

Research Article

Multipath Adaptive Tabu Search for a Vehicle Control Problem

Jukkrit Kluabwang,^{1,2} Deacha Puangdownreong,³ and Sarawut Sujitjorn¹

¹ Control and Automation Research Unit, Power Electronics, Machines, and Control Research Group, School of Electrical Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand

² Department of Electrical Engineering, Rajamangala University of Technology Lanna Tak, Tak 63000, Thailand

³ Department of Electrical Engineering, South-East Asia University, Bangkok 10160, Thailand

Correspondence should be addressed to Sarawut Sujitjorn, sarawut@sut.ac.th

Received 2 September 2011; Revised 27 October 2011; Accepted 28 October 2011

Academic Editor: Md. Sazzad Chowdhury

Copyright © 2012 Jukkrit Kluabwang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Tabu search has become acceptable worldwide as one of the most efficient intelligent searches applied to various real-world problems. There have been different modifications made to the generic tabu search in recent years to achieve better performances. Among those reviewed in the introduction of this paper, the adaptive tabu search (ATS) has incorporated the backtracking and the adaptive search radius mechanisms that help accelerate the search and release it from a local solution lock. The paper explains an enhancement made to the ATS to accomplish multipath ATS (MATS) algorithms. Performances of the ATS and the MATS are evaluated using surface optimization problems, and results are presented in the paper. Finally, the MATS is applied to solve a real-world vehicle control problem.

1. Introduction

The tabu search (TS) [1], one of the most widely used local search techniques, was proposed to solve the combinatorial optimization problems. The TS is based on the neighbourhood search approach and adaptive memory. With the common deterministic scheme, the search process of the TS is commonly based on the recency and the frequency approaches [2, 3]. The high-quality solution found by each search round is stored in the tabu list (TL). In case of entrapment (deadlock) problems caused by some local minima, the elite solutions stored in the TL are retrieved according to the aspiration criteria (AC) for starting a new search such that the search could be released from the deadlock. By literatures, the TS has been successfully applied to solve optimization problems in various fields, for examples, power systems [4, 5], transportation [6], flow shop [7], food processing [8], and so forth. However,

the conventionally simple TS cannot completely handle the deadlock problems. With this challenging task, some researchers have proposed modified versions of the original TS to find alternative, more potential versions to overcome such problems. Distinctively modified version of the TS include the reactive tabu search (RTS) [9, 10], the parallel tabu search (PTS) [11–13], the modified tabu search (MoTS) [14–16], the probabilistic tabu search (PrTS) [17–19], and the hybrid tabu search (HTS) [20–29].

Recently, the adaptive tabu search (ATS) has been launched [30] as a modified versions of the original TS. The ATS consists of the backtracking (BT) and the adaptive search radius (AR) mechanisms. As firstly introduced in 1996 [7], the BT regarded as a diversification strategy is used for the search to escape from the deadlocks caused by local minima. The AR concept introduced in 1989 [31] and regarded as an intensification strategy is applied to speed up the search process. The convergence analysis and the performance evaluation of the ATS were reported [30]. In addition, the ATS has been successfully applied to various engineering problems, for examples, electrical system protection [32], dynamical system identification [33], control synthesis [34], and acoustic signal processing [35].

This article proposes the multipath adaptive tabu search (MATS) and begins with the review of the ATS algorithms in Section 2. Section 3 explains the MATS in details and presents some algorithm analyses. Performance evaluations of the MATS compared with the ATS via surface optimization problems can be found in Section 4. An application of the proposed MATS to the control synthesis problem of a scaled vehicle (the Illinois Roadway Simulator system (IRS) [36]) is illustrated in Section 5. The application is considered as a special case ($n = 2$) of the n -dimensional continuous optimization problems. Conclusion follows in Section 6.

2. Review of the Adaptive Tabu Search

In this section, reviews of the adaptive tabu search (ATS) are provided. The section is divided into two parts, that is, the ATS algorithms review and its performance evaluation via some benchmark surface optimization problems.

2.1. Adaptive Tabu Search Algorithms

Regarding the original TS as the predecessor, the adaptive tabu search (ATS) consists of two additional mechanisms, that is, the backtracking (BT) regarded as a diversification strategy and the adaptive search radius (AR) regarded as an intensification strategy. The ATS begins the search with some random initial solutions belonging to an entire search space. In practice, the users may implement heuristically seeded initial solutions instead. All solutions in a neighbourhood search space within a certain search radius are evaluated via the objective function. The solution giving the minimum objective value (cost value) is set as a new starting point for the next search round and memorized in the tabu list (TL). According to the original TS algorithm and the TL property, the visited search spaces (or visited solutions) are prohibited for consecutively generated solutions. These search spaces are remarked as tabu. However, once the search is held by a local minimum (deadlock problem), the BT is activated. The BT looks up the TL and selects one of the memorized solutions as a new starting point. A new search could begin in a new direction. Hence, the search can possibly be released from the deadlock, and the search proceeds towards a better local minimum. Once the search approaches an elite local solution, the AR is invoked to speed up the search

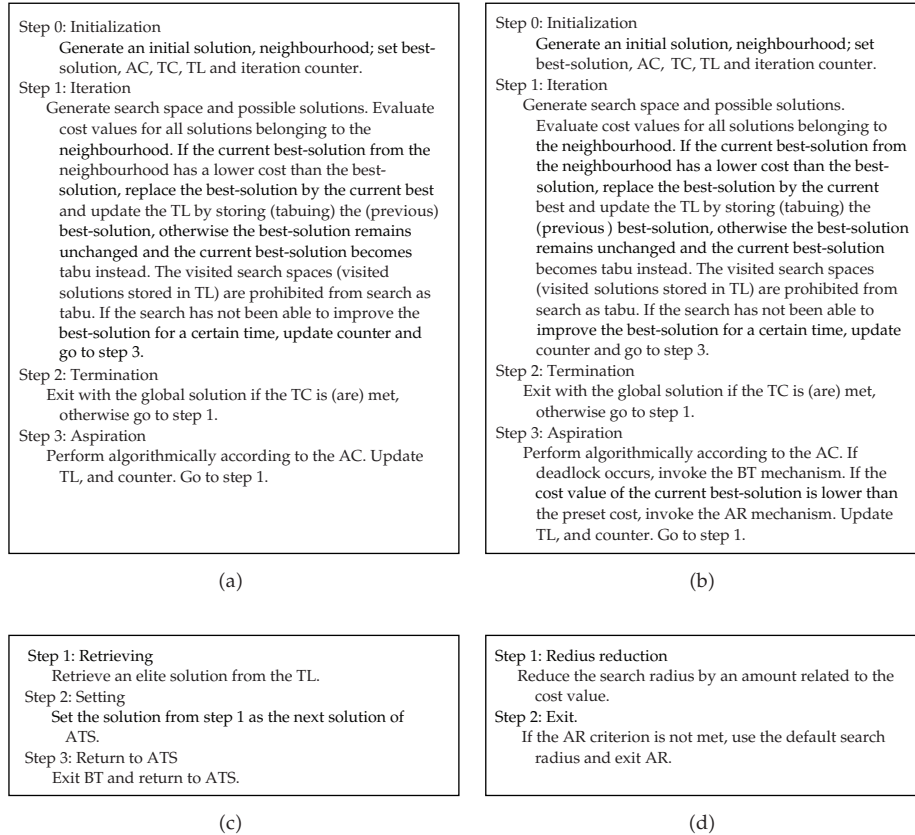


Figure 1: Generic algorithms of the TS and the ATS: (a) generic TS algorithms, (b) generic ATS algorithms, (c) the BT mechanism of the ATS, and (d) the AR mechanism of the ATS.

process. The search radius is subsequently decreased in accordance with the cost value of the current solution to provide finer and finer solutions within a short duration. The less the cost value, the smaller the search radius becomes. With these two mechanisms, a sequence of solutions obtained by the ATS is efficiently and rapidly converged towards the global minimum. Figure 1 expresses the generic algorithms of the original TS and the ATS with the BT and the AR mechanisms, respectively, while Figure 2 depicts the diagram representing some movements of the ATS. Both figures give a clear view for the reader to follow.

