



ADAPTIVE HARMONY SEARCH FOR OPTIMIZING CONSTRAINED RESOURCE ALLOCATION PROBLEM

Amol C. Adamuthe, Tushar R. Nitave

Department of Information Technology, Rajarambapu Institute of Technology, Rajaramnagar, MS, India
amol.admuthe@gmail.com, tushar.nitave@gmail.com

Paper history:

Received 4 September 2018
Received in revised form 8 November 2018
Accepted 28 December 2018
Available online 31 December 2018

Keywords:

Harmony Search Algorithm (HS);
Bin packing problem (BPP);
Virtual machine placement problem (VMP);
Meta-heuristic.

Abstract: Resource Allocation problem is finding the optimal assignment of finite available resources to tasks or users. Resource allocation problems refer to a wide range of applications such as production, supply chain management, transportation, ICT technologies, etc. Resource allocation problems are NP-hard in nature where the objective is to find the optimal allocations satisfying given constraints. Harmony search (HS) algorithm is a meta-heuristic population based algorithm found good for solving different optimization problems. This paper presents adaptive harmony search (AHS) for solving one-dimensional bin packing problem (BPP) and multi-objective virtual machine placement problem (VMP). The proposed real coded solution representation supports partial constraint satisfaction. Adaptive pitch adjustment rate (PAR) based on population diversity improves the performance of harmony search algorithm. Results show that proposed HS gives optimal solution for 50 BPP instances with 100 % success rate. The performance reduced for large instances of BPP. The proposed weighted AHS for multi objective VMP problem gives better results than genetic algorithm.

Copyright © Research Institute for Intelligent Computer Systems, 2018.
All rights reserved.

1. INTRODUCTION

Resource allocation problems are NP-hard problems in which the objective is optimal allocation of limited resources by satisfying constraints. Resources can be manpower, assets, raw material or anything that has limited supply which can be used to achieve goals and the goals can be minimizing the cost or maximizing the profit depending on the problem [1]. It has a wide range of applications in project management, economics, storage management, scheduling, etc. The bin packing problem is an NP-hard resource allocation problem in which objects of different sizes are to be packed into bins of homogenous or heterogeneous size in such a way that minimum bins are required. The constraint needs to be satisfied in such a way that the sum of weights of objects in a bin must be no more than its maximum capacity. Bin packing problem is a well-known constraint satisfaction problem because of its wide real-life applications such as timetabling problems, scheduling problems.

Cloud computing is one of the fast-growing technology. One of the most important uses of cloud computing is that, it provides scalable/elastic resources for the application hosted on it as well as it makes full utilization of the available resources. In order to do so, a good virtual machine allocation/provisioning strategy is needed. In virtual machine management whenever a user requests a machine, an instance of virtual machine is created and is delivered to the customer. This task of creating VMs and assigning it with maximum profit and minimal wastage is known as virtual machine management.

Different resource allocation problems when solved using traditional algorithms get trapped in local maxima/minima and hence it is difficult to solve complex problems. Many researchers experimented GPU parallelization to accelerate the performance of these algorithms [2]. Most of the traditional algorithms are problem specific, which is one of the important limitations, whereas, meta-heuristic algorithms can be applied to a wide range

of problems. To solve these problems various meta-heuristic algorithm, namely Ant colony optimization (ACO), Genetic algorithm (GA), Harmony search algorithm (HS), Artificial bee colony algorithm (ABC) have been experimented by researchers. Paper [3] shows that performance of metaheuristic algorithms improves with hybridization and parallel implementation using CUDA.

There are various popular meta-heuristic algorithms such as particle swarm optimization (PSO), GA, HS, artificial bee colony, differential evolution (DE), Tabu search, etc. Harmony search algorithm is a meta-heuristic population based algorithm in which the improvisation process is analogous to that of composer in a musical production. The number of publications about harmony search algorithm is increasing rapidly [4]. It has been used in solving various problems such as flow control [5], optimization of truss structures [6], etc. Music generally consists of a series of combined sounds that give rise to elegant effect. It takes combination of three factors - pitch, tone and intensity to produce any single sound. Music composer improvises the pitch of the instruments by search for perfect harmony.

This paper presents adaptive harmony search algorithm for resource allocation problem. The objective of this paper is to optimize one-dimensional bin packing problem and multi-objective virtual machine placement problem using harmony search algorithm.

The contribution of the paper is twofold:

1. Firstly, adaptive harmony search algorithm is proposed.
2. Secondly, initialization strategy that produces partially feasible solutions is offered.

The next section describes different strategies used for solving one-dimensional bin packing problem and multi-objective virtual machine placement problem. Section 3 describes the problems and section 4 describes the proposed harmony search algorithm. Finally, section 5 shows the results tested on various datasets. At last, conclusions are stated in section 6.

2. RELATED WORK

To solve the bin packing problem and virtual machine placement problem researchers have experimented several traditional as well as meta-heuristic algorithms.

