A MODIFIED HARMONY SEARCH FOR OFFICE-SPACE-ALLOCATION

Mohammed Azmi Al-Betar[1,2], Mohammed A. Awadallah[1], Ahamad Tajudin Khader[1], Phuah Chea Woon[1], Iyad Abu Doush[3]

[1] School of Computer Sciences, Universiti Sains Malaysia (USM), 11800, Pulau Pinang, Malaysia.
[2] Department of Computer Science, Jadara University, Irbid, Jordan.
[3] Department of Computer Sciences, Yarmouk University, Irbid, Jordan.
E-mail: mohbetar@cs.usm.my

Abstract

Harmony search (HS) algorithm is a recent evolutionary algorithm which mimics the musical improvisation process. It has been successfully applied to a wide range of combinatorial optimization problems. Office-Space-Allocation (OSA) is the process of distributing given limited spaces to given resources in accordance with given constraints: hard and soft. Hard constraints are mandatory to satisfy while the soft constraints are desired but not absolutely essential. The main aim is to find a solution with the least violations of soft constraints while the hard constraints are satisfied and the best usage of spaces is achieved. In this paper, HS is modified and adapted for OSA. The modification includes two HS operators: i) memory consideration where the global-best concept of Particle Swarm Optimization is borrowed and employed; and ii) the pitch adjustment has been designed to be as a local search agent with three effective neighbourhood structures. The proposed Modified Harmony Search Algorithm (MHSA) is evaluated using three datasets from Nottingham and Wolverhampton universities. Interestingly, MHSA obtained two new best results, and a competitively comparable result for the third dataset.

Keywords - Optimization. Harmony Search. Timetabling. Office-Space-Allocation.

1 INTRODUCTION

Office-Space-Allocation (OSA) problem is concerned with distributing a set of limited spaces among a set of resources, to ensure that all resources are given the necessary spaces for optimal space utilization. In academic institutions, the resources can be staff, students, lecture rooms, laboratories, and storage rooms. Notice that the addition or deletion of one of the resources results in reallocation of specific resource into another time slot. This problem is subject to two types of constraints: hard and soft. Fulfilling the hard constraint is required to consider the generated solution valid, while fulfilling the soft constrains are preferable but not mandatory. Practically, it is almost impossible to fulfil all constraints due to the combinatorial nature of the problem [1].

The space allocation problem has been studied by several researchers from the operations research and artificial intelligence fields. Different techniques have been used to solve the problem such as Integer Goal Programming [2], Linear Goal Programming [3], Hill Climbing [4], Genetic Algorithm [4], Simulating Annealing [4], and Memetic Algorithm [4].

Harmony Search (HS) algorithm is an evolutionary algorithm proposed by Geem et al., [5]. It has been successfully applied to many optimization problems such as shop scheduling, blocking permutation flow, the multi-objective flowshop scheduling, 0-1 Knapsack, nurse rostering, timetabling [6-13], and others in the surveys [14, 15]. HS algorithm contains many features that make it very good in exploring the solution search space: (i) it can be used for both continuous and discrete problems, (ii) initializing the decision variables not required, (iii) it iteratively generates a new solution by considering all existing solutions [16], and (iv) it has novel stochastic derivative [17]. The performance of HS algorithm is improved by tuning parameters [18, 19], or hybridizing it with other methods [8, 20-22].

The objective of this paper is to modify the HS algorithm for OSA. The memory consideration and pitch adjustment phases are modified to enhance search capabilities of the HS algorithm. The proposed Algorithm is called Modified Harmony Search algorithm (MHSA). For evaluating MHSA, a ***de***

facto dataset is used that includes two samples extracted from the University of Nottingham, and one from the University of Wolverhampton. Experimentally, the MHSA has successfully provided two new results for two datasets, and a comparable result for the third dataset.

