Management Agent for Search Algorithms with Surface Optimization Applications

Jukkrit Kluabwang[1] Deacha Puangdownreong[2] and Sarawut Sujitjorn[1*]
[1]School of Electrical Engineering, Institute of Engineering
Suranaree University of Technology
Nakorn Ratchasima, Thailand
[2]Department of Electrical Engineering, Faculty of Engineering
South-East Asia University
Bangkok, Thailand
* corresponding author: sarawut@sut.ac.th
http://www.sut.ac.th/engineering/electrical/carg/

Abstract: - This paper presents a management approach applied to search algorithms to achieve more efficient search. It acts as a management agent to a core search unit, in which the Adaptive Tabu Search (ATS) has been applied. The proposed management agent composes of partitioning mechanism (PM), sequencing method (SM), and discarding mechanism to speed up the search. It has been tested against Bohachevsky's, Rastrigin's and Shekel's foxholes functions, respectively, for surface optimization. The paper gives a review of the ATS, detailed explanations of the PM, SM, and DM, respectively. Comparison of the optimization results are elaborated.

Key-Words: - search algorithms, management agent, partitioning mechanism, discarding mechanism, sequencing method, adaptive tabu search

1 Introduction

For the first time in 1989, Glover introduced the Tabu Search (TS) to solve combinatorial optimization problems [1,2]. Recently, it has been applied to school timetabling [3], power system restoration [4], job-shop scheduling [5], etc. The TS possesses two main strategies namely intensification and diversification [6,7] which make the TS quite flexible for a diversity of problems. On the basis of the naïve TS, researchers at Suranaree Unviversity of Technology (SUT) launched a modified TS so called Adaptive Tabu Search (ATS) in 2004. The ATS contains two distinctive mechanisms denoted as back-tracking (BT) and adaptive radius (AR) mechanisms, respectively. The former can be regarded as one type of the diversification strategies, while the later as one of those intensification tactics. Performance assessment and convergence proof of the ATS have been reported [8,9]. The ATS has been applied to solve various optimization problems, e.g. power system protection [10], dynamic system identification [11,12], control system synthesis [13,14], and audio signal processing [15].

This paper reports our attempts to improve the performance of the ATS. Our approach is to have algorithms working as a management agent to distribute the search units over an entire search space, and to effectively terminate some search paths unlikely to hit the global solution. The effectiveness of the management agent (MA) is demonstrated by using the ATS as its search units. So, we firsty review the ATS in Section 2.1. Explanation of the MA follows in Section 2.2. Performance evaluation, results and discussions can be found in Sections 3 and 4, respectively. Section 5 provides conclusions.

2 Algorithms

2.1 Adaptive Tabu Search (ATS)

The ATS as an original search core or unit begins the search with some random initial solutions belonging to a neighbourhood search space. The objective functions of these solutions are evaluated such that one with the best objective value is kept. This recent solution serves as the starting point of the next search or move, and is recorded in the tabu list. Subsequent searches occur in this manner until the global solution is reached. As a matter of fact, search moves can be entrapped by some local solutions. Without an efficient escaping logic, the search could fall into a deadlock situation. The ATS possesses the BT mechanism which looks up the tabu list and selects one of its listed solution as a new starting point. A new search could begin in a new direction by using this mechanism. Hence, an entrapment by a local solution can be released. Once the search approaches the global solution, the AR mechanism is invoked. Commonly, the search radius is subsequently reduced to provide finer and finer solutions within a short search time. Fig. 1 illustrates some movements of the ATS while searches, and and Fig. 2 showsthe flow diagram of the ATS. In addition, the ATS algorithms can be described in a step-by-step manner as follows:

Step 1 Initialize a search space, *count* and $count_{max}$

Step 2 Randomly select an initial solution S_0 from the search space. Let S_0 be a current local minimum.

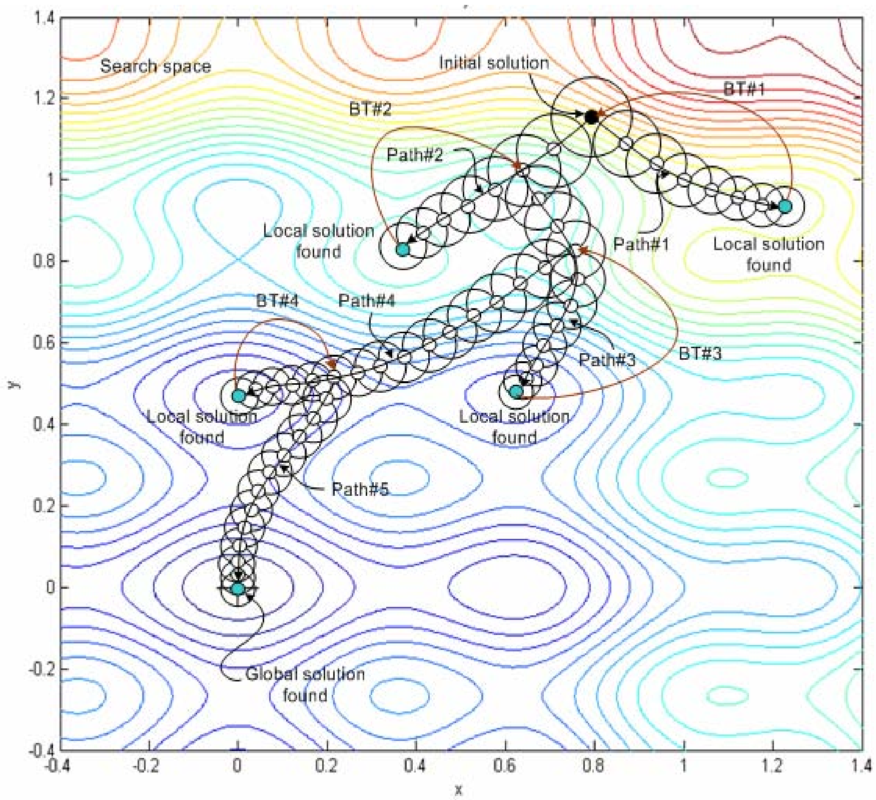


Fig. 1 Movements of the ATS.

Step 3 Randomly generate N solutions around S_0 within a certain radius R. Store the N solutions, called neighborhood, in a set X.

Step 4 Evaluate a cost function of each member in X. Set S_1 as a member that gives the minimum cost in X.

Step 5 If $S_1 < S_0$, put S_0 into the Tabu list and set $S_0 = S_1$, otherwise, store S_1 into the Tabu list instead.

Step 6 Activate the back-tracking mechanism , when solution cycling occurs.

Step 7 If the termination criteria: $count \geq count_{max}$, or desired specifications are met, then stop the search process. S_0 is the best solution, otherwise go to Step 8.

Step 8 Activate the adaptive search radius mechanism, when a current solution S_0 is relatively close to a local mini-Mum to refine searching accuracy.

Step 9 Update *count*, and go to Step 2.

To apply the ATS effectively, potential users should consider the following recommendations (i) the initial search radius, R, should be 7.5-15.0% of search space radius, (ii) the number of neighborhood members, N, should be 30-40, (iii) the number of repetions of a solution before invoking the back-tracking mechanism should be 5-15, (iv) the k^{th} backward solution selected by the back-tracking mechanism should be equal or close to the number of repetitions of a solution before invoking the back-tracking mechanism, (v) the adaptive search radius should employ 20-25% of radius reduction, and (vi) a well educated guess of the search space that is wide enough to cover the global solution is necessary, [8-13].

