 0.999 | $J$<5, $R$=0.5 | $J$<2, $R$=0.1 | |

***Table 2:*** *MA parameters in terms of DM arguments.*

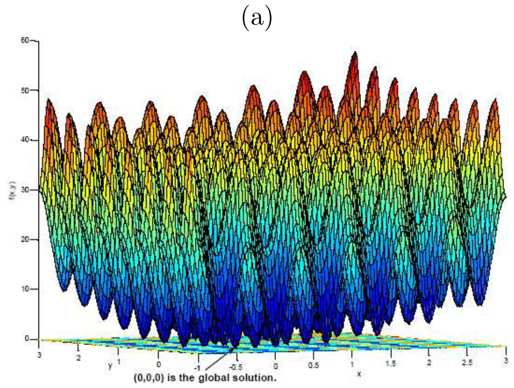
| Test functions | Iteration$^{th}$ at which DM is invoked. | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MA#2 | MA#4 | | MA#8 | | | MA#16 | | | | MA#32 | | | | | MA#64 | | | | | |
| | 1$^{st}$ stage | 1$^{st}$ stage | 2$^{nd}$ stage | 1$^{st}$ stage | 2$^{nd}$ stage | 3$^{rd}$ stage | 1$^{st}$ stage | 2$^{nd}$ stage | 3$^{rd}$ stage | 4$^{th}$ stage | 1$^{st}$ stage | 2$^{nd}$ stage | 3$^{rd}$ stage | 4$^{th}$ stage | 5$^{th}$ stage | 1$^{st}$ stage | 2$^{nd}$ stage | 3$^{rd}$ stage | 4$^{th}$ stage | 5$^{th}$ stage | 6$^{th}$ stage |
| BF | 250 | 20 | 200 | 15 | 25 | 100 | 5 | 10 | 15 | 20 | 5 | 10 | 15 | 20 | 25 | 1 | 3 | 5 | 7 | 9 | 15 |
| RF | 500 | 15 | 25 | 15 | 25 | 35 | 15 | 25 | 35 | 45 | 15 | 25 | 35 | 45 | 55 | 5 | 10 | 15 | 20 | 25 | 30 |
| SF | 50 | 20 | 40 | 10 | 20 | 50 | 5 | 10 | 15 | 20 | 5 | 10 | 15 | 20 | 25 | 1 | 3 | 5 | 7 | 9 | 15 |




(a)




(b)




(c)

***Fig.5:*** *3D problems for performance assessment (a) Bohachevskys surface (b) Rastrigins surface (c) Shekels Foxholes surface.*

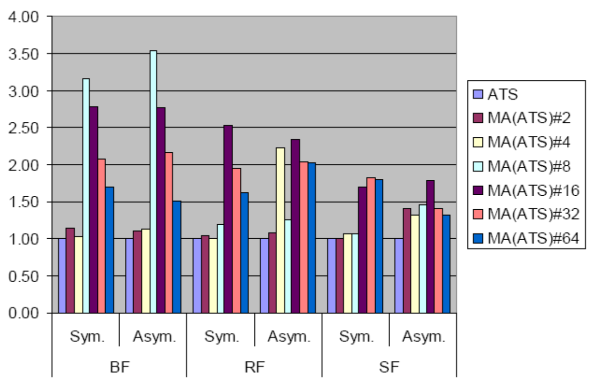



***Fig.6:*** *Speed up ratios for symmetrical and asymmetrical problems.*

*Table 3: Average search time.*

| Test functions | | average search time (seconds) | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | ATS | MA(ATS) | | | | | |
| | | | #2 | #4 | #8 | #16 | #32 | #64 |
| BF | Sym. | 12.8996 | 11.2039 | 12.5100 | 4.0728 | 4.6429 | 6.1919 | 7.6006 |
| | Asym. | 13.8378 | 12.5574 | 12.2330 | 3.9040 | 4.9901 | 6.3966 | 9.1036 |
| RF | Sym. | 14.7950 | 14.2214 | 14.5549 | 12.4603 | 5.8682 | 7.5916 | 9.1136 |
| | Asym. | 19.1713 | 17.6766 | 8.5792 | 15.2682 | 8.1967 | 9.4051 | 9.4274 |
| SF | Sym. | 3.9596 | 3.9266 | 3.7271 | 3.6978 | 2.3271 | 2.1720 | 2.1988 |
| | Asym. | 4.3564 | 3.0986 | 3.2957 | 2.9820 | 2.4380 | 3.1135 | 3.2915 |

