Journal of Computer Science & Computational Mathematics, Volume 8, Issue 4, December 2018
DOI: 10.20967/jcscm.2018.04.002

Capacity Allocation Optimization using Offline Learning Differential Evolution Hyper-Heuristic

Mohamad Fiqri*, Chin Jeng Feng

School of mechanical engineering, Universiti Sains Malaysia, Engineering campus, 14300 Nibong Tebal, Penang
*Corresponding author email: fiqri1992@gmail.com

Abstract: Hyper-heuristics are high-level computational techniques to select or generate heuristics for solving complex problems, such as capacity planning involving finite resource allocation. This research proposes a hyper-heuristics method based on two-level differential evolution (DE) termed as HMDE. It optimizes four parameters in the hyper stage, namely mutation factor, crossover rate, number of generations, and number of iterations to derive a comparatively superior capacity planning meta-heuristics, also in the form of DE. The effect of demand seasonality onto its performance is examined and benchmarked against Genetic Algorithm (GA). As it is statistically proven from the results, HMDE has shortened the runtime and produced higher profit on average in comparison to the GA over a test case containing twelve months of seasonal data.

Keywords: hyper-heuristic, differential evolution, genetic algorithm, capacity allocation.

1. Introduction

In today's business, a factory's production needs to adapt to different product demands which may vary throughout a year. Capacity allocation is therefore vital to maximize profit and to provide the best matching of resources to the product demands. A number of related studies have already been made on models and algorithms catered for different industries, manufacturing conditions, market environments and production characteristics [1-4].

This paper proposes a hyper-heuristic model termed as HMDE to harness the effectiveness in searching for capacity plans that maximize profit for a given stream of months. Hyper-heuristics are high-level computational techniques to select or generate heuristics to solve complex problems. The architecture commonly presents two levels of principally identical structure which is based on meta-heuristics.

Succinctly, meta-heuristic is an approximate technique used to find solutions as the substitution to or in the absence of exact methods. Most meta-heuristics are inspired by nature (based on a number of principles in physics, biology, or ethology), e.g. Boussaid *et al.* [5] and Mirjalili [6]. Meta-heuristics require little or no domain-specific knowledge [7]. Also, meta-heuristic trades off accuracy, precision, completeness and optimality with the computation time, memory consumption, randomness and limited domain knowledge-bias solution manipulation strategies. The first generation of meta-heuristics includes the ant colony, simulated annealing, tabu search, and genetic algorithms (GAs). Interests in metaheuristics have grown in the recent years, and comprehensive research has led to the development of new models and variants such as the jmetal method Durillo *et al.* [8] and the dragonfly algorithm method shown in Mirjalili [6].

The basic structure of meta-heuristics can generally be broken down into two stages, namely, the initialization of a solution population and a finite number of iterations to selectively manipulate the population to achieve incremental improvement. A case in point is the work of Colorni *et al.* [7] and Glover [9]. Manipulation, in this case, capitalizes on randomization, search history, domain-independent knowledge, or collectivism to produce two typical forms of search: intensification (exploitation) and diversification (exploration). Intensification, which limits the search in areas with potentially optimal solutions, is achieved by creating a small perturbation in the solution within a defined neighborhood, some guided with domain knowledge heuristic. By contrast, diversification diverts the search to other less explored areas, thus eliminating premature convergence to a local optimum.

A typical meta-heuristic does not retain any knowledge after the problem-solving, even the relevant problems were originated from the same source. Also, deciding on the values of parameters is itself a complex optimization problem. Often, they are prefixed and decided arbitrarily, e.g. trials-and-errors. In these lights, hyper-heuristics aim to complement a meta-heuristic by placing another one in the backdrop to gather knowledge and improve the performances of the primary meta-heuristic, e.g. by providing a more matching set of parameter values. In a long run, this mimics human learning.

The model contains two levels of meta-heuristic known as differential evolution (DE), respectively identified as Hyper-DE and Meta-DE. Hyper-DE optimizes the parameter values used in the Meta-DE offline and by using the recent problem cases. New parameter values are then fed into the meta-DE for the next run.

