Jestr

Journal of Engineering Science and Technology Review 13 (2) (2020) 127 - 134

JOURNAL OF
Engineering Science and Technology Review

Research Article

www.jestr.org

Order Allocation using Genetic Algorithm

Mariam EL HIRI*, Abdelali EN-NADI and Anas. CHAFI

Industrial Techniques Laboratory, Industrial Engineering Department, Faculty of Science and Technology, FEZ, Morocco

Received 30 August 2019; Accepted 17 March 2020

Abstract

In an increasingly competitive environment, purchasing is considered as a strategic function for all companies, so managers now give the purchasing function a central role in improving the company's performance. The latter is generally assessed in terms of cost, product quality and delivery time. Therefore, performance may vary depending on the percentage of order quantity allocated to each supplier. Indeed, the order allocation is considered one of the complex problems that belongs to the activities of the purchasing function. In this study we propose to ensure an optimal distribution of orders to enable companies to minimize risks both upstream and downstream of the supply chain and to increase their competitiveness. In this paper, we have developed a decision support model based on a genetic algorithm that is used to allocate order quantities according to their requirements in a multi-suppliers environment. The results showed that the model developed can generate efficient decisions compared to the three scenarios that purchasing managers can adopt.

Keywords: order, allocation, genetic algorithm, supplier

1. Introduction

Nowadays, when the strategy of sourcing from a multiple supplier base is most suitable, companies are facing increasingly enormous competitive pressures and challenges, whose purchasing costs have become a determining factor in their ability to compete. In most industries, the cost of raw materials is the main cost of a product, in some cases it can represent up to 80% of the total cost of the product.

After the supplier evaluation phase, the order allocation to selected suppliers is the most important activity in the management of a company's purchasing process. In general, the main objectives of this process are to reduce prices, ensure on-time deliveries and meet quality requirements. The order allocation problem determines how a company should determine the order quantities to be allocated in a multi-supplier environment. Research on supplier evaluation and selection is extensive, but the literature on order allocation is less frequent. Most studies have identified criteria for optimizing the order allocation according to the performance parameters required for each order. The criteria of cost, quality and time delivery are generally the most common [1].

Generally, in a multi-supplier environment, the purchase quantity is distributed among several suppliers, alternately or simultaneously [2]. A multiple sourcing strategy requires the determination of which suppliers will ensure an order and and the quantities allocated to each of these suppliers. The objective for the buyer is to find the best compromise between quality, delivery time and purchase price requirements. The purpose of this article is to study the problem of order allocation in a supply chain in a multi-supplier environment. This problem is defined as an assignment problem (AP). The latter and several of its variations have been widely discussed in the literature. In this article we study a specific class of this problem, in which each order is assigned to a group of collaborating suppliers. In other words, we must determine the optimal quantity of orders from the selected suppliers, taking into account the suppliers' capacity constraints and the requirements of the order. The main objective is to reduce the risks associated with quality and time delivery issues.

To this end, we developed a model for resolving our problem through a genetic algorithm and tested it to examine its effectiveness in terms of control requirements.

2. Review

The assignment problem exists in the literature in different forms, and in different fields. In recent years, many researchers have studied APs and proposed different methods to solve them [3-4]. To solve the problem of teacher assignment, a linear programming model of mixed integers is developed to balance teacher teaching load, while maximizing teacher preferences for courses according to their category [5].[6] proposed an intelligent decision support approach to the allocation problem. The aim is to present a hybrid decision support approach to facilitate the assignment of project reviewers, the resolution method incorporates heuristic knowledge of expert assignment and operational research models. [7] discussed in their article a specific assignment problem, in which each task is assigned to a group of agents. They indicated that the APs discussed in their article cannot be solved using Hungarian algorithms. They chose to solve their problem using the genetic algorithm (GA).

The choice of resolution method depends mainly on the type of problem and its scope of application. In this work, we study the problem of order allocation among several suppliers. The problem to be solved in our case is to assign the quantities of each order in a multi-supplier environment in order to minimize the cost of purchase and respect the level of service required by each order.