approach to achieve a desired pole-placement [26]. The vehicle models, and the controllers are summarized by Table 4. The transfer function $G_{irs} = \left.\frac{Y(s)}{R(s)}\right|_{V(s)=0}$ has its order as high as 27. So, its detail is omitted. The step response of the original control system is shown in Fig. 8. A smooth response with approximately 5 ms delay time can be observed. The response has rise-time of 0.231 s, settling time of 0.552 s, and steady-state value of 8.663 rad/s.

We have applied the MA(ATS) to seek for new controllers to improve the response. One approach is to seek 3 new controllers. Another is to adopt the original controller with an additional PID-controller in cascaded connection to the Gff.

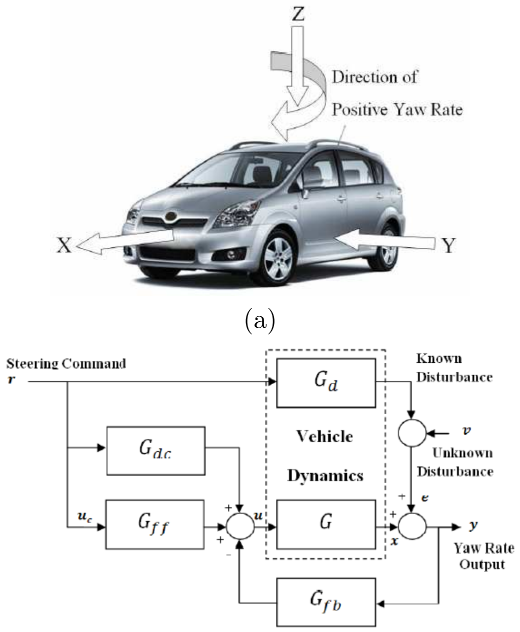



***Fig.7:*** *Scaled vehicle (a) reference frame (b) block diagram.*

## A. Seeking new controllers

The control problem assumes the original vehicle models with fixed parameters. The new controllers are designated as $G^*_{dc}, G^*_{ff}$ and $G^*_{fb}$ expressed as the following polynomial ratios in equations (5.a)-(5.c),

*Table 4: Models and original controllers [26].*

| | |
|---|---|
| forward dynamic $G$ | $G = \frac{2.802\times10^4 s + 2.306\times10^5}{1.5s^4 + 94.43s^3 + 7.308\times10^4 s^2 + 1.183\times10^5 s + 5.361\times10^5}$ |
| disturbance model $G_d$ | $G_d = \frac{8.024\times10^4 s + 6.710\times10^5}{1.3s^4 + 53.250s^3 + 1.943\times10^3 s^2 + 2.693\times10^4 s + 1.162\times10^5}$ |
| $G_{dc}$ | $G_{dc} = \frac{-1.2036\times10^5 s^5 - 8.584\times10^6 s^4 - 6.497\times10^8 s^3 -}{3.6426\times10^4 s^5 + 1.792\times10^6 s^4 + 6.671\times10^7 s^3 +}\cdots$ $\cdots\frac{-1.44\times10^{10} s^2 - 1.224\times10^{11} s - 3.598\times10^{11}}{+1.202\times10^9 s^2 + 9.464\times10^9 s + 2.679\times10^{10}}$ |
| $G_{ff}$ | $G_{ff} = \frac{103s^2 + 1.8542\times10^4 s + 8.34385\times10^5}{s^3 + 155.3s^2 + 3.727\times10^3 s + 2.0714\times10^4}$ |
| $G_{fb}$ | $G_{fb} = \frac{-5.181s^3 + 49.98s^2 + 5.294\times10^3 s + 4.8142\times10^4}{s^3 + 155.3s^2 + 3.727\times10^3 s + 2.0714\times10^4}$ |