2 PROBLEM DESCRIPTION

The OSA is defined as a problem of distributing a set of limited spaces among a set of resources subject to two types of constraints: ***Hard*** and ***Soft***. The satisfaction of hard constraints is required, while the violations of the soft constraints must be reduced as much as possible. A feasible solution must satisfy all hard constraints, and reduce the number of violated soft constraints. The setting for the space allocation used in this study is identified as below:

- ***No sharing*.** No any two resources can be double booked.
- ***Be located in*.** A particular room is assigned a specific resource.
- ***Be adjacent to*.** A particular resource should be allocated to room adjacent to another resource.
- ***Be away.*** No two resources should be placed next to each other.
- ***Be together with.*** The same room can simultaneously hold two particular resources.
- ***Be grouped with.*** A group of resources should be allocated close to each other.

Table 1. Notations used to formulate the office-space-allocation problem

Symbol	Description
M	Number of available rooms.
N	Number of resources.
ß	Set of available room *ß*=$\{r_1, r_2, \cdots, r_M\}$
£	Set of available resources *£*=$\{p_1, p_2, \cdots, p_N\}$
H	Number of hard constraints of the form Z(*r*) =true, where $1 \le r \le H$.
S	Number of soft constraints of the form Z(*r*) =true, where $1 \le r \le S$.
c_i	Capacity of the room *i.*
w_j	Space requirement of resource *j*.
$x_{i,j}$	1 if the resource *j* is allocated at room *i*, 0 otherwise.

The notations for the office-space-allocation problem are given in Table 1. The solution of the space allocation problem is represented by the two dimensional array $\boldsymbol{x}$ = (ß×£), where ß is the set of rooms, and £ is the set of the resources. The notations and formulations in this paper are adapted from [23].

Based on the Notation of the OSA parameters, the objective function is formulated as follows:

$$\textbf{Min} \; f(\boldsymbol{x}) = f_1(\boldsymbol{x}) + f_2(\boldsymbol{x})$$

Subject to

$$\sum_{j=1}^{N} \sum_{i=1}^{M} x_{i,j} = 1$$

And

$$Z(r) = true \quad for \quad r = 1, 2, \cdots, H$$

Where $f_1(x)$ the space misuse function and the violation of the soft constraints are computed by $f_2(x)$.

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{M} WP_i + \sum_{i=1}^{M} OP_i$$

$$f_2(\boldsymbol{x}) = \sum_{r=1}^{S} SCP_r$$

Note that the space misuse for each room is calculated using one of the following two states: wasted capacity or overused capacity of the room. Each room *i* has only one value for WP_i or OP_i, where *WP* refers to the wasted capacity and the *OP* refers to the capacity overused. The wasted capacity quantity for each room *i* is computed as follows:

$$WP_i = \max\left(0, c_i - \sum_{j=1}^{N} x_{i,j} . w_j\right)$$

$$OP_i = \max\left(0, 2\left(\sum_{j=1}^{N} x_{i,j} . w_j - c_i\right)\right)$$

3 MODIFIED HS ALGORITHM FOR OFFICE-SPACE-ALLOCATION

The Harmony Search (HS) algorithm is an optimization algorithm that mimics the musical improvisation process. On a musical setting, a musician plays the pitches of the instrument based on his/her historical skills or randomness. HS algorithm mimics this process by assigning values to decision variables according to accumulative search or random value within the range of decision variable. HS algorithm starts with a population of solutions kept in Harmony Memory (HM). At each iteration, the new solution (i.e. new harmony) is generated based on three rules: (i) *Memory Consideration*, this is similar to crossover in Genetic Algorithm (GA) which is used to make use of the accumulative search. (ii) *Random Consideration*, this is similar to mutation process in GA which is used as a diversification agent. And (iii) *Pitch Adjustment* which works is used as a local search agent. The new harmony will keep substituting the worst solution in HM if better, and this process is repeated until a stop criteria is met.

