Warith Scientific Journal

"خوارزمية ICA-GA الهجينة لحل مشاكل الامثلية متعددة الاهداف"
"A Novel ICA-GA Algorithm for Solving Multiobjective Optimization problems"

سرمد عبد الخالق صالح
Sarmad Abdulkhaleq Salih
قسم الرياضيات، كلية التربية للعلوم الصرفة، جامعة الحمدانية، الموصل، العراق
Department of Mathematics, College of Education for Pure Sciences, University of Hamdaniya, Mosul, Iraq,
sarmadsalih@uohamdaniya.edu.iq.

عمر رمزي جاسم
Omar Ramzi Jasim
قسم ادارة الاعمال، كلية الإدارة والاقتصاد، جامعة الحمدانية، الموصل، العراق
Department of Accounting ,College of Administration and Economics, University of Hamdaniya, Mosul, Iraq,
omar.ramzi89@uohamdaniya.edu.iq.

المستخلص

يُعد جدول الإنتاج الرئيسي من النماذج المهمة والذي يربط الاستخدام الفعال لموارد الإنتاج بخدمة العملاء والتي تعتبر بمثابة واجهة حاسمة بين التسويق والتصنيع. إحدى المشكلات الرئيسية المتعلقة بالعملية هي سوء إدارة (MPS) والتي يتم من خلالها تقليل رضا العملاء. تم في هذا البحث تطبيق خوارزمية تطورية هجينة (ICA-GA) لحل مشاكل متعددة الأهداف (MPS) تجمع بين أفضل ميزات الخوارزمية الجينية (GA) والخوارزمية التنافسية الإمبريالية (ICA). يتم تمثيل مستعمرات كل إمبراطورية من خلال عدد صغير من السكان في الخوارزمية المقدمة، وهم يتفاعلون مع بعضهم البعض من خلال العوامل الجينية. توضح النتائج العددية لخوارزمية ICA-GA والتي تم اختبارها على خمسة سيناريوهات إنتاج، فعالية وإمكانات الخوارزمية الهجينة في تحديد أفضل الحلول. عند مقارنتها بنتائج (GA) و (SA) في جميع مواقف الإنتاج مما تؤدي حلول ICA-GA إلى انخفاض مستوى المخزون، والحفاظ على مستوى عالٍ من رضا العملاء، وتتطلب قدرًا أقل من العمل الإضافي.

الكلمات المفتاحية: الخوارزمية الهجينة، الخوارزمية الجينية، خوارزمية التنافسية الإمبريالية، جدولة الإنتاج الرئيسي.

Abstract.

The Master Production Schedule (MPS), which connects effective utilization of production resources to customer service, is a "crucial interface between marketing and manufacturing". One of the main issues with operation is mismanagement of the (MPS), which has the ability to lower customer satisfaction. This work presents a hybrid evolutionary algorithm (ICA-GA) for solving multiobjective (MPS) problems that combines best features of the genetic algorithm (GA) and the imperialist competitive algorithm (ICA). The colonies of each empire are represented by a small population in the algorithm that is being given, and they interact with one another through genetic operators. The numerical results of the (ICA-GA) algorithm, which was tested on five production scenarios, demonstrate the effectiveness and potential of the hybrid algorithm in locating the best solutions.When compared to the outcomes of (GA) and (SA) in all production situations, the (ICA-GA) solutions result in a lower inventory level, maintain a high level of customer satisfaction, and require less overtime.

Keywords: hybrid algorithm, genetic algorithm, imperialist competitive algorithm, master production scheduling.

1. Introduction:

Production at the tactical level, or (MPS), is a well-established and extensively used industry paradigm. Creating tactical production plans can be difficult because demand varies over time for a variety of unanticipated reasons; however, because the industry lacks sufficient resources, it is difficult to predict demand precisely, and the production planner struggles greatly with these items. While it is possible to suggest increasing capacity during high demand periods, doing so will cost money and take time away from the business. Additionally, during low demand periods, the company will have even more idle capacity.When creating an (MPS), competing goals are taken into account, including minimizing inventory levels, optimizing service levels, and making effective use of resources. Regretfully, when the production scenario rises, the complexity and work required to create a master plan increase significantly, particularly when resources are scarce, as they are in the majority of enterprises. Because of this complexity, industries typically rely on straightforward heuristics that are put into spreadsheets and offer a fast plan, but they can also be expensive and inefficient. Thankfully, new concepts are frequently put forth by researchers to enhance production scheduling, such as the application of heuristics based on artificial intelligence [1].

Using artificial intelligence or meta-heuristic methods like (GA), simulated annealing, tabu search, etc., could identify solutions that are close to, or even better than, optimal in a reasonable amount of time [2].

The primary benefits of evolutionary algorithms are their ability to escape from local minima, their independence from the objective function's evaluation of gradients, and their ability to work with both differentiable and continuous objective functions [3].