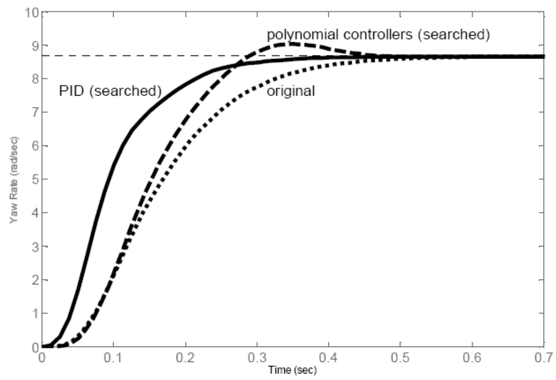



***Fig.8:*** *Responses of the controlled vehicle.*

respectively.

$$G^*_{dc} = \frac{b_1s^5 + b_2s^4 + b_3s^3 + b_4s^2 + b_5s + b_6}{a_1s^5 + a_2s^4 + a_3a^3 + a_4a^2 + a_5s + a_6} \quad (5.a)$$

$$G^*_{ff} = \frac{p_1s^2 + p_2^s + p_3}{r_1s^3 + r_2s^2 + r_3s + r_4} \quad (5.b)$$

$$G^*_{jb} = \frac{q_1s^3 + q_2^{s^2} + q_3s + q_4}{r_1s^3 + r_2s^2 + r_3s + r_4}. \quad (5.c)$$

The controller seeking problem is formulated as a combinatorial optimization problem to minimize the objective $J$ of the form

$$J = t_r + M_p + t_s \quad (6)$$

where $t_r$ = rise-time, $M_p$ = overshoot and $t_s$ = settling time of the step response. $t_r$, $M_p$ and $t_s$ are normalized to become unitless. A magnifying factor of 1,000 is also applied according to the penalty concept such that a fine quality solution could be obtained. The performance specifications are that $t_r < 0.231s$,

$M_p$ ¡ 5%, and the original DC-gain must be maintained.

It can be observed from equations (5.a)-(5.c) that 23 parameters are involved. Searching all these parameters is possible, but time-consuming. Some initial searches were thus established to identify a set of influential parameters. A mild termination criterion of $J < 90$ or $count_{max} = 20$ was used. Each parameter was searched one-by-one, while the others assumed their original values. As a result, two sets of parameters are identified. The first set consists of $r4, q2, q4, p3, a2, a3, a5, a6, b1$ and $b6$. The parameters belonging to this set are very stability sensitive. So, they assume the original values during the actual search process. The second set comprises $r1, r2, r3, q1, q3, p1, p2, a1, a4, b2, b3, b4$ and $b5$. These parameters are influential to the system performance, and robust to stability. As a further result from the initial searches, the search spaces for these parameters are established as follows:

parameters: [max min]
r1: [1.0 0.1]
r2: [155.3 0.1]
r3: [3.73 103 -3.73 103]
q1: [-5.18 -10.18]
q3: [5.29 103 -5.29 103]
p1: [103.0 -103.0]
p2: [1.85 104 -1.85 104]
a1: [3.64 104 0]
a4: [1.20 109 -1.20 109]
b2: [-8.58 106 -16.58 106]
b3: [-6.50 108 -7.50 108]
b4: [-1.44 1010 -2.44 1010]
b5: [-1.22 1011 -2.22 1011].