2.2. Performance Evaluation of the ATS

Although the performance evaluation of the ATS has appeared [30], the work reported here is an extension via six well-known surface optimization problems, that is, Bohachevsky function (BF), Rastrigin function (RF), Shekel's foxholes function (SF), Shubert function (ShF), large area Schwefel function (type-1 or ScF#1), and small area Schwefel function (type-2 or ScF#2), respectively. Details of these tested functions are summarized in Table 1. Among these, the ShF contains 18 global solutions while each of the rest has only one.

The ATS algorithms were coded by MATLAB, and run on the Intel Celeron(R), 2.6 GHz, 238 Mbytes SD-RAM, 30 Gbytes HDD. Referring to the Table 1, the global solutions

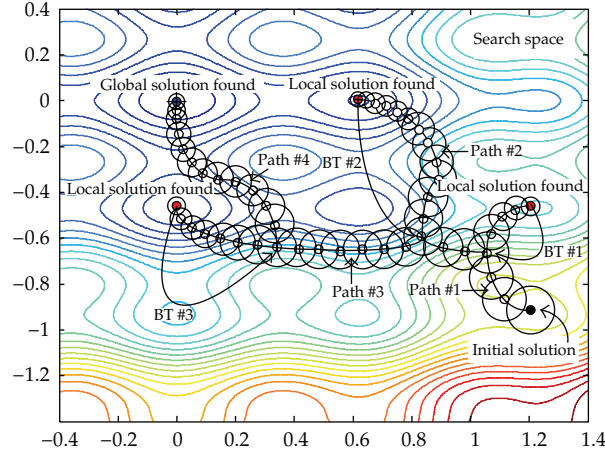


Figure 2: Movements of the ATS.

Table 1: Summary of the surface optimization problems.

Surface names	Surface functions	Search spaces	Sketches
Bohachevsky	$F(x, y) = x^2 + 2y^2 - 0.3 \cos(3\pi x) - 0.4 \cos(4\pi y) + 0.7$, $f_{\min}(0, 0) = 0$	$[-2, 2]$	
Rastrigin	$F(x, y) = x^2 + y^2 - 10 \cos(2\pi x) - 10 \cos(3\pi y) + 20$, $f_{\min}(0, 0) = 0$	$[-2, 2]$	
Shekel's foxholes	$f(x_1, x_2) = [1/500 + \sum_{j=1}^{25} (1/(j + \sum_{i=1}^2 (x_i - a_{ij})^6))]^{-1}$ Where $a_{ij} = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{pmatrix}$ $f_{\min}(-32, -32) = 1$	$[-40, 40]$	
Shubert	$f(x, y) = (\sum_{i=1}^5 i \cos((i+1)x_1 + i))(\sum_{i=1}^5 i \cos((i+1)y + i))$ $f_{\min}(x_j, y_j) = -186.7309, j \in \{1, 2, \dots, 17, 18\}$	$[-10, 10]$	
Schwefel I	$f(x_1, x_2) = 418.9829 \times 2 - \sum_{i=1}^2 (x_i \sin \sqrt{ x_i })$, $f_{\min}(421, 421) = 0$	$[-500, 500]$	
Schwefel II		$[400, 500]$	

of the tested functions are as follows: BF and RF- $\{x = y = 0$ with the cost value $f(0, 0) = 0\}$; SF- $\{x = y = -32$ with $f(-32, -32) = 0.998\}$; ShF- $\{x_j, y_j$ with $f(x_j, y_j) = -186.7309$, where $j = 1, 2, \dots, 18\}$, and ScF#1 and #2- $\{x = y = 421$ with $f(421, 421) = 0\}$. The variables' boundaries of all tested functions are set to form the corresponding search spaces as declared in the Table 1.

Following the recommendations [30], the parameter settings of the ATS for each tested function are declared in the Table 2, where Re_{\max} is the maximum allowance of the solution

Table 2: Setting of the ATS search parameters.

Test function	Search radius	no. of neighbours	ATS parameters					TC	
			BT		AR			Count _{max}	J _{min}
			Re _{max}	kth	Stage I	Stage II	Stage III		
BF	0.2 (5%)	30	5	5	$J < 0.1,$ $R = 0.002$	$J < 0.001,$ $R = 0.002$	—	10,000	1e-9
RF	0.2 (5%)	30	5	5	$J < 0.1,$ $R = 0.002$	$J < 0.001,$ $R = 0.002$	—	10,000	1e-8
SF	0.8 (1%)	30	5	5	$J < 5,$ $R = 0.2$	$J < 2,$ $R = 0.1$	—	10,000	0.999
ShF	1.0 (5%)	30	5	5	$J < -100,$ $R = 0.1$	$J < -150,$ $R = 0.015$	—	10,000	-186.7
ScF#1	100 (10%)	20	5	5	$J < 100,$ $R = 50$	$J < 10,$ $R = 0.01$	$J < 1,$ $R = 0.001$	10,000	1e-4
ScF#2	1.0 (1%)	20	5	5	$J < 10,$ $R = 0.01$	$J < 1,$ $R = 0.001$	—	10,000	1e-4

Table 3: Results of the ATS's performance evaluation.

Search results	Test functions					
	BF	RF	SF	ShF	ScF#1	ScF#2
(1) Search time (sec)	5.3940	8.2067	2.2837	3.3978	38.1037	0.2522
(2) Search rounds	556.06	759.80	108.22	342.54	336.24	31.24
(1)/(2) (sec/round)	0.0097	0.0108	0.0211	0.0099	0.1133	0.0081

cycling before invoking the BT, kth is the backward-selected solution by the BT, $count_{max}$ is maximum search round, R is search radius, J is the cost value, and J_{min} is minimum cost value used as the termination criteria (TC) for all tests in order to assure the global minimum found. The performance tests were conducted 50 trials against each tested function to obtain the average search round and time consumed. Each trial starts with a random initial solution generated by MATLAB using a seed number. The initial solution forms the center of a uniformly distributed hypersphere of a given radius, from which a neighbourhood is generated. The ATS stops when $J < J_{min}$ or $count \geq count_{max}$.

The results obtained are summarized in Table 3, where the first row stands for the average time consumed, the second row is the average search round, and the third row expresses the average time consumed per one search round. It is found that the ATS yields very satisfactory search performance in terms of solution convergence and the search time consumed for all surface optimization problems.

3. Multipath Adaptive Tabu Search (MATS)

In order to improve the search performance of the ATS, one may utilize an exploration strategy proposed by this paper. The implementation is suitable for use on a single CPU platform. The proposed MATS consists of partitioning, sequencing, and discarding mechanisms (PM, SM, and DM), respectively. Each mechanism is described next.

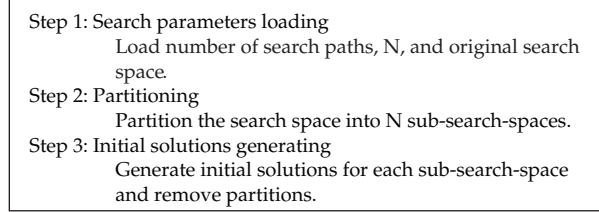


Figure 3: The PM procedures.