*E-mail address: mariam.elhiri@usmba.ac.ma
ISSN: 1791-2377
doi:10.25103/jestr.132.17

The literature does not present much work in this area, little research has been proposed on order allocation strategies. [2] developed a linear programming model for the distribution of order quantities among suppliers. [8] studied the problem of product allocation to suppliers at the operational level and tried to minimize purchasing costs based on service satisfaction history through a method based on simulation. An integrated decision-making model that combines a heuristic algorithm and an optimal dynamic programming algorithm has been developed [9] for optimal order allocation among several capacitated suppliers. A year later a Branch and Bound algorithm was developed for an optimal order allocation among suppliers offering quantity discounts [10]. [11] have pointed out how the order allocation problem is apparently a whole programming problem, solving such a NP-difficult problem usually takes a long time when variable dimensions increase. Other researchers have applied a mixed integer programming approach to solve the problem of sourcing and allocating orders with multiple products and suppliers in a supply chain with price discounts [12]. [13] proposed a fuzzy multi-objective linear programming model to overcome information inaccuracy to support order allocation decisions between suppliers. [14] have studied two different order allocation strategies, the production capacity-based strategy and the production load equilibrium-based strategy using a discrete event simulation. The simulation was also used to evaluate order allocation strategies under different market conditions in the work of [15]. Their objective was to show how an allocation strategy that takes into account the entire supply chain perspective leads to the sustainable development of supplier clusters. [16] address the problem of order allocation for a single product from several suppliers taking into account the risks associated with procurement in addition to trying to minimize transportation costs using a mathematical formulation is solved via LINGO software.

Although the problem of order allocation to suppliers is of great importance for supply chain management, there is less literature on this subject. The phase of orders allocation to suppliers must be carried out judiciously after the selection and evaluation of suppliers, to avoid any disruption related to the customer's requirement. Indeed, after selecting qualified suppliers, managers must examine the performance of each of them. Based on the results of the evaluations, the manager can allocate orders to suppliers [13]. They also indicated that decisions about the allocation of orders among each company's supplier base are so important that they can affect the efficiency of the entire chain.

The problem to be studied is considered as NP - difficult, since if the data of the problem increase the difficulty of the problem also increases, we intend to solve it by genetic algorithms (GA) which have demonstrated in the majority of cases a feasibility and efficiency in solving several problems.

The GA has been widely used to solve optimization problems in many applications. In the problem of order allocation, we must optimally achieve the minimum purchase cost, while respecting the level of service prescribed by each order.

GA is one of the non-traditional research techniques. It differs from traditional optimization techniques in various ways, it is considered a powerful technique to solve optimization problems. It follows the idea of the survival of the fittest. The best solutions evolve from previous generations until a near optimal solution is obtained [17]. Thus, he tries to find the right solution from one population of solutions to another rather than from one individual to another, to guide himself in the solution he adopts as a reference the fitness function information, and not the derivatives. GA use probabilistic transition rules instead of deterministic rules. Generally, by comparing with traditional optimization techniques, it has been proven that GA excels in solving combinatorial optimization problems [7]. As well, GA can simultaneously test many points throughout the solution space, optimize with discrete or continuous parameters, provide several optimal parameters instead of a single solution, and work with many different types of data [18]. Given the advantages already mentioned, we used the GA to solve our order allocation problem.

3. Genetic algorithms

The genetic algorithm is an optimization algorithm based on techniques derived from genetics and natural evolution. The fundamental principles of this algorithm were developed between 1960 and 1970 [19].

According to [20], a GA is made up of a population P of individuals, whose adaptation to their surroundings is measured by means of a fitness linked to the objective function to be optimized. The general principle of a GA is to evolve a population of individuals through evolutionary operators until a stopping condition is met (Fig. 1). Before presenting in detail how a GA works to generate an optimal solution, it is necessary to introduce some genetic terms often used in a GA:

- a gene is a unit of genetic information transmitted by an individual to his or her offspring,
- a chromosome is a structure containing a finite sequence of genes,
- An individual is a potential solution that can be presented by one or more chromosomes,
- A population is a set of individuals,
- A generation is a set of operations performed to move from one population to another.

Generally, these operations are: the selection of individuals from the current population, the application of genetic operators and the evaluation of individuals from the new population.

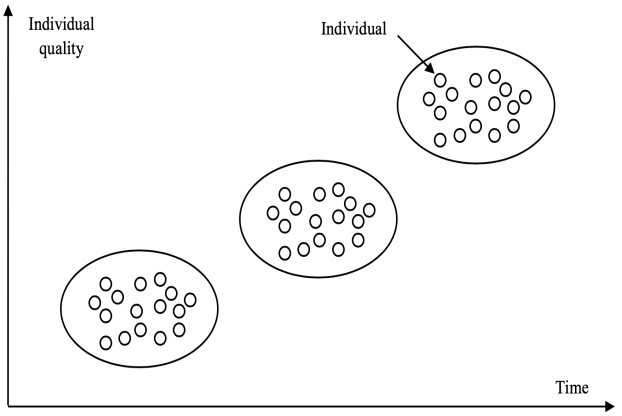

Fig. 1. The general principle of how the genetic algorithm works

The general scheme of a genetic algorithm is illustrated by the fig. 2.

