Performance Assessment of Search Management Agent under Asymmetrical Problems and Control Design Applications

Jukkrit Kluabwang[1] Deacha Puangdownreong[2] and Sarawut Sujitjorn[1*]
[1]School of Electrical Engineering, Institute of Engineering
Suranaree University of Technology
Nakhon 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: - The article presents the performance evaluation of the management agent (MA) containing the adaptive tabu search (ATS) as its search core. In particular, asymmetrical surface optimization problems have been considered. It has been found that symmetrical property of the problems has a significant effect on search performance of the ATS, but MA(ATS). As an average, the MA(ATS) is about 2 times faster than the ATS under both symmetrical and asymmetrical problems. The article also gives reviews on the ATS and the MA(ATS) algorithms. An application on controller design for a coupled system with three-degree-of-freedom is also elaborated.

Key-Words: - asymmetrical surface optimization problems, adaptive tabu search, management agent

1 Introduction

In recent years, intelligent search techniques have become major approaches for solving various optimization problems. Among those, tabu search (TS) has been widely applied for combinatorial optimization problems [1,2]. The TS consists of two main strategies: intensification and diversification [3,4]. Since the TS is very flexible to use for various problems, Sujitjorn et. al. [5,6] have launched a modified version of the TS namely adaptive tabu search (ATS). The ATS possesses the backtracking (BT) and adaptive radius (AR) mechanisms. The former is regarded as a diversification strategy, while the later as an intensification one. The convergence proof and performance assessment of the ATS can be found in [5,6]. Readers can find ATS's applications, such as electrical system protection in [7], system identification in [8,9], control synthesis in [10,11], and acoustic signal processing in [12]. In 2008, the authors proposed a management agent for search algorithms called shortly as $\mathrm{MA}(\cdot)$. $(\cdot)$ can be replaced by any search algorithms such as ATS, simulated annealing, genetic algorithms, etc. The MA contains three major mechanisms: partitioning mechanism (PM), discarding mechanism (DM), and sequencing mechanism (SM).

This article reports our computing results in order to assess the effects of symmetry of the problems on search performances of the MA(ATS) , i.e. the MA having the ATS as its search core. Section 2 of the paper reviews the ATS and the MA. Sections 3 and 4 present our problem formulation, results and discussions, respectively. An application on control system design appears in section 5. Conclusion is in section 6.

2 Reviews of Algorithms

2.1 Adaptive Tabu Search (ATS)

The ATS [5,6] preserves the main search mechanisms of the generic TS [3,4]. Regarding this, the tabu list (TL), the aspiration criteria (AC), and the termination criteria (TC) are effective. The additional mechanisms are the backtracking (BT), and the adaptive search radius (AR), respectively. The BT is for escaping the lock by local solutions via looking backward the TL for some previous solutions and adopting them for generating new search directions. The AR subsequently reduces the search radii in search of a high fidelity solution. The proofs elaborated in [5,6] confirm the ATS's rapid search performance, and convergence.

The following description summarizes the algorithms.

<u>STEP 0</u> Initialization: search radius R, TL, $count$, $count_{\max}$, TC, BT, and AR.
<u>STEP 1</u> Random an initial solution S_0. Set S_0 as a local solution. Generate the neighbourhood of radius R around S_0. Keep N solutions of the neighbourhood in the set X.
<u>STEP 2</u> Evaluate the cost, C, of each solution belonging to X. The solution having the minimum cost is designated as S_1.
If $C(S_1) < C(S_0)$, move S_0 into TL and assign $S_0 = S_1$, otherwise move S_1 into TL.
<u>STEP 3</u> If the solution cycling occurs, invoke the BT mechanism.
<u>STEP 4</u> Exit with S_0 as the best solution if the TC are met.
<u>STEP 5</u> If the current solution S_0 in the vicinity of the local or the global minimum, invoke the AR mechanism. Update count, and goto STEP 1.

Interested readers can find the useful recommendations with examples for the benefit of employing the algorithms from [5-12].

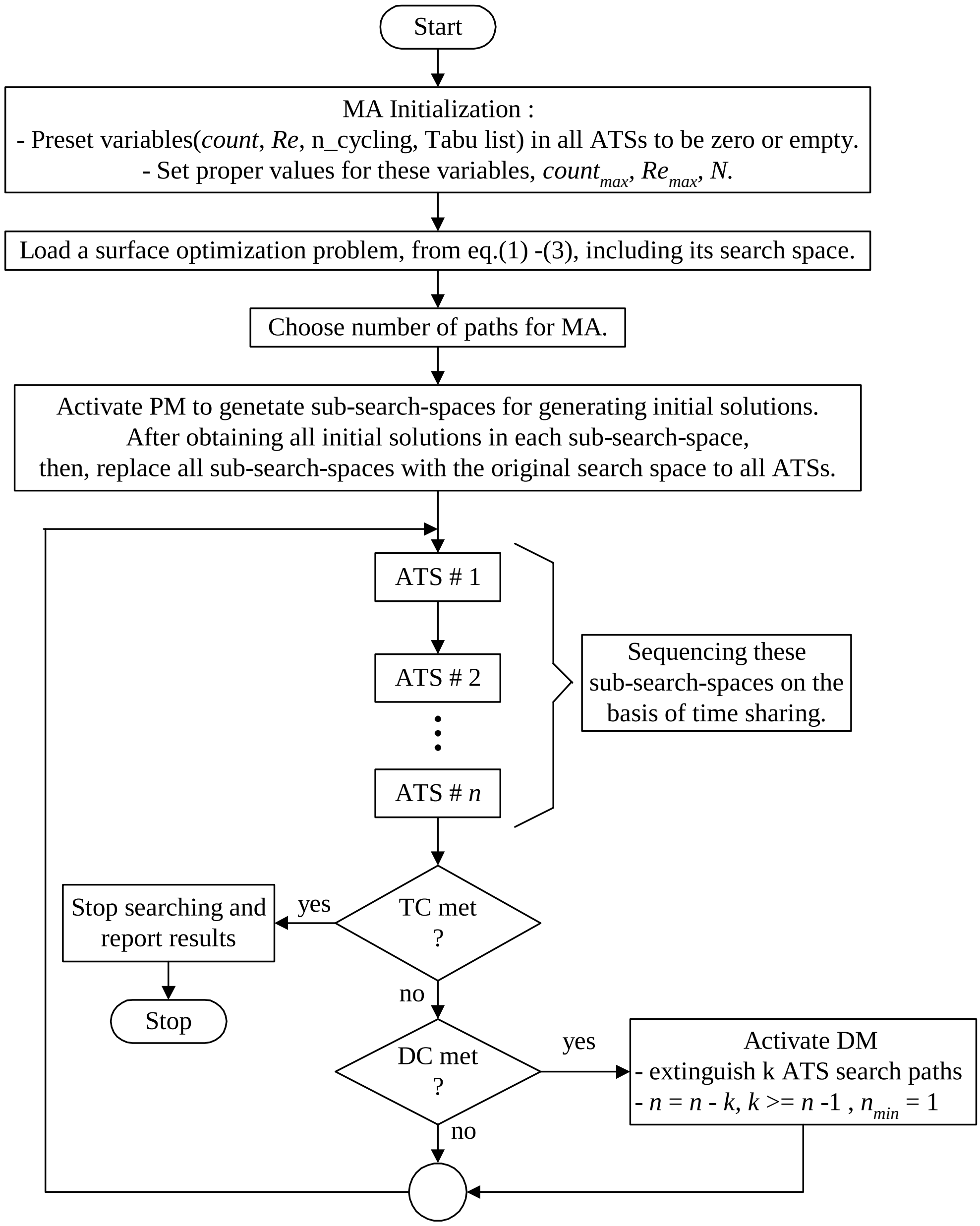


Fig. 1 Flow diagram of the MA. [13,14]

2.2 Management Agent for Search

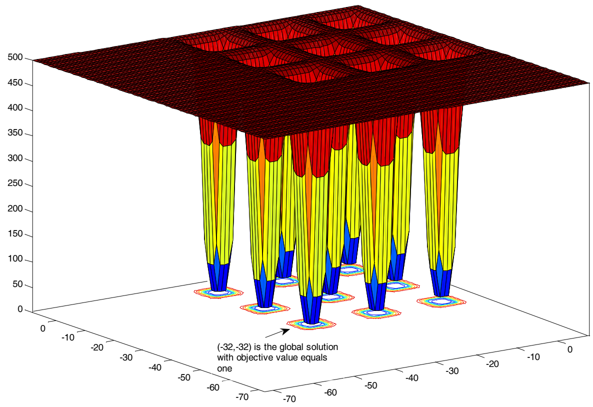
Recently, the authors have launched the search management agent (MA) flexible enough to be used with any search algorithms, particularly the meta-heuristic ones [13,14]. The MA does not intrude the main algorithms used as the search core unit. So, the convergence property of the search core is always preserved.