The motivation for combining two or more different algorithms into a single hybrid algorithm came from the possibility that this new algorithm will perform better than each of its component algorithms individually. This leads to the addition of a new family of algorithms to the hybrid algorithm strategies. The hybrid algorithm creates a combined algorithm with several benefits, such as speedier and/or higher-quality solution generation, by fusing the best aspects of the individual algorithms. Moreover, it can effectively handle problems with large input sizes, especially those involving (NP) difficulties [3].

This work presents the unique algorithm (ICA-GA) as a solution to multiobjective (MPS) production planning challenges.

The paper is organized as follows: Section 2 provides a condensed version of the production planning problem, and Section 3 presents the proposed algorithm for addressing it. In Section 4, the computational results of the proposed approach have been compared with those of other methods. Section 5 contains the study's conclusion.

2. Planning For Production :

Some of the basic ideas of production planning are explained in this section, with special emphasis on (MPS) and his mathematical model:

A. MPS: Foundations:

One role in integrated production planning and control is (MPS), which converts business planning's strategic goals into an expected statement of production, which serves as the basis for all other lower-level schedules [4]. A component of the wider production plan, the (MPS) is also an operational plan. As a result, [6] thought that (MPS) was the most crucial planning and control schedule in a company.

An MPS, according to the American Production and Inventory Control Society (APICS) [7], is a declaration of what a business intends to create. There are a number of planning considerations that direct the material requirements planning (MRP). It displays the exact dates, amounts, and configuration of the products the company intends to produce. Despite not being a sales prediction, this is one of the most important system data points in the master plan. Open orders, the availability of materials, available capacity, managerial policies and objectives, and more are important considerations that need to be made. It also shows the capacity used, available to promise (ATP), and expected inventory balance.

Wu et al. [8] developed a mathematical model of (MPS) and a (GA) that integrated several techniques in order to satisfy the constraints of developing an optimal (MPS) for production lines that incorporate both assembly and processing. To solve the (MPS) challenge in production planning, Vieira & Ribas [9] employed simulated annealing. This study exposes several drawbacks of simulated annealing, including bypassing the local optimum.

Soares & Vieira [10] presented a novel (GA) structure to address the (MPS) problem. The fitness function is created in this study to reduce inventory level, maximum service level (minimize requirement not satisfied), minimize overtime, and minimize inventory level below safety stock. Lastly, [11] has compared (GAs) with simulated annealing for (MPS) problems.

B. Mathematical Model of MPS:

A mixed integer program can be used to mathematically model the (MPS) problem as follows [10]:

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

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

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

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

$subject\ to$:

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

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

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

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

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

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

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

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

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

When (K): Denotes the total amount of various products. (R): The total amount of various producing resources. (P): The total amount of planning intervals. (TH): The entire planned horizon. (EI_{kp}): Product k's ending inventory level at period p. (RNM_{kp}): Product k's requirements were not fulfilled during period p. (BSS_{kp}): Product k's quantity below the safety inventory level at period p. (OC_{rp}): Excess capacity required during period p at resource r. (BI_{kp}): Product k's initial inventory level at time p. (OH_k): Starting inventory that is available at the start of the schedule period. (GR_{kp}): Product k's gross need at time p. (BS_{kp}): Product k's standard lot size during period p. (NR_{kp}): Net product requirement for product k at time p, taking infinite capacity into account. (SS_{kp}): Product k's safety inventory level at time p. (UR_{kr}) : Product k production rate (in units per hour) at resource r. (AC_{rp}): Hourly capacity available at resource r during period p. (BN_{kpr}): The number of standard lot sizes required at period p (resource r) for the manufacturing of product k (number of lots). (MPS_{kpr}): The total amount of the product k that must be produced at resource r during period p. $(MPST_{kp})$: The total amount of the product k that must be produced during period p, taking into account all available resources. (CUH_{rp}): Capacity utilized during period p from resource r. (CUP_{rp}): The percentage derived from the relationship between the quantity of hours used in period p to utilize resource r and the quantity of hours available for the same resource and period.

3. Procedure For Paper Submission :

The detailed presentation of the solution technique has been made in this section. This means that the conventional (ICA) and (GA) should be explained first. After this, the hybrid algorithm will be explained:

A. ICA:

ICA is a revolutionary evolutionary algorithm in the field of evolutionary computation, based on the socio-political evolution of humans. The process starts by generating a set of feasible random solutions inside the search space of the optimization problem. The world's countries, or the beginning population, are referred to as the randomly produced points. Both colonial and imperialist states fall within these two groups. The more powerful imperialists own more colonies. The cost function of the optimization problem determines a country's power. In order to create the first empires, some of the strongest founding nations those with the lowest cost function value become imperialists and start annexing other countries referred to as colonies [12].

The three main operators in this algorithm are Assimilation, Revolution, and Competition. This algorithm applies the assimilation policy. Based on this philosophy, the imperialists try to improve the political, cultural, and economic circumstances of their colonies. This policy allows the colony to be excited for the imperialists. Assimilation moves the colonies of each empire closer to the imperialist state in terms of socio-political characteristics (optimization search space). Due to revolution, the positions of some of the countries in the search space shift suddenly and arbitrarily. Through assimilation and revolution, a colony may rise to a higher position and have the chance to topple the imperialist regime now in place within the empire [13].