2.1 BIN PACKING PROBLEM

In [7] authors, applied genetic algorithm (GA) for bin packing problem and results showed that GA has better performance when compared with First Fit Decreasing (FFD) and Best Fit Decreasing (BFD) heuristic. Dokeroglu and Cosar [8], proposed

evolutionary grouping genetic meta-heuristic and bin-oriented heuristics which uses parallel computation techniques to solve large instances of one-dimensional BPP. Paper [9] used the concept of weight annealing for one-dimensional BPP. This strategy generated a high-quality solution in less time along with several optimal solutions.

A genetic algorithm which performs several perturbations for improving current solution was presented for one-dimensional bin packing problem [10]. This algorithm is tested on benchmark instances and yields good results compare to best known methods. In [11], proposed hybrid steady-state grouping genetic algorithm and improved version of Perturbations MBS' heuristic. A hyper-heuristic, which uses Evolutionary Algorithm (EA) to train a k-nearest neighbor algorithm was presented in [12] to solve one-dimensional bin packing problem.

Paper [13] proposed better-fit heuristic for one-dimensional BPP, which outperforms off-line best-fit-decreasing heuristic and has better performance than on-line best-fit heuristic. In [14], hybrid improvement procedure with the use of lower bounding strategies is proposed as well as generation of initial solutions by dual min-max problem, load distribution based on dominance, differencing and unbalancing and tabu search for one-dimensional bin packing problem. In [8, 14] hybrid algorithms for solving bin packing problem is used, the results on benchmark datasets are tested and good quality solutions are obtained.

A new variant of whale optimization algorithm known as improved Levy-based whale optimization algorithm was proposed for solving one-dimensional bin packing problem [15].

Scholl, Klein and Jürgens in [16] proposed BISON which is an exact hybrid solution that combines tabu search and branch and bound procedure for solving one-dimensional bin packing problem. Computational results show that BISON is very effective and outperforms existing approaches. In [17], authors proposed a graph-based algorithm for solving one-dimensional bin packing problem and the results tested on benchmark datasets showed that the performance of this algorithm was better than FFD. Paper [18] proposed several polynomial-time approximation algorithms for the one-dimensional bin packing problem. In [19], heuristics based on variable neighborhood search, minimal bin slack for one-dimensional bin packing problem is proposed.

2.2 VIRTUAL MACHINE PLACEMENT PROBLEM

In literature, virtual machine placement problem is studied for different objectives such as

maximizing providers' profit, maximizing resource utilization, minimizing power consumption, minimizing number of physical machines, etc. Many authors investigated performance of heuristic and metaheuristic algorithms for solving VMP problem in scientific literature.

Paper [20] presented review of different algorithms for solving virtual machine placement problem. Algorithms are categorized into different classes and their advantages & limitations are investigated. Paper [21] presented review of virtual machine placement problem in different aspects such as objectives, problem formulation and algorithms usage.

Paper [22] proposed a solution to reduce trade off by efficiently migrating VM to proper machine. Paper [23] presented SDN-based orchestration for live virtual machine management, which is capable of exploiting temporal network information for migrating virtual machines and minimizing the network-wide communication cost. In [24], authors proposed a two-level control system, which consists of an improved genetic algorithm with fuzzy multi-objective evaluation for efficiently searching large solution space and conveniently combining possibly conflicting objectives. In paper [25], VMP was investigated using differential evolution algorithm. Results indicate that for all instances DE algorithm is better than best-fit algorithm. Paper [26] presented crow search based algorithm for multiobjective VMP problem. The objectives were to minimize a number of physical machines, power consumption and maximize resource wastage. Performance of proposed algorithm is better than genetic algorithm and first fit decreasing algorithm. Proposed multiobjective genetic algorithm (GA) in paper [27] maximized the resource utilization of physical machines and minimized the energy consumption.

3. PROBLEM DESCRIPTION

This section presents problem definition for single objective bin packing problem and multi-objective virtual machine allocation problem.

3.1 SINGLE OBJECTIVE BIN PACKING PROBLEM

Single objective bin packing problem is well known problem. The problem description is presented in many papers [13-19].

In one-dimensional bin packing problem, items with varying weights are to be packed using homogeneous/heterogeneous bins. Consider a set of identical bins $B_1, B_2 \dots B_n$ each of them have capacity C and N objects have weights $W_1, W_2 \dots W_N$. The objective is to pack all the items into the bins in such a way that a minimum number of bins

are used. A solution is said to be optimal if it uses a minimal number of bins.

Objective function:

$$B = \sum_{i=1}^n z_i \quad (\text{minimize}), \quad (1)$$

subject to,

$$\sum_{j=1}^n w_j y_{ij} \leq C y_i \quad i \in N, \quad (2)$$

$$\sum_{i=1}^n y_{ij} = 1, \quad j \in N. \quad (3)$$

$$z_i = 0 \text{ or } 1, \quad i \in N,$$

$$y_{ij} = 0 \text{ or } 1, \quad i \in N, j \in N,$$

$$z_i \in \{0, 1\}, \quad i \in N,$$

$$y_{ij} \in \{0, 1\}, \quad i \in N, j \in N,$$

where,

$$z_i = 1 \text{ if bin } i \text{ is used,}$$

$$y_{ij} = 1 \text{ if item } j \text{ is put in bin } i.$$

3.2 MULTI-OBJECTIVE VIRTUAL MACHINE PLACEMENT

Various researchers proposed different problem formulations for virtual machine placement problem. In this paper, we considered problem formulation presented in paper [28].