The MA consisted of 3 main strategies namely the partitioning, the sequencing, and the discarding mechanisms (PM, SM, and DM), respectively. The PM serves to split an entire search space into many sub-search-spaces. It also initiates searches correspondingly to those sub-search-spaces. After the search initiations, the search boundaries, once arisen from the PM, are immediately removed. The search space is then the original one containing several to many search paths. The PM concept is not new, and has been applied to data fusion [15] and genetic algorithms [16], for instance. The SM on a single CPU platform is crucial because it organizes the search units to perform searches in a sequential manner without a conflict. The SM can be regarded as a time-sharing strategy. As soon as all search paths finish their searches at the k^{th} round, the DM interrupts all search paths, and interrogates them for the current solutions with associated cost values. The DM, apparently behaves similarly to the location management algorithms [17,18], is an important tactic to accelerate the search. It determines the chance of success of each search path, and eliminates those considered to have less chances. For instance, some search paths that are locked by local minima are rapidly deleted from the search plateau. The DM is activated upon the discarding criteria (DC). The DC is simply the measurement of errors between the cost of the global solution and that of the current one. The concept of MA becomes clearer with the flow diagram reproduced from [13,14] and illustrated in Fig. 1. Referring to the flow diagram, the search core implemented is the ATS. In addition to this, the algorithms MA(ATS) can be summarized as follows:

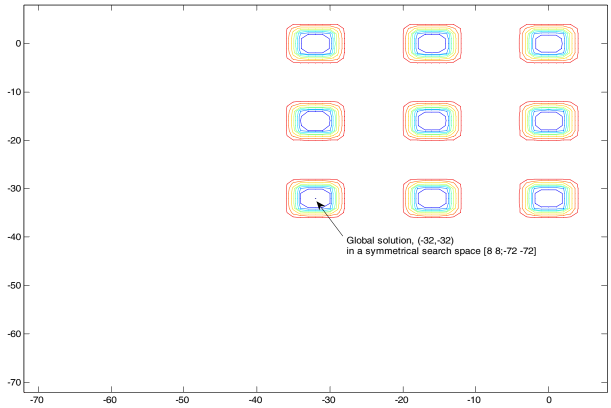
<u>STEP 0</u> Define search spaces and search paths. Generate initial solutions (random or well-educated guess). Activate PM.
<u>STEP 1</u> Invoke SM and ATS of multiple paths (or other search algorithms).
<u>STEP 2</u> Exit with the best solution if the TC are met, otherwise invoke DM.
<u>STEP 3</u> Update counter and goto STEP 1.

3 Performance Evaluations

Our evaluation was conducted against the Shekel's Foxholes function (see Figs. 2 and 3) expressed by equation (1). The global solution sits on (-32,-32) coordinates. Its cost value is 1. Fig. 4 shows the top view locations of the global solutions for the symmetrical and asymmetrical problems. Table 1 summarrizes the corresponding search spaces. Our MATLAB™ codes were run on a Pentium IV 2.4 GHz 640 Mbytes of SD-RAM.



(a)



(b)

Fig. 2 Shekel's Foxholes surface under symmetrical problem (a) 3-D and (b) 2-D.

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

in which

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

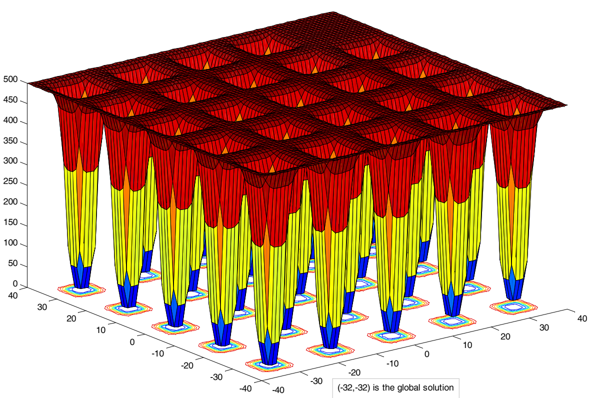
Our tests were also conducted against the Bohachevsky's, and the Rastrigin's functions. They are represented by the expressions (2) and (3), respectively, with their corresponding surface plots shown in Figs. 5 and 6. For asymmetrical problems, the global solutions are eccentric in the similar manner to what shown in Fig. 4(b), otherwise symmetrical.

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

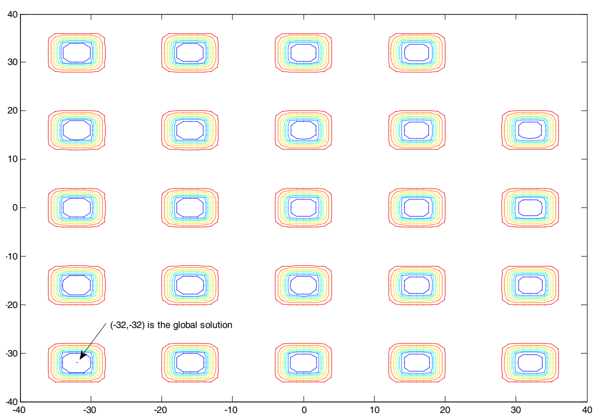
$$f(x,y) = x^2 + y^2 - 10\cos(2\pi x) - 10\cos(2\pi y) + 20 \quad (3).$$

Table 1 Corresponding search spaces.

Types	search spaces	
	BF & RF	SF
Sym.	[2 2;-2 -2]	[8 8;-72 -72]
Asym.	[3 3;-1 -1]	[40 40;-40 -40]

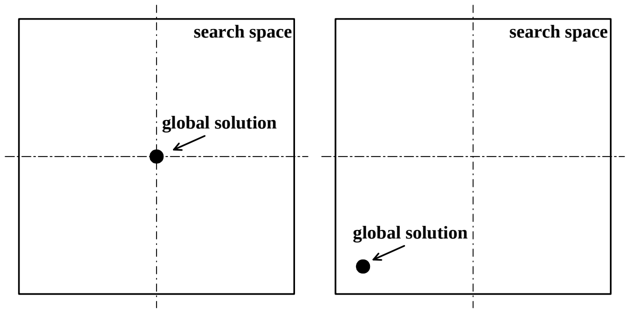


(a)



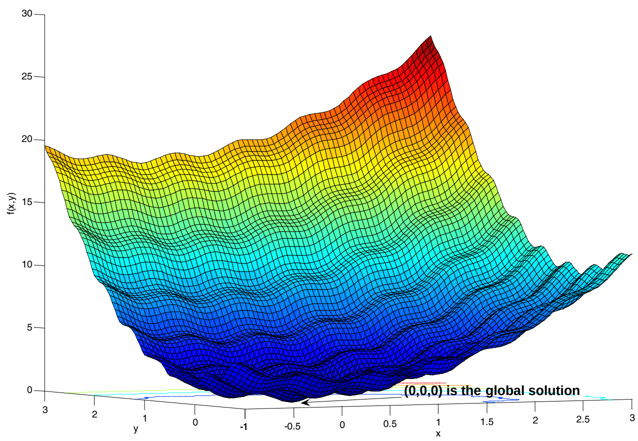
(b)

Fig. 3 Shekel's Foxholes surface under asymmetrical problem (a) 3-D and (b) 2-D.

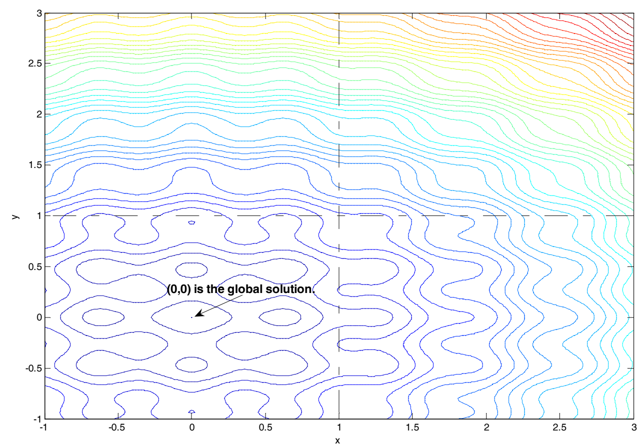


(a) (b)

Fig. 4 Top view locations of the global solutions (a) symmetrical and (b) asymmetrical problems.