Typically, capacity allocation involves the allocation of several product types to be processed on a collection of machines in the manufacturing setting [10]. Each product type requires different combinations of machines, and the times spent on these machines are product-dependent. Profits are also product-dependent. The product demands are largely seasonal and hence affect the allocation of the machine resources, with the intent to maximize the overall profit.

The remainder of the paper is organized as follows. In section 2, the previous research on capacity allocation is discussed. Then the proposed method is explained in section 3. While in section 4, the experimental procedure is explained and the range of the parameter used is shown. Section 5 shows the experimental result and the analysis of the proposed model. Finally, section 6 summarizes the paper.

2. Related research

Capacity allocation methods generally come in the form of mathematical models and meta-heuristics. The former translates identified issues or problems within a system and transforms them into mathematical formulations. The most prominent mathematical models include mixed-integer programming, linear programming, and Lagrangian relaxation. You, *et al.* [11] incorporated a two-stage linear stochastic programming approach into a multi-period planning model. Geng, *et al.* [12] proposed a scenario-based stochastic programming model for uncertain capacity allocation based on overall equipment effectiveness. Wang and Wang [13] devised a mathematical model that supports simultaneous resource investment and task allocation planning. Chen and Lu [14] developed a stochastic mixed-integer programming model to determine a suitable capacity allocation and expansion policy. The optimality of a solution is often domain-specific. A mathematical model often fails to generate a good number of solutions (or becomes stagnant at the local optimum) for multimodal problems with non-linear objective functions and numerous variables [15]. Moreover, mathematical models cannot easily handle large-scale complex systems, such as semiconductor manufacturing systems [16].

Meta-heuristics are prevalent in capacity allocation. Bilgin and Azizoglu [10] adopted a tabu search algorithm to find the optimum allocation of the scarce time, tool magazine capacities and the limited quantities of tool to meet the demand of the production. Wang and Hou [17] used genetic algorithm to effectively distribute resources such as tester, handlers, load board and tooling. Beside capacity allocation based on the tool and machine, material requirement planning, production and transportation have to be bought in for supply chain planning. Vahdani [18] attempted such problem by using large neighborhood search algorithm. Feng [19] considered the allocation of mostly used routes and less used routes in airline capacity planning.

3. Model description

DE was invented by Storn *et al.* [20] as the solution method for the Chebychev trial polynomial problem. It operates directly on real valued chromosomes and uses the differential operator. The operator works with real numbers in natural manner and fulfills the same purpose as the crossover operator in a standard genetic algorithm. Since then, DE is considered as one of the most powerful optimization meta-heuristics. Das *et al.* [21] provides an overview of different types of DE, relevant theoretical studies and applications to different optimization problems. Villela *et al.* [22] used twelve DE models for the low level heuristic to produce four selection mechanisms that choose the heuristic method.

The variant of DE used in this research is the DE/rand/1. It is based from the general notion used in DE from Storn and Price [20], that is DE/x/y where x refer to the method used to select the target vector and y indicate the number of difference vector used. In this variant, the target vector is selected randomly from the current population.

Figure 1 and 2 respectively show the process flows of HMDE at Hyper-DE and Meta-DE, as well as the collaborative relationship between them. Initial populations for both levels are randomly generated. Nevertheless, in Hyper-DE the good solutions are retained and brought forward to the next capacity allocation.

