

Solving multi-objective master production schedule problem using memetic algorithm

Shereen S. Sadiq¹, Adnan Mohsin Abdulazeez², Habibollah Haron³

¹Department of Information Technology, College of Informatics, Duhok Polytechnic University, Iraq

²Presidency of Duhok Polytechnic University, Iraq

³Applied Industrial Analytics Research Group (Department of Computer Science),
Universiti Teknologi Malaysia, Malaysia

Article Info

Article history:

Received Aug 15, 2019

Revised Oct 17, 2019

Accepted Oct 31, 2019

Keywords:

Genetic algorithm

Master production schedule

Memetic algorithm

Multi-objective optimization

Simulated annealing

ABSTRACT

A master production schedule (MPS) need find a good, perhaps optimal, plan for maximize service levels while minimizing inventory and resource usage. However, these are conflicting objectives and a tradeoff to reach acceptable values must be made. Therefore, several techniques have been proposed to perform optimization on production planning problems based on, for instance, linear and non-linear programming, dynamic-lot sizing and meta-heuristics. In particular, several meta- heuristics have been successfully used to solve MPS problems such as genetic algorithms (GA) and simulated annealing (SA). This paper proposes a memetic algorithm to solve multi-objective master production schedule (MOMPS). The proposed memetic algorithm combines the evolutionary operations of MA (such as mutation and Crossover) with local search operators (swap operator and inverse movement operator) to improve the solutions of MA and increase the diversity of the population). This algorithm has proved its efficiency in solving MOMPS problems compared with the genetic algorithm and simulated annealing. The results clearly showed the ability of the algorithm to evaluate properly how much, when and where extra capacities (overtime) are permitted so that the inventory can be lowered without influencing the level of service.

*Copyright © 2020 Institute of Advanced Engineering and Science.
All rights reserved.*

Corresponding Author:

Shereen S. Sadiq,

Department of Information Technology,

College of Informatics,

Duhok Polytechnic University, Duhok, Iraq.

Email:shireen.saleem@dpu.edu.krd

1. INTRODUCTION

The key to the success of a business is to be competitive. Now, to meet customer images over time, ensuring high quality products and services is not a difference, but a basic form. In this case, in general, with the best or near-optimal processes reducing costs, improving slowing and economic efficiency, particularly due to better use of product quality, resources, and better service levels, can be the shortest time and inventory levels available very complicated theme. As production plans increase, the complexity and hassle will need to create a master plan, grow faster, especially in the case of limited resources, as is the case in most companies [1]. In fact, Production scheduling problems are proved to be NP-hard sorts of problems and are not easily solved for large sizes [2]. Fortunately, new ideas and methods of improving production management, such as the use of heuristics AI, are frequently being proposed by researchers. In order to solve NP-hard issues, use of meta-heuristic methods requires both effective and efficient optimization techniques. A review of existing literature reveals that much work has yet to be reported in the application of meta-heuristic techniques to solve MPS problems [3]. Research in this field is continually being conducted,

with widely used nature-inspired meta-heuristic optimization techniques proving superior to traditional techniques. Although evolutionary computation methods provide solutions combining computational efficiency and good performance, evolutionary computational research has been criticized for considering artificial test problem scenarios that are much less complex than real-world manufacturing cases [4].

Multi-objective optimization is the concurrent technique that defines more than one contrasting goal under specific restrictions [5]. Use (MOEA) to solve multifunctional optimization issues. Some MOEA are population-based optimization, where repetition is performed on a group of solutions (population) and in each generation, multiple solution will return. For the following reasons MOEA is more popular to solve multi-objective optimization issues: (i) ease of implementation, (ii) robust and flexible, (iii) less chances of falling into local minimum, (iv) it gives the number of optimal solutions [6].

The researchers used a variety of advanced optimization techniques; Vieira et al [7] applied simulated annealing, Soares et al [1] proposed a different framework of genetic algorithms, Vieira [8] has compared simulated annealing and genetic algorithms for master production scheduling problems and Radhika et al [3, 9] applied differential evolution. The goal consideration was to minimize inventory level, maximum service level and minimize each of overtime and inventory level below safety stock.

Wu et al [11] suggested BP neural network to solve the demand predicate for master production scheduling. The goal is considered to maximize device usage and minimize loss for penalties for production inventory.