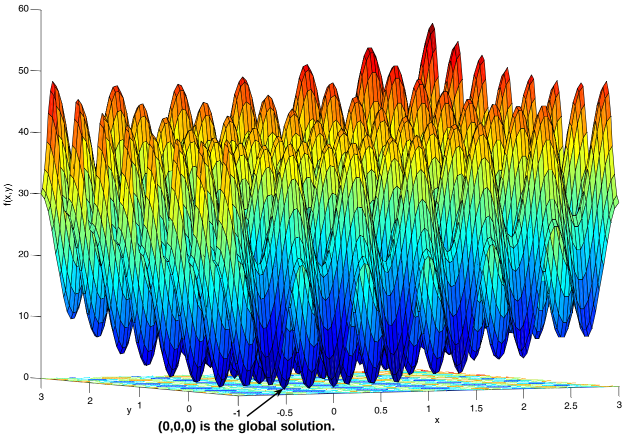


(a)

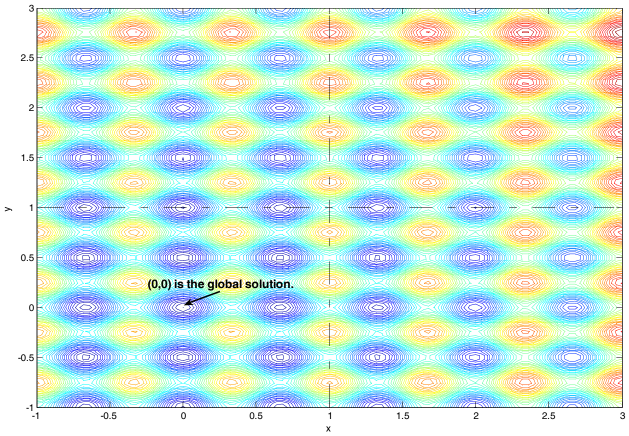


(b)

Fig. 5 Bohachevsky's surface under asymmetrical problem (a) 3-D and (b) 2-D.

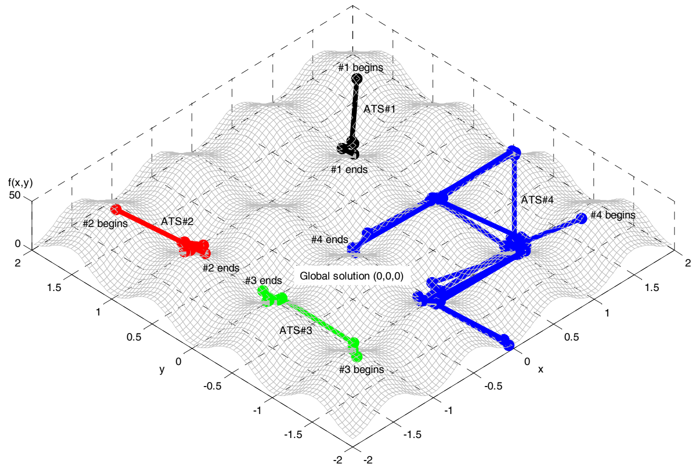


(a)

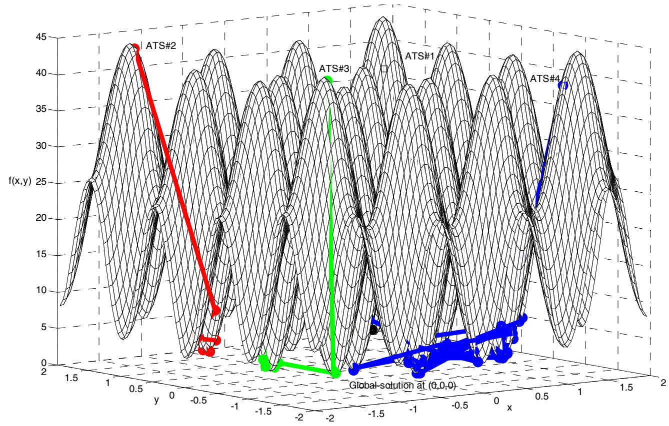


(b)

Fig. 6 Rastrigin's surface under asymmetrical problem (a) 3-D and (b) 2-D.



(a)



(b)

Fig. 7 Search trajectories (4 paths) on Rastrigin's surface (a) bird's eye view and (b) front view.

4 Results and Discussions

Fig. 7 demonstrates the case of 4 search trajectories on the Rastrigin's surface. The ATS#1 and #2 are discarded on the 15th iteration, subsequently ATS#3 on the 25th iteration. The ATS#4 spent 14.184 seconds to hit the global optimum by the 372nd iteration and at (4.08×10^{-6}, -3.71×10^{-6}), with the cost value of 0.578×10^{-8}. Moreover, Figs.8-10 illustrate the traces of search trajectories (64 paths) on the three contours of both symmetrical and asymmetrical problems.

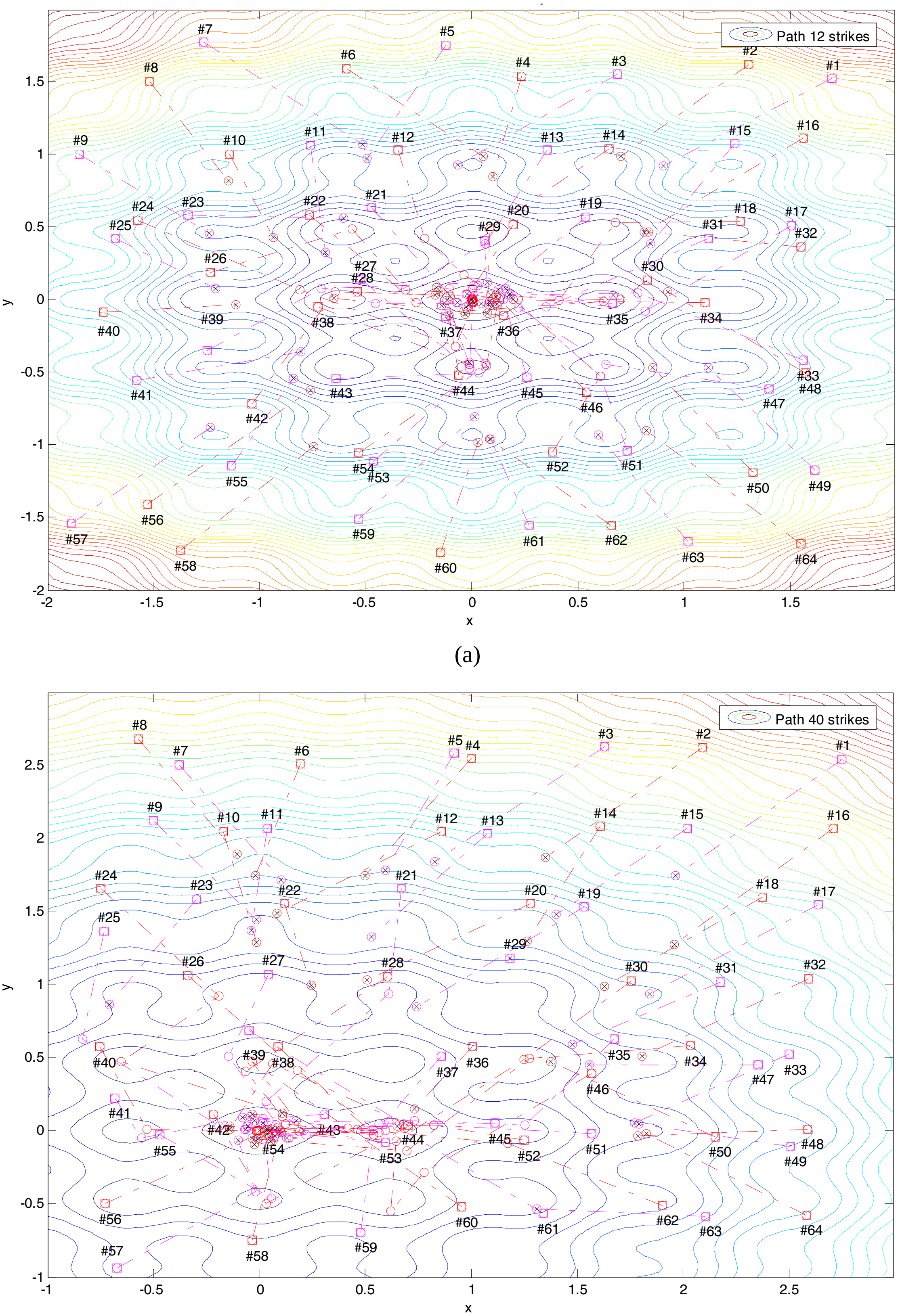
Table 2 Average search time.