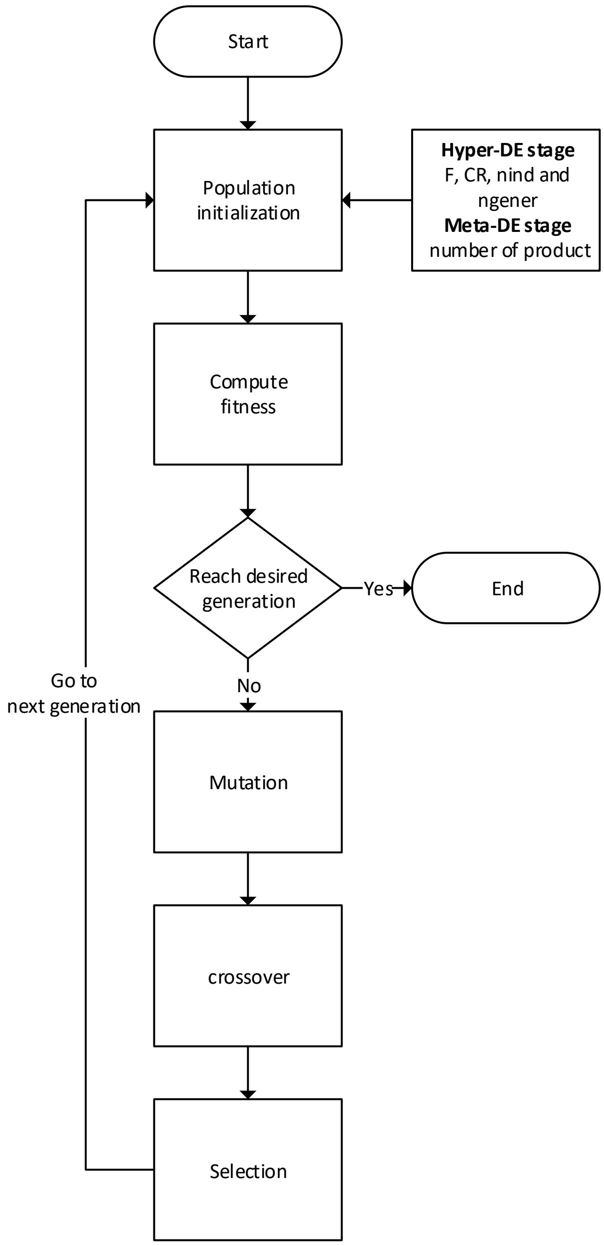

Figure 1. flowchart of HMDE

DE works by first generating the initial population by using a uniform probability distribution. The fitness of each individual in the population is then calculated using the fitness function. Next, the DE generates new solution vector called mutant vector by adding the multiplication result between the mutation factor and the weighted difference between two population vectors to a third vector. The third vector is the best solution (highest fitness value) from the initial population.

Later, crossover is performed to produce a trial vector by mixing the mutant vectors with another predetermined vector.

The crossover factor determines that at least one parameter from the mutant vector is to be mixed to the trial vector. In selection, if the trial vector yields a higher fitness value than the target vector, the trial vector will replace the target vector in the following generation. The population that has the highest fitness is then saved in the memory to be used for the second generation until a maximum number of generations is reached.

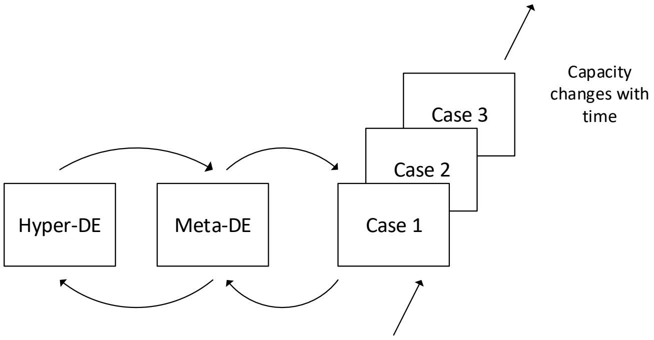

Figure 2. program flow of HMDE

Hyper-DE is run offline after Meta-DE completes its cycle. This provides flexibility as improvements on parameters can be done without interfering the computations of Meta-DE.

4. Experiment details

This section provides the detail of the problem description and experiment variables.

- Problem description
  Both algorithms were tested with capacity allocation case as described in Table 1:

Table 1. Problem description

No of product	**No of machine**	**Machine cycle time distribution type**	**Max. hour per week for each machine**	**No of month**
5	5	Normal distribution	168 hours	12

- Hardware
  Computer using AMD FX(tm)-6300 six-core processor with speed of 3.5GHz and 16GB of RAM. The operating system used is window 10 64-bit.
- Software
  MATLAB R2017a
- Algorithms
  - ➢ HMDE
  For HMDE, both DEs are based on Storn and Price (1997). Parameters involved are in table 2.
  - ➢ Genetic algorithm (GA)
  For GA, the parameters and the corresponding value are as illustrated in table 3.