Virtual machine and physical machine is abstracted as a node with three attributes, namely computing (CPU), memory (M) and bandwidth (B). The number and capacity of physical machines are known. Total numbers of virtual machines, their resource requirements and profit associated with them are known. The objective is to allocate all VMs to PMs that satisfies the VMs' resource requirements while fulfilling the following objectives.

1. Maximizing profit.
2. Maximizing load balancing.
3. Minimizing wastage of resources.

Two main constraints that must be satisfied are as follows:

1. Capacity Constraints: For all three attributes of a given physical machine, the sum of the resource requirements of all VMs placed on it should be less than or equal to the total available capacity.
2. Placement Guarantee Constraints: All virtual machines should be placed.

Resource wastage is calculated on the basis of all three attributes (CPU, M and B). Wasted residual resource on a server is calculated as the sum of

differences between the smallest normalized residual resource and the others [29].

4. HARMONY SEARCH ALGORITHM

Harmony search (HS) is relatively new population-based metaheuristic optimization algorithm that imitates the music improvisation process used by the musicians to improvise their instruments' pitch by searching for a perfect state of harmony. Researchers found solutions to different constraint satisfaction/optimization problem such as solving Sudoku [30], music composition [31], water network design [32], optimization of satellite heat pipe design [33], scheduling of multiple dam systems [34], vehicle routing [35] using HS. Paper [36] presents HS algorithm for solving combined heat and power economic dispatch problem. Paper [37] presents HS algorithm to solve the network reconfiguration problem to get optimal switching combination in the network that results in minimum loss. Paper [38] presents a hybrid harmony search algorithm (HHSA) to solve engineering optimization problems.

4.1 BASIC HARMONY SEARCH

HS is inspired from musicians' behavior in which they try to improvise pitches of their instruments to achieve pleasing harmony, which is measured by aesthetic standards. The fundamental steps involved in basic harmony search algorithm are [39]:

- Step 1: Initialize the parameters - harmony memory size (HMS), harmony memory consideration rate (HMCR), pitch adjusting rate (PAR) and maximum iterations.
- Step 2: Randomly initializing harmony memory (HM). The initial HM consists of randomly generated solutions.
- Step 3: Improvise a new solution from HM. Each component of the solution is obtained depending on the rate of HMCR. The HMCR is nothing but probability of choosing a component from initial HM for improvisation. On the other hand, PAR determines the probability of mutation for selected candidate.
- Step 4: Update the harmony memory. First, the improvised solution, which we get from step 3 is evaluated. If it is better than the solution in the HM, it will replace the latter. Otherwise, it is simply neglected.
- Step 5: Repeat steps 3 and 4 until the termination condition is satisfied. Generally, termination condition is maximum iterations.

4.2 ADAPTIVE HARMONY SEARCH

Meta-heuristic algorithms work on two important concepts: intensification and diversification, which can also be termed as exploration and exploitation. The proper balance between them has a great effect on overall efficiency of the algorithm. The HSA is efficaciously administered using two crucial parameters: harmony memory consideration rate (HMCR) and pitch adjusting rate (PAR). The value of the parameter HMCR ranges from 0 to 1 and is used for exploration and exploitation. On the other hand, value of PAR varies between 0 and 1 is analogous to local search mechanism. PAR has considerable effect on the quality of final solution.

In [40], Wang and Huang proposed harmony search algorithm in which value of PAR is automatically adjusted and low-discrepancy sequences is used for initialization of harmony memory. Results showed that the proposed algorithm is superior to original HS. In order to solve network reconfiguration problem to minimize active and reactive power loss, in [41] self-adaptive harmony search algorithm (SAHSA) is proposed, which uses stochastic search and local search instead of gradient search. In [42] authors proposed adaptive harmony search algorithm, which uses novel approach for adjusting HMCR and PAR for making optimization process efficient.

4.3 PROPOSED ALGORITHM

The effectiveness of harmony search algorithm greatly depends on the value of PAR. A new variant of harmony search algorithm is presented to solve resource allocation problems. Adaptive Harmony Search (AHS) algorithm uses first-fit heuristics for initialization of harmony memory and automatic tuning of PAR, which is based on harmony memory diversity.

Real coded Representation: A 1D representation as shown in Fig. 1. is used to solve the virtual machine placement and bin packing problem. Number of items to be packed indicates the size of the array. The values in the array indicate the resources to which the items are assigned. Fig. 1 shows the sample solution with 10 objects and 6 resources. For example, I5 is assigned to R3 and R7. This representation avoids violation of placement guarantee constraint.