Bakar, et al [12] developed solutions on Master Production Schedule (MPS) with the help of an algorithm that is based on the installation of two birds swarm algorithms namely Bat Algorithm (BA) and Gravitational Search Algorithm (GSA). A new multi-objective evolutionary algorithm Bat Algorithm and Gravitational Search Algorithm MOGSABAT was also proposed that considered the configuration performance of the system parameters. Therefore, solving the issue of the MPS using intelligent algorithms were carried out by a specific number of researchers.

A memetic algorithm (MA) has been proposed to solve the problem of multi-objective master production scheduling. Six scenarios have taken from previous studies and the result obtained are compared with two meta-heuristic algorithm simulated annealing and genetic algorithm.

The paper’s framework is structured as follows: Section 2 illustrate a brief revision of production planning problem, and the proposed algorithm has been explained in detail in Section 3. Section 4 shows the computational results, and finally, the outcome of this research has been provided in Section 5.

2. OVERVIEW OF PRODUCTION PLANNING OPTIMIZATION AND MEMETIC ALGORITHM

This section explains some basic concepts of production planning, particularly, master production scheduling and memetic algorithms.

2.1. Master Production Scheduling

As per the American Production and Inventory Control Society (APICS), a master production plan is a statement of what the company needs to produce, which in turn becomes a sequence of scheduling choices that guide the Material Requirements Planning (MRP) scheme. It represents what the company intends to produce and is expressed in configuration, quantities and specific dates. The master plan is not a sales forecast, which represents a demand declaration. It should take into consideration many others such as the requests, material availability, pending orders, projected ending inventory levels, capacity availability, managerial policies and goals, among others.

MPS problems usually bring about conflicting objectives, such as minimizing of inventory, maximizing of service levels, and set-up times. For these reasons, it is suggested that heuristics and meta-heuristics be used to solve these kinds of issues. Several artificial intelligence meta-heuristics were used for optimization, including tabu search, genetic algorithms, simulated annealing, ant colony and beam search [7].

2.2. Mathematical Model of MPS

The mathematical model of MPS problem can be represented as a mixed integer program as follows [1]:

$$Min EI = \frac{\sum_{k=1}^K \sum_{p=1}^P EI_{kp}}{TH} \tag{1}$$

$$Min RNM = \frac{\sum_{k=1}^K \sum_{p=1}^P RNM_{kp}}{TH} \tag{2}$$

$$\text{Min } BSS = \frac{\sum_{k=1}^K \sum_{p=1}^P BSS_{kp}}{TH} \quad (3)$$

$$\text{Min } OC = \sum_{r=1}^R \sum_{p=1}^P OC_{rp} \quad (4)$$

Subject to:

$$BI_{kp} = \begin{cases} OH_k & \text{if } (p = 1) \\ EI_{k(p-1)} & \text{if } (p > 1) \end{cases} \quad (5)$$

$$EI_{kp} = \max \left[0, \left((MPST_{kp} + BI_{kp}) - GR_{kp} \right) \right] \quad (6)$$

$$MPST_{kp} = \sum_{r=1}^R MPS_{kpr} \quad (7)$$

$$MPS_{kpr} = BN_{kpr} * BS_{kpr} \quad (8)$$

$$RNM_{kp} = \max \left[0, \left(GR_{kp} - (MPST_{kp} + BI_{kp}) \right) \right] \quad (9)$$

$$BSS_{kp} = \max \left[0, (SS_{kp} - EI_{kp}) \right] \quad (10)$$

$$CUH_{rp} = \sum_{k=1}^K \frac{(MPS_{kpr})}{UR_{kr}} \quad (11)$$

$$CUH_{rp} \leq AC_{rp} \quad (12)$$

$$OC_{rp} = \max \left[0, (CUH_{rp} - AC_{rp}) \right] \quad (13)$$

2.3. Memetic Algorithm

In this formulation, each character expresses the value as follows. K : Total quantities of separate products, R : Total quantity of separate productive resources, P : Total number of planning periods, TH : Total horizon of planning, EI_{kp} : Ending inventory level generated for product k at period p , RNM_{kp} : Requirements not met for product k at period p , BSS_{kp} : Quantity bellow safety inventory level for product k at period p , OC_{rp} : Over capacity needed at resource r at period p , BI_{kp} : Initial inventory level of the product k at period p , OH_k : Initial storage available (on-hand), in the first planning period, GR_{kp} : Gross requirement for product k at period p , BS_{kp} : Standard lot size for product k at period p , N_{kp} : Net requirement for product k at period p , considering infinity capacity, SS_{kp} : Safety inventory level of product k at period p , UR_{kr} : Production rate for product k at resource r (units per hour), AC_{rp} : Available capacity, in hours, at resource p at period p , BN_{kpr} : Quantity of standard lot sizes needed for the production of the product k at resource r , at period p (number of lots), MPS_{kpr} : Total quantity to be manufactured of the product k at resource r at period p , $MPST_{kp}$: Total quantity to be manufactured of the product k at period p (considering all available resources), CUH_{rp} : Capacity used from the resource r at period p .