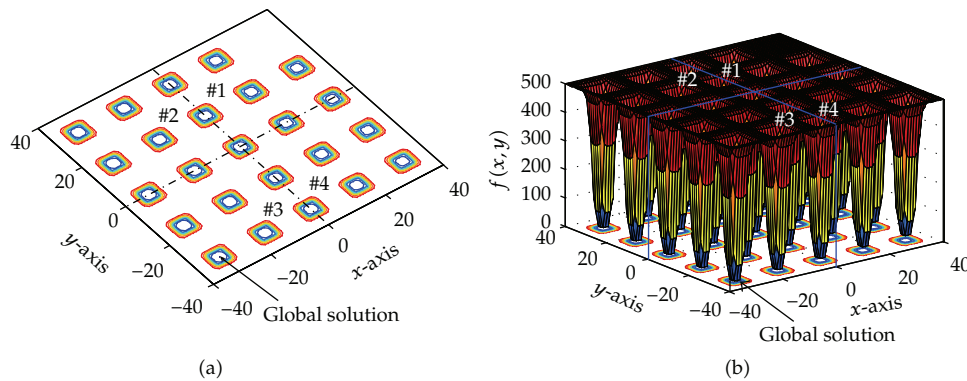


Figure 4: Partitioned surface of the SF: (a) 2D contour, (b) 3D surface.

Partitioning Mechanism (PM)

This mechanism operates only once at the beginning of the search. The PM starts with decomposing the entire search space into a number of sub-search-spaces defined by the user. The PM also defines explicit boundaries for the sub-search-spaces, each of which is given an individual initial solution and an independent ATS path. Since all ATS paths as the search cores of the MATS use the same set of search parameters, this approach is referred to as the multiple points single strategy (MPSS). Just before launching the search paths, the PM removes the predefined boundaries such that all ATS paths could search freely on the originally entire search space. This technique helps to reduce any conflicts that may arise during the search along the border lines. Figure 3 summarizes the PM procedures.

As an example, the surface of the Shekel's foxholes function (SF) partitioned into 4 sub-search-spaces is shown in Figure 4. The boundaries of the sub-search-spaces are defined in the form of ((upper limit of the 1st variable) (upper limit of the 2nd variable); (lower limit of the 1st variable) (lower limit of the 2nd variable)).

Regarding this notion, the entire search space can be defined as $[40\ 40;-40\ -40]$, and the sub-search-spaces #1, 2, 3, and 4 are confined within $[40\ 40;0\ 0]$, $[0\ 40;-40\ 0]$, $[0\ 0;-40\ -40]$ and $[40\ 0;0\ -40]$, respectively. In practice, any geometrical forms and coordinates are possible for the partitioning process to suit the applications. The number of sub-search-spaces must be finite and not too great, and the problem dimensions are not limited. Subsequently, an ATS path is assigned to each sub-search-space, and the boundaries are then removed.

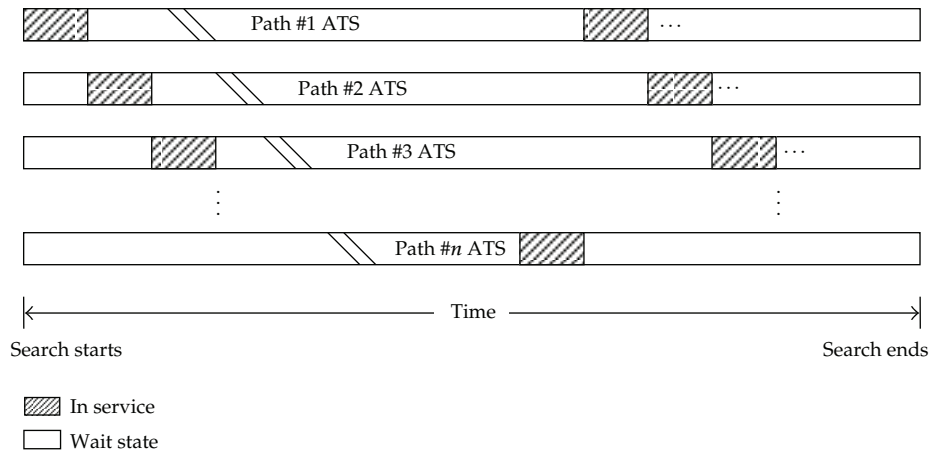


Figure 5: Diagram representing n paths of the ATS.

Sequencing Mechanism (SM)

As mentioned earlier, the proposed MATS is intended for the use on a single CPU platform. Since the MATS possesses a few to many search paths, it needs some organizing mechanisms for its correct sequential operations. The SM is implemented to accomplish this, and can be represented by the simple diagram in Figure 5. Assuming that there are n search paths organized by the SM, the first ATS path (#1ATS) begins its first trial. Afterward, it goes to the wait state. Once the CPU finishes its service to the #1ATS, it provides the service to the second path. Once the #2ATS finishes its first search trial, it has to wait. The operation goes on in this manner until the # n ATS finishes its first search trial. The CPU then returns to service the #1ATS for its second search trial, then the #2, #3, \dots and # n ATS, in sequential manner. The operation is repeated until one of the ATS paths hits the solution, and requests for exiting the search according to the satisfied termination criterion. Either an equal or an unequal number of search rounds constitutes one search trial. The works by this paper utilize an equal number of search rounds for all ATS paths to work on one search trial. Readers may notice that organizing the search this way, more search paths would result in more wait states making the search process very time consuming. Hence, the number of search paths employed must be limited. Moreover, another mechanism, namely, the discarding mechanism (DM) is introduced and explained in the next section. With the DM, as time goes by, more and more search paths will be cut off from the search process until only one path is left to continue searching for most cases. However, an individual ATS can request for a complete termination once the global solution is found at any time. The SM needs to communicate with the DM to learn about the existing and the forcedly terminated (or discarded) ATS paths. Thus, the SM can be summarized by the procedures shown in Figure 6.

Discarding Mechanism (DM)

The DM is designed to reduce the overall search time of the MATS based on the idea of forcing some low-quality search paths to stop as soon as possible. Various possible approaches can be used to implement this idea including evaluation of the search-path quality based on the cost values of the current best-solutions or a set of previous elite solutions of the specific

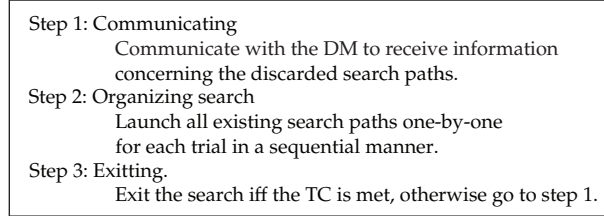


Figure 6: The SM procedures.

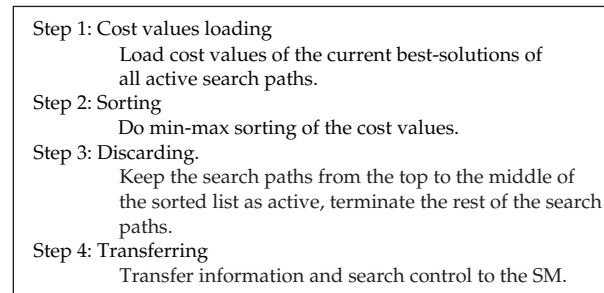


Figure 7: The DM procedures.

search path, the probability or the possibility of the search path to hit some higher quality solutions, and so forth. A simple implementation of the recency strategy via the evaluation of the cost values of the current best-solutions of the search paths is demonstrated. Each time the DM operates, the number of search paths is reduced by half. However, the number of search paths to be reduced can be arbitrarily adjusted to suit an application. After this forced termination made to the low-quality paths, the DM transfers the information concerning the being-terminated and the existing paths to the SM. Figure 7 summarizes the DM procedures.

According to the previous explanation, the MATS algorithms can be described by the list in Figure 8 due to the fact that the ATS is used as the core search algorithms.

4. Performance Evaluation

Evaluation of the performance of any search algorithms can be done in several ways, for example, using data mining, logistic, surface optimization problems, and so forth. The works reported here have applied the six surface optimization problems described in Section 2 for the purposes. These test functions are referred to shortly as the BF, RF, SF, ShF, ScF#1, and ScF#2, respectively, and summarized by the Table 1. Performance comparisons are made among the ATS, the MATS with 2, 4, 8, 16, 32, and 64 paths denoted as MATS#2, MATS#4, MATS#8, MATS#16, MATS#32 and MATS#64, respectively.