Table 2. HMDE parameter setup

Range for mutation factor (F)	Range for crossover rate (CR)	Range for number of population (nind)	Range for number of generation (ngener)
0-1	0-1	20-100	10-1000

Table 3. GA parameter setup

Stall generation	crossover fraction (CF)	Number of population (nind)	Number of generation (ngener)	Elite count
5	0.8	100	17	5

5. Results and discussion

The result for the twelve months run is tabulated in table 4, where it shows the average profit that the algorithm obtained. Furthermore the table also reveals the number of generations needed for each algorithm to reach the maximum profit for a particular month. The numbers of generations show the point of convergence for the algorithms.

Table 4. Profit and the number of generation

Month	HMDE		GA	
	Profit (RM)	No of generations	Profit (RM)	No of generations
1	187.92	8	160.31	8
2	1272.98	16	1011.46	7
3	764.56	19	704.38	8
4	694.17	29	573.43	8
5	1340.71	10	1070.74	9
6	1133.52	12	951.87	9
7	754.64	9	680.05	7
8	869.15	17	727.68	8
9	788.37	11	625.36	9
10	1561.00	11	1284.9	8
11	706.80	7	578.98	7
12	407.80	3	386.06	7

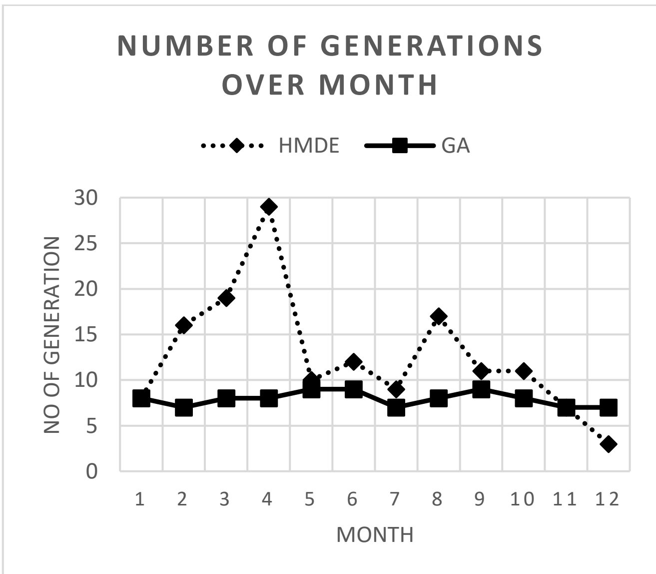

Figure 3. Number of generations taken by HMDE and GA to solve the capacity allocation problem

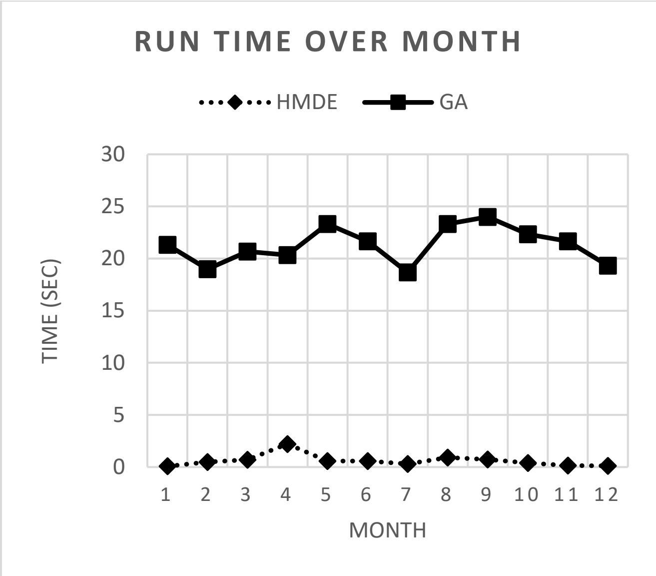

Figure 4. Run time for HMDE and GA over the twelve months

To verify that the results of the two methods are statistically different, an analysis of variance (ANOVA) is performed using Matlab. Figure 6 shows the comparison of the HMDE and GA profit means. The HMDE, represented by the dotted line, shows a higher mean compared to the GA represented in solid line.