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

Referring to the average search time in seconds summarized by table 2, the search time consumed by the ATS is longer than that used by the MA(ATS) for each case. Under asymmetrical problems, the ATS consumes longer search time than that used under symmetrical problems. For the MA(ATS), the following details are clarified. With the BF, 66.67% of MA(ATS), i.e., 2, 16, 32 and 64 paths, reach the global solution faster under symmetrical problems than under asymmetrical ones. Similar situations occur with the RF and the SF. 83.33% of MA(ATS), i.e., 2, 8, 16, 32 and 64 paths, perform faster searches with the symmetrical RF. 50% of MA(ATS), i.e., 16, 32 and 64 paths, perform faster searches with the symmetrical SF.

$$\text{speed up ratios} = \frac{\text{average search time by ATS}}{\text{average search time by MA(ATS)}} \quad (4)$$

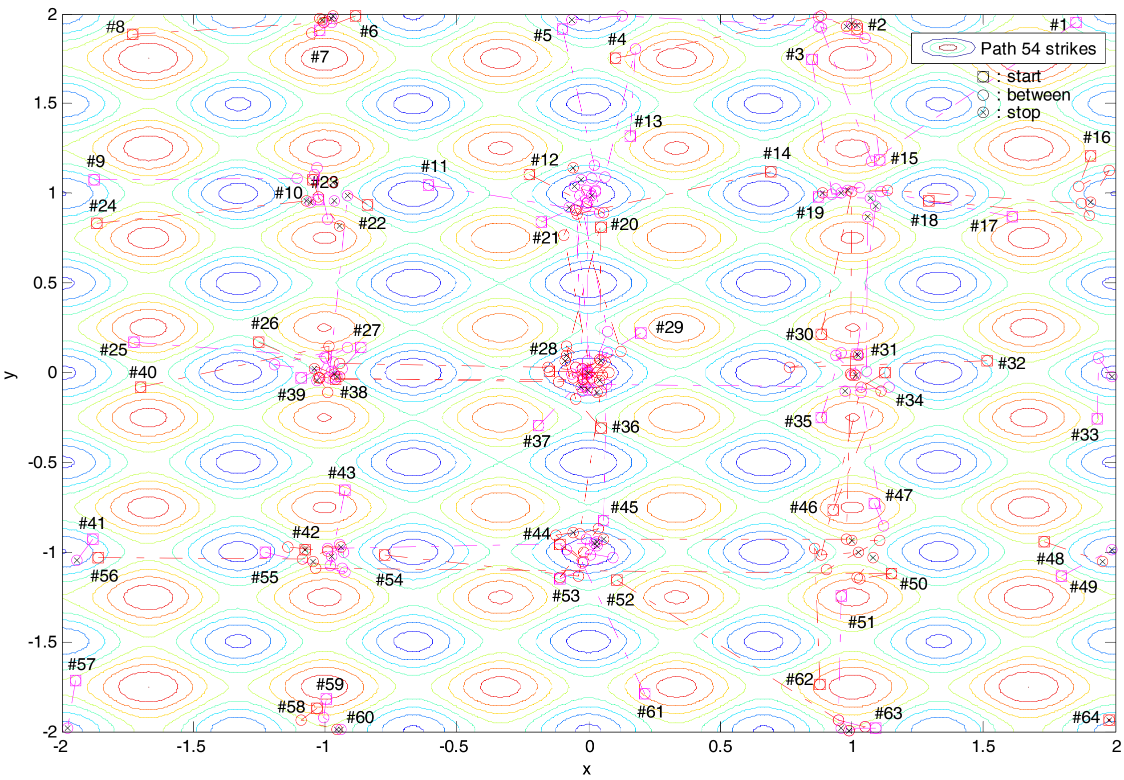
The speed up ratios according to (4) are calculated and summarized by the bar graphs shown in Fig. 11.



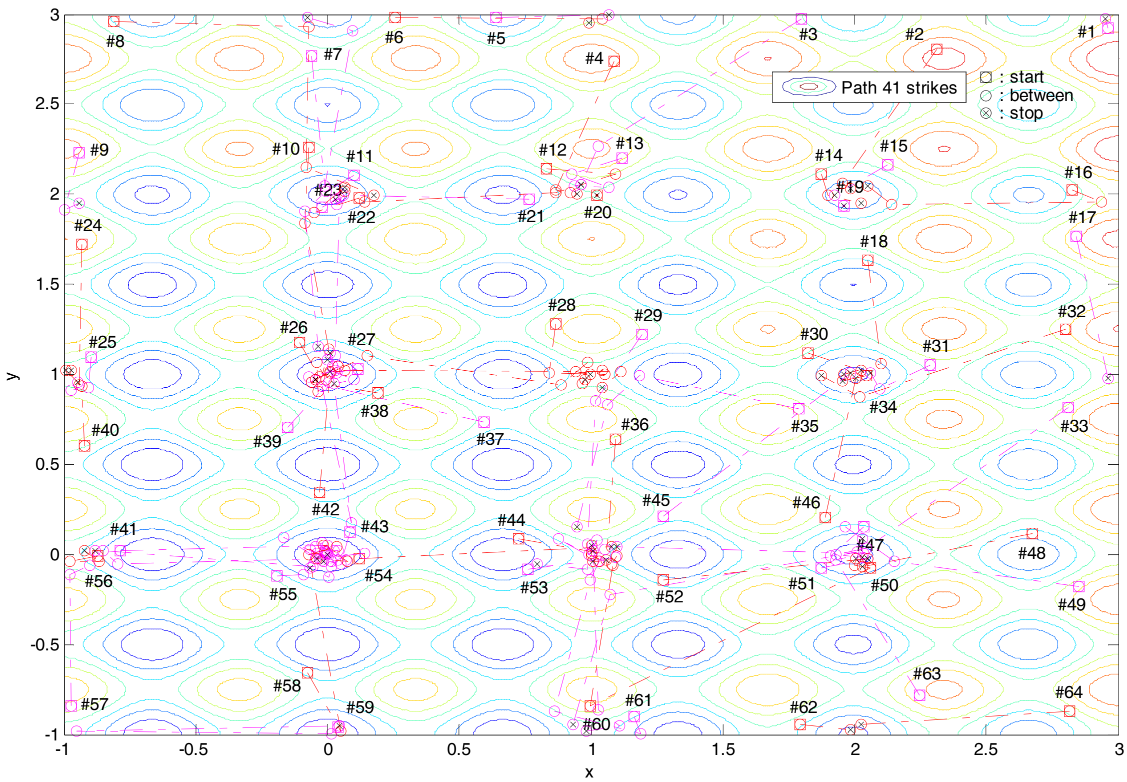
(a)

(b)

Fig. 8 Traces of search paths (64 paths) on Bohachevsky's contour (a) symmetrical case and (b) asymmetrical case.

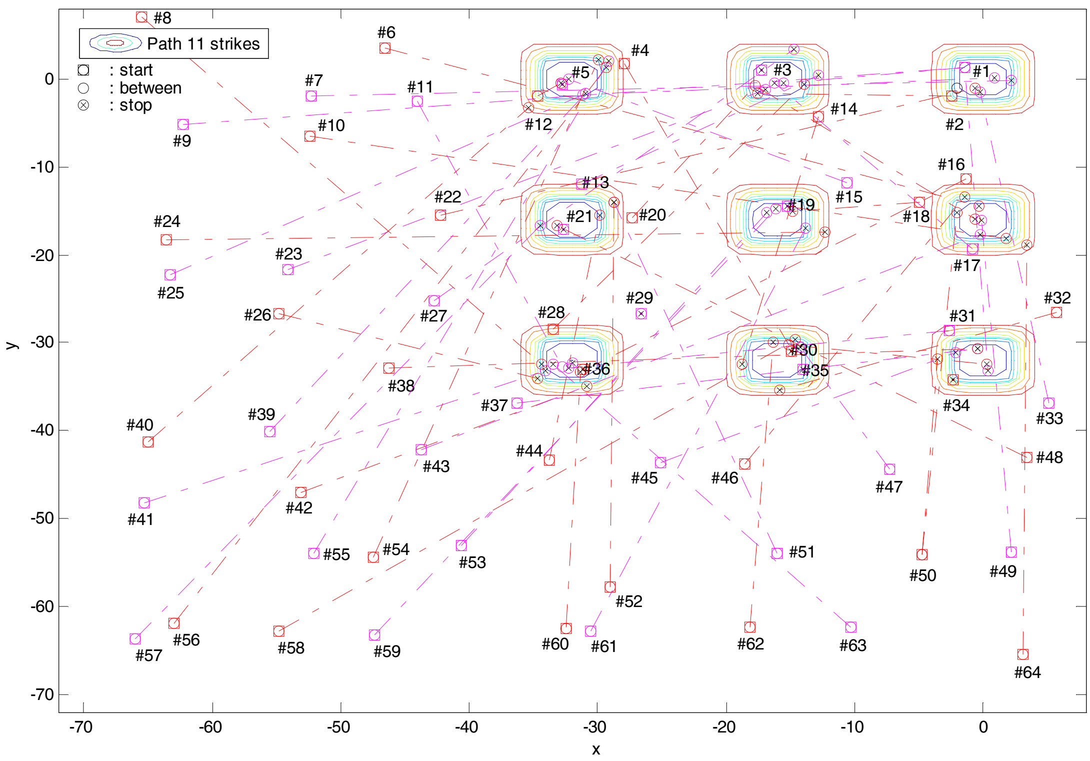


(a)