R ₁	R ₂	R ₃	R ₄	R ₅	R ₆	R ₇	R ₈	R ₉	R ₁₀
I ₃	I ₁	I ₅	I ₆	I ₁	I ₂	I ₅	I ₉	I ₃	I ₂

Figure 1 – Memory Representation

Adaptive PAR: As mentioned earlier value of PAR plays a crucial role in the performance of harmony search algorithm. Dynamic PAR helps in achieving results when complexity of dataset is higher. As shown in the Fig. 2. HM diversity is calculated after every iteration. PAR is modified when diversity goes less than set threshold. This ensures that the value of PAR is modified only when it is necessary.

The value of PAR is tuned as per following equation:

$$PAR = PAR + (PAR_{max} - \frac{(PAR_{max} - PAR_{min})}{NI} \times gn). \quad (4)$$

In equation (4), value of PAR is changed after every iteration depending on HM diversity. PAR_{max} and PAR_{min} are maximum and minimum values of PAR, which are set to 1 and 0 respectively, gn is the generation (current iteration) and NI represents maximum iterations.

Partially feasible initialization: Fig. 2. shows the pseudocode of the proposed harmony search algorithm. This algorithm benefits from heuristic inspired from first-fit strategy, which is used for initializing the harmony memory. In order to maintain diversity in the harmony population randomness is added in first-fit.

Objective function and Penalty Function: Bin packing problem is considered as minimization problem in which we have to minimize the number of bins used. Whenever there is violation of any constraints, penalty cost is added. The objective function for bin packing problem can be stated as follows:

$$P = \sum_{i=1}^n p_i = 0 \quad i \in N, \quad (5)$$

$$p_i = C y_i \sum_{j=1}^n w_j y_{ij} \quad j \in N, \quad (6)$$

$$y_{ij} \in \{0, 1\}, \quad i \in N, j \in N,$$

where p_i is the total penalty of i^{th} bin and P is the sum of total penalty of all the bins available.

The virtual machine placement problem is treated as maximization problem. The objective score of a solution is computed by assigning penalty costs (penalty points) for the violations of constraints. These penalty costs are specified according to the importance of the constraint. The objective function for individual 't' is calculated as follows.

$$O(t) = \sum \alpha_i H_i, \quad (7)$$

where H_i represents the number of hard constraints violated by a solution for i^{th} hard constraint and α_i represents penalty costs for violation of hard constraint.

5. RESULTS AND DISCUSSION

This section gives detailed explanation about the datasets used and the results obtained. The proposed algorithm is implemented using 'C' programming language and tested on a computer with the following specifications: Windows 7 Professional, Intel core i5-3210M CPU 2.5 GHz and 4 GB RAM. For every dataset harmony search algorithm was executed for 10 times.

Pseudocode: Adaptive Harmony Search

Input: A Number of VMs and Resource requirement of each VM. Number of PMs and capacity of each PM.
Output: Number of PMs required allocating all VMs.

```

define objective function;
define harmony memory size HMS;
define harmony memory consideration rate HMCR;
PARmin = 0
PARmax = 1
PAR = [0.4, 0.9]
/* Initialization strategy for partially feasible
solution for large datasets*/
while i ≤ HMS do
  while j ≤ number_of_objects do
    while k ≤ number_of_bins do
      if bin_cap(k) ≥ current_obj_size then
        HM(i,j) = current_bin;
        bin_cap(k) = bin_cap(k) - current_obj_size
      else
        HM(i,j) = rand[0, max_bins];
        calculate_penalty();
      end while
    end while
  end while
  /* Improvise the harmony memory */
  while i ≤ max_iterations do
    while j ≤ HMS do
      while k ≤ no_of_obj do
        if rand[0,1] < HMCR then
          memory_consideration(j);
          if(HM_diversity < threshold)
            PAR = PAR + (PARmax - ((PARmax-PARmin) x i) /
max_iterations)
            if rand[0,1] < PAR then
              pitch_adjustment();
            else
              random_solution();
          accept_new_solution_if_better_than_previous;
        end while
      end procedure

```

Figure 2 – Proposed Adaptive Harmony Search

5.1 BIN PACKING DATASET AND RESULTS

Dataset [43] consists of four instances in which the number of object varies from 9 to 33.

The proposed algorithm was compared with best-fit. Results from Table 1 show that AHS has better performance than best-fit heuristic.

Table 1. Results comparison for dataset [43]

Dataset from [43]	Number of objects	Bins capacity	Best Fit (BF)	AHS
P04	33	524	8	7
P03	10	100	4	3
P02	14	100	7	7
P01	9	100	4	4

Dataset [44] consists of six instances with varying number of objects and bin size. Table 2 shows comparison of AHS with GA [44]. For all the instances AHS gives optimal solutions like that of GA. Success rate of AHS is 100%.

Dataset [45] is a benchmark dataset, which consists of 50 objects and bin size of 100. Complexity of these datasets is more than [43] and [44]. We have experimented with 40 instances mentioned in Table 3. Optimal solutions are obtained for all the 40 instances. But the success rate is reduced to 90%.