Headings In the mid of 1960s and 1970s, various researchers laid the foundations of the evolutionary algorithms (EAs) we now know. Since this period, and despite some difficult beginnings, most researchers have been interest in searching or optimizing existential and theoretical perspectives to understand and accept the existence and authenticity of these technologies. The same is true for other related techniques, for instance simulated annealing (SA), tabu search (TS), and the like. The name metaheuristics is used to collectively term these specific technologies. At the end of the 1980s, the term "memetic algorithms" (MAs) was given to refer to meta-heuristic family, and they tried various concepts of close-up disintegration-in those days-families like EAs and SA. The word "memetic" comes from the term "meme" written by Dougs in 1976 for the analogy of genes in the context of cultural evolution [14].

The memetic algorithm was named by Moscato [15] after a series of stochastic global search techniques that were incorporated into the framework of evolutionary algorithms in a wide range of lines, "problem-specific local search heuristics" and multi-agent systems [16]. The web treatment method using local search provides several names in the research paper. The most meme algorithm is the mixture of local search heuristic and genetic algorithm. This method is also called genetic local search, hybrid genetic algorithm, hybrid evolution algorithm. And other names. This hybrid approach has been applied to a large number of optimization problems with only great success [17]. For example, the memetic algorithm for

multi-objective optimization has been used to address home healthcare [18] and Constrained Weapons assignment (WTA) problem [19]. Different to traditional evolutionary computation (EC) methods, MAS fundamentally concern the exploding of all existing knowledge about the problem being studied. The problem domain of problem solving is not an optional mechanism, but the basic characteristics of MAs's features [20].

MAs are like a EAs theory based on population. This implies that the algorithm holds a number of solutions for the problem at hand, in particular a pool of multiple solutions at the same time. These solutions are inspired by EA terminology, following the nature on which these technologies are based. In the context of MAs the two terms will be used interchangeably. Each individual provides a prudent solution to the problem being solved. These solutions are subject to the process of reciprocal cooperation and competition, which seems to make the behavioral patterns of similar individuals clearer at this point. The first thing to consider is the high-level template of the basic population events: a generation, as shows in Figure 1.

As is evident, each generation comprises the updating of a population of individuals, which it is hopeful, will give rise to better solutions for the problem being addressed. There are three primary components: selection, reproduction, and replacement.

```

Process Do-Generation (↓↑ pop: individual [ ])
variables
    breeders, newpop : Individual [ ];
begin
    breeders ← Select-From-Population(pop);
    newpop ← Generate-New-Population(breeders);
    pop ← Update-Population (pop, newpop)
end
    
```

Figure 1. The generational template

As is evident, the selection uses the information provided by the fitness function, which takes responsibility for the population's competition. The information obtained by the temporary management function (fitness function) is used to evaluate the benefits of the individual in the population; then a group of population is selected for reproduction (reproduction is controlled by crossover and mutation) according to such goodwill. This selection can be made in a variety of ways. The most popular techniques are rank-based methods, fitness scale methods, and game-based methods (selecting individuals based on direct competition in a small subgroup of individuals). Replacement is closely related to this competitive aspect. This component ensures that the size of the group remains the same. To this end, individual in the new population are replaced by certain criteria for new solution. In most cases, this can be achieved by getting the best (guiding function) from new population (the so-called "plus" refresh strategy), or simply by capturing the best individual of the new pop and recording them. In pop to complete from the smallest ("comma" strategy) [21-26].

3. PROPOSED MEMETIC ALGORITHM FOR MOMPS

The Meta-algorithms are study strategies using a population-based approach where a group of collaborating agents and competitors participate in periods of individual enhancement to resolve intermittently. MA relates to a metaheuristics family whose primary theme is hybridization and is fundamentally interested in exploiting all accessible information about the issue under study [20]. Each local search algorithm investigates distinct processes for reaching individuals [22]. The suggested algorithm contains nine comprehensive steps:

Step 1: Define MPS Parameters

For MPS, the software implemented takes into consideration the following parameters (K , P , TH , OH_k , GR_{kp} , SS_k).