The colonies start to converge on their imperialists as the imperialists attempt to obtain more colonies in the competitive operator. Every empire wants to annex and take over the colonies of other empires. The power of an empire is derived from the power of its colonists and imperialists. At each stage of the algorithm, each empire has an equal chance, depending on their might, of taking control of one or more of the colonies of the weaker empire. Consequently, the contest will upend the weak imperialists and empower the stronger ones. Eventually, all of the smaller empires will fall and only one powerful empire will survive. The weaker empires will lose all of its colonies and their imperialists will become colonists of the other empires. The imperialists randomly assign each colony to a different group. Imperialists with more power take more colonies [14]. The algorithm continues to run until a predetermined stop condition for example, the eradication of every imperialist is satisfied. Imperialism and its colonies will then be on an equal footing [12], [13].

B.GA:

The GAs are a collection of computational models developed by Holland [15] based on the concepts of natural biological evolution. GA codes a single chromosome as a potential fix for a specific problem. The initial population of chromosomes, or the set of initial search locations in the problem's solution space, is where the approach begins. Next, genetic operators such as crossover, selection, and mutation are employed to produce a new generation of chromosomes. It is expected that chromosomal quality will rise with each generation due to the operators' guiding principle of "survival of the fittest, extinction of the unfitness". This process is repeated until the termination condition is satisfied, at which point the best chromosome from the most recent generation is proclaimed as the final solution [1].

C. An ICA Hybrid:

This section presents a novel hybrid algorithm that combines two algorithms. The pseudo-code for the (ICA-GA) is shown as follows:

Protocols of the (ICA-GA) :

Step 1: Setting up the (ICA-GA) algorithm's settings.

Step 2:

2.1: Identify the (MPS) problem's parameters.

2.2: Create a few arbitrary nations and then determine the EImax, RNMmax, and BSSmax.

2.3: Establish empires out of the most strong nations.

2.4: Assign remaining nations at random to various empires based on equality.

Step 3: "Nd=Nd+1" decade loop.

Step 4: Perform for i=1,2,...,N_imp; %Procedures for Genetic Algorithms %

4.1: "Selection".

4.2: "Crossover".

4.3: "Mutation".

Step 5: %% Imperialist Competitive Algorithm Procedures%%

5.1: Integrate colonies with their imperialist system.

5.2: Revolutions in countries.

5.3: "If need", swap an imperialist for the finest colony.

5.4: Determine the entire cost of an empire.

5.5: Competitive imperialism.

5.6: Destroy the weak empires.

Step 6: Terminating Criterion Control; Continue from Steps $(3-6)$ until a final requirement is met.

1. **Input Data.** The software used for the master scheduling optimization considers as many factors as it can that are present in real-world industrial settings, including:
   - Product count and description.
   - Product count and description (production lines, workstations, machines, production cells).
   - Product count and description.
   - Time period count and duration (various length periods are acceptable).
   - starting (on-hand) inventories: product amounts at the start of the planning period.
   - gross requirements, which are calculated from forecasts and customer orders and represent the required quantity per product per period.
   - Production rate refers to the quantity of a product that a resource can produce in a given amount of time.
   - Batch sizes are the production standard lot sizes per product per period.
   - Safety inventory levels are the amount of inventory per product per period.
   - Setup times are the amount of time required for each product, regardless of the order of operations.
   - The monthly capacity that each resource can possess.

2. **The objective Function.**

The following is a definition of the MPS objective function and its limitations:

$$Min\ Z = c_1 * EI + c_2 * RNM + c_3 * BSS + c_4 * OC \qquad (14)$$

where each (MPS) performance measure's relevance is indicated by the coefficients $(c_1, c_2, c_3$ and $c_4)$. The goal of the objective function is to reduce the average of the following: over capacity (OC), below safety stock (BSS), requirement not met (RNM), and ending inventory level (EI). Because the objectives and goals on (14) have values that fall into entirely separate ranges, min-max normalization is used to place them on the interval $(0,1)$. Consequently, the goal function mentioned above becomes:

$$Min\ Z = c_1 * \frac{EI}{EI_{max}} + c_2 * \frac{RNM}{RNM_{max}} + c_3 * \frac{BSS}{BSS_{max}} + c_4 * OC \quad (15)$$

where the maximum values of the relevant goals, which are calculated from the pre-processing stage (warm-up period) in the proposed algorithm runs, are denoted by the variables EI_{max}, RNM_{max} and BSS_{max} .

3. Country National Structure Developed. The content and shape of the nation (chromosome) in (MPS) issues fluctuate in the manner of representation, in contrast to most representations found in the literature, which consider a single chromosome represented by a single bit vector structure.

(Fig. 1) [16] depicts the basic model of the structure that is implemented for a scenario with three goods, four resources, and three periods. An alphabetic set of integer positive numbers serves as the structural representation of a nation. Every sphere in the structure represents a gene. A gene set creates a branch that shows how amounts of a specific product to be produced at various resources will vary over time.