Table 2. Results comparison for dataset [44]

Dataset [44]	No. of Objects	Bin Capacity	GA	AHS	Optimal [44]
1	50	100	23	23	23
2	40	70	19	19	19
3	20	45	10	10	10
4	9	14	6	6	6
5	4	6	2	2	2
6	10	20	6	6	6

Table 3. Results comparison for dataset [45], Bin capacity =100 and no. of objects =50

Dataset from [45]	AHS	Optimal [45]	Dataset from [45]	AHS	Optimal [45]
N1C1W1 A	25	25	N1C1W1 K	35	35
N1C1W1 B	31	31	N1C1W1 L	31	31
N1C1W1 C	20	20	N1C1W1 M	30	30
N1C1W1 D	28	28	N1C1W1 N	33	33
N1C1W1 E	26	26	N1C1W1 O	29	29
N1C1W1 F	27	27	N1C1W1 P	33	33
N1C1W1 G	25	25	N1C1W1 Q	36	36
N1C1W1 H	31	31	N1C1W1 R	34	34
N1C1W1 I	25	25	N1C1W1 S	37	37
N1C1W1 J	26	26	N1C1W1 T	38	38
N1C1W2 A	29	29	N1C1W2 K	35	35
N1C1W2 B	30	30	N1C1W2 L	31	31
N1C1W2 C	33	33	N1C1W2 M	30	30
N1C1W2 D	28	28	N1C1W2 N	33	33
N1C1W2 E	36	36	N1C1W2 O	29	29
N1C1W2 F	30	30	N1C1W2 P	33	33
N1C1W2 G	30	30	N1C1W2 Q	36	36
N1C1W2 H	33	33	N1C1W2 R	34	34
N1C1W2 I	35	35	N1C1W2 S	37	37

N1C1W2 J	34	34	N1C1W2 T	38	38
----------	----	----	----------	----	----

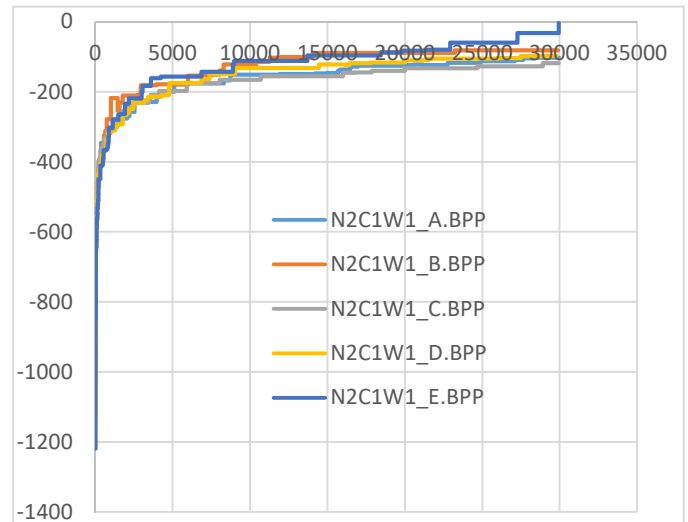


Figure 3 – Performance of AHS for dataset [45], no. of objects = 100

Graph 3 shows the performance of AHS for dataset [45] with no. of objects 100. Results show that the performance of the AHS has reduced significantly.

5.2 VIRTUAL MACHINE PLACEMENT DATASET AND RESULTS

Data Generator: Five random problem instances are generated using data generator implemented in C programming language. For experimentation, we assumed homogeneous physical machines and heterogeneous virtual machines. Datasets contain combination of CPU intensive, memory intensive and bandwidth intensive virtual machines. Table 4 shows small sample dataset.

Table 4. Sample data set

	CPU intensive		Memory intensive		Bandwidth intensive	
	VM ₁	VM ₂	VM ₃	VM ₄	VM ₅	VM ₆
CPU	165	180	42	30	54	41
Memory	13	37	160	111	51	13
Bandwidth	28	41	1	18	148	148
Profit	1660	1823	484	348	590	444

The VM requirements were generated as a mix of different ranges of uniformly distributed random numbers as follows:

<0.7-0.9, 0.1-0.3, 0.1-0.3>: CPU intensive VMs

<0.1-0.3, 0.7-0.9, 0.1-0.3>: Memory intensive VMs

<0.1-0.3, 0.1-0.3, 0.7-0.9>: Bandwidth intensive VMs

Five such configuration files of 60 VMs each were generated for input.

Fig. 4. shows the comparison between original harmony search, harmony search with first-fit and harmony search with first-fit and adaptive PAR (AHS). Partially feasible initialization and adaptive

PAR shows significant improvement on the performance of harmony search algorithm.

Table 5 shows the results of adaptive harmony search for multi-objective VMP.