Step 2: Define the Objective Function

By using min-max normalization approach, MPS objective function can be defined as bellow:

$$Min Z = c_1 * \frac{EI}{EI_{max}} + c_2 * \frac{RNM}{RNM_{max}} + c_3 * \frac{BSS}{BSS_{max}} + c_4 * \frac{OC}{OC_{max}} \tag{14}$$

where, the coefficient c_1, c_2, c_3 and c_4 are used to indicate the significance of each MPS performance measure. $El_{max}, RNM_{max}, BSS_{max}$ and $OC_{max-max}$ are the maximum values of corresponding goals which are estimated from pre-processing step (warm-up period) in proposed algorithm runs. Since those values about objectives goals have totally different ranges, memetic algorithm is performed to place them on interval (0, 1).

Step 3: Create Initial Population

In this step, generating a population (Pop) of N_p individuals. In our implementation, this population is randomly generated according to the uniform distributions. The pseudo code of multi (resources, products and periods) may be written as follows:

```

for k=1:K
  for r=1:R
    if UR(k,r)≠ 0
  for p=1:FP
IP=randi([0,round(GR(k,p)/BS(k,p))],nPop,1)*BS(k,p);
    Pop=[Pop IP];
    IP=[];
  end
  else
  for p=1:P
    IP=zeros(nPop,1);
    Pop=[Pop IP];
    IP=[];
  end
  end
end
end
end

```

Step 4: Stopping Conditions

In this algorithm, stop the procedures and return the results after a given number of iterations (Niter).

Step 5: Selection of Pc and PL

Selection is a process in which solutions are chosen according to their objective function value. In this algorithm, a highest rank selection is used to select population for applying crossover (Pc) and population for local search (PL) from Pop. In this method, we select the first individual that has greatest fitness (minimum Z), after sorting the population in ascending order, while the other is selected randomly. This method allows to improve the individuals to carry some good properties from the individual, which has the highest fitness.

Step 6: Apply Arithmetic Crossover to Pc

Output:

The crossover produces new offspring individuals from parent individuals.

- Create two new individual's crossovers by exchanging some genes of two parent individuals.
- Produce two new offspring by combine two linear crossovers with two parent individual vectors based on the following equation:

$$\text{offspringone} = b * \text{Parentone} + (1-b) * \text{Parenttwo} \quad (1)$$

$$\text{offspringtwo} = (1-b) * \text{Parentone} + b * \text{Parenttwo} \quad (2)$$

Where b is a random weighting factor (chosen before each crossover operation).

Step 7: Apply Local Search Operators to PL

The operators are the Local Search's building blocks, since they are used to travel from solution to solution. The choice of operators controls the richness of a solution's neighborhood. Using a large variety of operators increases the size of the solution space that can be reached during each iteration. This algorithm is using swap operator and inverse movement operator to improve the solutions of PL.

Step 8: Evaluate Z (x) for PC and PL

After we apply arithmetic crossover to PC and Local Search Operators to PL, we will evaluate each individual x in PC and PL using MPS objective function which defines in (14).

Step 9: Select the new Population from (Pop, PC, PL)

The selection is carried out to obtain the next generation after combining the individuals in pop with PC and PL. Then elitism is ensured, and sorting is carried out on the basis of the objective function of the recent population. The new generation is consequently filled from the best solutions until the size of the population exceeds the existing size. Next, go to Step 4

4. RESULTS AND ANALYSIS

The production scenarios (S1, S2, S3, S4, S5, S6) are selected from Sultan [23-24], Ribas [25] and Supriyanto [13] to study applicability of memetic algorithm for solving MPS problem. Each scenario consists of (K, R and P) where K, R, P represent the numbers of production, production lines and periods respectively. All scenarios are tested using the software written in MATLAB 8.1 programming languages. The results obtained for solving MPS problem using proposed algorithm are compared with GA and SA algorithms as shown in Table 1.

Master production schedules created by GA, SA and MA algorithms. Other measures, like computer execution time, although quite important for some types of problems, have not been considered in the present study. Results obtained for these scenarios, for each technique used, are shown in Table 2. The solutions of SA are worse than GA in all production scenarios which is the same conclusion of Viera et al. [8]. The Tukey’s method was used to test the results of these algorithms. In this method, the results must be normalized as follows:

$$Normalized Ri = (Ri - \min(R1: R3)) / \min(R1: R3) \tag{15}$$

Where Ri is final solution of ith algorithm. These normalized finding should be evaluated using the methods of Tukey. The error portion of these interval plots has been presumed to be 0.05. From Figure 2, we notice that the normalized means of results of three algorithms have clear distinctions.