HMDE reduces the time taken for the solutions to converge by using the DE/rand/1 mutation strategy on the population consisting of F, CR, nind and ngener. This allows the algorithm to produce improved populations both at the hyper-DE stage and the meta-DE stage.

The performance variation between HMDE and GA in term of profits, is significant. The average difference in profit for the 12 months period between HMDE and GA are RM 144.59 with the maximum are RM 276.10 in October.

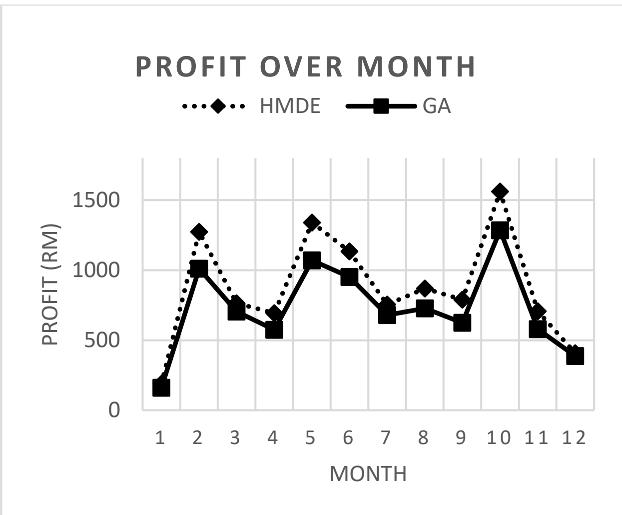

Figure 5. Profit generated by HMDE and GA over the twelve months

Table 5. Runtime taken to solved the capacity allocation

Month	*HMDE*	*GA*
	Time (Sec)	
1	0.074	21.333
2	0.492	19.000
3	0.709	20.667
4	2.221	20.333
5	0.575	23.333
6	0.574	21.667
7	0.303	18.667
8	0.925	23.333
9	0.742	24.000
10	0.402	22.333
11	0.147	21.667
12	0.118	19.333

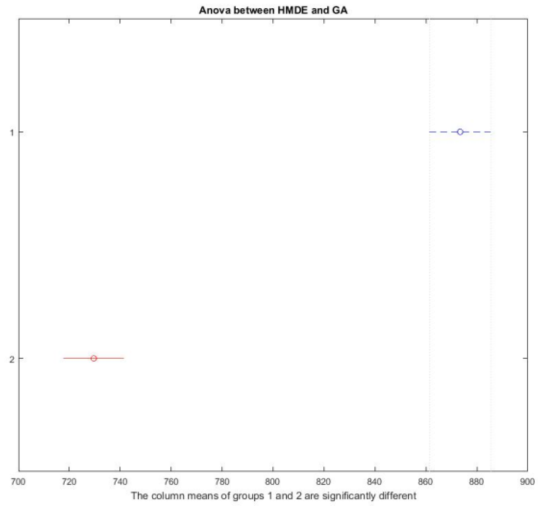

Figure 6. ANOVA for HMDE and GA means

6. Conclusion

In this paper, HMDE is proposed as an offline hyper-heuristic for capacity planning problem. The model employs two levels of DE with the hyper-heuristic stage evolves the four parameters which in turn is being used at the meta-heuristic stage for direct capacity planning. The benchmarked GA model with fixed parameters were used to compare with HMDE. Their performances were compared on a case study of twelve months data. The findings reveal that HDME outperformed benchmarked GA models in term of profits, the number of generations and runtime to reach convergence.

Acknowledgement

We wish to acknowledge Fundamental Research Grant Scheme (FRGS) (no: 6071276) issued by the ministry of higher education, Malaysia as the financial sponsor of this research.

References

[1] R. Q. Zhang, "Research on Capacity Planning under Stochastic Production and Uncertain Demand", *Systems*

Engineering-Theory & Practice, vol. 27, no. 1, pp. 51-59, 2007.

[2] U. K. Yusof, S. Deris, "Optimizing machine utilization in semiconductor assembly industry using constraint-chromosome genetic algorithm", in *International Symposium in Information Technology (ITSim)*, pp. 601-606, 2010.

