Meta-heuristic algorithm for binary dynamic optimisation problems and its relevancy to timetabling

Ayad Mashaan Turky[1,2], Salwani Abdullah[1] and Nasser R. Sabar[3]
[1]Data Mining and Optimisation Research Group,
Centre for Artificial Intelligence Technology,
Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia
[2]Faculty of Information & Communication Technologies,
Swinburne University of Technology, Victoria 3122, Australia.
[3]The University of Nottingham Malaysia Campus, Jalan Broga, 43500 Semenyih Selangor, Malaysia.
ayadalrashid@gmail.com, salwani@ukm.edu.my, Nasser.Sabar@nottingham.edu.my

1. Introduction

Many real world problems are dynamic in the sense that changes occur during the optimisation process.These problems are more convincing in real world applications than the static ones. This is due to the fact that most of the real world applications are dynamic as the problems differ in the changes that occur in the optimisation environment or the size of the problem increases from time to time [1, 2]. This phenomenon can be illustrated by the following example of a delivery company having to render a service to a set of customers where usually, the number of customers to be served changes on the service schedule due to the length of the contract period. Furthermore, the service that is demanded from the customer could also vary over time. This sort of situation could be considered to be a dynamic problem because the parameters would only be revealed during the delivery process where the number of customers or the demand of the product may increase or decrease.

Much effort has been made to solve dynamic optimisation problems over the recent decade [3]. In solving this problem, a solution method that is able to keep track of the changes is much needed. In addition the solution method should be adaptable in line with the current changes. In contrast to static optimisation problems (where the aim is to find the global optima), the goal of dynamic optimisation problems is to find not only the global optima but also to keep track of changes that usually occur during the optimisation process.

Generally, one could easily remark that the success of these algorithms is due to the incorporated mechanism that manages to maintain the population diversity when dealing with the changes [4]. Even though at present, there are a number of population-based methods applied on dynamic optimisation problems, there is still plenty of room for further research work, since the nature of this problem usually requires an efficient and effective algorithm that would quickly respond to changes.

Harmony search algorithm has been used to successfully solve a number of static optimization problems [5-8]. In this work, we investigate the applicability of the harmony search algorithm in tackling binary dynamic optimisation problems where four standard binary test functions are used. Based on the performance obtained, the proposed approach will later be employed on dynamic combinatorial optimisation problems such as dynamic vehicle routing problems and dynamic job shop scheduling problems.

2. Solution Approach

In this section, we present our proposed Harmony Search Algorithm (HSA) for solving binary dynamic optimisation problems. In this work, different mechanisms have been used to maintain the population diversity and their hybridisation with the HSA. This is referred to as Hybrid Harmony Search. HSA is one of the recent stochastic population-based meta-heuristic optimisation algorithms proposed by Geem et al. [9]. HSA has five steps as depicted in Fig. 1.