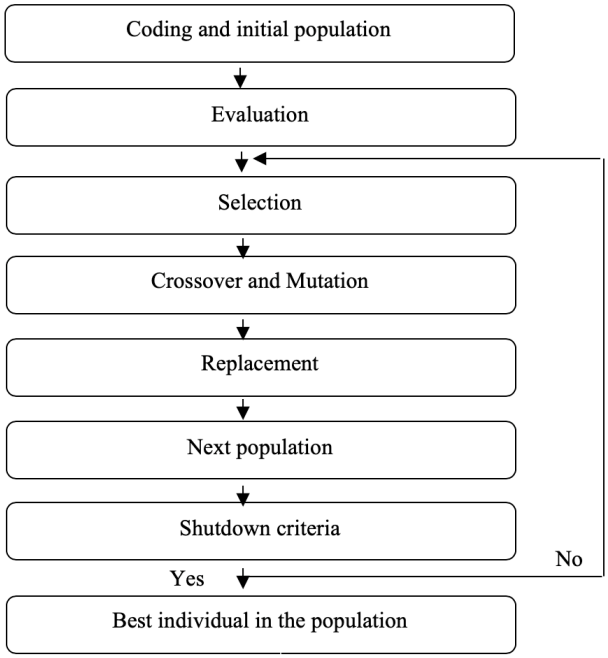

Fig. 2. General diagram of a genetic algorithm [20].

4. Problem description

We consider a manufacturing company which adopts a multi-suplier policy. The company receives orders from various customers in different categories. To ensure an order, the company must purchase products from its suppliers. Each order is characterized by a specific quantity of products to be ordered, a required level of quality, and a deadline not to be exceeded.

To maintain relationships with its suppliers, the company must provide a minimum percentage α% of the order to each supplier able to ensure it. The value of α% is determined by the company in such a way as to obtain supplies from all suppliers able to ensure an order, while respecting the capacity of each supplier.

The objective for this company is to allocate a quantity α% of the order to each supplier able to ensure a specific order. The remainder (R) of the order must be allocated to the same suppliers in order to meet the desired level time delivery.

4.1 Problem formulation

We assume that Cj = {C1, C2,....., Cn} is the set of orders that a company must place with its supplier base which constitutes the set Fi = {F1, F2,.....,Fm}. n and m are respectively the number of orders and the number of the suppliers.

- Each Cj command is characterized by:

Q: The order quantity, Nq: Quality level required, D: deadline not to be exceeded.

- Each Fi supplier is characterized by:

Cui: unit cost for each quantity ordered from supplier Fi, Ai: quantity allocated to each supplier Fi, Capi : capacity of the supplier Fi, di: % of the delivery time respected by the supplier Fi, bi: variable indicating whether the supplier can ensure both types of quality levels (medium and high) or just one quality level (medium or high). If the supplier provides both types of quality levels bi= 0, 1 for suppliers who just provide high quality and -1 for medium quality.

The quantity Ai allocated to each supplier Fi able to ensure a Cj order is given by the following equation:

$$Ai = \alpha\% * Q + \beta i\% * R \quad (1)$$

α% *Q : is the minimum quantity that the company must allocate to each supplier able to ensure an order.
R : the remainder of the order quantity after the allocation of the percentage α% of the order to all suppliers able to ensure an order.

$$R = Q - \sum_{i=1}^{k} \alpha\% * Q \quad (2)$$

K : is the number of suppliers selected to provide a Cj order.
Bi% : a random percentage allocated to suitable suppliers of the remainder of the order quantity after the allocation of the percentage α% of the order to all suppliers able to secure an order.

An illustrative example of the problem to be solved is shown in the figure below

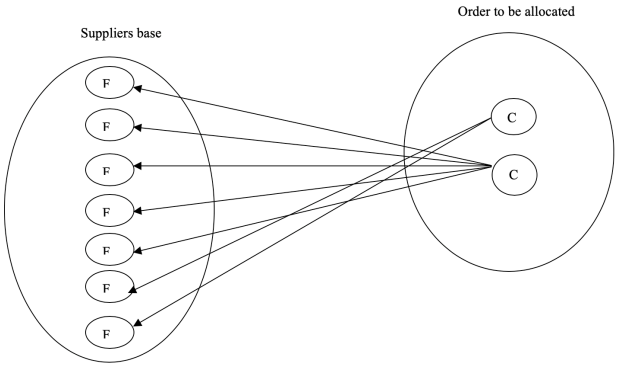

Fig. 3. Example of assigning orders to a company's supplier base

In Figure 3, the C1 order is allocated to the set of suppliers {F1, F5, F6, F7}, while the C2 order is assigned to the set of suppliers {F1, F2, F3, F4, F7}.