The PM and the DM settings are rather complicated for the performance investigations. The SM simple settings can be either fixed or adjustable search rounds for each ATS path. The fixed numbers of search rounds as well as the same ATS search parameters have been used throughout the studies. As mentioned earlier, the proposed MATS employs the multiple points single strategy (MPSS) method.

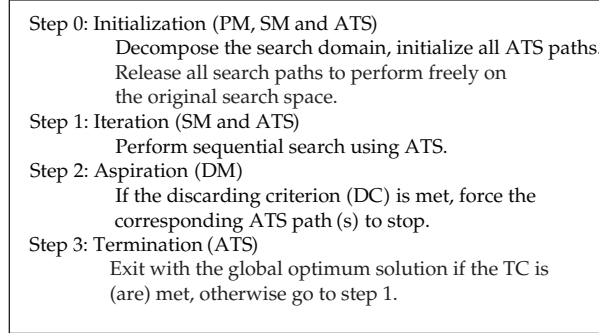


Figure 8: The MATS algorithms.

The PM settings are now explained by referring to Table 4, which declares the boundaries of the sub-search-spaces for 2, 4, ..., 64 paths. As an example, consider the ShF having its entire space defined by [10 10; -10 -10]. For MATS#32, the ShF is decomposed into #1[10 10; 7.5 5] for the 1st region, #2[7.5 10; 5 5] for the 2nd, #3[5 10; 2.5 5] for the 3rd, and so on. The similar approach for space partitioning can be applied to the other surface functions. In addition, other geometric partitioning techniques and coordinates could be employed to suit particular applications.

The DM employs the recency strategy by comparing the cost values of the current best-solutions of the ATS paths. Only the paths with the cost values lower than a preset limit are kept, otherwise they are discarded. The number of the paths being discarded can be set in the manner of either a gradual cut or a sudden cut. For the MATS with many search paths, the sudden cut is preferable such that the wait-state time be minimum. For a moderate to a few numbers of search paths inside the MATS, the gradual cut is more suitable. Table 5 reveals the DM settings in details. Let us consider the SF case of 8, 32, and 64 paths as examples. For the case of SF with MATS#8, the DM becomes active firstly at the 2nd iteration, and 4 search paths are discarded. The DM becomes active again at the 4th and 6th iterations; 2 and 1 paths are forced to stop, respectively. Eventually, there is 1 path left to continue searching. This is one example of using the gradual cut technique. Different approaches, that is, the sudden cuts, are found for the cases of MATS#32 and #64. At the 1st iteration of both cases, 31 and 63 search paths are discarded correspondingly. This means that only one path is left after the 1st iteration to track down the solution as for the cases of MATS#32 and #64. It is found in some cases, that is, the ShF problem, that good performances are obtained via having more than one path left for searching and competing for solutions. Note that, for an effective operation, the q th iteration at which the DM is invoked should be less than half of the average search rounds consumed by the single path ATS (\overline{SR}_{ATS}). The \overline{SR}_{ATS} data are presented in row 2, Table 3. Due to the fact that the MATS utilizes more than 1 search path, the inequality $\sum_{i=1}^k (dm_i \times n_i) < \overline{SR}_{ATS}$ must hold. Consider the case of BF with MATS#4, $\sum_{i=1}^k (dm_i \times n_i) = (5 \times 4) + (10 \times 2) < 556$, for instance.

The investigations of the search performance of the proposed MATS were conducted against the six surface optimization problems on the Intel Celeron(R), 2.6 GHz, 250 MB-RAM and 30 GB-HDD platform. Each figure of the results reported herein is an average of 50 trials. Tables 6–9 reveal the detailed results. The averages of search rounds are summarized in Table 6. However, the figures do not disclose the actual activities performed by the ATS used as the search core of the MATS. These figures are therefore converted into the equivalent

Table 4: Sub-search-spaces partitioned by PM.

Test functions	Sub-search-spaces defined by PM				
	MAATS#2	MAATS#4	MAATS#8	MAATS#16	MAATS#64
BF and RF [0 2; -2 -2]	#1[2 2; 0 -2]	#1[2 2; 0 0]	#1[2 2; 1 0]	#1[2 2; 1 1]	#1[2 2; 1.5 1]
	#2[0 2; -2 -2]	#2[0 2; -2 0]	#2[1 2; 0 0]	#2[1 2; 0 1]	#2[1.5 2; 1 1]
		#3[0 0; -2 -2]	#3[0 2; -1 0]	#3[0 2; -1 1]	#3[1 2; 0.5 1.5]
		#4[2 0; 0 -2]	#4[-1 2; -2 2]	#4[-1 2; -2 1]	#4[0.5 2; 0 1.5]
			#5[-1 0; -2 -2]	#5[-1 1; -2 0]	#5[0 2; -0.5 1.5]
SF [40 40; -40 -40]	#1[40 40; 0 -40]	#1[40 40; 0 0]	#1[40 40; 20 0]	#1[40 40; 20 20]	#1[40 40; 30 30]
	#2[0 40; -40 -40]	#2[0 40; -40 0]	#2[20 40; 0 0]	#2[20 40; 0 20]	#2[30 40; 20 30]
		#3[40 0; 0 -40]	#3[0 40; -20 0]	#3[0 40; -20 20]	#3[20 40; 10 30]
		#4[0 0; -40 -40]	#4[-20 40; -40 0]	#4[-20 40; -40 20]	#4[10 40; 0 30]
			#5[40 0; 20 -40]	#5[-20 20; -40 0]	#5[0 40; -10 30]
ShF [10 10; -10 -10]	#1[10 10; 0 -10]	#1[10 10; 0 0]	#1[10 10; 5 0]	#1[10 10; 5 5]	#1[10 10; 7.5 7.5]
	#2[0 10; -10 -10]	#2[0 10; -10 0]	#2[5 10; 0 0]	#2[5 10; 0 5]	#2[7.5 10; 5 7.5]
		#3[0 0; -10 -10]	#3[0 10; -5 0]	#3[0 10; -5 5]	#3[5 10; 2.5 7.5]
		#4[10 0; 0 -10]	#4[10 0; 0 -10]	#4[-5 10; -10 0]	#4[2.5 10; 0 7.5]
			#5[-5 0; -10 -10]	#5[-5 5; -10 0]	#5[0 10; -2.5 7.5]
ScF#1 [500 500; -500 -500]	#1[500 500; 0 -500]	#1[500 500; 0 0]	#1[500 500; 250 0]	#1[500 500; 250 250]	#1[500 500; 375 375]
	#2[0 500; -500 -500]	#2[0 500; -500 0]	#2[250 500; 0 0]	#2[250 500; 0 250]	#2[375 500; 250 375]
		#3[500 0; 0 -500]	#3[0 500; -250 0]	#3[0 500; -250 250]	#3[250 500; 125 375]
		#4[0 0; -500 -500]	#4[-250 500; -500 0]	#4[-250 500; -500 250]	#4[125 500; 0 375]
			#5[500 0; 250 -500]	#5[-250 250; -500 0]	#5[0 500; -125 375]
ScF#2 [500 500; 400 400]	#1[500 500; 450 400]	#1[500 500; 450 450]	#1[500 500; 475 450]	#1[500 500; 475 475]	#1[500 500; 487.5 487.5]
	#2[450 500; 400 400]	#2[450 500; 400 450]	#2[475 500; 450 450]	#2[475 500; 450 475]	#2[487.5 500; 475 487.5]
		#3[500 450; 450 400]	#3[450 500; 425 450]	#3[450 500; 425 475]	#3[475 500; 450 475]
		#4[450 450; 400 400]	#4[425 500; 400 450]	#4[425 500; 400 475]	#4[462.5 500; 450 475]
			#5[500 450; 475 400]	#5[425 475; 400 450]	#5[450 500; 437.5 450]