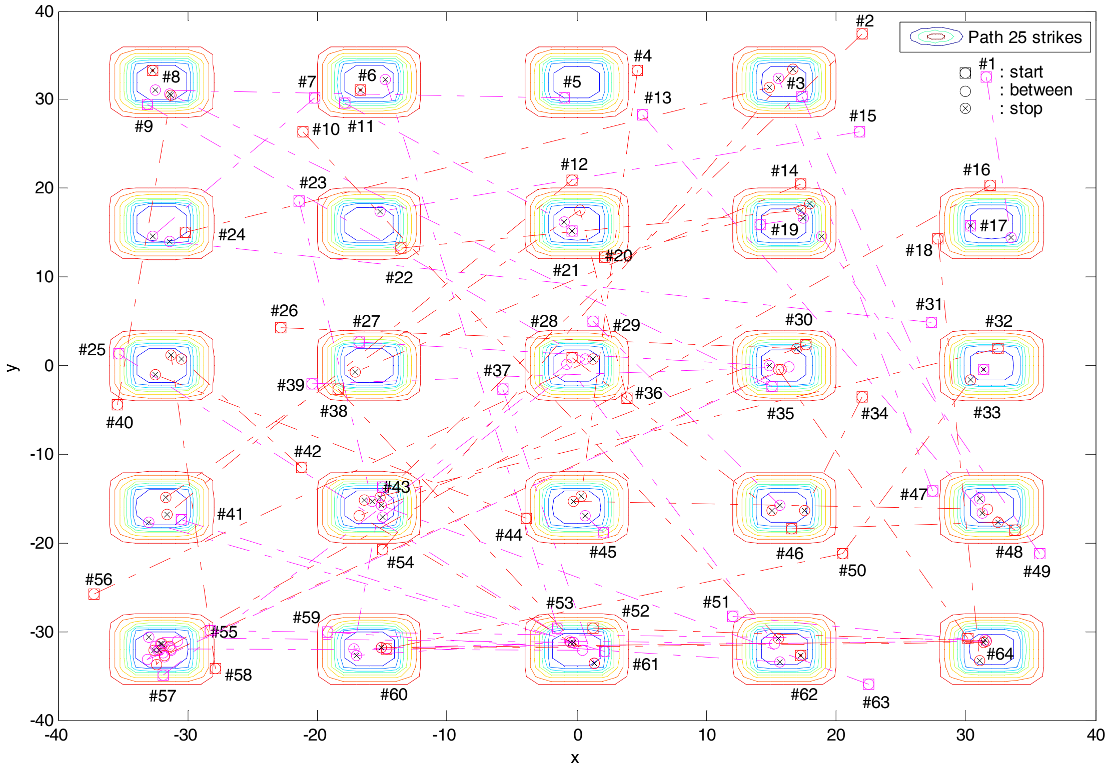


(b)

Fig. 9 Traces of search paths (64 paths) on Rastrigin's contour (a) symmetrical case and (b) asymmetrical case.



(a)



(b)

Fig. 10 Traces of search paths (64 paths) on Shekel's foxhole contour (a) symmetrical case and (b) asymmetrical case.

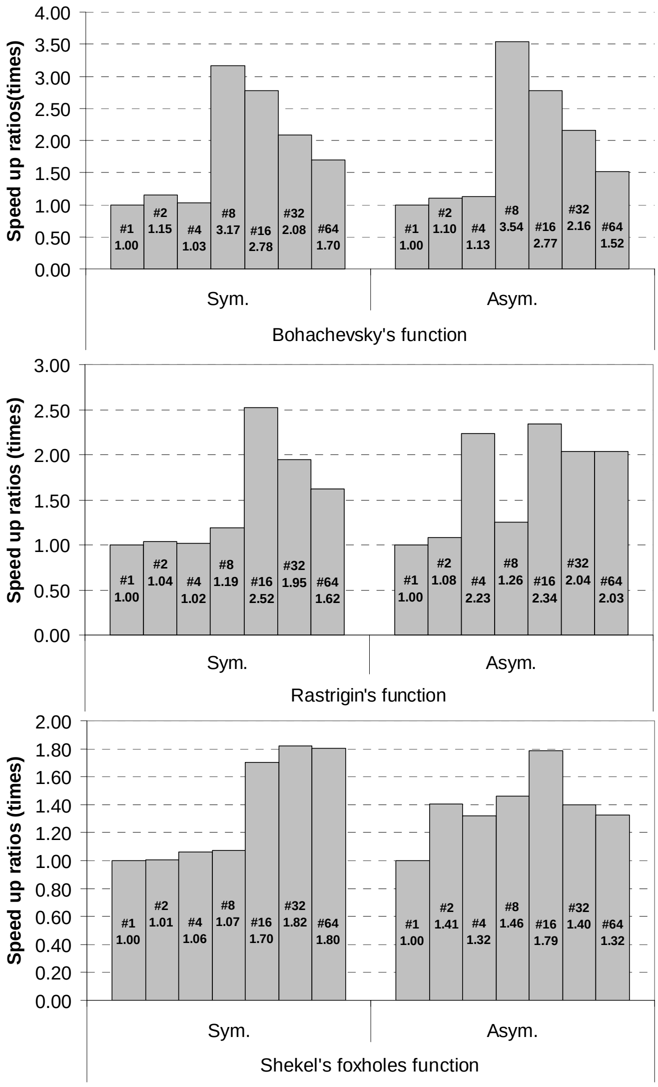


Fig. 11 Speed up ratios of the MA(ATS).

The top window of Fig. 11 shows the results of the MA(ATS) searching through the Bohachevsky's surface. The numeric figures embedded in each bar, for instance, (#1, 1.00) means that it is of the case single path ATS, thus having the speed up ratio equal to 1. As another example, consider the asymmetrical case appearing in the same window, the bar with (#8, 3.54) means the case of 8-path ATS having the speed up ratios of 3.54 times that of the single path ATS. The similar explanation is applied to the rest of the results. Noticeably, the speed up ratios of the symmetrical and the asymmetrical problems bear very similar figures. This means that the symmetrical property of the problems does not greatly affect the search performance of the MA(ATS). To give an average figure, the MA(ATS) performs search in sequential manner about 1.713 times faster than the ATS does. Average speed up ratios of 2,4,...,64 paths are summarized by table 3. The figures indicate that the MA(ATS) of 16 paths provides the fastest search.

Table 3 Average speed up ratios.

no. of paths	**2**	**4**	**8**	**16**	**32**	**64**
average speed up ratios (times)	**1.13**	**1.30**	**1.95**	**2.32**	**1.91**	**1.67**

5 Applications

Control engineering applications are discussed herein. The problem is to fine tune an existing control system such that optimally seeked controllers could be obtained. The control system of the Illinois Roadway Simulator (IRS) system [19,20] in which the yaw rate control (see Fig. 12) is of prime objective.

Fig. 12 Car reference frame.

The system can be represented by the block diagram shown in Fig. 13. Referring to Fig.13, two s-polynomial ratios namely G and G_d represent the vehicle dynamics, and expressed by

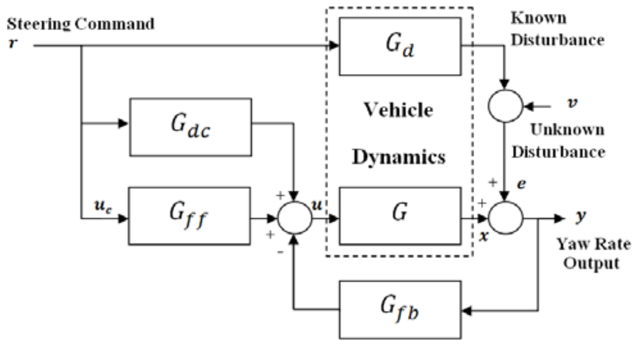


Fig. 13 Block diagram representing the yaw rate control.

$$G = \frac{2.802\times10^4 s + 2.306\times10^5}{1.5s^4 + 94.43s^3 + 7.308\times10^4 s^2 + 1.183\times10^5 s + 5.361\times10^5} \quad (5)$$

$$G_d = \frac{8.024\times10^4 s + 6.710\times10^5}{1.3s^4 + 53.250s^3 + 1.943\times10^3 s^2 + 2.693\times10^4 s + 1.162\times10^5} \quad (6).$$

The control system has three degrees-of-freedom, and possesses the following controllers:

$$G_{dc} = \frac{-1.2036\times10^{5}s^{5} - 8.584\times10^{6}s^{4} - 6.497\times10^{8}s^{3} -}{3.6426\times10^{4}s^{5} + 1.792\times10^{6}s^{4} + 6.671\times10^{7}s^{3} +}\cdots$$
$$\cdots\frac{-1.44\times10^{10}s^{2} - 1.224\times10^{11}s - 3.598\times10^{11}}{+1.202\times10^{9}s^{2} + 9.464\times10^{9}s + 2.679\times10^{10}} \quad (7)$$

$$G_{ff} = \frac{103s^{2} + 1.8542\times10^{4}s + 8.34385\times10^{5}}{s^{3} + 155.3s^{2} + 3.727\times10^{3}s + 2.0714\times10^{4}} \quad (8)$$

$$G_{fb} = \frac{-5.181s^{3} + 49.98s^{2} + 5.294\times10^{3}s + 4.8142\times10^{4}}{s^{3} + 155.3s^{2} + 3.727\times10^{3}s + 2.0714\times10^{4}} \quad (9).$$