The five main steps of the proposed Modified Harmony Search algorithm (MHSA) used for tackling OSA problem are described below:

3.1 Initializing space-allocation variables and MHSA parameters

Space allocation problem variables are extracted from the problem datasets. The dataset contains information about rooms, the room capacity, the room capacity required by each resource, and the hard and soft constraints. Moreover, during this step the objective function $f(\boldsymbol{x})$ is formulated to evaluate the solution improvised by HS algorithm. Fig.1 shows the representation of the solution vector $\boldsymbol{x}$.

decision variable →	x_1	x_2	x_3	x_4	…	x_{N-1}	x_N

resource →	p_1	p_2	p_3	p_4	$\cdots$	p_{N-1}	p_N
room →	r_5	r_3	r_{30}	r_1	$\cdots$	r_M	r_2

Fig. 1. Solution $\boldsymbol{x}$ representation

The next step is to set the parameters of the MHSA: (i) The Harmony Memory Consideration Rate (HMCR), the percentage of selecting a value that either comes from memory or generated randomly. (ii) The Harmony Memory Size (HMS), specifies the number of solutions stored in HM (i.e., population size). (iii) The Pitch Adjustment Rate (PAR), specifies the local improvement rate. And (iv) the Number of Iterations (NI) represents the number of iterations determined to solve this problem.

3.2 Initializing the Harmony Memory

The Harmony Memory (HM) is a matrix used to keep a set of solution vectors as determined by HMS (see (1)). During this step each row in the HM represents a complete space allocation solution which is generated using the peckish method. Peckish method is a heuristic method proposed in [10] to generate feasible space allocation solutions. The pseudo-code of the peckish method is presented in Table 2. The solution has to comply with all hard constraints and all resources must be allocated in the suitable rooms.

$$HM = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_N^{HMS} \end{bmatrix} = \begin{bmatrix} f(\boldsymbol{x}^1) \\ f(\boldsymbol{x}^2) \\ \\ f(\boldsymbol{x}^{HMS}) \end{bmatrix}. \tag{1}$$

Table 2. Peckish heuristic method pseudo-code

```
While all resources are not allocated
   K = N / 3
   Randomly select an unallocated resource j
   Randomly select a number of K rooms which satisfy 1/2 × w_j ≤ w_j ≤ 3/2 × w_j
   Select the best room from K rooms with the minimum penalty
   Allocate the resource j to the best room
End loop
```

3.3 Initializing the Harmony Memory

During this step, a new harmony is generated by relying on three operators: memory consideration, random consideration, and pitch adjustment. The feasibility of the New Harmony $\boldsymbol{x}' = (x'_1, x'_2, x'_3, \cdots, x'_N)$ must be preserved (i.e., satisfy hard constraints). If the feasibility is not achieved during the search, the improvisation process will take over again.

Memory Consideration. This operator determines the rate of selecting the value of the decision variable x'_j from the best solution $\boldsymbol{x}^{best}$ stored in HM, with probability HMCR, where HMCR $\in$ [0,1]. If this operator fails to obtain any feasible value, random consideration will take care of that value.

Random Consideration. The decision variable x'_j with probability 1-HMCR, will be assigned a room from the set of the available rooms $x'_j \in$ *ß*.

Pitch Adjustment. The decision variable x'_j assigned by the memory consideration will be locally stepped with probability PAR, where PAR $\in$ [0,1]. For OSA problem, this operator is divided into three procedures for improving the solution locally: *Move*, *Swap*, and *Interchange*, each of which controls the range of PAR as follows:

$$x'_j \leftarrow \begin{cases} Move & 0 \le rand < (PAR/3), \\ Swap & (PAR/3) \le rand < (2.PAR/3), \\ Interchange & (2.PAR/3) \le rand < PAR. \end{cases}$$

Where rand $\in$ [0,1]. The three pitch adjustment procedures work as follows:

- ***Move.*** With a probability range [0, PAR/3), the resource of the allocation x'_j moves from the current room to another room selected randomly.
- ***Swap.*** With a probability range [PAR/3, 2.PAR/3), the room of the allocation x'_j interchanges with the room of the resource x'_t selected randomly.
- ***Interchange.*** With a probability range [2.PAR/3, PAR], all resources interchange between two randomly-selected rooms.

Note that any local changes that do not improve the new harmony solution or generate infeasible solution will be discarded.