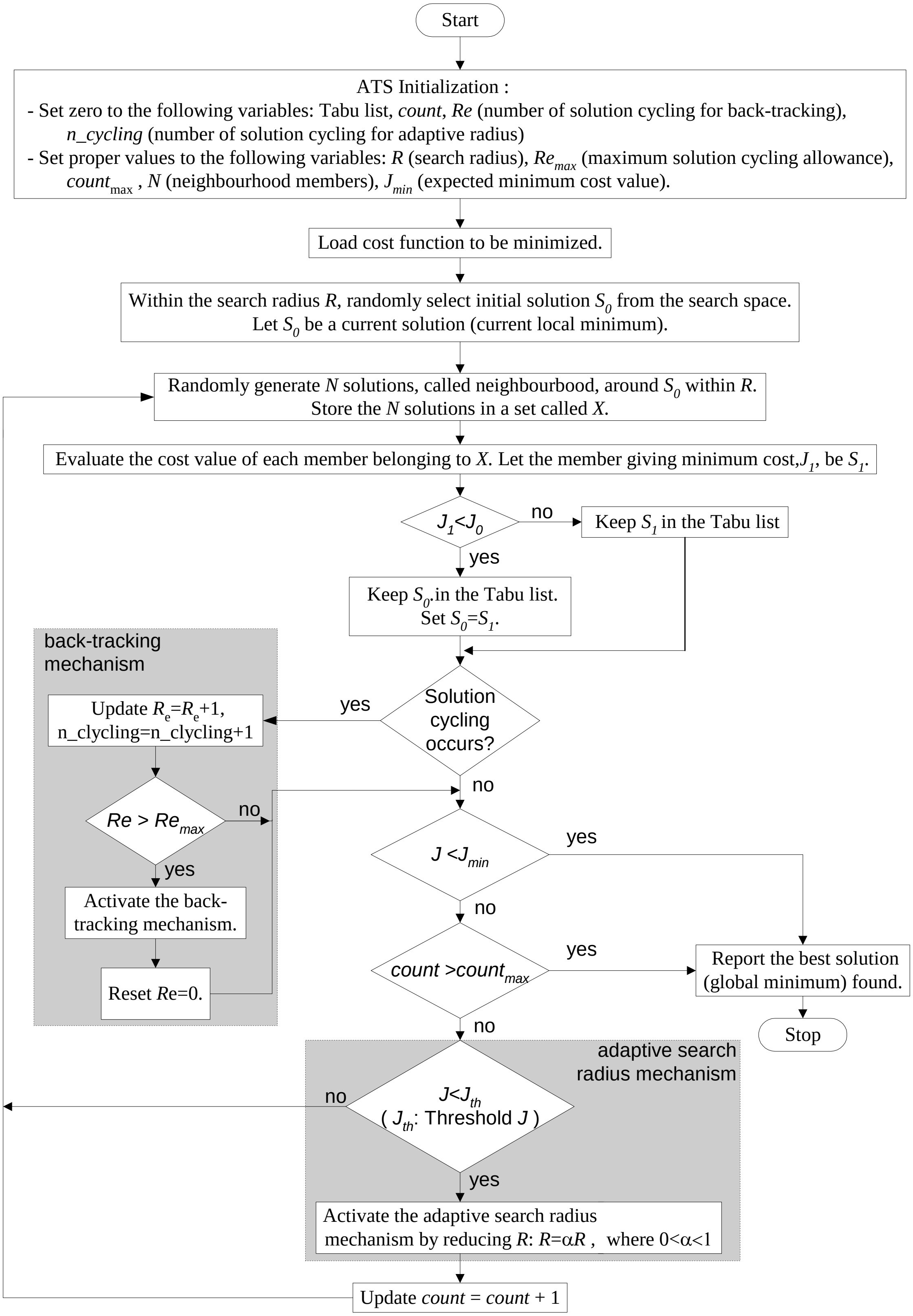


Fig. 2 Flow diagram of the ATS.

2.2 Management Agent (MA)

The MA organizes corresponding search units to achieve a global solution within a rapidly finite search time. It contains three main strategies namely sequencing method (SM), partitioning and discarding mechanisms (PM and DM), respectively. The PM assists on dividing the entire search space into several to many sub-search-spaces. Also, PM initiates searches among those spaces. The DM serves to identify some unlikely to be successful search moves, and extinguish them.

By going through the DM process repeatedly within a finite time, the most likely to be successful search path would be singled out. The particular search path eventually hits the global solution. More descriptions of PM and DM appear in the topics 2.2.1 and 2.2.3, respectively. The SM is explained in the topic 2.2.2. Since the ATS has been chosen as a core search unit, our proposed MA can be represented by the flow diagram in Fig. 3.

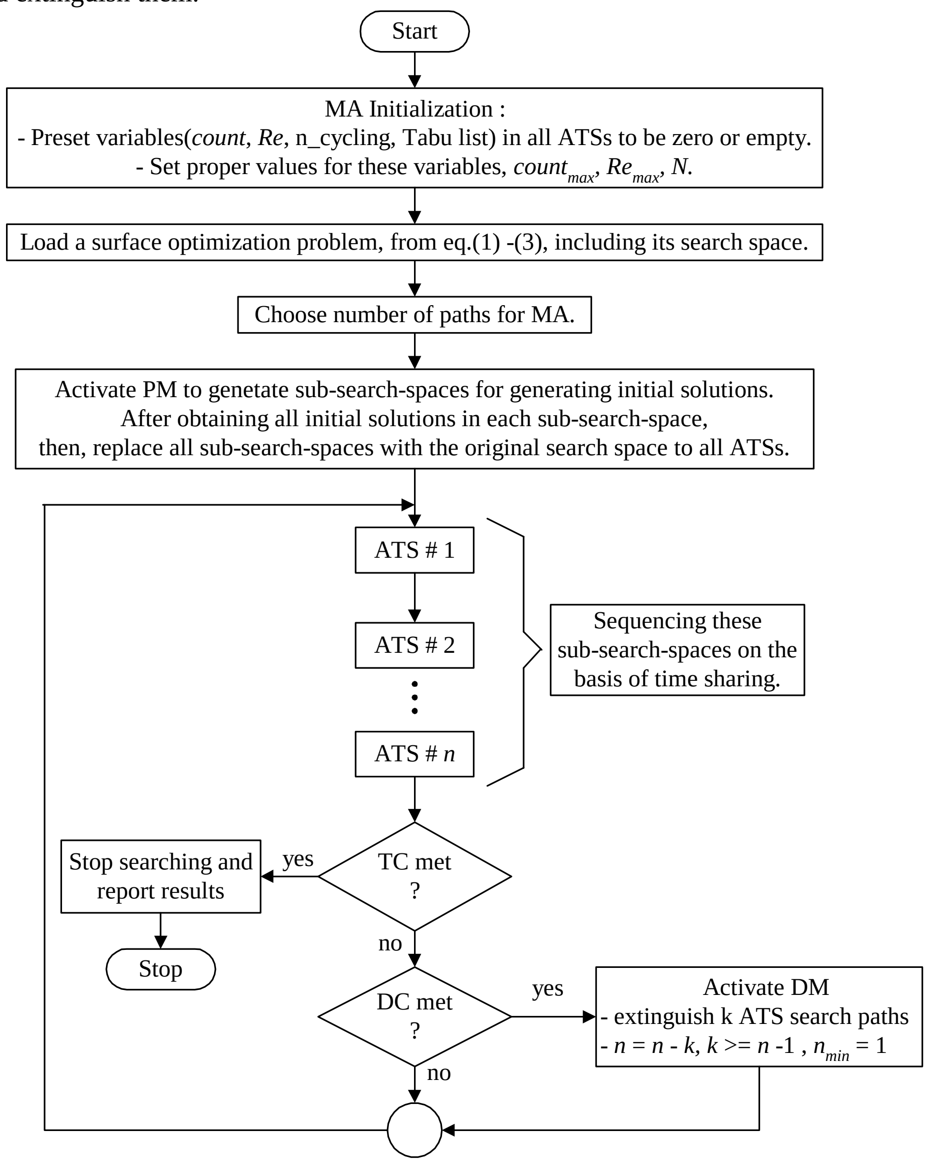


Fig. 3 Flow diagram of the MA.