The transfer function $G_{irs} = \left.\frac{Y(s)}{R(s)}\right|_{V(s)=0}$ can be written as

$$G_{irs} = \frac{3.075\times10^{11}s^{23} + 1.721\times10^{14}s^{22} + 4.525\times10^{16}s^{21} +}{1.598\times10^{5}s^{27} + 9.423\times10^{7}s^{26} + 2.671\times10^{10}s^{25} +}\cdots$$

$$\cdots\frac{7.624\times10^{18}s^{20} + 9.327\times10^{20}s^{19} + 8.798\times10^{22}s^{18} +}{4.957\times10^{12}s^{24} + 6.805\times10^{14}s^{23} + 7.331\times10^{16}s^{22}}\cdots$$

$$\cdots\frac{6.618\times10^{24}s^{17} + 4.055\times10^{26}s^{16} + 2.048\times10^{28}s^{15}}{6.423\times10^{18}s^{21} + 4.678\times10^{20}s^{20} + 2.873\times10^{22}s^{19}}\cdots$$

$$\cdots\frac{8.590\times10^{29}s^{14} + 3.006\times10^{31}s^{13} + 8.779\times10^{32}s^{12}}{1.503\times10^{24}s^{18} + 6.729\times10^{25}s^{17} + 2.590\times10^{27}s^{16}}\cdots$$

$$\cdots\frac{2.136\times10^{34}s^{11} + 4.307\times10^{35}s^{10} + 7.143\times10^{36}s^{9}}{8.580\times10^{28}s^{15} + 2.445\times10^{30}s^{14} + 5.979\times10^{31}s^{13} +}\cdots$$

$$\cdots\frac{9.645\times10^{37}s^{8} + 1.048\times10^{39}s^{7} + 9.037\times10^{39}s^{6}}{1.251\times10^{33}s^{12} + 2.225\times10^{34}s^{11} + 3.346\times10^{35}s^{10} +}\cdots$$

$$\cdots\frac{6.075\times10^{40}s^{5} + 3.107\times10^{41}s^{4} + 1.165\times10^{42}s^{3}}{4.219\times10^{36}s^{9} + 4.417\times10^{37}s^{8} + 3.797\times10^{38}s^{7} +}\cdots$$

$$\cdots\frac{3.016\times10^{42}s^{2} + 4.813\times10^{42}s + 3.565\times10^{42}}{2.643\times10^{39}s^{6} + 1.462\times10^{40}s^{5} + 6.265\times10^{40}s^{4} +}\cdots$$

$$\cdots\frac{}{2.001\times10^{41}s^{3} + 4.480\times10^{41}s^{2} + 6.265\times10^{41}s + 4.115\times10^{41}}$$

(10).

Fig. 14 shows the simulation results of the original control system. The yaw rate response due to unit-step input is quite smooth and contains a delay time of 0.005 sec. It is noticed that the system has the DC-gain of 8.6638 rad/sec.

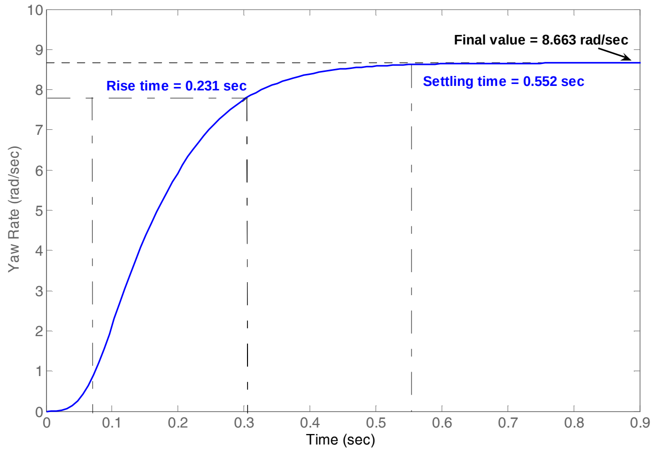


Fig. 14 Yaw rate response of the original system [19,20].

```
function J=obj(r1,r2,r3,r4,s1,s2,s3,s4,t1,t2,t3,a1,a2,a3,a4,a5,a6, b1, b2,…
         b3,b4,b5,b6)
% Declarations
G=tf([0.2802e5 2.3057e5],[1.5 94.4284 7.3078e3 1.1831e5 5.3616e5]);
Gd=tf([0.8024e5 6.7103e5],[1.3 53.2495 1.9425e3 2.6928e4 1.1617e5]);
Gdc=tf([b1 b2 b3 b4 b5 b6],[a1 a2 a3 a4 a5 a6]) ;
Gff=tf([t1 t2 t3],[r1 r2 r3 r4]);
Gfb=tf([s1 s2 s3 s4],[r1 r2 r3 r4]);
Girs=(Gd+G*Gdc+G*Gff)/(1+G*Gfb);
a=real(pole(Girs));                    % Poles of overall Closedloop System.
  if a<0                              % Only Stable System allowed come in.
      [y,t]=step(Girs);                      % Step response data recorded.
      if y(length(y))>8.66 & y(length(y))<8.67     % Only same DC gain.
        i=find(y>0.1*(y(length(y))-y(1))&y<0.9*(y(length(y))-y(1)));
        tr=t(i(length(i)))-t(i(1));
        trj=(tr*100)/0.23096393759434;                        % Find trj.
        Mpj=100*(max(y)-y(length(y)))/y(length(y));          % Find Mpj.
        for i=1:length(y)                                     % Find tsj.
            y(length(y)+1)=y(length(y));
            dy(i)=y(i+1)-y(i);
        end
        i=find(~(dy<1e-3));
        ts=t(i(length(i)))-t(i(1));
        tsj=(ts*100)/(0.23096393759434*5);
        if trj<100                 % Constraint 1, trj must be less than 100.
        trj=trj;                      % Constraint 1, otherwise trj = 1,000.
        else
        trj=1000;
        end
        if Mpj<5                   % Constraint 2, Mpj must be less than 5.
        Mpj=1;                                % Constraint 2, then Mpj = 1.
        else                          % Constraint 2, otherwise Mpj = 1,000.
        Mpj=1000;
        end
        if tsj<5*trj                      % Constraint 3, tsj must be < 5*trj.
        tsj=1;                                   % Constraint 3, then tsj = 1.
        else                             % Constraint 3, otherwise tsj = 1,000.
        tsj=1000;
        end
        J=trj+Mpj+tsj;                              % Objective function, J.
    else
        J=1000;
    end
 else
    J=1000;
 end
return  % Note that the original system has the objective function J = 102.
```

Fig. 15 Objective function coding.

To obtain 3 controllers that render an optimum response is not a trivial task. In conventional practice, the controllers have to be designed one-by-one with an expectation of having a satisfactory response of the complete system. Synthesis all the controllers simultaneously is possible via an intelligent search method. This control synthesis can be formulated as a combinatorial optimization problem to minimize J, where J is a combined objective function namely

$$J = t_{rj} + M_{pj} + t_{sj} \qquad (11).$$

t_{rj}, M_{pj} and t_{sj} in the equation (11) are normalized, hence unitless, and stand for rise time, overshoot and settling time, respectively. In our software implementation, the penalty concept of using 1,000 as a magnification factor is applied to t_r, M_p and t_s in order to obtain a fine quality solution. The step performance specfications are t_r < 0.231 sec, M_p < 5 %, and t_s defined as the first time the response stops oscillating completely. Moreover, it is mandated that the system maintains the DC-gain. Defining such performance specifications, the MA(ATS) is expected to produce three polynomial controllers optimally rendering a better response than that of the original. Fig. 15 provides the code list of the objective function, J.