The MA(ATS), hence, consists of 13 ATS-paths each of which employs its own initial solution. However, they share the same set of search parameters, i.e. $N\_neighbour = 5, R = 0.1, count_{max} = 100, J < 70$,
BT: $N\_re\_max = 5$, $kth\_backword = 5$,
AR: $J < 90 \Rightarrow R = 0.5R, J < 80 \Rightarrow R = 0.1R$, and
DM: at $5^{th}$ iteration, reduce search paths to 7,
at $10^{th}$ iteration, reduce search paths to 3,
at $15^{th}$ iteration, reduce search paths to 1, respectively. The search was completed in 71.53 s on a Pentium 4 with $J = 69.59$. The solutions were provided by the $11^{th}$ path. The obtained controllers are expressed as

$$G^*_{dc}=\frac{-1.203\times10^5s^5-8.875\times10^6s^4-6.516\times10^8s^3-}{3.333\times10^4s^5+1.792\times10^6s^4+6.671\times10^7s^3+}\cdots$$
$$\cdots\frac{1.451\times10^{10}s^2-1.308\times10^6s^4-3.598\times10^{11}}{+1.077\times10^9s^2+9.464\times10^9s+2.679\times10^{10}} \quad (6.a)$$

$$G^*_{fb}=\frac{87.922s^2+1.854\times10^4s+8.344\times10^5}{0955s^3+153.194s^2+2.541\times10^3s+2.071\times10^4}. \quad (6.b)$$

$$G^*_{fb}=\frac{-5.391s^1+49.98s^2+4.174\times10^3s+4.814\times^4}{0955s^3+153.194s^2+2.541\times10^3s+2.071\times10^4}. \quad (6.c)$$

**B. Cascaded PID**

A PID-controller is considered due to its anticipative property. As the energy in the main control flows in the forward direction, a PID-controller is inserted in cascade connection with the original controller Gff. The present control problem becomes simpler than the previous topic since there are only three controller parameters to be searched, while the original controllers exist and remain unchanged.

The search problem assigned to the MA(ATS) has 3 ATS-paths to search for the real values of $K_P, K_I$ and $K_D$ . The controller is assumed to have a theoretical transfer function of

$$G^*_c = (K_D s^2 + K_p s + K_1)/s. \quad (7)$$

Each of the parameters has the [max, min] range of [2,0]. The search parameters are as follows: $N\_neighbour = 5$, $R = 0.2$, $count_{max} = 1,000$, $J = 65$,
BT: $N\_re\_max = 5$, $k_b^{th}ackword = 5$,
AR: $J < 90 \Rightarrow R = 0.5R$, $J < 70 \Rightarrow R = 0.5R$, and
DM: at 10th iteration, reduce search paths to 2,
at $50^{th}$ iteration, reduce search paths to 1, respectively.

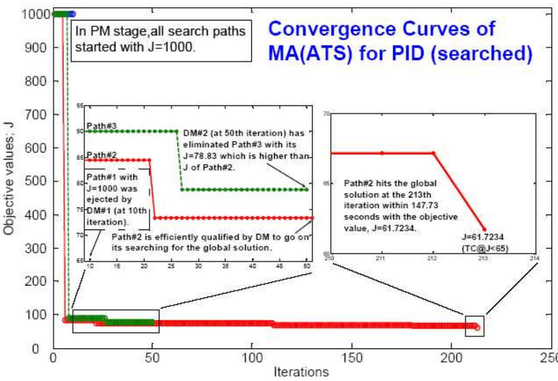



***Fig.9:*** *Convergence curves*

In addition, the performance specifications, and the formulation of the objective function are similar to those described earlier. The search stopped with $J = 61.72$ at $213^{th}$ iteration, and rendered the solutions $K_P = 0.9993, K_I = 5.084810 - 4$ and $K_D = 0.0751$. Fig. 9. shows the corresponding convergence curves. Step response of the controlled system is also shown in *Fig. 8*. It can be noticed that the rise-time = 0.161 s, the settling time = 0.401 s, the delay time = 0.4 ms, and no overshoot.

## 5. CONCLUSIONS

The article presents the adaptive tabu search (ATS) with its parallel counter part named management agent (MA). The algorithms are detailed in an easy-to-follow manner. The performance assessment is also elaborated. The calculated speed up ratios are given to confirm the superiority of the MA(ATS). Two applications in optimalizing control of a scaled vehicle are presented.

## 6. ACKNOWLEDGMENT

Financial support from Suranaree University of Technology is greatly appreciated.