The MA contains 7 major steps as follows:

Step 1 Identify search spaces, and obtain initial solutions.

Step 2 Define the number of search units or search paths.

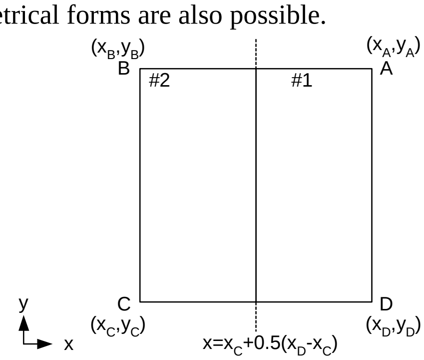
Step 3 Invoke the PM.

Step 4 Invoke the ATS#1, ATS#2,…,ATS# n $(n = n - k,\ k \leq n-1, n_{\min} = 1)$
Step 5 Evaluate termination criteria (TC). Completely stop some paths of the ATS according to the TC.
Step 6 Invoke the DM.
Step 7 Update counter. Goto step 4 until the global solution is reached.

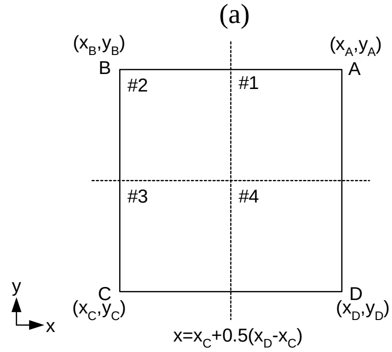
As the ATS possesses the convergence pro-perty [8,9], the proposed MA having the ATS as its core always converges to the global solution. PM, one of the main strategies of the MA, is explained in the topic 2.2.1 as follows.

2.2.1 Partitioning Machanism (PM)

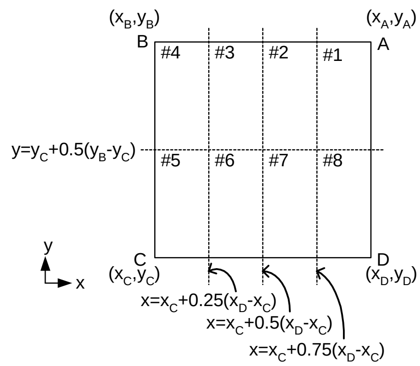
The concept of problem partitioning is well known, and recently been applied to data fusion [16] and genetic algorithms [17]. Our work also utilizes the PM to provide higher successful rates of the search hitting the global solution. The PM, once invoked, starts dividing the entire search space into a few to many sub-search-spaces. From our previous tests against some surfaces, whose details are given in Section 3, it was found that the searches had been slow if the number of sub-search-spaces was more than 8. To aid the understanding, let the entire search space be a 2D-rectangle, ABCD, as shown in Fig. 4(a). Figs. 4(a)-(c) depict the partitioning of rectangular forms into 2, 4, and 8 parts, while other geometrical forms are also possible.



(a)



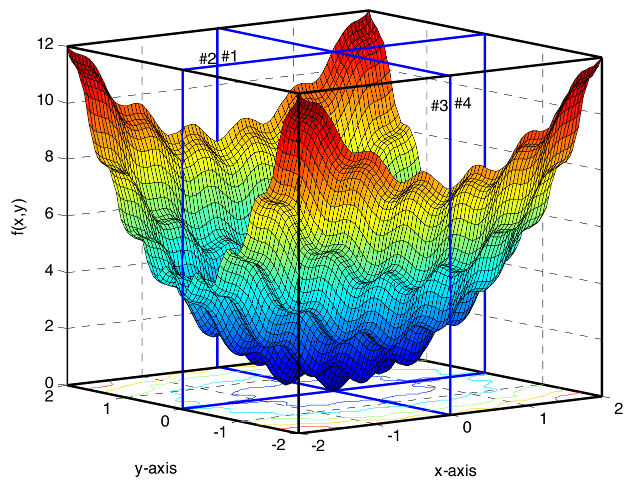
(b)



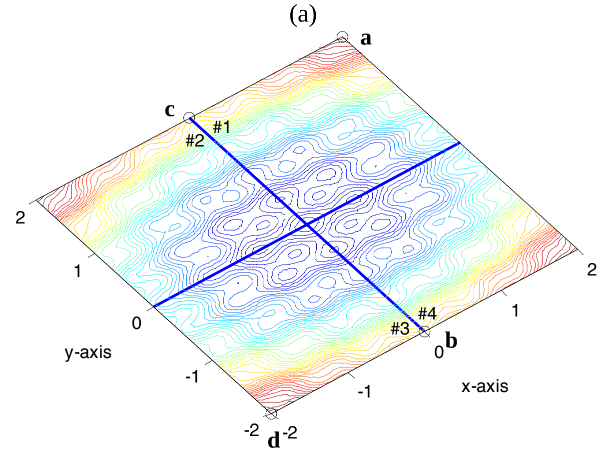
(c)

Fig. 4 Partitioning 2D-search-space (a) 2 regions (b) 4 regions and (c) 8 regions.

Partitioning the 3-D Bohachevsky's function into 4 rectangular cylinders is shown in Fig. 5.



(a)



(b)

Fig. 5 Partitioning 3D-search-space (PM) (a) Bohachevsky's surface (b) 2-D map.

After successful partitioning, the PM creates initial solutions with their corresponding neighbourhoods for all sub-search-spaces. Then, the partitioning boundaries are removed, and all search paths run freely over the entire search space. At this stage, symmetrical and non-overlapping partitioning is assumed. Asymmetrical and overlapping partitioning is possible and open for further investigations.

An important issue concerning the use of PM is the maximum number of search units. The number of partitioned regions must be finite. For the ATS and 2D problem confined within a square, it is defined by

$$N_{\max} = \left(\frac{l}{2R}\right)^2 \qquad (1)$$

, where $N_{\max}$ is the maximum number of partitioned regions, l is the length (m) of the side of the square, and R is the ATS search radius (m). Readers are reminded that the ATS employs circular neighbourhoods with the radii R. For example, a search problem fitted in a square of 30 cm x 30 cm, and the ATS having its search radius of 1.5 cm, $N_{\max}$ is equal to 100.

2.2.2 Sequencing Method (SM)

Referring to Fig. 3, between the PM and the DM, there exists the sequencing method (SM). The SM is a time-sharing tactic which organizes the search units to run in a sequential manner. Readers are reminded that the ATS and the MA are sequential algorithms. The SM organizes the ATS paths to run one-by-one on a single iteration at a time. This process is repeated k times. Then, their results are transferred to the DM. After the DM completing its task (described in the next section), the remaining search paths are transferred back to the SM. These sequential process do not require a parallel platform. Nonetheless, a parallel platform with multitasking operating system (OS) would result in a more rapid search. By the SM, each ATS unit performs its search one-by-one with its own initial solutions. This process is depicted in Fig. 6. By the end of each iteration, i.e. the ATS#*n* performs a complete search at each iteration, the DM is invoked. Its mechanism is described next.

2.2.3 Discarding Mechanism (DM)

The DM assists on eliminating some unlikely to be successful search paths. For example, some search paths could be locked by local solutions, convergence of some paths may be slow, etc. The criterion for an activation of the DM is called discarding criterion (DC). The DC is set upon the threshold of the $|\varepsilon|$. $|\varepsilon|$ is defined by the absolute difference between the objective value of the current solution and the goal objective value. In some aspects, the DM may behave similarly to the location management algorithms in [18,19].