Table 5. Results of AHS on VMP

Runs	Dataset 1			Dataset 2			Dataset 3			Dataset 4		
	Profit	Load Balance	Wastage	Profit	Load Balance	Wastage	Profit	Load Balance	Wastage	Profit	Load Balance	Wastage
1	38395	7361	3740	40050	5835	2895	39289	6872	2474	39510	7110	3110
2	40369	4855	2831	40869	3860	2054	38620	8409	3857	40302	6382	2618
3	36872	9033	4064	40577	4914	3302	40866	5632	2732	39991	7152	3138
4	39864	5550	2530	39974	6566	3632	39287	7604	2734	41105	5369	2375
5	38842	6821	3587	39655	5642	3013	39889	6550	2663	41171	6117	2891
6	40000	5248	2498	33968	11348	2768	38215	8286	3512	40641	6682	2665
7	39029	6607	3254	40705	4741	2621	36887	10584	4202	38630	8230	2093
8	38194	5936	2345	40929	4101	2085	38879	7715	3017	38249	8760	3203
9	39362	5430	2828	39473	6463	3134	34019	12035	4529	34790	10830	3673
10	40053	5703	2352	41257	3640	1841	40341	6505	2972	40289	5956	2341

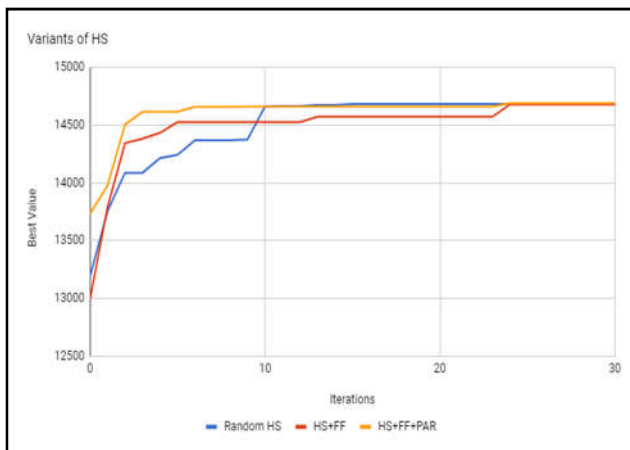


Figure 4 – Comparison of variants of HS



Figure 6 – Results of GA for VMP from [28]



Figure 5 – Results of AHS for VMP

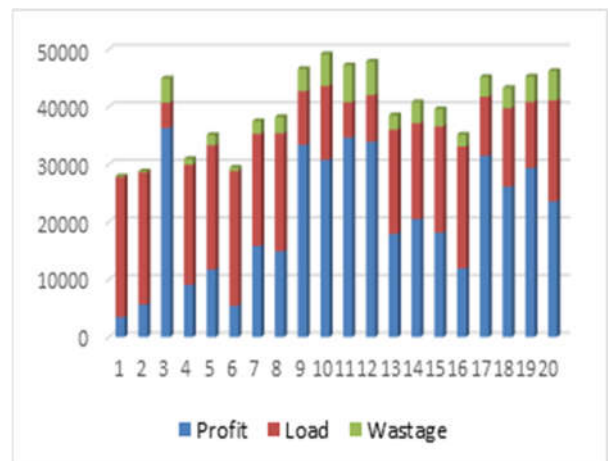


Figure 7 – Results of NSGA-II for VMP from [28]

Fig. 5 shows the results of real coded adaptive harmony search algorithm for multi-objective virtual machine placement problem. It is clear from the graph that profit, which needs to be maximized is maximum in all the runs with load balance being compromised. On the other hand, wastage is always minimized. Fig. 6 and 7 show results obtained from genetic algorithm and NSGA-II for VMP from [28]. Results lead to following conclusions:

- NSGA-II shows good exploration capability for all three objectives than AHS and GA. The pareto front obtained by NSGA-II algorithms gives importance to all three objectives under consideration. AHS shows better exploration capability as compared to GA.
- Results obtained by NSGA-II are better than GA and AHS. Better resource wastage is obtained by AHS than GA.

6. CONCLUSIONS

Paper presents adaptive harmony search algorithm for constrained resource allocation problem. The proposed algorithm presents two modifications in the original harmony search: first, adaptive PAR and second, partially feasible initialization strategy with real coded solution representation. Experiments were performed to study one-dimensional bin packing problem and multi-objective virtual machine placement problem. 55 instances of BPP dataset are used for testing performance of the algorithm. AHS performs better than best-fit heuristic and original harmony search. AHS gives optimal results with 100% success rate for BPP with 50 objects. The performance reduced for BPP with 100 objects. Results of virtual machine problem show that AHS has good performance as compared to GA but has less exploration capability than NSGA-II.

Future work: Performance of AHS reduced significantly for larger instances of BPP. There is scope to improve the strategy for adaptive PAR.