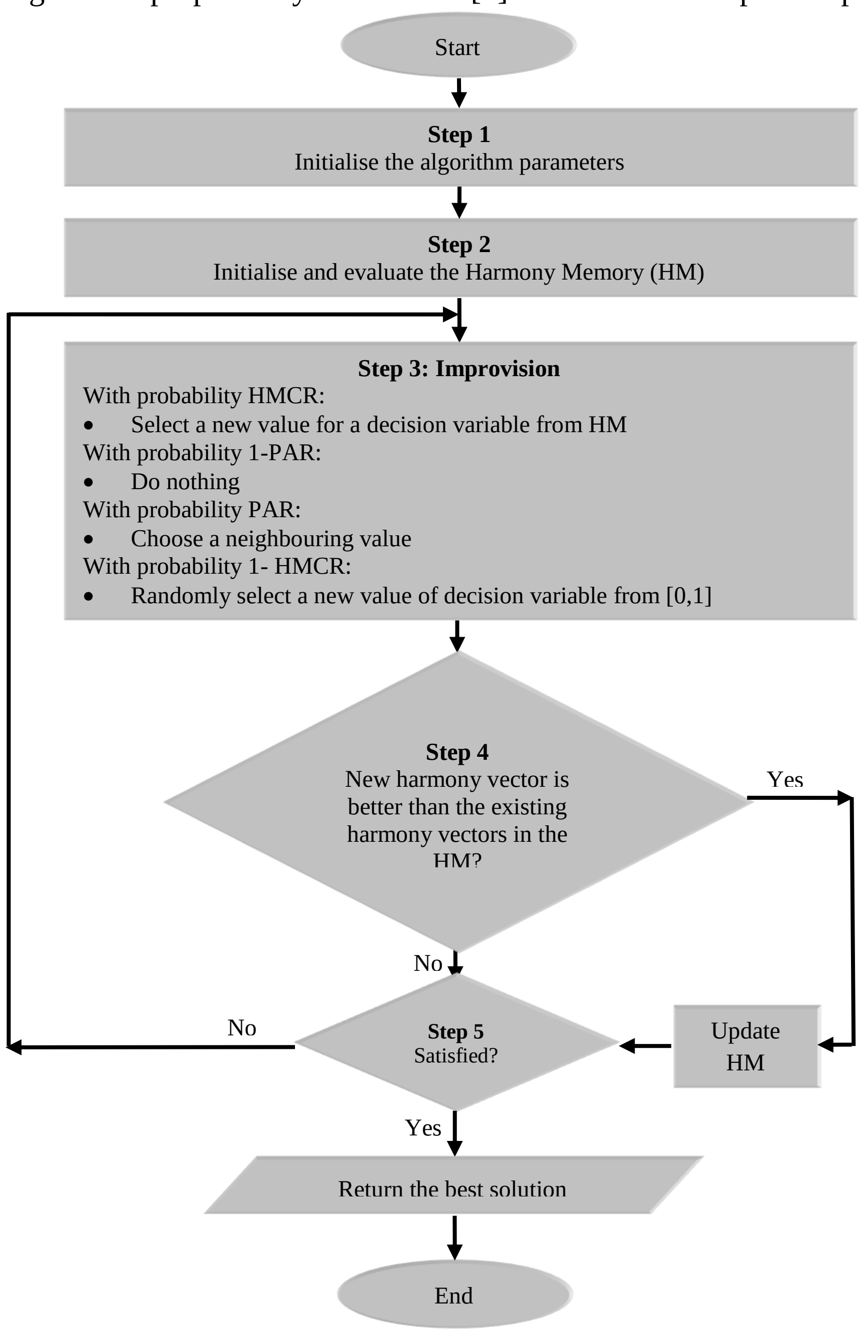

Figure1: Steps in HSA

In Step 1, parameters of HSA (as in Table 1) are initialised. Harmony Memory (HM) is initialised and evaluated in Step 2. Step 3 is an improvisation process where Harmony Memory Consideration Rate (HMCR) parameter is used during the improvisation process in order to determine whether the value of a decision variable of a new harmony will be selected from the HM or will it be generated at random from the possible range that takes a value between [0, 1]. The probability of randomly selecting the decision variable value from the possible range is given as 1-HMCR. Pitch Adjusting Rate (PAR) parameter is used to decide either the values of decision variables (that have been selected from the HM) will be modified or maintained. PAR takes a value between [0, 1]. In Step 4, the HM will be updated and if the termination criterion is satisfied then the process will be terminated (Step 5).

In order to cope with the dynamic changes, harmony search algorithm needs to keep track of the changes during the search process. This is needed because the changes in the problem may change the current local optima into global optima and vice versa [1]. In addition, it is also shown in the literature that the developed algorithms for stationary problems cannot be directly used to solve dynamic problems [1, 10].

Therefore, to handle this problem, the HSA has been hybridised with three population diversity mechanisms, (i) HSA with random immigrant, HSA-I, (ii) HSA with memory mechanism, HSA-M, and (iii) HSA with memory based immigrant mechanism, HSA-MI.

- HSA-I: First mechanism where HSA is hybridised with random immigrant. In this approach, at each of the generation a subset of solutions is generated at random and is used to replace the worst solutions in the HM. In this paper, the number of solutions are fixed to be replaced at every iteration as *rs*=HMS*0.2 where *rs* represents the number of replaced solutions.

- **HSA-M:** Second mechanism where HSA is hybridised with a memory mechanism. In this approach, a subset of best solutions is kept and will be re-inserted in the HM once changes are detected.

- **HSA-MI:** In the third mechanism where HSA is hybridised with a random immigrant and a memory based mechanism in order to maintain the diversity of HM.

3. Results and Discussions

The performance of the proposed approaches is verified on four well-known binary dynamic optimisation test functions i.e. OneMax, Plateau, Royal Road, and Deceptive. The parameter values of HSA which is based on our preliminary tests are presented in Table 1.