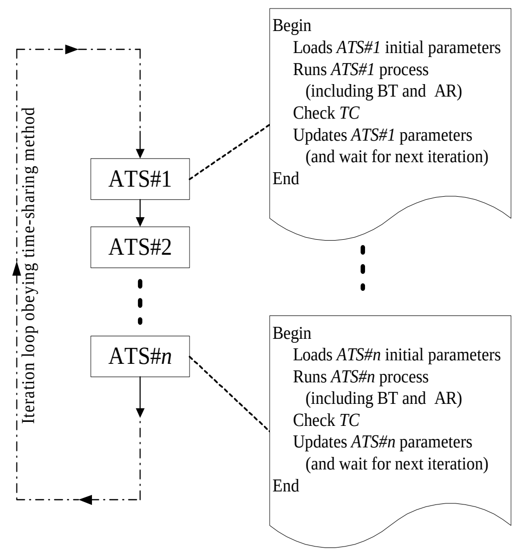


Fig. 6 Diagram of the sequencing method (SM).

The proposed DM performs a min-max sort to the objective values, keep only half of the search paths with smaller objective values, and eliminate the rest. The number of times invoking the DM depends on the number of search paths. As summarized in Table 1, the MA#16 possesses 16 search paths and requires 4 cycles of operation of the DM. By the first stage, the DM extinguishes the search paths by half, thus only 8 paths remain. By the forth stage, only a single path remains and particularly expected to strike the global solution.

Table 1 DM and number of agents.

Types of MA	number of agents when DM invoked											
	1st stage		2nd stage		3rd stage		4th stage		5th stage		6th stage	
	in	out	in	out	in	out	in	out	in	out	in	out
MA#2	2	1										
MA#4	4	2	2	1								
MA#8	8	4	4	2	2	1						
MA#16	16	8	8	4	4	2	2	1				
MA#32	32	16	16	8	8	4	4	2	2	1		
MA#64	64	32	32	16	16	8	8	4	4	2	2	1

3 Performance Evaluations

The proposed MA has been tested against surface optimization problems of the following functions: Bohachevsky's (see Fig. 5(a)), Rastrigin's (see Fig. 7), Shekel's foxholes (see Fig. 8), etc. They are described by the mathematical expressions in the equations (2)-(4), respectively.

$$f(x,y) = x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7 \quad (2),$$

$$f(x,y) = x^2 + y^2 - 10\cos(2\pi x) - 10\cos(2\pi y) + 20 \quad (3),$$

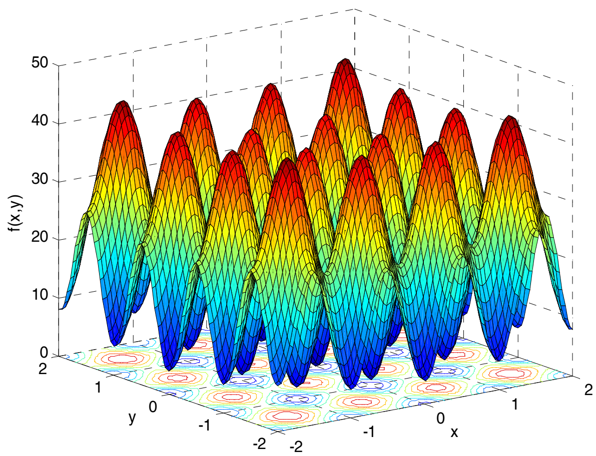


Fig. 7 Rastrigin's surface

$$f(x_1, x_2) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right]^{-1} \quad (4),$$

in which

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

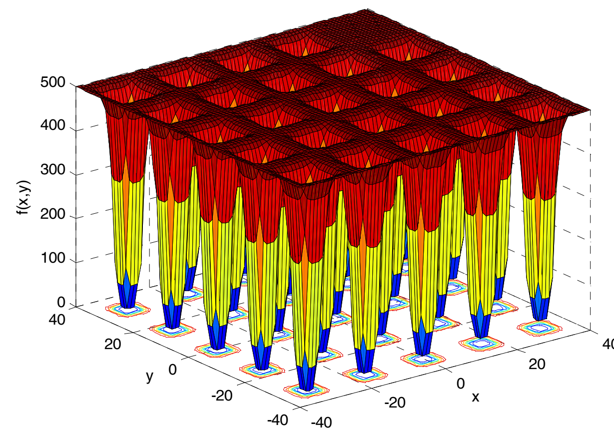


Fig. 8 Shekel 's foxholes surface

Table 2 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	$1\text{x}10^{-9}$	J<$1\text{x}10^{-1}$,R=$2\text{x}10^{-3}$	J<$1\text{x}10^{-3}$,R=$2\text{x}10^{-4}$	Up to PM
RF	0.2 (5%)	30	5	10,000	$1\text{x}10^{-8}$			
SF	0.8 (1%)	30	5	10,000	0.999	J<5 ,R=0.5	J<2 ,R=0.1	

Table 3 MA parameters in terms of DM arguments.

Test functions	Iterationth at which DM is invoked.																				
	MA#2	MA#4		MA#8			MA#16				MA#32					MA#64					
	1st stage	1st stage	2nd stage	1st stage	2nd stage	3rd stage	1st stage	2nd stage	3rd stage	4th stage	1st stage	2nd stage	3rd stage	4th stage	5th stage	1st stage	2nd stage	3rd stage	4th stage	5th stage	6th 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

The Bohachevsky's and Rastrigin's functions have their global minima of zero values at (0,0), while the minimum of the Shekel's foxholes function is at (-32,-32) and equal to 1. These surfaces are very difficult to search. Mostly, conventional search and optimization methods fail to find their minima, even some unconventional and intelligent search methods fail too.

Our algorithms were coded in MATLAB™, and run on a Pentium IV 2.4 GHz 640 Mbytes of SD-RAM. Performance of the original ATS has been compared with those of the MA. The search parameters need to be predefined for the MA and the ATS. Tables 2, 3 and 4, in the next page, summarize those parameters in accordance with the functions to be searched in which BF, RF and SF stand for Bohachevsky's, Rastrigin's and Shekel's foxholes functions, respectively.

Table 4 MA parameters in terms of PM outputs.