[3] X. Zeng, W.-K. Wong, S. Y.-S. Leung, "An operator allocation optimization model for balancing control of the hybrid assembly lines using Pareto utility discrete differential evolution algorithm", *Computers & Operations Research*, vol. 39, no. 5, pp. 1145-1159, 2012.

[4] R. Zhang, J. Ding, "Non-linear optimal control of manufacturing system based on modified differential evolution", in *IMACS Multiconference on Computational Engineering in Systems Applications*, pp. 1797-1803, 2006.

[5] I. Boussaïd, J. Lepagnot, P. Siarry, "A survey on optimization metaheuristics", *Information Sciences*, vol. 237, pp. 82-117, 2013.

[6] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems", *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053-1073, 2016.

[7] J. J. Durillo, A. J. Nebro, "jMetal: A Java framework for multi-objective optimization", *Advances in Engineering Software*, vol. 42, no. 10, pp. 760-771, 2011.

[8] F. Glover, "Tabu search and adaptive memory programming—advances, applications and challenges", in *Interfaces in computer science and operations research*, ed: Springer, pp. 1-75, 1997.

[9] A. Colorni, M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini, M. Trubian, "Heuristics from nature for hard combinatorial optimization problems", *International Transactions in Operational Research*, vol. 3, no. 1, pp. 1-21, 1996.

[10] S. Bilgin, M. Azizoğlu, "Operation assignment and capacity allocation problem in automated manufacturing systems", *Computers & Industrial Engineering*, vol. 56, no. 2, pp. 662-676, 2009.

[11] F. You, J. M. Wassick, I. E. Grossmann, "Risk management for a global supply chain planning under uncertainty: models and algorithms", *AIChE Journal*, vol. 55, no. 4, pp. 931-946, 2009.

[12] N. Geng, Z. Jiang, F. Chen, "Stochastic programming based capacity planning for semiconductor wafer fab with uncertain demand and capacity", *European Journal of Operational Research*, vol. 198, no. 3, pp. 899-908, 2009.

[13] K. J. Wang, S.-M. Wang, "Simultaneous resource portfolio planning under demand and technology uncertainty in the semiconductor testing industry", *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 5, pp. 278-287, 2013.

[14] T.-L. Chen, H.-C. Lu, "Stochastic multi-site capacity planning of TFT-LCD manufacturing using expected shadow-price based decomposition", *Applied Mathematical Modelling*, vol. 36, no. 12, pp. 5901-5919, 2012.

[15] R. Rao, V. Patel, "An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems", *International Journal of Industrial Engineering Computations*, vol. 3, no. 4, pp. 535-560, 2012.

[16] J. Lin, Q. Long, "Development of a multi-agent-based distributed simulation platform for semiconductor manufacturing", *Expert Systems with Applications*, vol. 38, no. 5, pp. 5231-5239, 2011.

[17] K.-J. Wang, T. Hou, "Modelling and resolving the joint problem of capacity expansion and allocation with multiple resources and a limited budget in the semiconductor testing industry", *International Journal of Production Research*, vol. 41, no. 14, pp. 3217-3235, 2003.

[18] B. Vahdani, S. T. A. Niaki, S. Aslanzade, "Production-inventory-routing coordination with capacity and time window constraints for perishable products: Heuristic and meta-heuristic algorithms", *Journal of Cleaner Production*, vol. 161, pp. 598-618, 2017.

[19] B. Feng, Y. Li, H. Shen, "Tying mechanism for airlines' air cargo capacity allocation", *European Journal of Operational Research*, vol. 244, no. 1, pp. 322-330, 2015.

[20] R. Storn, K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces", *Journal of global optimization*, vol. 11, no. 4, pp. 341-359, 1997.

[21] S. Das, P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art", *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4-31, 2011.

[22] J. C. Villela Tinoco, C. A. Coello Coello, "hypDE: A Hyper-Heuristic Based on Differential Evolution for Solving Constrained Optimization Problems", in *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, Springer Berlin, Heidelberg, pp. 267-282, 2013.