Table 1 HSA parameter values

Parameters	***Description***	***Tested range***	***Suggested value***
HMS	Harmony memory size HMS= 1 to 100	10-200	100

HMCR	Harmony memory consideration rate... $0 < HMCR < 1$	0.1-0.99	0.6
RCR	Random consideration rate	-	RCR=1-HMCR
PAR	Pitch adjustment rate $0 < PAR < 1$	0.1-0.99	0.3
NI	Number of improvisations or iterations	-	500000 function evaluations

Our hybridisation approaches are compared against the well-known methods in the literature. The algorithms in comparison are presented in Table 2.

Table 2 Acronyms of compared methods

#	*Symbol*	*References*
1	MIGA	[6]
2	MEGA	[7]
3	AHMA	[1]
4	MRIGA	[6]

In order to measure the performance of our proposed algorithm the overall offline performance (the best-of-generation fitness) is calculated over 30 runs (with different initial solutions and seeds) based maximisation of Eq. 1.

$$\overline{F}_{BOG} = \frac{1}{G}\sum_{i=1}^{G}\left(\frac{1}{N}\sum_{j=1}^{N} F_{BOG_{ij}}\right) \tag{1}$$

where G is the total number of generations, N is the total number of runs and F_{BOGij} is the best of generation fitness of generation i of run j. Our results as well as other methods in comparison are presented in Table 3.

Table 3: Comparison of Results

Function name	**HSA-I**	**HSA-M**	**HSA-MI**	**% Deviation**	**MIGA**	**MEGA**	**AHMA**	**MRIGA**
OneMax	91.67	90.42	**96.01**	**	94.0	79.3	95.89	80.8
Plateau	72.21	68.41	**84.91**	**	-	-	62.88	-
Royal Road	64.76	63.96	**66.19**	**	-	-	52.52	-
Deceptive	76.39	73.11	**85.97**	**	71.1	83.1	85.75	68.6

'-': no results are reported. '**': our algorithm is better than others.

As shown the in Table 3, HSA-MI outperforms other methods in all test functions (presented in bold), in which we believe this is due to the idea of of hybridising a random immigrant and a memory based mechanism in order to maintain the harmony memory (population) diversity.

4. Dynamic Optimisation Problems and Its Relevancy to Timetabling Problems

Dynamic optimisation problems however, present a greater challenge to the research community since the problem parameters are either revealed or change during the course of the on-going optimization [4]. These problems are more convincing in real world applications than the static ones. This is due to the fact that most of the real world applications are dynamic, as the problems in the sense that the environment is subjected to changes or the size of the problem increases from time to time [11].

Timetabling problems have been frequently studied because of their wide range of applications such as school timetabling, transport scheduling, job shop scheduling, vehicle routing, and patient admission scheduling problems. In timetabling problems, two main difficulties are encountered: (i) often over-constrained and optimisation criteria are hard to define; (ii) intrinsically dynamic where activities, resources or constraints are sometimes unknown or can often change at the last moment [12].

The relevancy between dynamic optimisation problems with timetabling problems can be expressed through examples on:

- School timetabling: the dynamic part of the schedule is more related to logistic needs and unexpected events such as link with timetabling of other years in terms sharing common resources and, inside and outside teachers availability [12].
- Train scheduling: the information such as train arrival times, train lengths, train speeds are available before solving the problem in the static scheduling environment. However, in dynamic scheduling environment (which mimics the real-world problem), the information of only arrived trains is considered known, then the schedule of the new train and the trains currently in the network should be generated, given no information of later trains [13].
- Job shop scheduling: most manufacturing systems operate in dynamic environment where unexpected disruptions occur during the manufacturing process such as machine breakdowns, material shortage, random job releases, and job cancellations and due date and time processing changes. The disruption will produce uncertainty in the sequence of operation, i.e. the time taken to repair the broken machine [14].
- Dynamic vehicle routing: dynamic scenarios have become more common in vehicle routing. The most common source of dynamism in vehicle routing is the online arrival of customer *requests* (demand for goods and services) during the operation, dynamic travel and service time and vehicle availability, and breakdown of vehicles. These source of dynamism caused schedulers to update the generated timetable [15, 16].
- Patient admission scheduling: it concerns in assigning patients to bed in a hospital. In order to tailor real scenario, several real-world features, such as the presence of emergency patients, uncertainty in the length of stay, and the possibility of delayed admissions are included [17].