7. REFERENCES

- [1] S. K. Chaharsooghi, A. H. Meimand Kermani, "An effective ant colony optimization algorithm (ACO) for multi-objective resource allocation problem (MORAP)," *Applied Mathematics and Computation*, vol. 200, no. 1, pp. 167-177, 2008.
- [2] A. Borisenko, M. Haidl, S. Gortlatch, "A GPU parallelization of branch-and-bound for multiproduct batch plants optimization," *The Journal of Supercomputing*, vol. 73, no. 2, pp. 639-651, 2017.
- [3] A. Borisenko, S. Gortlatch, "Comparing GPU-parallelized metaheuristics to branch-and-bound for batch plants optimization," *The Journal of Supercomputing*, pp. 1-13, 2018.
- [4] D. Weyland, "A critical analysis of the harmony search algorithm – How not to solve Sudoku," *Operations Research Perspectives*, vol. 2, pp. 97-105, 2015.
- [5] S. Sivasubramani, K. S. Swarup, "Multi-objective harmony search algorithm for optimal power flow problem," *International Journal Electric Power Energy System*, vol. 33, no. 3, pp. 745-52, 2011.
- [6] A. Kaveh, S. Talatahari, "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures," *Computers & Structures*, vol. 87, no. 5-6, pp. 267-283, 2009.
- [7] S. A. Hussain, V. U. K. Sastry, "Application of genetic algorithm for bin packing," *International Journal of Computer Mathematics*, vol. 63, no. 3-4, pp. 203-214, 1997.
- [8] T. Dokeroglu, A. Cosar, "Optimization of one-dimensional bin packing problem with island parallel grouping genetic algorithms," *Computers & Industrial Engineering*, vol. 75, pp. 176-186, 2014.
- [9] K.-H. Loh, B. Golden, E. Wasil, "Solving the one-dimensional bin packing problem with a weight annealing heuristic," *Computers & Operations Research*, vol. 35, no. 7, pp. 2283-2291, 2008.
- [10] S. Abidi, S. Krichen, E. Alba, J. M. Molina, "Improvement heuristic for solving the one-dimensional bin-packing problem," *Proceedings of the IEEE 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*, 2013, pp. 1-5.
- [11] A. Singh, A. K. Gupta, "Two heuristics for the one-dimensional bin-packing problem," *OR Spectrum*, vol. 29, no. 4, pp. 765-781, 2007.
- [12] K. Sim, E. Hart, B. Paechter, "A hyper-heuristic classifier for one dimensional bin packing problems: Improving classification accuracy by attribute evolution," *Proceedings of the International Conference on Parallel Problem Solving from Nature*, Springer, Berlin, Heidelberg, 2012, pp. 348-357.
- [13] A. K. Bhatia, M. Hazra, S. K. Basu, "Better-fit heuristic for one-dimensional bin-packing problem," *Proceedings of the IEEE International Advance Computing Conference IACC 2009*, 2009, pp. 193-196.
- [14] Alvim, Adriana CF, Celso C. Ribeiro, Fred Glover, and Dario J. Aloise, "A hybrid improvement heuristic for the one-dimensional

- bin packing problem,” *Journal of Heuristics*, 2004, vol. 10, no. 2, pp. 205-229.
- [15] M. Abdel-Basset, G. Manogaran, L. Abdel-Fatah, S. Mirjalili, “An improved nature inspired meta-heuristic algorithm for 1-D bin packing problems,” *Personal and Ubiquitous Computing*, 2018, pp. 1-16.
- [16] A. Scholl, R. Klein, C. Jürgens, “Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem,” *Computers & Operations Research*, vol. 24, no. 7, pp. 627-645, 1997.
- [17] D. Sensarma, S. S. Sarma, “A novel graph based algorithm for one dimensional bin packing problem,” *International Journal of Global Research in Computer Science*, vol. 5, no. 8, pp. 1-4, 2014.
- [18] N. Karmarkar, R. M. Karp, “An efficient approximation scheme for the one-dimensional bin-packing problem,” *Proceedings of the IEEE 23rd Annual Symposium on Foundations of Computer Science SFCS’08*, 1982, pp. 160-164.
- [19] K. Fleszar, K. S. Hindi, “New heuristics for one-dimensional bin-packing,” *Computers & operations research*, vol. 29, no. 7, pp. 821-839, 2002.
- [20] M. Masdari, S. S. Nabavi, V. Ahmadi, “An overview of virtual machine placement schemes in cloud computing,” *Journal of Network and Computer Applications*, vol. 66, pp. 106-127, 2016.
- [21] F. Lopez-Pires, B. Baran, “Virtual machine placement literature review,” *arXiv preprint arXiv:1506.01509*, 2015.
- [22] S. Vahora, R. Patel, H. Patel, S. Patel, “Efficient virtual machine management based on dynamic workload in cloud computing environment,” *Proceedings of the IEEE International Advance Computing Conference (IACC)*, 2015, pp. 603-608.
- [23] R. Cziva, S. Jouët, D. Stapleton, F. P. Tso, D. P. Pazaros, “SDN-based virtual machine management for cloud data centers,” *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 212-225, 2016.
- [24] J. Xu, J. A. Fortes, “Multi-objective virtual machine placement in virtualized data center environments,” *Proceedings of the IEEE/ACM International Conference on Green Computing and Communications & International Conference on Cyber, Physical and Social Computing*, 2010 Dec 18, pp. 179-188.
- [25] A. C. Adamuthe, J. T. Patil, “Differential evolution algorithm for optimizing virtual machine placement problem in cloud computing,” *International Journal of Intelligent Systems and Applications*, vol. 10, no. 7, pp. 58-65, 2018.
- [26] A. Satpathy, S. K. Addya, A. K. Turuk, B. Majhi, G. Sahoo, “Crow search based virtual machine placement strategy in cloud data centers with live migration,” *Computers & Electrical Engineering*, vol. 69, pp. 334-350, 2018.
- [27] F.-H. Tseng, X. Wang, L.-D. Chou, H.-C. Chao, and V. C. M. Leung, “Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm,” *IEEE Systems Journal*, vol. 12, no. 2, pp. 1688-1699, 2018.
- [28] A. C. Adamuthe, R. M. Pandharpatte, G. T. Thampi, “Multiobjective virtual machine placement in cloud environment,” *Proceedings of the IEEE International Conference on Cloud & Ubiquitous Computing & Emerging Technologies (CUBE)*, 2013, pp. 8-13.
- [29] J. Xu and J. Fortes, “Multi-objective virtual machine placement in virtualization data center environments,” *Proceedings of the IEEE/ACM International Conference on Green Computing and Communications (GreenCom’10) & International Conference on Cyber, Physical and Social Computing (CPSCom’10)*, Dec. 18-20, 2010, pp. 179-188.
- [30] Z. Geem, “Harmony search algorithm for solving sudoku”, In: B. Apolloni, R.J. Howlett, L. Jain (eds) *Knowledge-based intelligent information and engineering systems, Lecture Notes in Computer Science*, vol. 4692. Springer, Berlin/Heidelberg, 2007, pp. 371–378.
- [31] Z. W. Geem, J.-Y. Choi, “Music composition using harmony search algorithm,” *Proceedings of the Workshops on Applications of Evolutionary Computation*, Springer, Berlin, Heidelberg, pp. 593-600, 2007.
- [32] Z. W. Geem, “Particle-swarm harmony search for water network design,” *Engineering Optimization*, vol. 41, no. 4, pp. 297-311, 2009.
- [33] Z. W. Geem, H. Hwangbo, “Application of harmony search to multi-objective optimization for satellite heat pipe design,” *Proceedings of US-Korea Conference on Science, Technology, & Entrepreneurship (UKC 2006)*, Teaneck, NJ, USA, 2006, pp. 1-3.
- [34] Z. W. Geem, “Optimal scheduling of multiple dam system using harmony search algorithm,” *Proceedings of the International Work-Conference on Artificial Neural Networks*, Springer, Berlin, Heidelberg, 2007, pp. 316-323.
- [35] Z. W. Geem, K. S. Lee, Y. Park, “Application of harmony search to vehicle routing,”