A group of branches that together form a branch group represents the entire distribution of quantities to be made of all the products at every resource, in a given time period. A group of branch groups constitutes an (MPS) individual in its entirety. The number of time periods in the master plan horizon establishes the length of the set. The population of (MPS) countries will vary in accordance with the (ICA-GA) configuration as it searches for the ideal country (master schedule or solution).

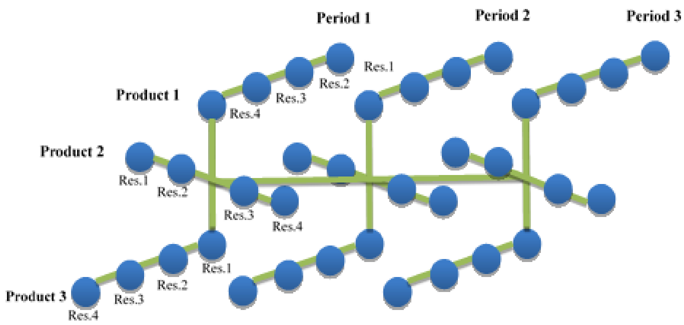

Fig. 1 Structure of the (MPS) Country.

4.Initial Population Creation Function.

The population's number of countries and the initial population's generation technique have a significant influence on the algorithm's performance and the quality of the results. To improve the search space, as many different countries as feasible should be included. The following is the pseudo code for the population formation function with multiple resources, multiple products, and numerous periods:

```
"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"
```

where the (m*n) matrix with integer values taken from the discrete uniform distribution on the interval ("imin","imax") is returned by "randi([imin,imax],m,n)" [10]. This heuristic method fills in as much diversity as feasible while ensuring that values consistently adhere to the normal lot size constraint. Assume the following hypothetical scenario: the first country in the population in the first period would have the genes "{0;500;1,000;1500}," the second country would have the genes "{2,000; 3,000; 2,500; 500}," and so on, sequentially for every individual in the population. The gross requirement for a given product at a given time period (time bucket) would be (3,000) units, the standard lot size would be (500) units, and there would be four possible productive resources available to make the product.

The best countries (those with the lowest costs) in the original population, N_imp, were chosen to be the imperialists, and the remaining countries, N_col, were equally distributed as colonists among the various empires at random.

5. **GA was applied to every empire's territories.** Genetic operators of selection, crossover, and mutation are applied to colonies, drawing inspiration from (GA), with the goal of diversifying the imperialist population.
   a) **Operator for selecting the highest rank:** Colonies can be modified to inherit certain advantageous qualities from the most fit colony by employing this procedure, which selects the first colony with the highest fitness (lowest cost) and chooses the other at random.
   b) **Single-point crossover operator**: One crossing site (k) is uniformly chosen at random within the interval $[1, 2, \ldots, \mathrm{N}_{\mathrm{var}} - 1]$ in single-point crossover. The variables are then swapped between the colonies around this point, resulting in the production of two new colonies. A nation's superior genetic plan is passed down from one generation to the next at the preset crossover probability (p_c). through the crossover process. A kid nation has the opportunity to jump from the local optimum when crossover operation occurs between two parent colonies that are situated at local optima. When two colonies cross over and produce suboptimal results, this algorithm returns to the original colonies before the crossover.
   c) **Mutation operator:** Population diversity may be lost during the (ICA-GA) 's evolutionary process, and premature convergence is a constant. When population variety needs to be restored, mutation comes in rather handy. Include changing a particular colony variable by adding or removing one production batch size. At the predefined mutation probability (p_m). the colonies are chosen. When a mutation creates a terrible colony, this algorithm returns to the colony that existed prior to the mutation.
6. **Moving the colonies of an empire toward the imperialist (assimilating).** Imperialists countries started to improve their colonies. This fact has been modeled by moving all the colonies toward the imperialist. Through this movement, some parts of a colony's structure will be similar to the empire's structure. The assimilation operator can be modeled as:

$$\{x\}_{new} = \{x\}_{old} + round\left(\frac{\beta * d * \{rand\} \otimes \{V_1\} + U(-1,1) * tan(\theta) * d * \{V_2\}}{BS}\right) * BS \qquad (16)$$

Where $\{x\}_{new}$ is a new position of colony, $\{x\}_{old}$ is a previous position of colony, β is assimilation parameter, and d is the distance between colony and imperialist, θ is a random amount of deviation added to the direction of movement, $\{rand\}$ is a random vector, $\{V_1\}$ is the base vector starting the previous location of colony and directing to the imperialistic.

The length of this vector is set unity, $\{V_2\}$ is orthogonal vector on colony-imperialist ($\{V_1\}.\{V_2\} = 0$). Colonies are moved to a new site in the (ICA-GA) process using different random values, as shown in (Fig. 2).