Test functions	Sub-search-spaces from PM					
	MA#2*	MA#4*	MA#8*	MA#16*	MA#32*	MA#64*
BF and RF	1#[2 2; 0 -2] 2#[0 2;-2 -2]	1#[2 2;0 0] 2#[0 2;-2 0] 3#[0 0;-2 -2] 4#[2 0;0 -2]	1#[2 2;1 0] 2#[1 2;0 0] 3#[0 2;-1 0] 4#[-1 2;-2 0] 5#[-1 0;-2 -2] 6#[0 0;-1 -2] 7#[1 0;0 -2] 8#[2 0;1 -2]	1#[2 2;1 1] 2#[1 2;0 1] 3#[0 2;-1 1] 4#[-1 2;-2 1] 5#[-1 1;-2 0] 6#[0 1;-1 0] 7#[1 1;0 0] 8#[2 1;1 0] 9#[2 0;1 1] 10#[1 0;0 1] 11#[0 0;-1 1] 12#[-1 0;-2 1] 13#[-1-1;-2 -2] 14#[0 -1;-1 -2] 15#[1 -1;0 -2] 16#[2 -1;1 -2]	1#[2 2;1.5 1] 2#[1.5 2;1 1] 3#[1 2;0.5 1] 4#[0.5 2;0 1] 5#[0 2;-0.5 1] 6#[-0.5 2;-1 1] 7#[-1 2;-1.5 1] 8#[-1.5 2;-2 1] 9#[2 1;1.5 0] 10#[1.5 1;1 0] 11#[1 1;0.5 0] 12#[0.5 1;0 0] 13#[0 1;-0.5 0] 14#[-0.5 1;-1 0] 15#[-1 1;-1.5 0] 16#[-1.5 1;-2 0] the rest is omitted.	1#[2 2;1.5 1.5] 2#[1.5 2;1 1.5] 3#[1 2;0.5 1.5] 4#[0.5 2;0 1.5] 5#[-0 2;-0.5 1.5] 6#[-0.5 2;-1 1.5] 7#[-1 2;-1.5 1.5] 8#[-1.5 2;-2 1.5] 9#[-1.5 1.5;-2 1] 10#[-1 1.5;-1.5 1] 11#[-0.5 1.5;-1 1] 12#[0 1.5;-0.5 1] 13#[0.5 1.5;0 1] 14#[1 1.5;0.5 1] 15#[1.5 1.5;1 1] 16#[2 1.5;1.5 1] the rest is omitted.
SF	1#[40 40; 0 -40] 2#[0 40;-40 -40]	1#[40 40; 0 0] 2#[0 40;-40 0] 3#[40 0;0 -40] 4#[0 0;-40 -40]	1#[40 40; 20 0] 2#[20 40;0 0] 3#[0 40;-20 0] 4#[-20 40;-40 0] 5#[40 0;20 -40] 6#[20 0;0 -40] 7#[0 0;-20 -40] 8#[-20 0;-40 -40]	1#[40 40; 20 20] 2#[20 40;0 20] 3#[0 40;-20 20] 4#[-20 40;-40 20] 5#[-20 20;-40 0] 6#[0 20;-20 0] 7#[20 20;0 0] 8#[40 20;20 0] 9#[40 0; 20 -20] 10#[20 0;0 -20] 11#[0 0;-20 -20] 12#[-20 0;-40 -20] 13#[-20 -20;-40-40] 14#[0 -20;-20 -40] 15#[20 -20;0 -40] 16#[40-20;20 -40]	1#[40 40; 30 20] 2#[30 40;20 20] 3#[20 40;10 20] 4#[10 40;0 20] 5#[0 40;-10 20] 6#[-10 40;-20 20] 7#[-20 40;-30 20] 8#[-30 40;-40 20] 9#[-30 20; -40 0] 10#[-20 20;-30 0] 11#[-10 20;-20 0] 12#[0 20;-10 0] 13#[10 20;0 0] 14#[20 20;10 0] 15#[30 20;20 0] 16#[40 20;30 0] the rest is omitted.	1#[40 40; 30 30] 2#[30 40;20 30] 3#[20 40;10 30] 4#[10 40;0 30] 5#[0 40;-10 30] 6#[-10 40;-20 30] 7#[-20 40;-30 30] 8#[-30 40;-40 30] 9#[-30 30; -40 20] 10#[-20 30;-30 20] 11#[-10 30;-20 20] 12#[0 30;-10 20] 13#[10 30;0 20] 14#[20 30;10 20] 15#[30 30;20 20] 16#[40 30;30 20] the rest is omitted.

* 2, 4, 8, 16, 32 and 64 regions partitioned.

4 Results and Discussions

Tables 5 and 6 summarize the results obtained from 50 trials. MA#2, #4, #8 #16 #32 and #64 in these tables stand for 2, 4, 8, 16, 32 and 64 regions partitioned by the PM, respectively.

Table 5 Average search times.

Test functions	average search time (seconds)						
	ATS	MA					
		MA#2	MA#4	MA#8	MA#16	MA#32	MA#64
BF	3.9672	3.1231	3.7378	1.4087	1.5482	2.4709	3.9618
RF	5.0669	4.8881	4.3737	4.1637	3.1396	3.7825	4.6094
SF	3.7800	2.5297	1.4616	1.2141	3.2082	3.2082	2.1206

In Table 4, readers should refer to the partitioning coordinates illustrated in Fig. 4 as well as the rectangularly cylindrical partitioning concept in Fig. 5. As an example, #1[2 2;0 -2] and #2[0 2;-2 -2] for MA#2 in Table 4 are referred to the coordinates of the points #1[a;b] and #2[c;d], respectively, in accordance with the partitioning shown in Fig. 5(b).

Table 6 Average search rounds.

Test functions	average search rounds						
	ATS	MA					
		MA#2	MA#4	MA#8	MA#16	MA#32	MA#64
BF	661.98	402.14	465.48	104.60	132.64	167.48	172.06
RF	811.02	504.68	397.14	222.64	266.26	165.00	196.36
SF	118.44	60.40	49.40	26.22	35.54	10.16	30.22

"Speed up ratios" in the Fig. 9 is defined by

$$speed\ up\ ratios = \frac{averge\ search\ time\ by\ ATS}{averge\ search\ time\ by\ MA(ATS)} \quad (5)$$

where *MA(ATS)* stands for the MA having the ATS as its search core. It can be noticed that the MA considerably reduces the search time for all cases. Confirmed by the results in Fig. 9, the *MA(ATS)* performs searches 1.04-3.11 times faster than the ATS solely does. To apply the proposed algorithms effectively, potential users are recommended to explore the PM for specific problems to find highly effective number of regions partitioned.

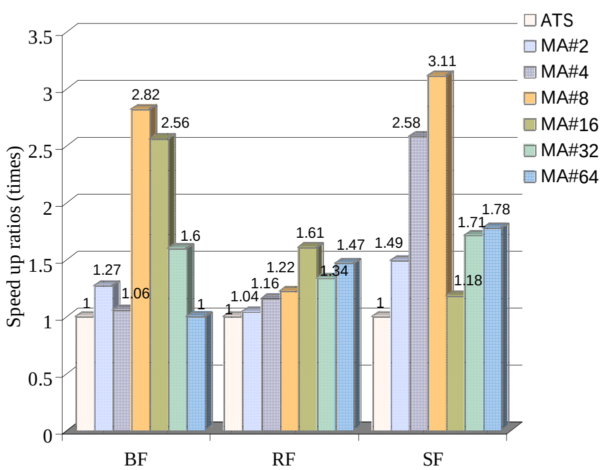


Fig. 9 Comparative bar charts of speed up ratios.

Some of the results are disclosed herewith to support our claims of the MA's performance. Fig. 10 illustrates the top view of the BF surface and the movement directions of the 32 search paths denotes by #1, #2, …, and #32. It can be noticed that almost all of the paths move towards the valley where the global solution sits in. Fig. 11 discloses the operation of the DM, and the convergence of one particular search path. At the 1st stage of the operation, the DM ceases 16 ATS paths. By the 2nd stage, only 8 paths remain, and finally only 1 path remains by the 5th stage. This last search path, which is found to be the ATS#8, strikes the global solution at the 50th iteration with $J<1x10^{-9}$ as the TC, and within 1.633 seconds. This path, ATS#8, begins its seach at the coordinate (-1.49,1.50) quite far away from the location of the global solution with its J=6.9919. It strikes the global solution at ($-0.05x10^{-6}$, $3.915x10^{-6}$) with $J=5.1476x10^{-10}<1x10^{-9}$. Figs. 12 and 13 illustrate the case of RF. The same concepts of understanding the graphical results previously explained for Figs. 10 and 11 can be applied. The case of RF is more difficult than that of the BF as this can be noticed from Fig.12 that reveals some local locks. As our results summarized by Fig. 13, the ATS#12 of 32 paths hits the global solution at the 442th iteration, with $J=4.645x10^{-6}<1x10^{-8}$ (TC), at the coordinates ($3.533x10^{-6}$, $-3.307x10^{-6}$) and within 6.049 seconds. Figs. 14 and 15 illustrate similar situations for the case of SF surface to the previus ones. It can be concluded that among the 32 search paths the ATS#18 hits the global solution at the 30th iteration, with J=0.9984<0.999 (TC), at the coordinates (-32.015,-31.7282) and within 2.3617 seconds.