### References

[1] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, Vol. 2202, No. 4598, pp. 671-680, 1983.

[2] G. Ducek and T. Scheuer, "Threshold Accepting: A General Purpose Optimization Algorithm," *Journal of Computational Physics*, Vol. 90, pp. 161-175,1990.

[3] F. Glover, and M. Laguna, *Tabu Search*, Kluwer Academic, 1997.

[4] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[5] P. Moscato, "Memetic Algorithms: A Short Introduction," New Ideas in Optimization (D. Corne, M. Dorigo, and F. Glover Ed.), McGraw-Hill, 1999.

[6] R. Marti, M. Laguna, and F. Glover, "Principle of Scatter Search," *European Journal of Operation Research*, Vol. 169, pp. 359-372, 2006.

[7] D.B. Fogel, *Evolutionary Computation-Toward a New Philosophy of Machine Intelligence*, em IEEE Press, 1995.

[8] H.-G. Beyer, H.-P. Schwefel, and I. Wegener, "How to Analyse Evolutionary Algorithms," *Theoretical Computer Science*, Vol. 287, pp. 101-130, 2002.

[9] A.H. Mantawy, Y.L. Abdel-Magid, and S.Z. Selim,"Unit Commitment by Tabu Search," *IEEE Proc.-Gener. Transm. Distrib.*, Vol. 45, No. 1, pp. 56-64,1998.

[10] T. Kulworawanichpong, and S. Sujitjorn, "Optimal Power Flow Using Tabu Search," *IEEE Power Engineering Review*, Vol. 22, No. 6, pp. 37- 40, 2002.

[11] J.-F. Cordean, and G. Laporte, "A Tabu Search Heuristic for the Static Multi-Vehicle Dial-a-Ride Problem," *Transportation Research Part B: Methodological*, Vol. 37, No. 6, pp. 579-594, 2003.

[12] E. Nowicki, and C. Smutnicki, "A Fast Tabu Search Algorithm for The Flow Shop Problem," *European Journal of Operational Research*, Vol. 91, pp. 160-175, 1996.

[13] G. Zhang, W. Habenitch, and W.E.L. SpieB, "Improving the Structure of Deep Frozen and Chilled Food Chain with Tabu Search Procedure," *Journal of Food Engineering*, Vol. 60, No. 1, pp. 67-79, 2003.

[14] S. Sujitjorn, T. Kulworawanichpong, D. Puangdownreong, and K. N. Areerak, "Adaptive Tabu Search and Applications in Engineering Design," *Frontiers in Artificial Intelligent and Applications*, Vol. 149, pp. 233-257, 2006.

[15] Y.A. Kochetov, and E.N. Goncharov, "Probabilistic Tabu Search Algorithm for The Multi-Stage Uncapacitated Facility Location Problem," *Operation Research Proceedings*, pp. 65-70, 2000.

[16] R. Battiti, and G. Tecchiolli, "The Reactive Tabu Search," *ORSA Journal on Computing*, Vol. 6, No. 2, pp. 126-140, 1994.

[17] K.-N. Areerak, and S. Sujitjorn, "Performance Comparison between Genetic Algorithm and Tabu Search Methods" *Suranaree J. Sci. Technol.*, Vol.9, pp. 61-68, 2002.

[18] D. Puangdownreong, and S. Sujitjorn, "Image Approach to System Identification," *WSEAS Transactions on Systems*, Vol. 5, No. 5, pp. 930-938, 2006.

[19] D. Puangdownreong, K. N. Areerak, K. L. Areerak, T. Kulworawanichpong, and S. Sujitjorn, "Application of Adaptive Tabu Search to System Identification," *Proc. IASTED Int. Conf. MIC.*, pp. 178-183, 2005.

[20] T. Kulworawanichpong, K. N. Areerak, K. L. Areerak, and S. Sujitjorn, "Harmonic Identification for Active Power Filters Via Adaptive Tabu Search Method," *LNAI3215- PartIII*, pp. 687-694, 2004.