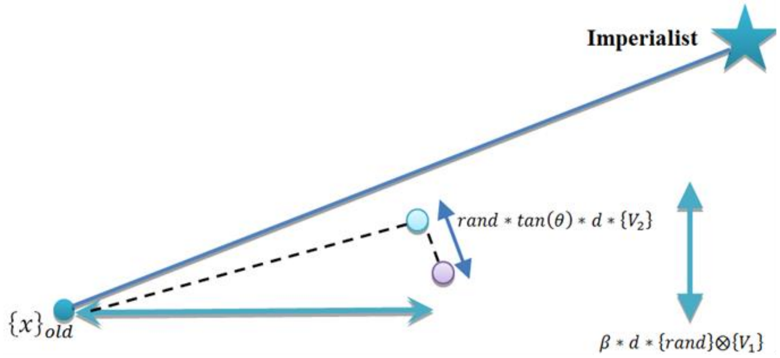

Fig. 2. Colonies moving to new place within (ICA-GA).

7. Revolution. Every time we iterate, we randomly swap out some of the weakest colonies' genes for new ones. One possible model for the revolution operator is:

$$Col(j).Pos(k) = round\left(\frac{unifrnd\big(VarMin(k), VarMax(k)\big)}{BS}\right) * BS \qquad (17)$$

where $Col(j)$, $VarMin(k)$ and $VarMax(k)$ indicate the lowest and higher bounds for gene (k), respectively, and $Pos(k)$ represents gene (k) in colony (j). $unifrnd\,(A, B)$ yields a random integer that is produced from continuous uniform distributions with the upper and lower ends designated by (A and B), respectively [10]. The revolution rate, abbreviated as (p_r), is the replacement rate.

8. swapping places between a colony and an imperialist. As these colonies get closer to becoming imperialists and in certain cases have revolutions, it's likely that some of them will become more powerful than their respective imperialists. In this case, the colony and its relevant imperialists take the opposite viewpoint. Moving forward, this new country will be employed as the imperialist in the algorithms.

9. Total power of an empire. The power of the imperialist nation mostly determines an empire's overall power, whereas the power of an empire's colonies also has a small but significant impact. As a result, the total cost equation is:

$$T.C_n = cost(imperialist) + \varepsilon * mean\{\, costs(colonies\ of\ empire)\} \quad (18)$$

Where $(T.C_n)$ represents the total cost of the n^{th} empire and ε is a positive number which is considered to be less than 1.

10. Imperialistic competition. The most important (ICA) process is imperialist rivalry, in which all empires aim to subjugate each other's colonies. Eventually, stronger empires wrest colonies from weaker ones. This procedure is repeated by choosing the weakest colony of the weakest empire and moving it to the appropriate empire, which is chosen by competition among all empires. A schematic of this process is shown in (Fig. 3).

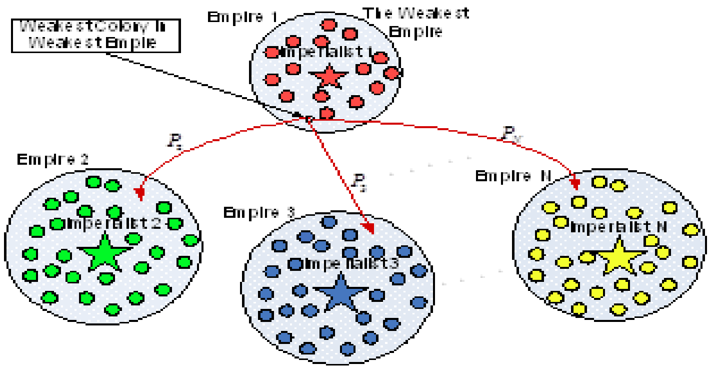

Fig. 3. Imperialistic competition.

An empire's likelihood of owning the weakest colony of the weakest empire increases with itspower.

Empire 1 is seen as the weakest empire in this picture, with one of its colonies undergoing a competitive process. The empires ranging from (2 to n) are vying for control of it. First, the possession probability was determined by taking into account the empire's entire might in order to start the competition. A roulette-wheel-like process is employed to assign the chosen colony to one of the empires based on a proportionate likelihood, given the possession probability of each empire.

11. Eliminating the powerless empires. In the imperialistic struggle, weak empires will fall and their colonies will be divided among the stronger powers. Different parameters can be identified for considering an empire helpless in the simulation of collapse mechanisms. We will assume in this work that the loss of all colonies results in the collapse of an empire.

12. Stopping criteria. In theory, the competition can go on until there is just one individual left in the search area. Nevertheless, certain criteria can be chosen as end points, such as completing a certain number of iterations or seeing a very small improvement in the objective function.

4. Numerical Rsults:

Several production situations were tried using software built in (Matlab 8.1) programming languages and run on the Intel Core 2 Duo in order to confirm the viability of the suggested technique to solve the multiobjective (MPS) issues. (2.20) megahertz. The outcomes of the (MPS) issue solution are contrasted with those of alternative algorithms.

Five production scenarios are used to test the proposed (ICA-GA). In contrast, these (5) scenarios are likewise subjected to the execution of (GA) and SA [2], [9]. The production scenarios' details are displayed in Table 1. The (ICA-GA) algorithm's parameter settings are explained as follows: (N_{imp}) is a multiple of (0.1) of the population size, crossover rate is (0.9), mutation rate is (0.7), assimilation coefficients are set to (2.0), θ are set to (45°), and revolutionary rates are set to (0.1). and (500) generations is the maximum number of iterations that may be made without any improvement. Twenty runs of the algorithm are performed for every production scenario.