5 Conclusions

This paper has presented the management agent (MA) to improve the effectiveness of the ATS regarded as the search core. The proposed MA composes of partitioning mechanism (PM), sequencing method (SM), and discarding mechanism (DM). Both MA and ATS are sequential algorithms that do not need any parallel computing platform. Performance of the MA has been evaluated against the surface optimization problems using the BF, RF and SF. The results indicate that the MA(ATS) searches 1.04-3.11 times faster than the ATS does.

Acknowledgements

The authors thanks are due to Suranaree University of Technology and Rajamangala University of Technology Lanna, Tak Campus, for the financial supports.

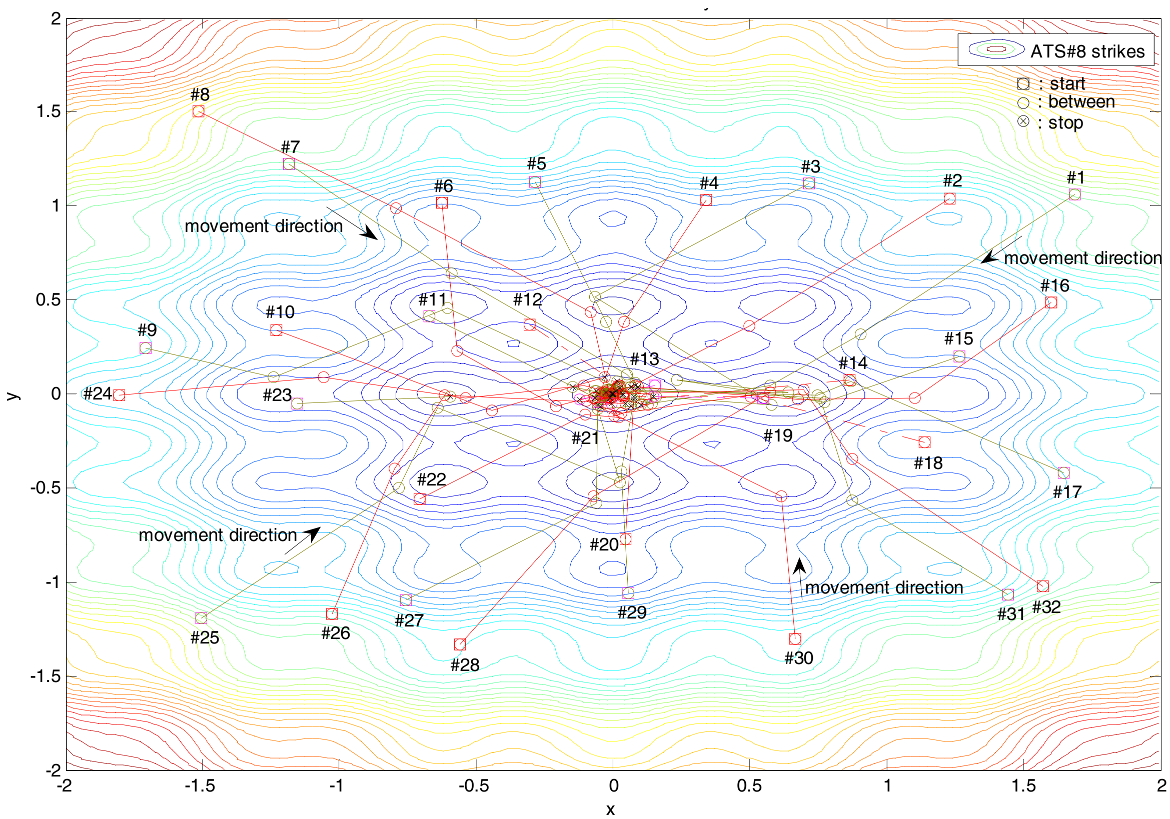


Fig. 10 BF contour with 32 ATS paths (MA#32).

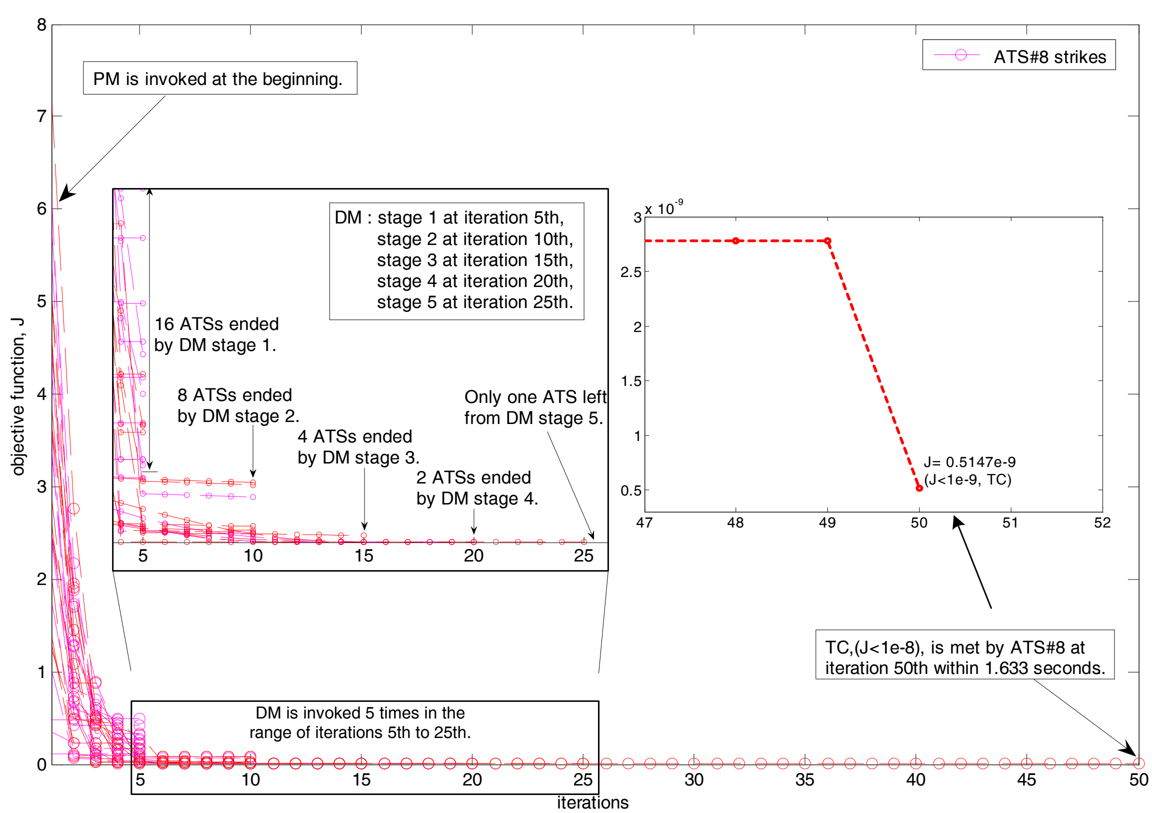


Fig. 11 Convergence curves (search on the BF surfaces).