Our control problem assumes the nominal plant models whose parameters are fixed. The three controllers namely G_{dc}, G_{ff} and G_{fb} are rewritten as follows:

$$G_{dc} = \frac{b_1 s^5 + b_2 s^4 + b_3 s^3 + b_4 s^2 + b_5 s + b_6}{a_1 s^5 + a_2 s^4 + a_3 s^3 + a_4 s^2 + a_5 s + a_6} \qquad (12)$$

$$G_{ff} = \frac{p_1 s^2 + p_2 s + p_3}{r_1 s^3 + r_2 s^2 + r_3 s + r_4} \qquad (13)$$

$$G_{fb} = \frac{q_1 s^2 + q_2 s + q_3}{r_1 s^3 + r_2 s^2 + r_3 s + r_4} \qquad (14)$$

whose parameter r's, q's, p's, a's and b's are to be searched. Regarding to the original design, these 23 parameters are summarized in table 4. Some initial trials referred to as pre-search process were carried out in order to establish search ranges of all influential parameters. The TC for the pre-search process was either J<50 or $count_{max}$=20. The parameters were searched one-by-one. During the pre-search, one parameter was searched while the others were kept constant as their original values in order to investigate its influence on the system time-domain performance, stability as well as max-min range for search. Referring to the table 4, the parameters shown in bold characters are those influencing the performance while being stability insensitive. Some shown in ordinary characters do not have much influence on the performance, but they are stability sensitive. Table 4 also shows the max-min ranges of the parameters obtained from the pre-search. These ranges are further used as search spaces for the corresponding parameters. Fig. 16 depicts the 13 influential parameters affecting the system performance in terms of the percent reduction in rise time. These parameters printed in bold in the table 4 are to be tuned via our MA(ATS).

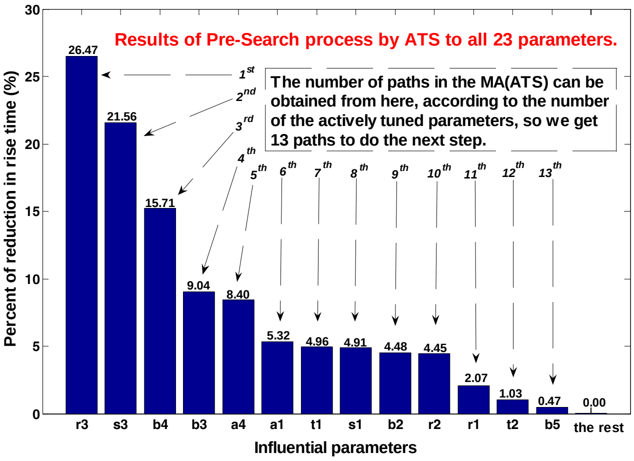


Fig. 16 Results of the pre-search process.

The search for these 13 parameters are distributed among 13 ATS paths with the corresponding search spaces shown by bold characters in the table 4. At the beginning, all ATS paths employ different initial solutions and the following heuristically set search parameters: N_neighbour =5, R=0.1, AR_1-if J<90 then R=0.5R, AR_2-if J<80 then R=0.1R, k^{th}_backward=5, N_re_max =5, max_count =100, TC if J<70 or max_count hit, DM_1-at 5^{th} iteration reduces 13 to 7 paths, DM_2-at 10^{th} iteration reduces 7 to 3 paths and DM_3-at 15^{th} iteration reduces 3 to 1 paths.

Referring to the table 4, 13 ATS paths perform a mixed type of multiple points single strategy (MPSS) and single point single strategy (SPSS) searches. The PM of the MA(ATS) designates the search in the following manner:

- within the 1^{st} neighbourhood, ATS#1 to #13 perform independently search for the individual parameter r_1, r_2, r_3, q_1, q_3, p_1, p_2, a_1, a_4, b_2, b_3, b_4, and

Table 4 Summary of controllers 's parameters and corresponding search spaces.

no.		parameters of original controllers	13 parameters obtained from MA(ATS)	max-min ranges of parameters	ATS (at PM)
1	**r1**	**1.0000**	**0.9548**	**[1 0.1]**	**ATS#1**
2	**r2**	**155.3000**	**153.1941**	**[155.3 0.1]**	**ATS#2**
3	**r3**	**3.727×10^3**	**2.541×10^3**	**[3.727×10^3 -3.727×10^3]**	**ATS#3**
4	r4	2.0714×10^4	2.0714×10^4	[2.0714×10^4 2.0714×10^4]	
5	**q1**	**-5.1810**	**-5.3916**	**[-5.1810 -10.1810]**	**ATS#4**
6	q2	49.98	49.98	[49.98 49.98]	
7	**q3**	**5.294×10^3**	**4.1739×10^3**	**[5.294×10^3 -5.294×10^3]**	**ATS#5**
8	q4	4.8142×10^4	4.8142×10^4	[4.8142×10^4 4.8142×10^4]	
9	**p1**	**103**	**87.9221**	**[103 -103]**	**ATS#6**
10	**p2**	**1.8542×10^4**	**1.5448×10^4**	**[1.8542×10^4 -1.8542×10^4]**	**ATS#7**
11	p3	8.34385×10^5	8.34385×10^5	[8.34385×10^5 8.34385×10^5]	
12	**a1**	**3.6426×10^4**	**3.3336×10^4**	**[3.6426×10^4 0]**	**ATS#8**
13	a2	1.7918×10^6	1.7918×10^6	[1.7918×10^6 1.7918×10^6]	
14	a3	6.6708×10^7	6.6708×10^7	[6.6708×10^7 6.6708×10^7]	
15	**a4**	**1.2024×10^9**	**1.0774×10^9**	**[1.2024×10^9 -1.2024×10^9]**	**ATS#9**
16	a5	9.4644×10^9	9.4644×10^9	[9.4644×10^9 9.4644×10^9]	
17	a6	2.6788×10^{10}	2.6788×10^{10}	[2.6788×10^{10} 2.6788×10^{10}]	
18	b1	-1.2036×10^5	-1.2036×10^5	[-1.2036×10^5 -1.2036×10^5]	
19	**b2**	**-8.5835×10^6**	**-8.8750×10^6**	**[-8.5835×10^6 -16.5835×10^6]**	**ATS#10**
20	**b3**	**-6.4974×10^8**	**-6.5163×10^8**	**[-6.4974×10^8 -7.4974×10^8]**	**ATS#11**
21	**b4**	**-1.4397×10^{10}**	**-1.4516×10^{10}**	**[-1.4397×10^{10} -2.4397×10^{10}]**	**ATS#12**
22	**b5**	**-1.2241×10^{11}**	**-1.3084×10^{11}**	**[-1.2241×10^{11} -2.2241×10^{11}]**	**ATS#13**
23	b6	-3.5978×10^{11}	-3.5978×10^{11}	[-3.5978×10^{11} -3.5978×10^{11}]	

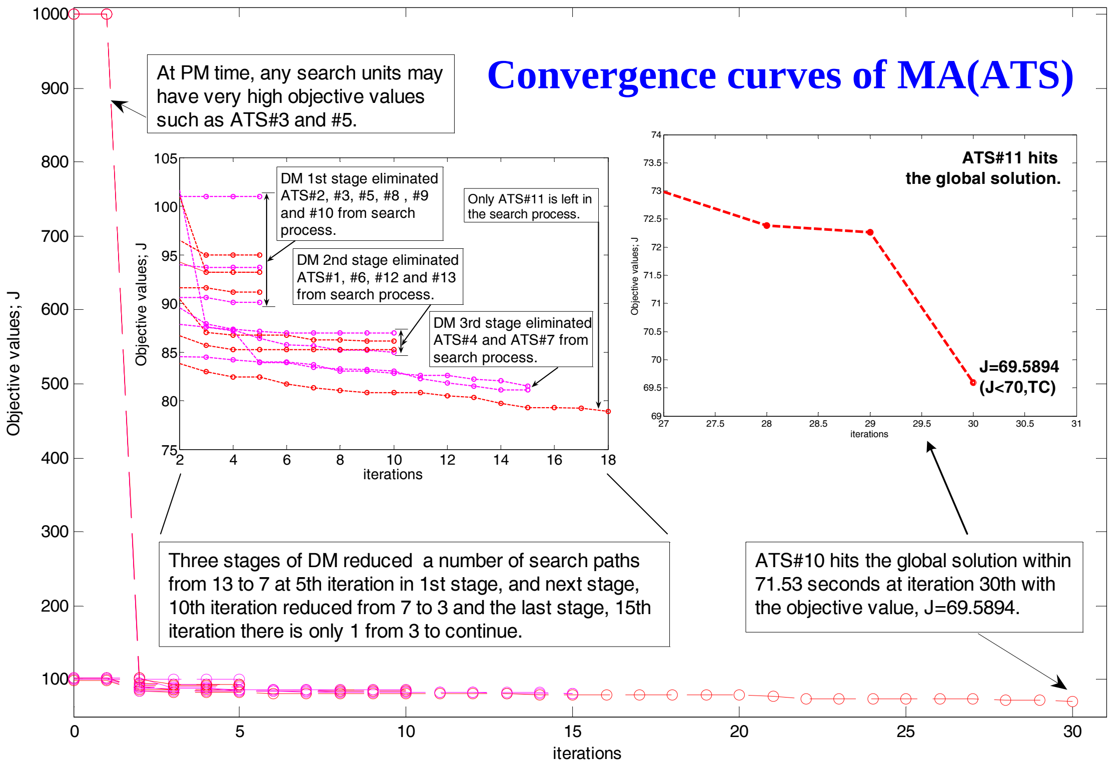


Fig. 17 Convergence curves of the control design problem.

b_5, respectively; during this search stage, the 22 parameters not being subjected to the search are kept constant as the original values; each ATS path would obtain its best solution to be used as an initial solution for a new search generation;

- after finishing the 1st neighbourhood search, all ATS paths with their corresponding initial solutions for the parameters are released to search freely for all 13 parameters.