3.4 Update the Harmony Memory

If the new harmony has better fitness value than the worst solution stored in HM, the worst solution will then be substituted by the new one.

3.5 Check the Stop condition

The Steps 3.3 and 3.4 of MHSA are repeated as many as NI.

4 EXPERIMENTS AND RESULTS

Three datasets are used to study the effectiveness of the proposed MHSA to solve OSA problems. Two datasets form the Nottingham university (i.e. Nott1, Nott1d), and one dataset from the university of Wolverhampton (i.e. Wolver1) are used. The characteristics of the three datasets are shown in Table 3. The proposed MHSA is coded using C# under windows Vista on an Intel Machine with CoreTM processor 2.66GHz, and 4GB RAM.

Table 3. Datasets characteristics

	Nott1		Nott1d		Wolver1	
Resources	158		56		115	
Rooms	131		56		115	
Constraints	Hard	Soft	Hard	Soft	Hard	Soft
Not sharing	100	58	10	0	115	0
Be allocated in	0	35	0	9	0	0
Be adjacent to	5	15	4	9	0	0
Be away from	6	14	1	2	0	0
Be together with	0	20	0	0	0	0
Be grouped with	0	10	0	9	0	0

A series of experiments when implementing MHSA to solve OSA problems has been conducted. The parameters of MHSA are set as follows: HMS=50, HMCR,0.99, and PAR=0.5, NI=10,000.

The best results obtained by the MHSA for the three datasets are compared with other methods previously proposed for these datasets. Table 4 shows the best results obtained by the MHSA, and other methods best results. Note that the best results among the different methods are highlighted in bold. The MHSA obtained high quality solutions for the three datasets, where the MHSA achieved the best results for the Nott1d and Wolver1 datasets, and comparable results for the Nott1 dataset. Here, the MHSA is able to obtain best results for small problems (i.e. Nott1d, and Wolver1) with a small number of resources and constraints. However, MHSA obtained the fourth best result in Nott1 dataset due to the fact that Nott1 is the hardest among the three datasets based on a number of resources and constraints.

Table 4. Compering the results of MHSA with other methods

Method	Reference	Nott1	Nott1d	Wolver1
Simulated Annealing	Landa-Silva and Burk[23]	543.7	-	-
Tabu Search		491.2	-	-
Asynchronous cooperative local search		**482.2**	-	-
Hybrid Genetic Algorithm		525.9	-	-
Integer programming	Űlker and Landa-Silva [24]	-	202.7	634.20
Mathematical programming		-	202.73	634.19
MHSA		539.35	**200.1**	**634.19**

5 CONCLUSION

In this paper, the Harmony Search (HS) algorithm is modified for Office-Space-Allocation problem. The modification encompasses two HS operators: i) the process of memory consideration is altered to select from the best solution in the population during the search and ii) the pitch adjustment is designed to function as a local search agent. Office-Space-Allocation problem requires distributing a set of limited spaces to a set of resources and abiding by two types of constraints: hard and soft. The hard constraints have to be met while the soft constraints are expected to be met as much as possible. The quality of the solution is determined based on satisfaction of the soft constraints and best usage of spaces. The harmony search algorithm is an evolutionary algorithm which mimics the musical improvisation process. The harmony search starts with a set of solutions stored in harmony memory. At each iteration, the new harmony is generated based on three operators: memory consideration, random consideration, and pitch adjustment. The new harmony will replace the worst solution in harmony memory if better. The proposed Modified Harmony Search Algorithm (MHSA) is evaluated using three datasets from Nottinghamand Wolverhampton universities. The MHSA is able to provide best results for two datasets, and a comparable result for others.

6 ACKNOWLEDGMENT

This work has been supported by Grant number 1001/PKOMP/817047.

7 REFERENCES

[1] Kellerer, H. and U. Pferschy, *Cardinality constrained bin-packing problems.* Annals of Operations Research, 1999. **92**: p. 335-348.

[2] Ritzman, L., J. Bradford, and R. Jacobs, *A multiple objective approach to space planning for academic facilities.* Management Science, 1979: p. 895-906.