The above examples show the important on tackling dynamic optimisation problems since most of the real world problems are dynamic in nature. These disruptions or random occurrences lead timetable officers to develop a new schedule from scratch or reschedule the existing schedule in order to cater the changes.

5. Conclusion and Future Work

The overall goal of the work presented in this paper is to investigate the performance of the hybrid harmony search algorithm in maintaining the population diversity in addressing binary dynamic optimisation problems. In this work, three kinds of population diversity mechanisms are presented i.e. the random immigrant, memory mechanism, and memory based immigrant mechanism. Initial experiments show that the harmony search with memory based immigrant mechanism outperforms two hybrid approaches presented here, and also managed to obtain better offline performance in comparison to other available approaches in the literature. For future work, the proposed method will be investigated on other dynamic combinatorial optimization problems such as dynamic vehicle routing and dynamic job shop scheduling problems.

References

[1] H. Wang, D. Wang, and S. Yang, "A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems," *Soft Computing,* vol. 13, pp. 763-780, 2009.

[2] E.-G. Talbi, "A taxonomy of hybrid metaheuristics," *Journal of heuristics,* vol. 8, pp. 541-564, 2002.

[3] S. Yang, Y. S. Ong, and Y. Jin, *Evolutionary computation in dynamic and uncertain environments* vol. 14: Springer, 2007.

[4] S. Yang, "Non-stationary problem optimization using the primal-dual genetic algorithm," in *The 2003 Congress on Evolutionary Computation. CEC'03.* , 2003, pp. 2246-2253.

[5] M. Hadwan, M. Ayob, N. R. Sabar, and R. Qu, "A harmony search algorithm for nurse rostering problems," *Information Sciences,* vol. 233, pp. 126-140, 2013.

[6] A. M. Turky and S. Abdullah, "A multi-population harmony search algorithm with external archive for dynamic optimization problems," *Information Sciences,* vol. 272, pp. 84-95, 2014.

[7] N. R. Sabar and G. Kendall, "Using Harmony Search with Multiple Pitch Adjustment Operators for the Portfolio Selection Problem.," *2014 IEEE Congress on Evolutionary Computation (IEEE CEC 2014), 6-11 July 2014, Beijing, China.,* 2014.

[8] A. M. Turky, S. Abdullah, and N. R. Sabar, "A Hybrid Harmony Search Algorithm for Solving Dynamic Optimisation Problems.," *The International Conference on Computational Science-ICCS, 10-12 June 2014, Cairns, Australia.,* 2014.

[9] Z. W. Geem, J. H. Kim, and G. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation,* vol. 76, pp. 60-68, 2001.

[10] S. Yang, "Genetic algorithms with memory-and elitism-based immigrants in dynamic environments," *Evolutionary Computation,* vol. 16, pp. 385-416, 2008.

[11] J. Branke and H. Schmeck, "Designing evolutionary algorithms for dynamic optimization problems," in *Advances in evolutionary computing*, ed: Springer, 2003, pp. 239-262.

[12] H. Cambazard, F. Demazeau, N. Jussien, and P. David, "Interactively solving school timetabling problems using extensions of constraint programming," in *Practice and Theory of Automated Timetabling V*, ed: Springer, 2005, pp. 190-207.

[13] J. Rodriguez, "A constraint programming model for real-time train scheduling at junctions," *Transportation Research Part B: Methodological,* vol. 41, pp. 231-245, 2007.

[14] G. Chryssolouris and V. Subramaniam, "Dynamic scheduling of manufacturing job shops using genetic algorithms," *Journal of Intelligent Manufacturing,* vol. 12, pp. 281-293, 2001.

[15] A. Haghani and S. Jung, "A dynamic vehicle routing problem with time-dependent travel times," *Computers & operations research,* vol. 32, pp. 2959-2986, 2005.

[16] H. N. Psaraftis, "Dynamic vehicle routing: Status and prospects," *annals of Operations Research,* vol. 61, pp. 143-164, 1995.

[17] S. Ceschia and A. Schaerf, "Modeling and solving the dynamic patient admission scheduling problem under uncertainty," *Artificial intelligence in medicine,* vol. 56, pp. 199-205, 2012.