[21] D. Puangdownreong, C. U. Thaiwasin, and S. Sujitjorn, "Optimized Performance of a 2-Mass Rotary System Using Adaptive Tabu Search," *WSEAS Transactions on Systems*, Vol. 5, No. 3, pp. 339-345, 2006.

[22] S. Sujitjorn, and S. Khawn-on, "Learning Control via Neuro-Tabu-Fuzzy Controller," *LNAI4251*,pp. 833-840, 2006.

[23] W. Ngernbaht, K.-L. Areerak, and S. Sujitjorn, "Resonance and Friction Compensation in a Micro Hard Drive," *WSEAS Transactions on Systems*, Vol. 8, No. 2, pp. 179-188, 2009.

[24] N. Sriyingyong, and K. Attakitmongcol, "Wavelet-Based Audio Watermarking Using Adaptive Tabu Search," *IEEE Proc. 1st Int. Symp. on Wireless Pervasive Computing*, pp 1-5, 2006.

[25] J. Kluabwang, D. Puangdownreong and S. Sujitjorn, "Management Agent for Search Algorithms with Surface Optimization Applications," *WSEAS Transactions on Computers*, Vol. 7, No. 6, pp. 791-803, 2008.

[26] S. Brennan, and A. Alleyne, "Using a Scale Testbed: Controller Design and Evolution,"

*IEEE Control Systems Magazine*, Vol. 21, No. 3, pp. 15-26, 2001.

**Sarawut Sujitjorn** received his B.Sc. (Hons) in electrical engineering from the Royal Thai Air Force (RTAF) Academy in 1984, and Ph.D. in electronic and electrical engineering from the University of Birmingham, UK, in 1987. He joined the Department of Electrical Engineering, RTAF Academy, in 1987. Since 1993, he has been with Suranaree University of Technology (SUT), Thailand, where he was the cofounder of the Center for Scientific and Technological Equipment. At SUT, Dr. Sujitjorn, while remains active in research, has served the university under various administrative positions including Vice Rector for Academic Affairs, Deputy Dean of the Institute of Industrial Technology, Deputy Director of the Center for Scientific and Technological Equipment, and Chairperson of the School of Electrical Engineering. At present, he is a professor of electrical engineering, and Director of the Institute of Research and Development. His research interests include control and identification, applied AI, search algorithms, harmonics detection algorithms, hybrid power filters, and electric drives. He has held 14 filed patents, 4 software copyrights, authored 1 book and 1 monograph, co-translated 1 book, and published as authors and coauthors of more than 100 research and technical articles in peer-reviewed journals and conference proceedings nationally and internationally.

**Jukkrit Kluabwang** was born in Bangkok, Thailand, in 1973. He received his BEng (EE) from Chiangmai University in 1995, and MEng (EE) from Chulalongkorn University in 2002. He is currently pursuing his PhD (EE) at Suranaree University of Technology, Thailand, through a scholarship from Rajamangala University of Technology Lanna Tak. His research interests include search algorithms, power electronics, control and identification.

**Deacha Puangdownreong** was born in Pranakhonsri Ayutthaya, Thailand, in 1970. He received the B.Eng. degree in electrical engineering from South-East Asia University (SAU), Bangkok, Thailand, in 1993, the M.Eng. degree in control engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand, in 1996, and the Ph.D. degree in electrical engineering from Suranaree University of Technology (SUT), Nakhon Ratchasima, Thailand, in 2005, respectively. Since 1994, he has been with the faculty of engineering, South-East Asia University, where he is currently an assistant professor in the department of electrical engineering. His research interests include linear and non-linear control, system identification, and applications of AI techniques to various engineering problems. His contributions include 20 research articles in peer-reviewed journals and conference proceedings, and one software copyright.

**Nuapett Sarasiri** was born in Chachoengsao, Thailand, in 1984. She received her BEng (EE) from Suranaree University of Technology (SUT), Thailand, in 2007, and MEng (EE) from the same university in 2009. She is currently working towards her PhD (EE) at SUT. Her research interests include power electronics, control, and search algorithms.