Table 1. Production scenarios (S1-S5)

Production Scenario	*(K,R,P)*	N_{var}^{*}	(c1,c2,c3,c4)	Source
S1	(2,9,6)	108	(1,1,1,1)	Sultan [16]
S2	(4,4,7)	112	(1,1,1,1)	Ribas [2]

S3	(4,4,10)	160	(1,1,1,1)	Supriyanto [17]
S4	(4,4,20)	320	(1,1,1,1)	Supriyanto [17]
S5	(20,4,13)	1040	(1,1,1,1)	Ribas [2]

* $N_{var}=K*R*P$

Ultimately, Table 2 presents a comparison of performance metrics between the (GA), (SA), and (ICA-GA) solutions for the specified production situations. Similar to Viera et al. [11], the final (GA) solution outperforms the (SA) solution in all production scenarios; however, the computational cost of (GA) is significantly higher. The hybrid method outperforms the other two algorithms in terms of both computing time and final solution.

Table 2 .presents a comparison of the computing time and final solution of three algorithms.

S.	*Alg.*	*EI*	*RNM*	*BSS*	*OC*	*Z*	*CPU (sec)*
S1	GA	35588	0	448.9	0.13	1.142	1197
	SA	35424	0	517.7	0.31	1.241	1173
	ICA-GA	**34073**	**0**	**1731.7**	**0**	**0.918**	**1068**
S2	GA	5226	986	585	4.33	1.740	1202.7
	SA	5464.3	0	13.81	23.51	1.719	1179.2
	ICA-GA	**4428.5**	**942.8**	**528.5**	**0.33**	**1.431**	**1104.3**
S3	GA	1221	685	954	6.43	1.727	1687.2
	SA	1354	718.4	469	8.66	1.799	1654.3
	ICA-GA	**999.1**	**421.5**	**169.4**	**0**	**0.406**	**1440.1**
S4	GA	1201	160	172	4.13	0.865	3306
	SA	1165	354	75.6	4.9	0.983	3241.6
	ICA-GA	**1074**	**0**	**69.5**	**0**	**0.239**	**3060.5**
S5	GA	5959.1	3981.5	2913	0	1.039	5082
	SA	5562.6	1048.9	66.41	101.83	1.844	4876
	ICA-GA	**5549**	**121.5**	**2736**	**0**	**0.681**	**4720**

Fig.4 averagely, illustrates the graph of convergence of S2 and S4 production scenarios. It can be seen from this fig. that the convergence of the (ICA-GA) is faster than the (SA) and (GA).

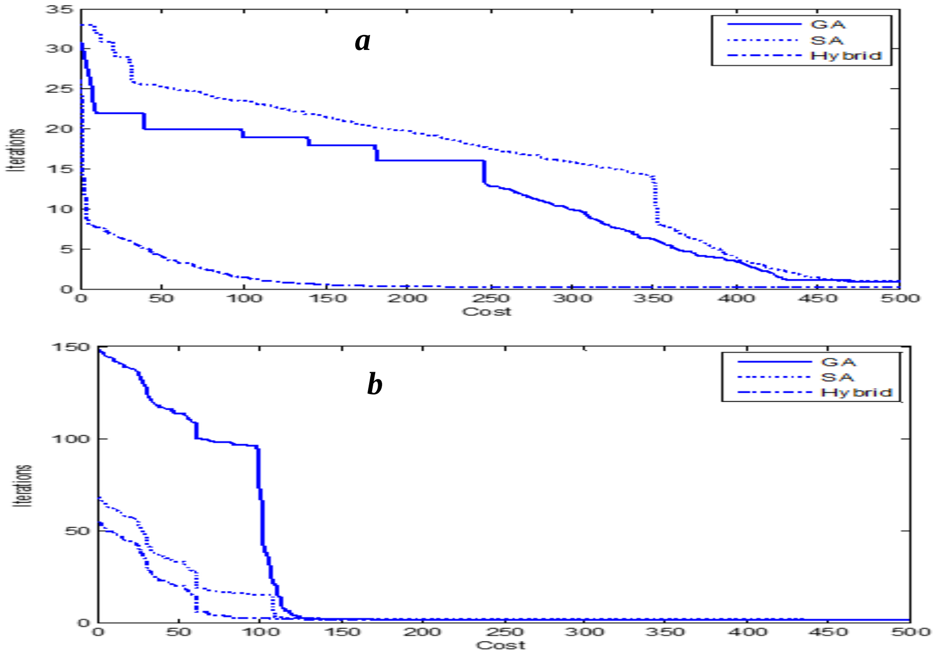

Fig. 4. The average of convergence of (a) S4 production scenario (b) S2 production scenario

Tukey's approach has been applied to test the outcomes of three algorithms. This approach has made use of the normalized data. The following is how these normalized data were calculated:

$$Normalized\ (S_i) = ((S_i) - \min(S_1 : S_3))/(\min(S_1 : S_3)) \qquad (18)$$

where (S_i) is the (i^{th}) algorithm's final solution or computing time. Tukey's technique should be used to test these adjusted results. For the final answer and the calculation time, an interval plot is taken into consideration. It has been assumed that these interval graphs have an error ratio of (0.05). These algorithms' interval charts are shown in (Fig. 5a) and (Fig. 5b), respectively, for the computational time and the outcome. We discovered that there are substantial discrepancies between the means of the normalized computational outputs of the three methods in (Fig. 5a), and there are also several significant differences between the final solutions of (GA), (SA), and (ICA-GA) in (Fig. 5b).