- American Journal of Applied Sciences*, vol. 2, no. 12, pp. 1552-1557, 2005.
- [36] A. Vasebi, M. Fesanghary, S. M. T. Bathaee, "Combined heat and power economic dispatch by harmony search algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 29, no. 10, pp. 713-719, 2007.
- [37] R. S. Rao, S. V. Lakshmi Narasimham, M. Ramalinga Raju, A. Srinivasa Rao, "Optimal network reconfiguration of large-scale distribution system using harmony search algorithm," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1080-1088, 2011.
- [38] M. Fesanghary, M. Mahdavi, M. Minary-Jolandan, and Y. Alizadeh, "Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 33-40, pp. 3080-3091, 2008.
- [39] X. Wang, X.-Z. Gao, and K. Zenger, *An Introduction to Harmony Search Optimization Method*, Springer International Publishing, 2015.
- [40] C.-M. Wang, Y.-F. Huang, "Self-adaptive harmony search algorithm for optimization," *Expert Systems with Applications*, vol. 37, no. 4, pp. 2826-2837, 2010.
- [41] D. S. Rani, N. Subrahmanyam, M. Sydulu, "Self adaptive harmony search algorithm for optimal network reconfiguration," *Proceedings of the IEEE Power and Energy Conference at Illinois (PECI'2014)*, 2014, pp. 1-6.
- [42] M. P. Saka, O. Hasancebi, "Adaptive harmony search algorithm for design code optimization of steel structures," *Proceedings of the Harmony Search Algorithms for Structural Design Optimization*, Springer, Berlin, Heidelberg, 2009, pp. 79-120.
- [43] https://people.sc.fsu.edu/~jburkardt/datasets/bin_packing/bin_packing.html
- [44] N. Mohamadi, "Application of genetic algorithm for the bin packing problem with a new representation scheme," *Mathematical Sciences*, vol. 4, no. 3, pp. 253-266, 2010.
- [45] <https://www2.wiwi.unijena.de/Entscheidung/binpp/bin1dat.htm>



Dr. Amol C. Adamuthe received Ph.D. from Mumbai University and Master of Technology in Computer Engineering from Dr. B. A. Technological University, Lonere, MS, India. He is currently an Assistant Professor at Rajarambapu Institute of Technology, Sakharale, Sangli, MS, India. His areas of interest are technology forecasting, optimization and cloud computing.



Tushar R. Nitave completed the Bachelor of Engineering in Computer Engineering, from Rajarambapu Institute of Technology, Sakharale, Sangli, MS, India. His areas of interest are heuristic algorithms and optimization.