Table 5: The DM settings.
 q th iteration at which the DM is invoked and the number of discarded path(s)

Test functions	MATS																		
	#2			#4			#8			#16			#32			#64			
	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd	4th	5th	1st	2nd	3rd	4th	5th	1st	2nd	3rd
BF	5	10	15	5	10	15	5	10	15	20	25	5	10	15	20	25	1	3	5
no.dis.pth.	1	1	1	8	4	2	1	8	4	2	1	16	8	4	2	1	32	16	15
RF	5	—	6	2	4	6	2	4	6	8	10	2	4	6	8	10	1	—	—
no.dis.pth.	1	3	1	8	4	2	1	8	4	2	1	16	8	4	2	1	63	—	—
SF	5	4	6	2	4	6	2	4	6	8	—	1	—	—	—	—	1	—	—
no.dis.pth.	1	1	1	8	4	2	1	8	4	2	1	31	—	—	—	—	63	—	—
ShF	1	—	—	3	5	—	1	3	5	—	—	1	—	—	—	—	1	—	—
no.dis.pth.	1	3	—	8	4	—	2	8	4	—	—	28	—	—	—	—	60	—	—
ScF#1	5	3	5	1	3	5	1	3	5	—	—	1	—	—	—	—	1	—	—
no.dis.pth.	1	2	1	8	4	3	1	8	4	3	—	31	—	—	—	—	63	—	—
ScF#2	1	—	—	1	—	—	1	—	—	—	—	1	—	—	—	—	1	—	—
no.dis.pth.	1	3	—	7	—	—	7	—	—	—	—	31	—	—	—	—	63	—	—

Table 6: Average values of search rounds.

Test functions	Average search rounds obtained from 50 trials						
	ATS 1-path	#2	#4	#8	#16	#32	#64
BF	556.06	530.20	507.94	186.82	142.70	202.34	173.76
RF	759.80	685.40	603.78	713.26	327.32	313.48	200.32
SF	108.22	79.84	72.96	64.62	30.62	54.04	36.16
ShF	342.54	291.56	282.16	75.68	73.02	61.68	70.12
ScF#1	336.24	304.88	294.44	241.56	268.22	263.18	131.10
ScF#2	31.24	30.02	29.20	26.48	24.26	21.58	19.96

Table 7: Equivalent average search rounds with respect to the ATS.

Test functions	Average search rounds equivalent to those of the ATS						
	ATS 1-path	#2	#4	#8	#16	#32	#64
BF	556.06	535.20	527.94	241.82	272.70	487.34	328.76
RF	759.80	690.40	753.78	735.26	379.32	427.48	263.32
SF	108.22	84.84	80.96	86.62	82.62	85.04	99.16
ShF	342.54	292.56	285.16	310.72	336.08	274.72	340.48
ScF#1	336.24	309.88	299.44	256.56	303.22	295.18	194.10
ScF#2	31.24	30.02	32.20	33.48	39.26	52.58	82.96

averages with-respect-to the ATS as shown in the Table 7. Noticeably, all cases, but the ScF#2, the MATS search rounds are actually less than those performed by the single path ATS.

The average values of search time tabulated in Table 8 also confirm this. The figures indicating search time (in seconds) in Table 8 are converted into percent reduction of search time using the following relation for comparison purposes:

Percent reduction of search time

$$= \frac{(\text{average search time of ATS} - \text{average search time of MATS})}{\text{average search time of ATS}} \times 100 \quad (4.1)$$

The percent reduction figures summarized in Table 9 reflect the merits of the MATS. The proposed MATS can effectively reduce the search time in the range of 4.08–56.79%, except that it is not efficient for the ScF#2 problem. Among the six test functions, only the ScF#2 is actually smooth over the entire search space whereas the MATS is not aimed for. The conventional optimization methods should be favourable to this kind of problems. The contemporary search methods, for example, the MATS and those of the local search algorithms, and so forth, are more suitable to the problems having Pareto front, nonsmooth surfaces, and multiple global solutions in nature. The numeric figures in Table 9 are displayed

Table 8: Average values of search time in seconds.

Test functions	Average search time in seconds obtained from 50 trials						
	ATS	MATS					
	1-path	#2	#4	#8	#16	#32	#64
BF	5.3940	5.0516	4.8901	2.3435	2.3309	4.1309	3.3029
RF	8.2067	6.5863	6.9925	7.2175	4.5316	5.0078	7.3221
SF	2.2837	1.8818	1.7697	1.8134	1.3918	1.7944	1.8431
ShF	3.3978	2.9819	3.1369	2.4038	2.6099	2.2640	2.9433
ScF#1	38.1037	30.3475	34.6200	28.1975	33.7100	34.8431	29.2327
ScF#2	0.2522	0.2419	0.2609	0.3009	0.3213	0.4350	1.5179

Table 9: Percent reduction of search time.

Test functions	Percent reduction of search						
	ATS	MATS					
	1-path	#2	#4	#8	#16	#32	#64
BF	0.0000	6.35	9.34	56.55	56.79	23.42	38.77
RF	0.0000	19.74	14.80	12.05	44.78	38.98	10.78
SF	0.0000	17.60	22.51	20.59	39.06	21.43	19.29
ShF	0.0000	12.24	7.68	29.25	23.19	33.37	13.38
ScF#1	0.0000	20.36	9.14	26.00	11.53	8.56	23.28
ScF#2	0.0000	4.08	-3.45	-19.31	-27.40	-72.48	-501.86

as bar graphs in Figure 9 to give a clear view of the merits of the MATS whereas the negative values indicate the nonproductive domain of the MATS.

To complement the revealed results, the cost values are also monitored and plotted as the convergence curves illustrated in Figure 10, for instance. Referring to Figure 10, the accepted solutions must have the costs of less than $1e-9$; the ATS hits the solution within 5.7650 s.; path#1 of the MATS#2 succeeds within 5.3440 s.; path#3 of the MATS#4 succeeds within 4.7500 s., and so forth. At present, it can be said that the proposed MATS performs faster than the single-path ATS; its speed for the best case (on a single CPU platform) is twice as much of the ATS speed; for the worst case its performance is about the same as the ATS, and the MATS can effectively reduce the search time by 30% as an average figure.

5. An Application: Car Control System in Roadway Simulator

This section presents the application of the proposed MATS algorithms for solving a real-world control problem of a car model.

The Illinois Roadway Simulator (IRS) has been proposed [36] to study the effects of road surface on car dynamics. The IRS is a reduced scale car model for a laboratory use. It