[3] Benjamin, C.O., I.C. Ehie, and Y. Omurtag, *Planning facilities at the University of Missouri-Rolla.* Interfaces, 1992: p. 95-105.

[4] Burke, E., et al., *Three methods to automate the space allocation process in UK universities.* Practice and Theory of Automated Timetabling III, 2001: p. 254-273.

[5] Geem, Z.W. and J.H. Kim, *A new heuristic optimization algorithm: harmony search.* Simulation, 2001. **76**(2): p. 60-68.

[6] Al-Betar, M., A. Khader, and J. Thomas, *A combination of metaheuristic components based on harmony search for the uncapacitated examination timetabling.* Practice and Theory of Automated Timetabling, ISBN, 2010: p. 08-538.

[7] Al-Betar, M.A., A.T. Khader, and F. Nadi. *Selection mechanisms in memory consideration for examination timetabling with harmony search*. in *the 12th annual conference on Genetic and evolutionary computation*. 2010. ACM.

[8] Al-Betar, M.A., A.T. Khader, and M. Zaman, *University Course Timetabling Using a Hybrid Harmony Search Metaheuristic Algorithm.* Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2012(99): p. 1-18.

[9] Awadallah, M.A., et al., *Harmony Search with Greedy Shuffle for Nurse Rostering.* International Journal of Natural Computing Research (IJNCR), 2012. **3**(2): p. 22-42.

[10] Awadallah, M.A., et al., *Global best Harmony Search with a new pitch adjustment designed for Nurse Rostering.* Journal of King Saud University-Computer and Information Sciences, 2012.

[11] Frosolini, M., M. Braglia, and F.A. Zammori, *A modified harmony search algorithm for the multi-objective flowshop scheduling problem with due dates.* International Journal of Production Research, 2011. **49**(20): p. 5957-5985.

[12] Wang, L., Q.K. Pan, and M. Fatih Tasgetiren, *Minimizing the total flow time in a flow shop with blocking by using hybrid harmony search algorithms.* Expert Systems with Applications, 2010. **37**(12): p. 7929-7936.

[13] Zou, D., et al., *Solving 0-1 knapsack problem by a novel global harmony search algorithm.* Applied Soft Computing, 2011. **11**(2): p. 1556-1564.

[14] Alia, O.M. and R. Mandava, *The variants of the harmony search algorithm: an overview.* Artificial Intelligence Review, 2011. **36**(1): p. 49-68.

[15] Ingram, G. and T. Zhang, *Overview of applications and developments in the harmony search algorithm.* Music-inspired harmony search algorithm, 2009: p. 15-37.

[16] Lee, K.S. and Z.W. Geem, *A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice.* Computer methods in applied mechanics and engineering, 2005. **194**(36): p. 3902-3933.

[17] Geem, Z.W., *Novel derivative of harmony search algorithm for discrete design variables.* Applied mathematics and computation, 2008. **199**(1): p. 223-230.

[18] Geem, Z.W. and K.B. Sim, *Parameter-setting-free harmony search algorithm.* Applied Mathematics and Computation, 2010. **217**(8): p. 3881-3889.

[19] Yadav, P., et al., *An Intelligent Tuned Harmony Search algorithm for optimisation.* Information Sciences, 2012.

[20] Al-Betar, M.A., et al., *Novel selection schemes for harmony search.* Applied Mathematics and Computation, 2011.

[21] Fesanghary, M., et al., *Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems.* Computer methods in applied mechanics and engineering, 2008. **197**(33): p. 3080-3091.

[22] Kaveh, A. and S. Talatahari, *Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures.* Computers & Structures, 2009. **87**(5-6): p. 267-283.

[23] Landa-Silva, D. and E.K. Burke, *Asynchronous Cooperative Local Search for the Office-Space-Allocation Problem.* INFORMS Journal on Computing, 2007. **19**(4): p. 575-587.

[24] Ulker, O. and D. Landa-Silva, *A 0/1 Integer Programming Model for the Office Space Allocation Problem.* Electronic Notes in Discrete Mathematics, 2010. **36**: p. 575-582.