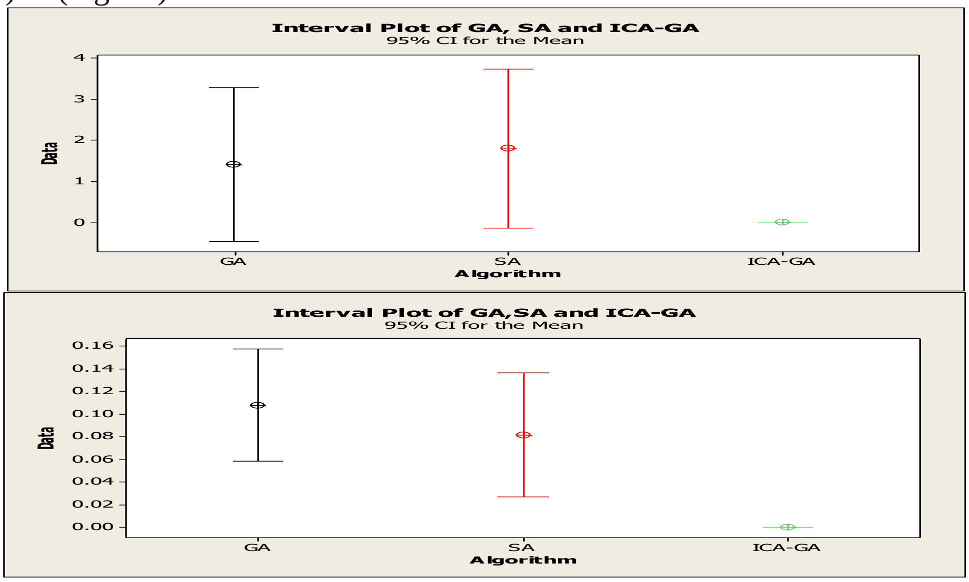

Figure 5: Confidence intervals of three algorithms for the final answer and computation time.

Three algorithms' outcomes are then proven based on the findings of Tukey's approach. After carrying out these three algorithms, it is evident to us how much more successful the suggested hybrid method is than the others.

5. Discussion:

For a variety of optimization problems, a large proportion of hybridizing population-based meta-heuristics have been presented. This study introduced (ICA-GA), a novel hybrid technique that combines (ICA) and (GA). In comparison to (GA) and (SA), the viability and effectiveness of (ICA-GA) for resolving (MPS) issues in various production settings are examined. The suggested algorithm can find globally optimal solutions in a comparatively short number of iterations, according to the results.

The biggest inventory level and overtime are often produced by the (GA) and (SA) solutions. As we can see in (Fig. 6a), It appears that these approaches would not be able to assign overtime appropriately (the "where and when" question is not handled accurately). The graph illustrates how the available resources are not equally allocated to meet the whole need capacity of the (S3) production scenario utilizing the (GA) solution. A portion of the resources are underutilized, a portion are completely employed, and some are overloaded. In theory, inventory levels should be able to be lowered provided overtime is assigned to the appropriate resources at the appropriate times. On the other hand, (ICA-GA) solutions generate less inventory over time than (GA) and (SA) systems. As we can see in (Fig. 6b), it appears that (ICA-GA) can successfully handle when the additional capacity must be substituted, how much it is needed, and at which resource should be added.

Exploration and exploitation are two major issues in the evolutionary search process that can be appropriately handled using (ICA-GA). Mutation is frequently seen as an exploratory operator in (GA) since it provides fresh information in an objective manner [18]. This is the purpose of revolution policy in (ICA). This purpose is fulfilled by assimilation policy in (ICA), and crossover, which recombines the older content of the parents into new configurations, can be seen as an experimental operator in a similar manner [18].