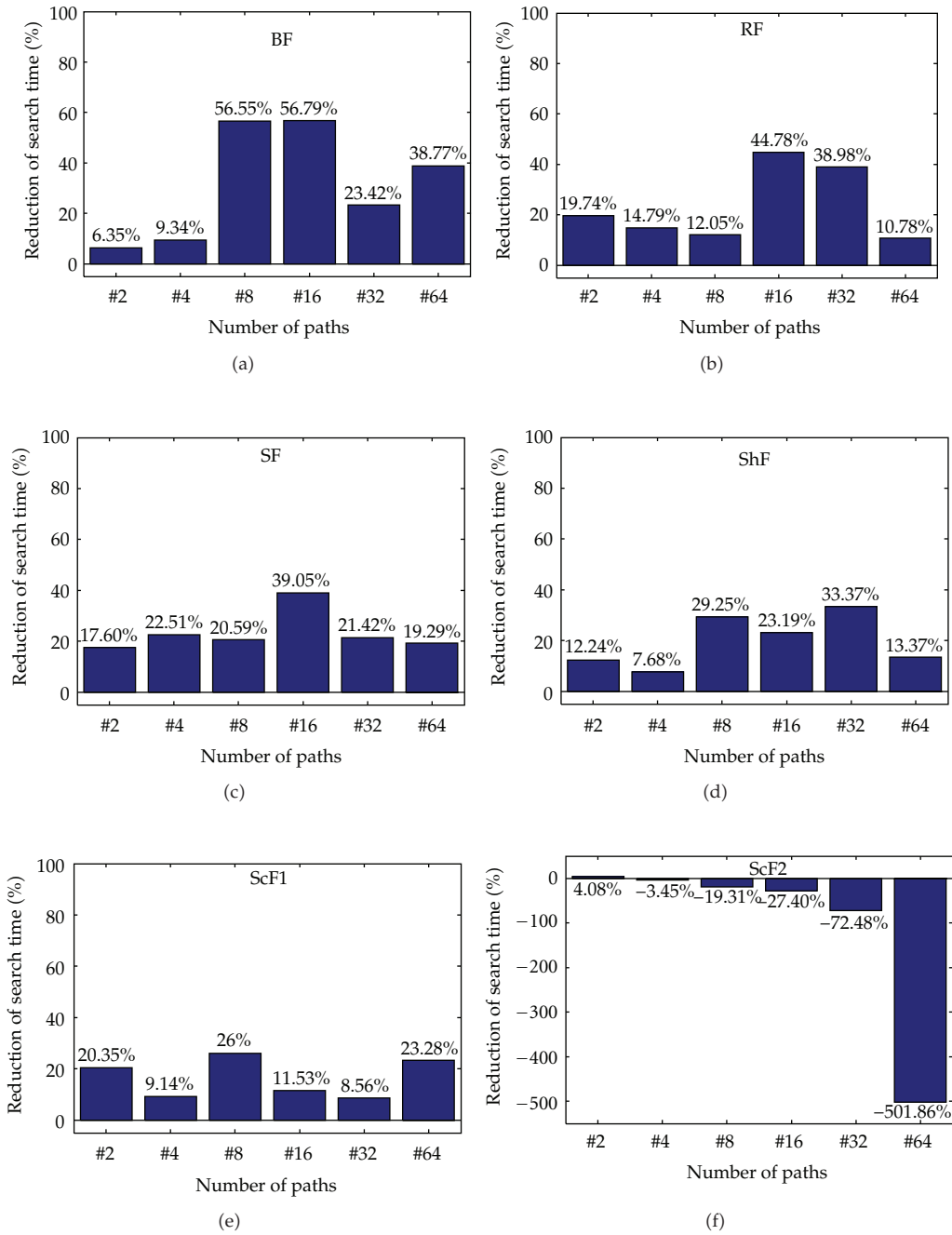


Figure 9: Bar graphs of the reduction of search time in percentages.

possesses the dynamics very similar to those of an actual car. In this, the yaw rate control is the issue of interest. The transfer function models of the IRS have been developed by applying the Buckingham-Pi theorem as well as system identification. The obtained transfer functions consist of the plant dynamic (G) and the disturbance dynamic (G_d) constituting

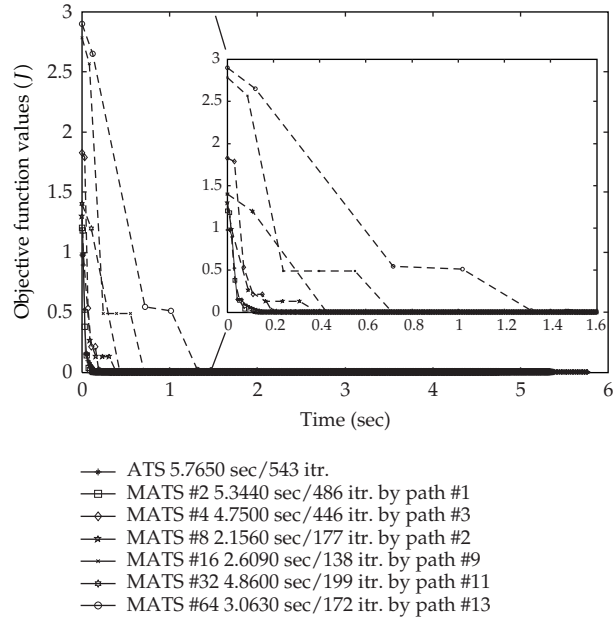


Figure 10: Convergence curves (BF problem).

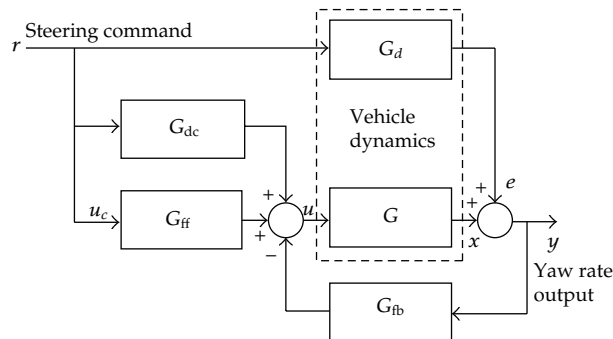


Figure 11: Block diagram representing the original yaw-rate control problem.

the vehicle dynamics. The block diagram shown in Figure 11 indicating the 3-DOF control structure represents the original system.

The three controllers, namely, G_{dc} , G_{ff} , and G_{fb} are designed via the model reference control (MRC) method. The vehicle models and the controllers are listed as follows:

$$G = \frac{2.802e4s + 2.306e5}{1.5s^4 + 94.43s^3 + 7.308e4s^2 + 1.183e5s + 5.361e5}$$

$$G_d = \frac{8.024e4s + 6.710e5}{1.3s^4 + 53.250s^3 + 1.943e3s^2 + 2.693e4s + 1.162e5}$$

$$\begin{aligned}
G_{dc} &= \frac{-1.2036e5s^5 - 8.584e6s^4 - 6.497e8s^3 - 1.44e10s^2 - 1.224e11s - 3.598e11}{3.6426e4s^5 + 1.792e6s^4 + 6.671e7s^3 + 1.202e9s^2 + 9.464e9s + 2.679e10}, \\
G_{ff} &= \frac{103s^2 + 1.8542e4s + 8.34385e5}{s^3 + 155.3s^2 + 3.727e3s + 2.0714e4}, \\
G_{fb} &= \frac{-5.181s^3 + 49.98s^2 + 5.294e3s + 4.8142e4}{s^3 + 155.3s^2 + 3.727e3s + 2.0714e4}.
\end{aligned} \tag{5.1}$$

Note that the controllers G_{ff} and G_{fb} have the same poles. Referring to Figure 11, the system transfer function, G_{irs} can be expressed as

$$G_{irs}(s) = \frac{Y(s)}{R(s)} = \frac{G_d + G \cdot G_{dc} + G \cdot G_{ff}}{1 + G \cdot G_{fb}}, \tag{5.2}$$

and hence

$$G_{irs} = \frac{b_{23}s^{23} + b_{22}s^{22} + \dots + b_1s + b_0}{a_{27}s^{27} + a_{26}s^{26} + \dots + a_1s + a_0}, \tag{5.3}$$

where the coefficients $a_i, i = 0, 1, 2, \dots, 27$ and $b_i, i = 0, 1, 2, \dots, 23$ are summarized in Table 10.

The unit step response of the original control system is depicted in Figure 12. The response possesses 40 ms delay time, and smoothly rises to the final value in 0.552 s.