The cost values monitored are presented as the convergence curves in Fig. 17. Among the 13 ATS paths, the 11th path succeeded in reaching the solutions with J=69.5894, and within 71.53 sec. The parameters obtained from search are shown in bold characters in the table 4. They result in the following controllers

$$G_{dc}^{*} = \frac{-1.203\times10^{5}s^{5} - 8.875\times10^{6}s^{4} - 6.516\times10^{8}s^{3} - }{3.333\times10^{4}s^{5} + 1.792\times10^{6}s^{4} + 6.671\times10^{7}s^{3} + }\cdots$$

$$\cdots\frac{1.451\times10^{10}s^{2} - 1.308\times10^{11}s - 3.598\times10^{11}}{+1.077\times10^{9}s^{2} + 9.464\times10^{9}s + 2.679\times10^{10}} \quad (15)$$

$$G_{ff}^{*} = \frac{87.922s^{2} + 1.854\times10^{4}s + 8.344\times10^{5}}{0.955s^{3} + 153.194s^{2} + 2.541\times10^{3}s + 2.071\times10^{4}} \quad (16)$$

$$G_{fb}^{*} = \frac{-5.391s^{3} + 49.98s^{2} + 4.174\times10^{3}s + 4.814\times10^{4}}{0.955s^{3} + 153.194s^{2} + 2.541\times10^{3}s + 2.071\times10^{4}} \quad (17).$$

The overall transfer function of the system having the controllers in (15)-(17) is expressed by

$$G_{irs}^{*} = \frac{-7.628\times10^{8}s^{24} - 1.877\times10^{11}s^{23} + 2.525\times10^{13}s^{22} + }{1.333\times10^{5}s^{27} + 8.060\times10^{7}s^{26} + 2.295\times10^{10}s^{25} + }\cdots$$

$$\cdots\frac{1.784\times10^{16}s^{21} + 3.947\times10^{18}s^{20} + 5.535\times10^{20}s^{19} + }{4.240\times10^{12}s^{24} + 5.772\times10^{14}s^{23} + 6.144\times10^{16}s^{22} + }\cdots$$

$$\cdots\frac{5.672\times10^{22}s^{18} + 4.504\times10^{24}s^{17} + 2.861\times10^{26}s^{16} + }{5.307\times10^{18}s^{21} + 3.801\times10^{20}s^{20} + 2.291\times10^{22}s^{19} + }\cdots$$

$$\cdots\frac{1.477\times10^{28}s^{15} + 6.267\times10^{29}s^{14} + 2.197\times10^{31}s^{13} + }{1.173\times10^{24}s^{18} + 5.137\times10^{25}s^{17} + 1.930\times10^{27}s^{16} + }\cdots$$

$$\cdots\frac{6.391\times10^{32}s^{12} + 1.544\times10^{34}s^{11} + 3.094\times10^{35}s^{10} + }{6.238\times10^{28}s^{15} + 1.734\times10^{30}s^{14} + 4.144\times10^{31}s^{13} + }\cdots$$

$$\cdots\frac{5.120\times10^{36}s^{9} + 6.947\times10^{37}s^{8} + 7.654\times10^{38}s^{7} + }{8.493\times10^{32}s^{12} + 1.488\times10^{34}s^{11} + 2.217\times10^{35}s^{10} + }\cdots$$

$$\cdots\frac{6.762\times10^{39}s^{6} + 4.705\times10^{40}s^{5} + 2.513\times10^{41}s^{4} + }{2.794\times10^{36}s^{9} + 2.953\times10^{37}s^{8} + 2.592\times10^{38}s^{7} + }\cdots$$

$$\cdots\frac{9.908\times10^{41}s^{3} + 2.709\times10^{42}s^{2} + 4.568\times10^{42}s + }{1.862\times10^{39}s^{6} + 1.074\times10^{40}s^{5} + 4.847\times10^{40}s^{4} + }\cdots$$

$$\cdots\frac{3.565\times10^{42}}{1.643\times10^{41}s^{3} + 3.922\times10^{41}s^{2} + 5.863\times10^{41}s + 4.115\times10^{41}} \quad (18).$$

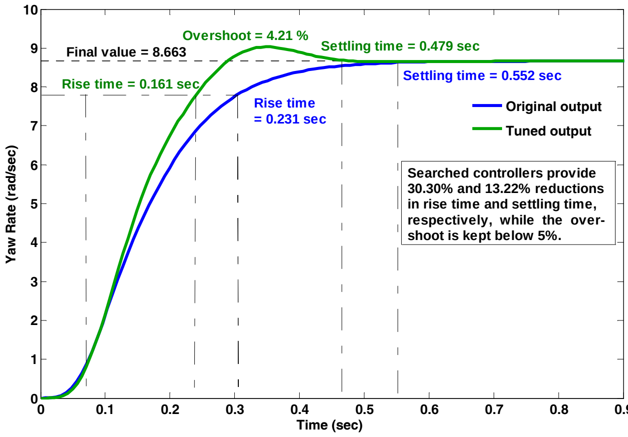


Fig. 18 Comparison of time-domain performances.

Fig. 18 shows the time-domain performances of the original and the proposed systems for comparison. It can be noticed that the new controllers provide much better performance in terms of 30.30% and 13.22% reductions in rise time and settling time, respectively, while the overshoot of 4.21% is acceptable.

6 Conclusions

The performance assessment of the management agent for search, in particular MA(ATS), has been presented. Computing experiments were conducted against symmetrical and asymmetrical problems. In this context, a symmetrical problem has its global solution situated at the center of the search space, while that of an asymmetrical problem is eccentric. Surface optimization problems have been utilized based on the Bohachevsky's, Rastrigin's, and Shekel's Foxholes functions, respectively. The derived speedup ratios indicated that (i) the MA(ATS) under symmetrical problems is 1.01-3.16 times faster than the ATS, and (ii) under asymmetrical problems the MA(ATS) is 1.08-3.54 times faster than the ATS. As an average, the MA(ATS) is about 1.713 times faster than the ATS. The MA(ATS) of 16 paths has the most rapid performance, i.e. 2.32 times the ATS, on surface optimization problems. From the speed up ratios of both cases, we can conclude that the MA(ATS) performs extremely well under symmetrical and asymmetrical problems, it is superior to the ATS, and the symmetrical property of the problems does not have a significant effect on search performance of the MA(ATS). The method has been successfully applied for a control design problem that requires three controllers. The MA(ATS) with 13 ATS paths spent 71.53 sec. on a Pentium IV platform to finish the search. The control system having the new controllers obtained from search has better step response than the original as elaborated in section 5 of this paper. Opportunities open for further investigations of the search performance of having different search algorithms as the search core of the $\mathrm{MA}(\cdot)$.

Acknowledgements

The authors express their thanks to Asst. Prof. Dr. Arthit Srikeaw, School of Eletrical Engineering, Suranaree University of Technology for his valuable suggestion on the symmetrical property of search problems. Financial supports from Rajamangala University of Technology Lanna, Tak Campus, and Suranaree University of Technology are very much appreciated.

References:

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

[2] F. Glover, Parametric tabu search for mixed integer programs, *Computers and Operations Research*, Vol.33, No.9, 2006, pp. 2449-2494.

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

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

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

[6] 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)

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

[8] 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)

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

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

[11] 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.

[12] 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.

[13] J. Kluabwang, D. Puangdownreong and S. Sujitjorn, Management agent for search algorithms with surface optimization applications, *WSEAS Transactions on Computers*, Vol.7, No.6, 2008, pp. 791-803.

[14] J. Kluabwang, D. Puangdownreong and S. Sujitjorn, Management agent for search algorithms, *In Proc. 12th WSEAS Int . Conf. on-Computers*, Vol.2, No.2, 2008, pp.375-382.

[15] 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.

[16] L. Phillip and R. Bart, Problem partitioning in hybrid genetic algorithms, *WSEAS Transactions on Systems*, Vol.6, No.2, 2007, pp. 395-399.

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

[18] 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.

[19] S. Brennan, and A. Alleyne, The Illinois roadway simulator: a mechatronic testbed for vehicle dynamics and control, IEEE/ASME Transaction on Mechatronics, Vol.5, No.4, 2000, pp.349-359.

[20] S. Brennan, and A. Alleyne, Using a scaled testbed: controller design and evaluation, IEEE Control Systems Magazine, Vol.21, No.3, 2001, pp.15-26.