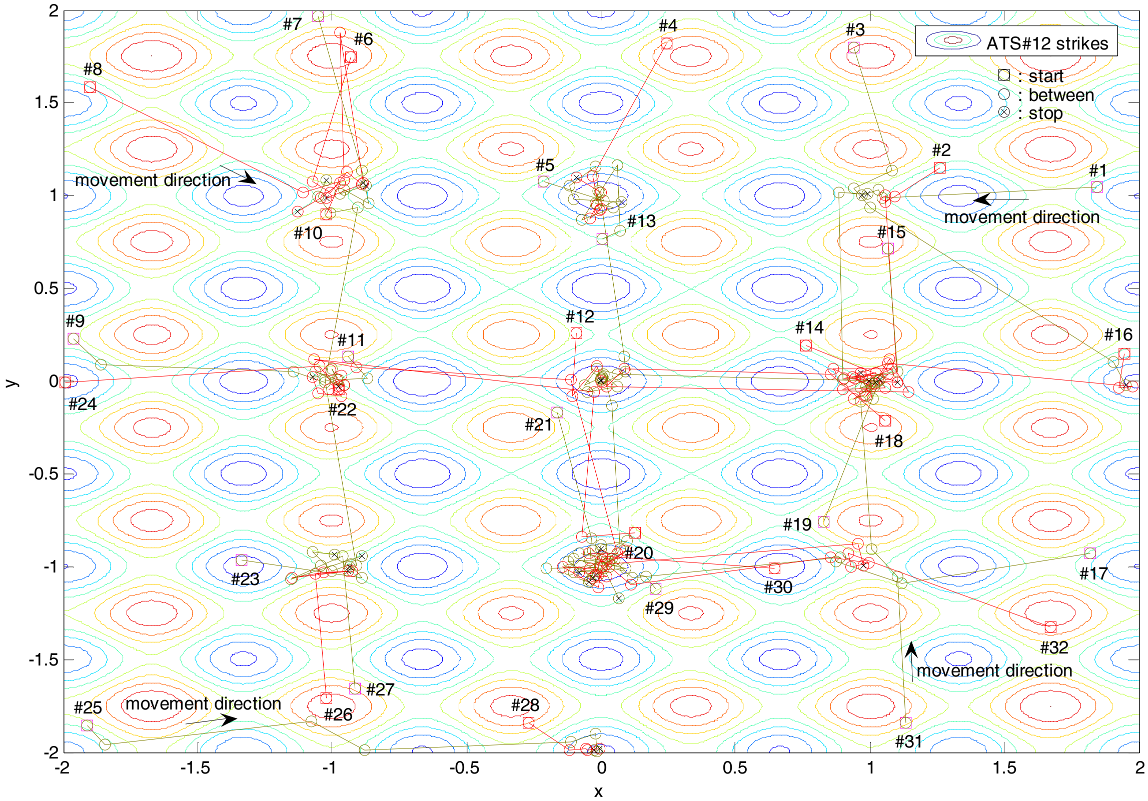


Fig. 12 RF contour with 32 ATS paths (MA#32).

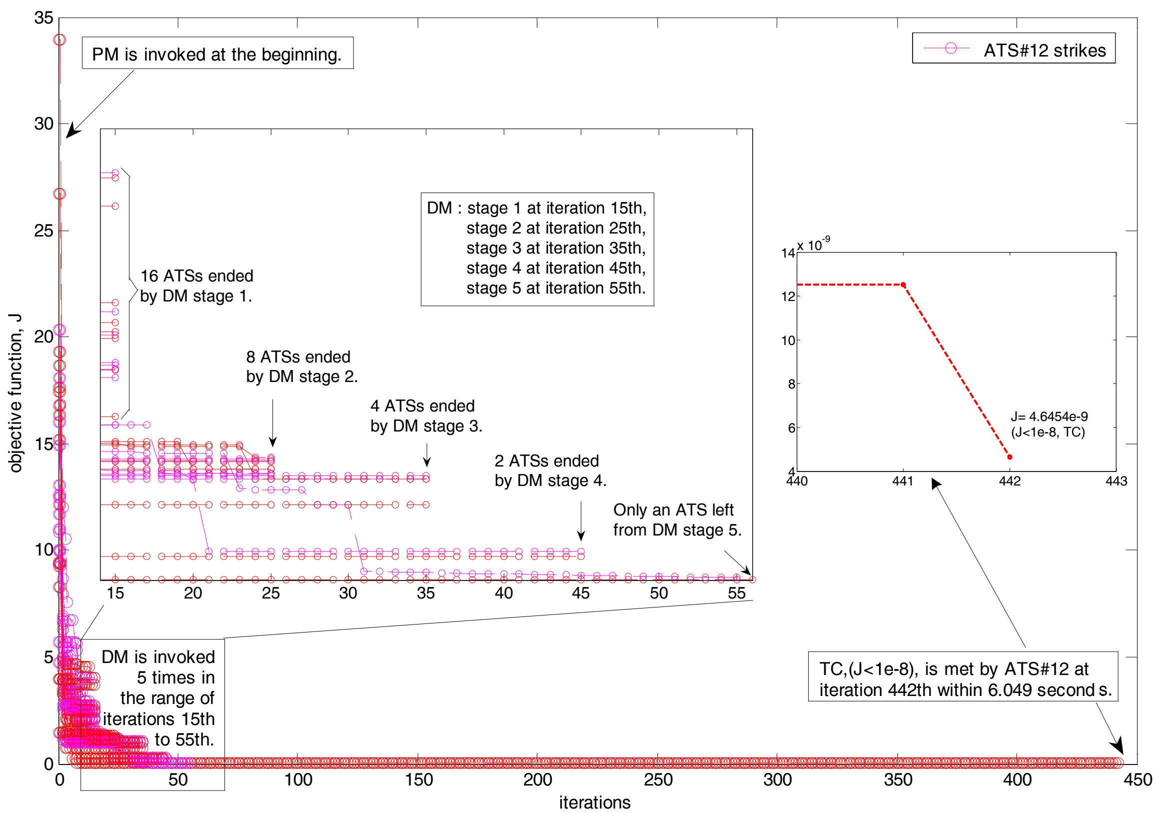


Fig. 13 Convergence curves (search on the RF surfaces).

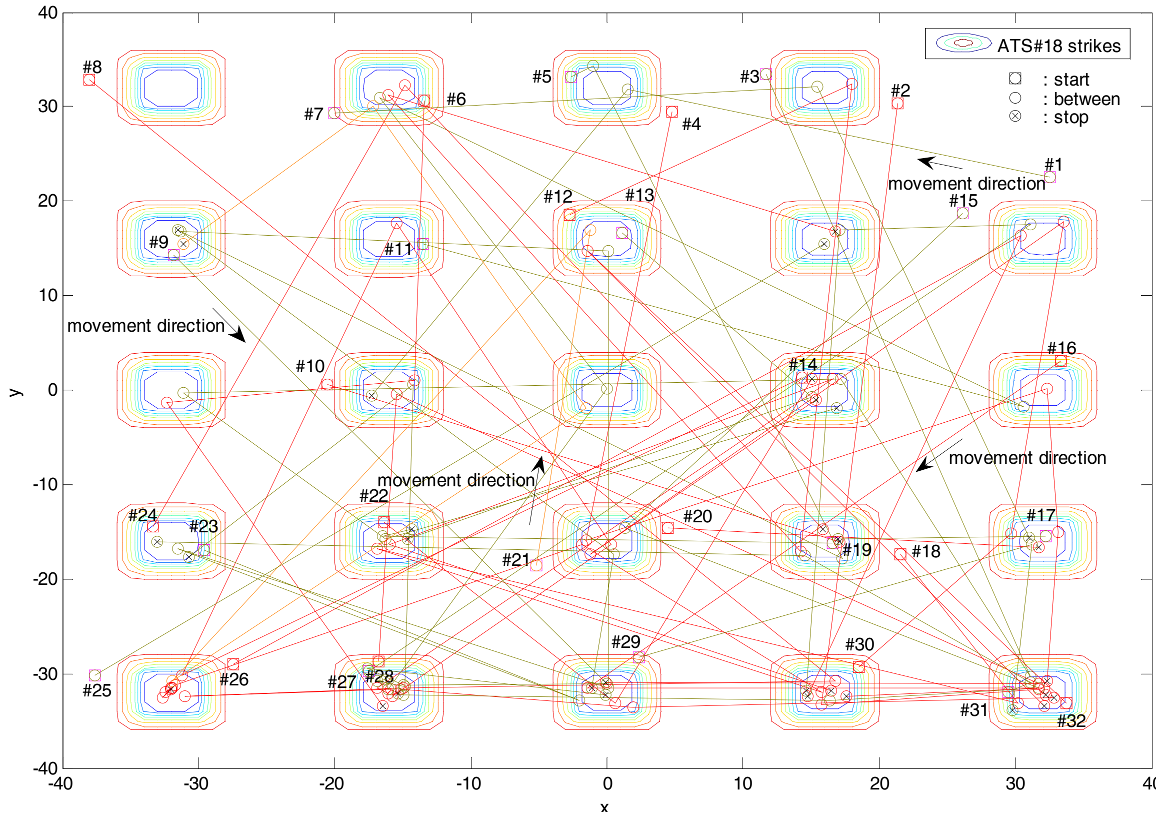


Fig. 14 SF contour with 32 ATS paths (MA#32).