To compensate for the delay time, an additional PD-controller is proposed in cascade connection to the G_{ff} . The theoretical PD-controller of the form $G_{PD} = K_D s + K_P$ is considered. Thus, the G_{ff} block is modified to be $G_{PD} \cdot G_{ff}$. The design of the PD-controller is not straight-forward because of the complexity of the existing control system structure. So, the search method has been applied to find an optimum PD-controller such that the following objectives are satisfied: $P.O. < 0.1\%$ (no overshoot preferred), rise-time (t_r) < 0.231 s., and settling-time (t_s) $< 5t_r$ s. The control design problem becomes a multiobjective optimization problem in which the surrogate objective function, J , is formed such that

$$J = \alpha_1 t_{sj} + \alpha_2 t_{rj} + \alpha_3 P.O., \tag{5.4}$$

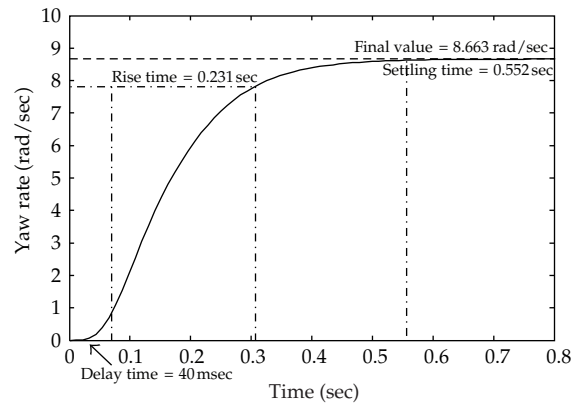
where t_{rj} = % rise-time (normalized), t_{sj} = % settling-time (normalized), $P.O.$ = % overshoot, and α_1, α_2 , and α_3 = weighting factors.

In this application, the factors $\alpha_1 = \alpha_2 = \alpha_3 = 1$, and the penalty concept is applied to $t_{rj}, t_{sj}, P.O.$ and J , respectively. Figure 13 provides the list of the objective function procedures.

The max-min ranges of both K_P and K_D for the search are (2, 0). The MATS with two ATS-paths is used with the following settings: [2 2; 1 0] and [1 2; 0 0] for the PM, and the q th iteration = 50 for the DM. The initial solutions for the ATS#1 and #2 are ($K_P = 1.824, K_D = 0.153$) and ($K_P = 0.329, K_D = 0.776$), respectively. The search parameters of the ATS are as follows: initial $R = 0.2$, $N = 5$, $N_{re,max} = 5$, k th backward for the $BT = 5$, ($J < 90$) \rightarrow ($R = 0.1$) and ($J < 70$) \rightarrow ($R = 0.05$) for the AR, and $count_{max} = 10,000$ or $J < 65$ for the TC.

Table 10: The coefficients of the transfer function in (5.3).

The coefficients					
a_i			b_i		
$a_0 = 4.115e41$	$a_{10} = 3.346e35$	$a_{20} = 4.678e20$	$b_0 = 3.565e42$	$b_8 = 9.645e37$	$b_{16} = 4.055e26$
$a_1 = 6.265e41$	$a_{11} = 2.225e34$	$a_{21} = 6.423e18$	$b_1 = 4.813e42$	$b_9 = 7.143e36$	$b_{17} = 6.618e24$
$a_2 = 4.480e41$	$a_{12} = 1.251e33$	$a_{22} = 7.331e16$	$b_2 = 3.016e42$	$b_{10} = 4.307e35$	$b_{18} = 8.798e22$
$a_3 = 2.001e41$	$a_{13} = 5.979e31$	$a_{23} = 6.805e14$	$b_3 = 1.165e42$	$b_{11} = 2.136e34$	$b_{19} = 9.327e20$
$a_4 = 6.265e40$	$a_{14} = 2.445e30$	$a_{24} = 4.957e12$	$b_4 = 3.107e41$	$b_{12} = 8.779e32$	$b_{20} = 7.624e18$
$a_5 = 1.462e40$	$a_{15} = 8.580e28$	$a_{25} = 2.671e10$	$b_5 = 6.075e40$	$b_{13} = 3.006e31$	$b_{21} = 4.525e16$
$a_6 = 2.643e39$	$a_{16} = 2.590e27$	$a_{26} = 9.423e7$	$b_6 = 9.037e39$	$b_{14} = 8.590e29$	$b_{22} = 1.7521e14$
$a_7 = 3.797e38$	$a_{17} = 6.729e25$	$a_{27} = 1.598e5$	$b_7 = 1.048e39$	$b_{15} = 2.048e28$	$b_{23} = 3.075e11$
$a_8 = 4.417e37$	$a_{18} = 1.503e24$				
$a_9 = 4.219e36$	$a_{19} = 2.873e22$				

**Figure 12:** Original system response.

As a result shown in Figure 14, path#1 of the MATS#2 spent 217.1648 s to track down the solutions $K_P = 1.0$ and $K_D = 0.07$ at the 326th iteration with the cost $J = 61.7234$. Figure 15 illustrates the step responses of the original and the enhanced systems for comparison purposes. It can be observed that the delay time is reduced by 97%, and the rise-time and settling-time are reduced by 30%, approximately.

6. Conclusion

The development of the multipath adaptive tabu search (MATS) has been reported by this paper. The algorithm incorporates the partitioning, sequencing, and discarding mechanisms, respectively, to the ATS, and can be run efficiently on a single CPU platform as being confirmed by the detailed performance assessment results using the well-known surface optimization problems. According to the assessment results, the MATS is about two-times faster than the ATS as the best case. Moderately, the MATS renders 30% reduction in the search time compared to the ATS. However, the MATS is not suitable for some smooth problems as it consumes more search time than the single ATS does. The paper has been

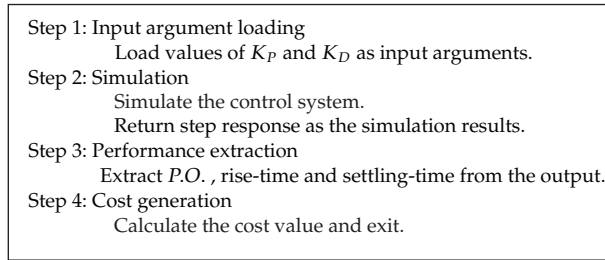


Figure 13: The objective function procedures for the control problem.

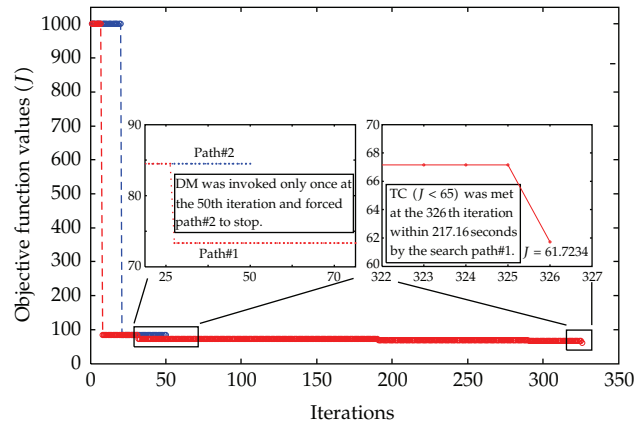


Figure 14: Convergence curves of the PD-controller design problem.

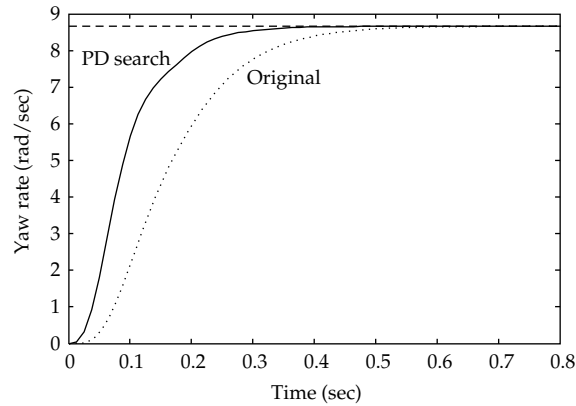


Figure 15: Comparison of the step responses.

closed by an illustrative example of using the MATS to solve the multiparameter control problem of a reduced scale vehicle in which the enhanced system becomes 4-DOF. The very satisfactory results obtained confirm the effectiveness and the usefulness of the proposed algorithms. The proposed algorithmic mechanisms described herein are open enough for applications to other search algorithms.