Table 1. Production Scenarios (S1-S6)

Production Scenario	(K, R, P)	Nvar(K*R*P)	Source
S1	(4,6,4)	96	Sultan et. Al. [23]
S2	(2,9,6)	108	Sultan [24]
S3	(4,4,7)	112	Ribas [25]
S4	(4,4,10)	160	Supriyanto [13]
S5	(4,4,20)	320	Supriyanto [13]
S6	(20,4,13)	1040	Ribas [25]

Table 2. The Computational Results of Comparison Algorithms

PS	ALG	EI	RNM	BSS	OC	Z
s1	SA	1843	908	320	6.33	7.125
s1	GA	1745	854	548	4.8	6.062
s1	MA	1093	205.5	196	1.8	2.241
s2	SA	35424	234	2517.7	9.31	10.073
s2	GA	35588	314	2485.7	14.33	15.101
s2	MA	34073	123	1731.96	3.8	4.446
S3	SA	5464.3	0	13.81	23.51	23.525
S3	GA	5226	986	585	4.33	4.532
S3	MA	4428.5	942.8	528.5	0.33	0.514
S4	SA	1354	718.4	469	8.66	9.284
S4	GA	1221	685	954	6.43	7.569
S4	ME	999.1	421.5	169.4	0	0.254
S5	SA	1165	354	75.6	4.9	5.076
S5	GA	1201	160	172	4.13	4.383
S5	MA	1074	0	69.5	0	0.089
S6	SA	5562.6	1048.9	66.41	101.83	101.865
S6	GA	5959.1	3981.5	2913	0	0.534
S6	MA	5549	121.5	2736	0	0.430

The results of the comparison algorithms are thus proven according to the outputs of Tukey's method. Considering the execution of these three algorithms, we can clearly conclude the effectiveness of the proposed memetic algorithm compared to the other algorithms.

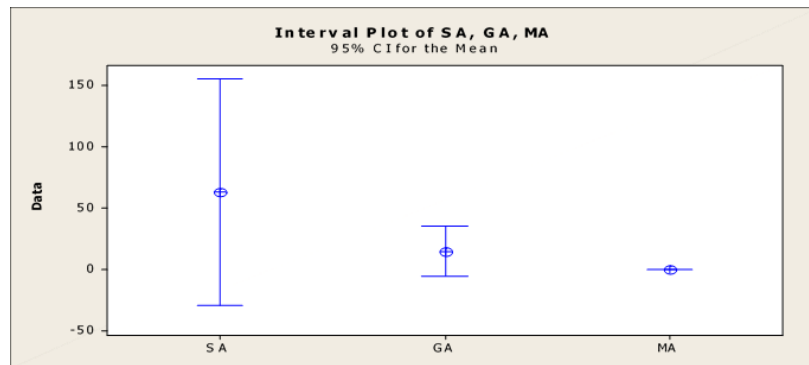


Figure 2. The Confidence Interval of Three Algorithms for Final Solutions

5. CONCLUSION

This paper proposes Memetic Algorithm combines evolutionary operators with local search operators for solving production planning, in particular, multi-objective MPS problems and its performance is evaluated by different production scenarios. The MA can solve MPS ' multi-objective model effectively owing to its balance between exploring and exploiting a search scope. It has the ability to intelligently determine how much, when, and where additional capacities are required so that the inventory can be reduced without affecting the level of customer service. The comparison results show the efficiency and capabilities of the proposed algorithm in finding the optimum especially with a problem of large number of variables. The performance achieved is quite satisfactory and promising for solving MPS models.

REFERENCES:

- [1] Soares, M.M. & Vieira, G.E., 2008. "A new multi-objective optimization method for master production scheduling problems based on genetic algorithm". *International Journal of Advanced Manufacturing Technology*, 41(5-6), pp.549-567. Available at: <http://link.springer.com/10.1007/s00170-008-1481-x>.
- [2] Garey, M.R. & Johnson, D.S., 1979. "Computers and intractability: a guide to the theory of NP-hardness".
- [3] Radhika, S. et al., 2016. "Multi-Objective Optimization of Master Production Scheduling Problems using Jaya Algorithm". (December), pp.1729-1732.
- [4] Ahmed, J.A., Mohsin, A. & Brifceni, A., 2015. "A New Internal Architecture Based on Feature Selection for Holonic Manufacturing System". 9(8), pp.1549-1552.
- [5] Wang, Y., 2013. "Constraint Cellular Ant Algorithm for the Multi-Objective Vehicle Routing Problem". *JSW*, 8(6), pp.1339-1345.
- [6] Guliashki, V., Toshev, H. & Korsevov, C., 2009. "Survey of evolutionary algorithms used in multiobjective optimization". *Problems of engineering cybernetics and robotics*, 60(1), pp.42-54
- [7] Ernani Vieira*, G. & Ribas, P.C., 2004. "A new multi-objective optimization method for master production scheduling problems using simulated annealing". *International Journal of Production Research*, 42(21), pp. 4609-4622.
- [8] Vieira, G.E., Favaretto, F. & Ribas, P.C., 2004. "Comparing genetic algorithms and simulated annealing in master production scheduling problems". In Proceeding of 17th International Conference on Production Research.
- [9] Radhika, S., Rao, C.S. & Pavan, K.K., 2013. "A differential evolution-based optimization for master production scheduling problems". *International Journal of Hybrid Information Technology*, 6(5), pp.163-170.
- [10] Sajja, R. & Rao, C.S., 2014. "A New Multi-Objective Optimization of Master Production Scheduling Problems Using Differential Evolution". *International Journal of Applied Science and Engineering*, 12(1), pp.75-86.
- [11] Wu, Z., Zhang, C. & Zhu, X., 2012. "An ant colony algorithm for master production scheduling optimization. *Computer Supported Cooperative Work in Design (CSCWD)*", 2012 IEEE 16th International Conference, pp. 775-779.
- [12] Bakar, M.R.A. et al., 2017. "Solution for Multi-Objective Optimization Master Production Scheduling Problems Based on Swarm Intelligence Algorithms". *Journal of Computational and Theoretical Nanoscience*, 14(11), pp.5184-5194. Available at: <http://www.ingentaconnect.com/content/10.1166/jctn.2017.6729>.

- [13] Supriyanto, I. & Noche, B., 2011. "Fuzzy multi-objective linear programming and simulation approach to the development of valid and realistic master production schedule". *Logistics Journal*, 7(1). Available at: <http://www.logistics-journal.de/proceedings/2011/3103> [Accessed April 15, 2018].
- [14] Moscato, P., Cotta, C. & Mendes, A., 2004. "Memetic algorithms". In *New optimization techniques in engineering*. Springer, pp. 53-85.
- [15] Moscato, P., 1989. "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms". *Caltech concurrent computation program*, C3P Report, 826, p.1989.
- [16] Cho, J. et al., 2017. "A Survey on Modeling and Optimizing Multi-Objective Systems", 19(3), pp. 1867-1901.
- [17] Eesa, A.S. et al., 2015. "Cuttlefish Algorithm-A Novel Bio-Inspired Optimization Algorithm", (June 2013).
- [18] Decerle, J. et al., 2019. "A memetic algorithm for multi-objective optimization of the home health care problem". *Swarm and Evolutionary Computation*, 44(August 2018), pp.712-727. Available at: <https://doi.org/10.1016/j.swevo.2018.08.014>.
- [19] Wang, J., Luo, P. & Zhou, J., 2017. "A Memetic Algorithm for Constrained Weapon Target Assignment Problems", pp.182-188.
- [20] Moscato, P. & Cotta, C., 2003. "A gentle introduction to memetic algorithms". In *Handbook of metaheuristics*. Springer, pp. 105-144.
- [21] Burke, E.K. & Silva, J.D.L., 2005. "The design of memetic algorithms for scheduling and timetabling problems". In *Recent Advances in Memetic Algorithms*. Springer, pp. 289-311.
- [22] Luca, B., 2018. "Local search algorithms for memetic algorithms: understanding behaviors using biological intelligence". In pp. 553-558.
- [23] Sultan, J.A., Jasim, O.R. & Salih, S.A., 2016. "An improved Genetic Algorithm for Fuzzy Production Planning Problems with Application". *University of Human Development*, 3, pp.390-396.
- [24] Sultan, J.A., 2013. "Proposed Hybrid Techniques for Solving Fuzzy Multi-Objective Linear Programming with Application". *University of Mosul*.
- [25] Ribas, P.C., 2003. "Análise do uso de t mpera simulada na otimiza o do planejamento mestre da produ o". Pontif cia Universidade Cat lica do Paran , Curitiba.
- [26] Wijayaningrum, V. N., & Mahmudy, W. F. (2016). "Optimization of Ship's Route Scheduling Using Genetic Algorithm". *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, 2(1), 180-186.