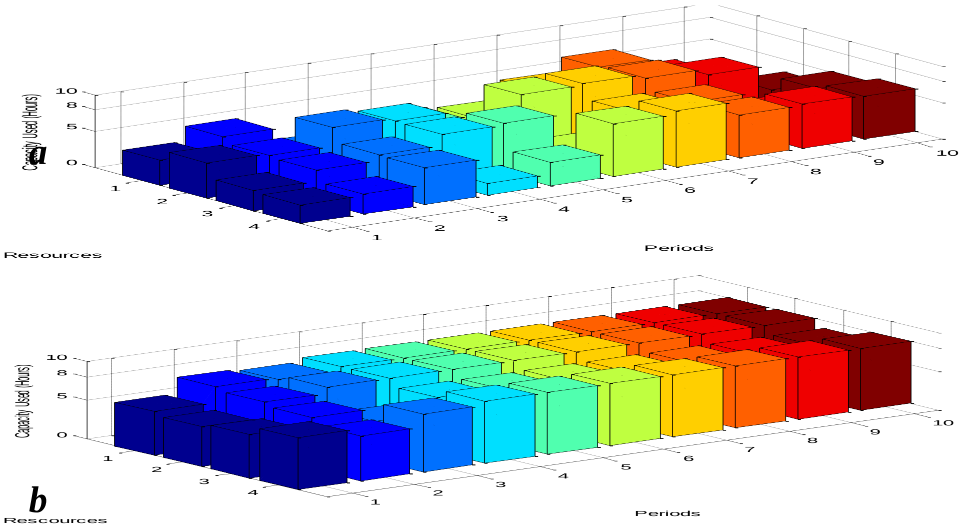

Fig. 6. The total requirement capacities of (S3) production scenario using (a) (GA) (b) (ICA-GA).

6. Conclusion:

In order to solve multiobjective (MPS) issues, this research presents a novel hybrid approach that combines (ICA) and (GA). Its effectiveness is assessed using a range of production scenarios. The results of the simulations show that the suggested algorithm performs exceptionally well in terms of convergence speed and accuracy of the global optimization solution; that is, it can intelligently decide "how much, when, and where" additional overtimes are needed in order to reduce inventory without compromising customer service standards within a reasonable time frame. The outcomes demonstrate the new hybrid algorithm's effectiveness and capacity to discover the optimal solution. Its performance is remarkably superior to those of other algorithms like (GA) and (SA). All manufacturing scenarios can benefit from the highly satisfactory and promising performance attained.

References:

[1] . D.E. Goldberg, Genetic Algorithms in Search Optimization & Machine Learning, Addison-Wesley. Reading,1989.

[2] P.C. Ribas, "Análise do uso de têmpera simulada na otimização do planejamento mestre da produção" Pontifícia Universidade Católica do Paraná, Curitiba, 2003.

[3] T. Bäck, Evolutionary algorithms in theory and practice, New York, Oxford Univ Press, 1996.

[4] P. Higgins, J. Browne, "Master production scheduling: a concurrent planning approach" Production Planning & Control, vol. 3, no.1, 1992 pp. 2-18.

[5] T.E. Vollmann, W.L. Berry, D.C. Whybark, Manufacturing planning and control system, New York McGraw-Hill,1997.

[6] N. S. Slack, Operation Management, Prentice Hall, 3rd Edition, New Jersey, USA, 2001.

[7] J. F. Cox, J. H. Blackstone, APICS Dictionary, 10th Editio, APICS, Alexandria, 2001.

[8] Y. Wu, M. Liu, C. Wu, "A genetic algorithm for optimizing the MPS of a processing-assembly production line with identical machines", Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, 2002.

[9] G. E. Vieira, C. P. Ribas, "A new multi-objective optimization method for master production scheduling problems using simulated annealing" International Journal of Production Research, vol. 42, No. 21, 2003, pp. 4609-4622.

[10] M.M. Soares, G.E. Vieira, "A New multi-objective optimization method for master production scheduling problems based on genetic algorithm, International Journal of Advanced Manufacturing Technology, DOI 10.1007/s00170-008-1481-x, 2008.

[11] G.E. Vieira, F. Favaretto,P.C. Ribas, "Comparing genetic algorithms and simulated annealing in master production scheduling problems" Proceeding of 17th International Conference on Production Research, Blacksburg, Virginia, USA, 2004.

[12] E. A. Gargari, C. Lucas, "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition", IEEE Congress on Evolutionary Computation, 2007, pp. 4661–4667.

[13] E. A. Gargari, F. Hashemzadeh, R. Rajabioun, C. Lucas, "Colonial Competitive Algorithm: A novel approach for PID controller design in MIMO distillation column process" International Journal of Intelligent Computing and Cybernetics (IJICC), vol.1, no.3, 2008, pp. 337–355.

[14] M. Garey, D. Johnson, Computer, complexity and intractability. A guide to theory of NP-Completeness, Freeman, San Francisco, USA, 1979.

[15] J. H. Holland, Adoption in Natural and Artificial Systems, University of Michigan, Ann Arbor, 1975.

[16] J. A. Sultan, Proposed Hybrid Techniques for Solving Fuzzy Multi-Objective Linear Programming with Application, Master thesis, university of Mosul, Mosul, Iraq, 2013.

[17] S. Supriyanto, Fuzzy Multi-Objective Linear Programming and Simulation Approach to the Development of Valid and Realistic Master Production Schedule, Doctor thesis, University of Duisburg, Essen, Germany, 2011.

[18] L. Hansheng, K. Lishan, "Balance between exploration and exploitation in genetic search", Journal of Natural Sciences, vol. 4, 1999, pp. 28–32.