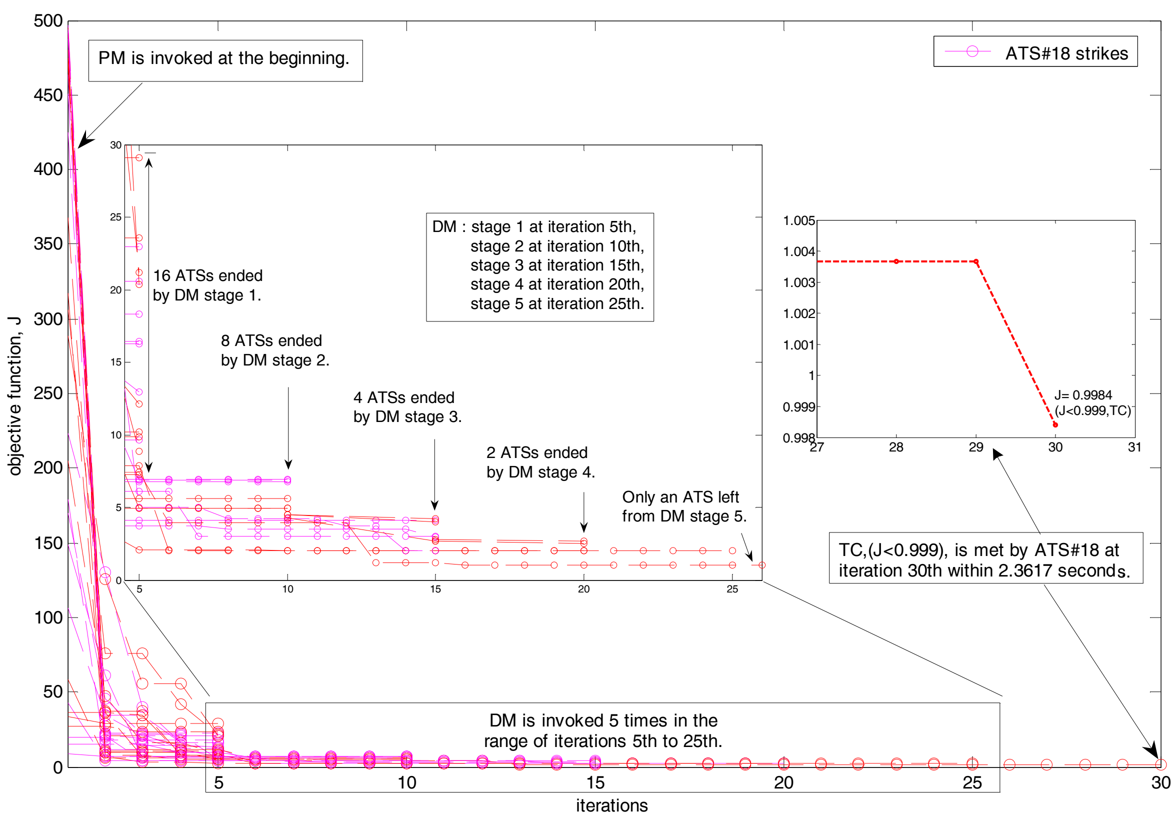


Fig. 15 Convergence curves (search on the SF surfaces).

References:

[1] F. Glover, Tabu search-Part I, *ORSA Journal on Computing*, Vol.1, No.3, 1989, pp. 190-206.

[2] F. Glover, Tabu search-Part II, *ORSA Journal on Computing*, Vol.2, No.1, 1990, pp. 4-32.

[3] F-S. Juan, A-P. Federico, L. Monica, G-S. Cinhtia, and G-R. Luis, Solving the timetabling problem with three heuristics, *WSEAS Transactions on Computer*, Vol.5, No.11, 2006, pp. 2849-2855.

[4] X-Y. Fang and X-Y. Cai, Power system restoration using a Tabu Search algorithm, *WSEAS Transactions on Circuits and systems*, Vol.4, No.10, 2005, pp. 1403-1409.

[5] I. Yoshitomo, O. Hironari and K. Norihisa, An integrated tabu search with probabilistic rule application for large-scale job shop scheduling problems with group constraints, *WSEAS Transactions on Information Science and Applications*, Vol.2, No.12, 2005, pp. 2031-2037.

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

[7] F. Glover, Parametric Tabu Search for Mixed Integer Programs, *Computers and Operations Research*, Vol.33, No.9, 2006, pp. 2449-2494.

[8]S. Sujitjorn, T. Kulworawanichpong and D. Puangdownreong and K-N Areerak, Adaptive tabu search and applications in engineering design, *Book Chapters in Integrated Intelligent Systems for Engineering Design (ed. X. F. Zha and R..J. Howlett)*, IOS Press, The Netherlands, 2006, pp. 233-257.

[9]D. Puangdownreong, T. Kulworawanichpong and S. Sujitjorn, Finite convergence and performance evaluation of adaptive tabu search, *Lecture Notes in Computer Science*, Springer-Verlag Heidelberg, 2004. (online)

[10] K-N. Areerak, T. Kulworawanichpong and S. Sujitjorn, Moving towards a new era of intelligent protection through digital relaying in power systems, *Lecture Notes in Computer Science*, Springer-Verlag Heidelberg, 2004. (online)

[11] T. Kulworawanichpong, K-L. Areerak, K-N. Areerak and S. Sujitjorn, Harmonic identification for active power filters via adaptive tabu search, *Lecture Notes in Computer Science*, Springer-Verlag Heidelberg, 2004. (online)

[12] D. Puangdownreong and S. Sujitjorn, Image approach to system identification, *WSEAS Transactions on Systems*, Vol.5, No.9, 2006, pp. 930-938.

[13] D. Puangdownreong and S. Sujitjorn, Optaining an optimum PID controller using adaptive tabu search, *Lecture Note in Computer Science*, Springer-Verlag Heidelberg, 2007. (online)

[14] D. Puangdownreong, T. Kulworawanichpong and S. Sujitjorn, Input weighting optimization for PID controllers based on the adaptive tabu search, *Proc. IEEE TENCON 2004*, Vol.4, 2004, pp. 451-454.

[15] N. Sriyingyong, and K. Attakitmongcol, Wavelet-based audio watermarking using adaptive tabu search, *Proc. Wireless Pervasive Computing, 2006 1st International Symposium on*, pp. 1-5.

[16] X. Haidong and L. Jianhua, Partition of computing granularity of security data fusion under high performance computing environment, *WSEAS Transactions on Computers*, Vol.5, No.1, 2006, pp. 61-68.

[17] L. Phillip and R. Bart, Problem partitioning in Hybrid Genetic Algorithms, *WSEAS Transactions on Systems*, Vol.6, No.2, 2007, pp. 395-399.

[18] R-B. Patel, M. Nikos and K. Garq, Mobile agent location management in global networks, *WSEAS Transactions on Computers*, Vol.4, No.7, 2005, pp. 697-710.

[19] R.B. Patel, and M. Nikos, Locating mobile agents in a wide distributed system: A dynamic approach, *WSEAS Transactions on Mathematics*, Vol.4, No.3, 2005, pp. 246-259.