Acknowledgments

The financial support from Suranaree University of Technology (SUT) and the Office of the Higher Education Commission under NRU project of Thailand are greatly acknowledged. The authors' thanks are also due to Witchupong Wiboonjaroen for his general assistance for preparing some parts of the paper.

References

- [1] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers and Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.
- [2] F. Glover, "Tabu search—part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.
- [3] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, 1997.
- [4] A. H. Mantawy, Y. L. Abdel-Magid, and S. Z. Selim, "Unit commitment by Tabu search," *IEEE Generation Transmission and Distribution*, vol. 1, no. 45, pp. 56–64, 1998.
- [5] T. Kulworawanichpong and S. Sujitjorn, "Optimal power flow using Tabu search," *IEEE Power Engineering Review*, vol. 22, no. 6, pp. 37–40, 2002.
- [6] J.-F. Cordeau and G. Laporte, "A Tabu search heuristic for the static multi-vehicle dial-a-ride problem," *Transportation Research B*, vol. 37, no. 6, pp. 579–594, 2003.
- [7] E. Nowicki and C. Smutnicki, "A fast taboo search algorithm for the job shop problem," *Management Science*, vol. 42, no. 6, pp. 797–813, 1996.
- [8] G. Zhang, W. Habenicht, and W. E. L. Spieß, "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.
- [9] R. Battiti and G. Tecchiolli, "The reactive Tabu search," *ORSA Journal on Computing*, vol. 6, no. 2, pp. 126–140, 1994.
- [10] N. Wassan, "A reactive Tabu search for the vehicle routing problem," *Journal of the Operational Research Society*, vol. 57, no. 1, pp. 111–116, 2006.
- [11] T. Crainic, M. Toulouse, and M. Gendreau, "Toward a taxonomy of parallel Tabu search heuristics," *INFORMS Journal on Computing*, vol. 9, no. 1, pp. 61–72, 1997.
- [12] E. G. Talbi, Z. Hafidi, and J.-M. Geib, "A parallel adaptive Tabu search approach," *Parallel Computing*, vol. 24, no. 14, pp. 2003–2019, 1998.
- [13] S. M. Sait, H. Youssef, H. R. Barada, and A. Y. Ahmad, "Parallel Tabu search algorithm for VLSI standard-cell placement," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '00)*, pp. 581–584, May 2000.
- [14] E. Nowicki and C. Smutnicki, "An advanced Tabu search algorithm for the job shop problem," *Journal of Scheduling*, vol. 8, no. 2, pp. 145–159, 2005.
- [15] D. Zhang, Z. Fu, and L. Zhang, "An improved TS algorithm for loss-minimum reconfiguration in large-scale distribution systems," *Electric Power Systems Research*, vol. 77, no. 5-6, pp. 685–694, 2007.
- [16] F. Glover, "Parametric Tabu-search for mixed integer programs," *Computers and Operations Research*, vol. 33, no. 9, pp. 2449–2494, 2006.
- [17] J. Xu, S. Y. Chiu, and F. Glover, "Probabilistic Tabu search for telecommunications network design," *Journal of Combinatorial Optimization*, vol. 1, no. 1, pp. 69–94, 1997.
- [18] Y. Kochetov and E. Goncharov, "Behavior of a probabilistic Tabu search algorithm for the multi stage uncapacitated facility location problem," in *Proceedings of the Operations Research*, pp. 65–70, Springer, 2000.
- [19] D. Ghosh, *A Probabilistic Tabu Search Algorithm for the Generalized Minimum Spanning Tree Problem*, Indian Institute of Management, Ahmedabad, India, 2003, Working Paper no. 2003-07-02.
- [20] W. Ongsakul and P. Bhasaputra, "Optimal power flow with FACTS devices by hybrid TS/SA approach," *International Journal of Electrical Power and Energy Systems*, vol. 24, no. 10, pp. 851–857, 2002.
- [21] D. Gong, Y. Zhou, X. Guo, X. Ma, and L. Ming, "Study on an adaptive Tabu search genetic algorithm," in *Proceedings of the 4th IEEE World Congress on Intelligent Control and Automation*, pp. 3063–3065, June 2002.
- [22] H. Mori and H. Tani, "Fast hybrid method of parallel Tabu search and ordinal optimization for on-line capacitor control in distribution systems," in *Proceedings of the IEEE Power Engineering Society General Meeting*, vol. 3, pp. 1553–1558, July 2003.

- [23] A. Kalinli, "Component value selection for active filters using parallel Tabu search algorithm," *International Journal of Electronics and Communications*, vol. 60, no. 1, pp. 85–92, 2006.
- [24] R. Swarnkar and M. K. Tiwari, "Modeling machine loading problem of FMSs and its solution methodology using a hybrid Tabu search and simulated annealing-based heuristic approach," *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 3, pp. 199–209, 2004.
- [25] Y. Liu, Y. Liu, L. Wang, and K. Chen, "A hybrid Tabu search based clustering algorithm," *Lecture Notes in Computer Science*, vol. 3682, pp. 186–192, 2005.
- [26] J. P. Pedroso and M. Kubo, "Hybrid Tabu search for lot sizing problems," *Lecture Notes in Computer Science*, vol. 3636, pp. 66–77, 2005.
- [27] J.-S. Chen, J. C.-H. Pan, and C.-K. Wu, "Minimizing makespan in reentrant flow-shops using hybrid Tabu search," *International Journal of Advanced Manufacturing Technology*, vol. 34, no. 3-4, pp. 353–361, 2007.
- [28] T. A. A. Victoire and A. E. Jeyakumar, "A Tabu search based hybrid optimization approach for a fuzzy modelled unit commitment problem," *Journal of Electric Power Systems Research*, vol. 76, no. 6-7, pp. 413–425, 2006.
- [29] M. A. Tahir, A. Bouridane, and F. Kurugollu, "Simultaneous feature selection and feature weighting using hybrid Tabu search/K-nearest neighbor classifier," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 438–446, 2007.
- [30] S. Sujitjorn, T. Kulworawanichpong, D. Puangdownreong, and K.-N. Areerak, *Adaptive Tabu Search and Applications in Engineering Design, Frontiers in Artificial Intelligent and Applications*, IOS Press, Amsterdam, The Netherlands, 2006.
- [31] M. Widmer and A. Hertz, "A new heuristic method for the flow shop sequencing problem," *European Journal of Operational Research*, vol. 41, no. 2, pp. 186–193, 1989.
- [32] K.-N. Areerak, T. Kulworawanichpong, and S. Sujitjorn, "Moving towards a new era of intelligent protection through digital relaying in power system," in *Knowledge-Based Intelligent Information and Engineering Systems*, vol. 3215 of *Lecture Notes in Artificial Intelligence*, pp. 1255–1261, Springer, New York, NY, USA, 2004.
- [33] D. Puangdownreong and S. Sujitjorn, "Image approach to system identification," *WSEAS Transactions on Systems*, vol. 5, no. 5, pp. 930–938, 2006.
- [34] D. Puangdownreong and S. Sujitjorn, "Obtaining an optimum PID controller via adaptive Tabu search," *Lecture Notes in Computer Science*, vol. 4432, no. 2, pp. 747–755, 2007.
- [35] N. Sriyingyong and K. Attakitmongkol, "Wavelet-based audio watermarking using adaptive Tabu search," in *Proceedings of the 1st International Symposium on Wireless Pervasive Computing*, pp. 1–5, January 2006.
- [36] S. Brennan and A. Alleyne, "Using a scale testbed: controller design and evaluation," *IEEE Control Systems Magazine*, vol. 21, no. 3, pp. 15–26, 2001.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