4.2 Steps to solve the problem

Once the company has selected its base of N suppliers. We will filter the suppliers able to ensure a Cj order according to the required quality level by evaluating the bi variable associated with each supplier Fi. Then, an allocation of α% of the order is made for all k filtered suppliers. We then calculate the remainder after allocating α% of the order. If this value is zero, the problem is solved. Otherwise, the remainder of the order must be allocated in such a way as to minimize the cost, meet the required deadline and not exceed the suppliers capacities. The resolution steps are shown in Fig. 4.

4.3 Mathematical formulation of the problem

In this section, a mathematical model of the problem is developed to determine the optimal allocation of order quantities to the selected k suppliers.

Minimize

$$\sum_{i=1}^{k} Cui * Ai \quad (1)$$

Subject to :

$$\sum_{i=1}^{k} di * Ai \geq Di \quad (2)$$

$$Ai \leq Capi \quad (5)$$

Such as :

$$Ai = \alpha\% * Q + \beta i\% * R \quad (3)$$

$$R = Q - \sum_{i=1}^{k} \alpha\% * Q \quad (4)$$

Equation (3) represents the objective function that aims to minimize costs.
Equation (4) represents the constraint that meets the required deadline
Equation (5) represents the constraint that respects the capacity of suppliers.

The following section presents our approach based on the genetic algorithm we propose to solve the problem of allocating optimal order quantities.

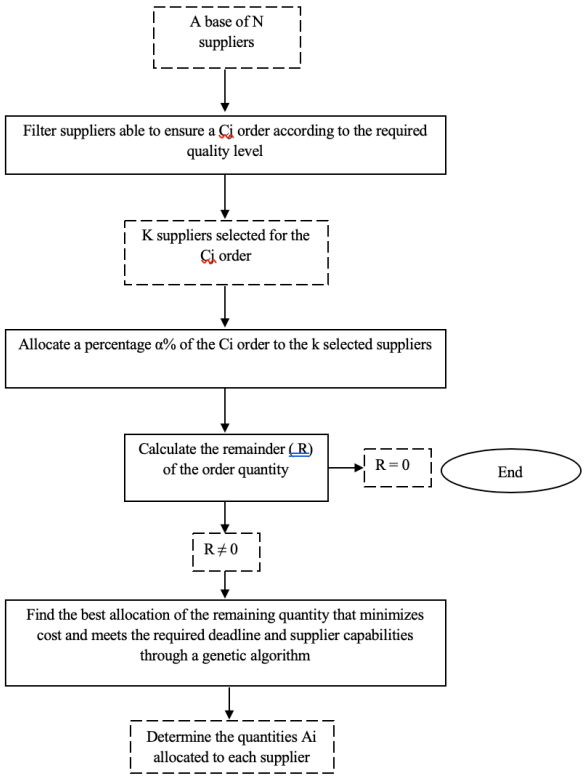

Fig. 4. Steps to resolve the allocation of order quantities

4.4 Resolution model by Genetic Algorithm

In this section we propose an adaptation of the genetic algorithm for the problem of order allocation quantity in a multi-supplier environment. We begin by determining the type of coding used, and then we detail how the initial population is generated. We will then present the evaluation phase. To develop the population of solutions in such a way as to obtain individuals who adapt to the solution of the problem, the operators crossover and mutation are defined. Finally, we explain the replacement step for the generation of individuals participating in the next iteration. The resolution model was developed by the C programming language using DEV C++ software.

To illustrate the different steps of the resolution approach, we have tried to solve the following problem:

For the manufacture of a product X. A company must place an order for a quantity Q= 300000 units, of average quality and a deadline satisfaction rate that must be greater than or equal to 0.87. The company must allocate at least 10% of this order to its supplier base, which is composed of six suppliers. Table 1 shows the characteristics of each supplier.

Table 1 Supplier characteristics

	deadline satisfaction rate	Quality	Capacity	Unit cost
Supplier 1	0,87	0	140000	1,9
Supplier 2	0,91	1	150000	2,3
Supplier 3	0,88	-1	70000	1,9
Supplier 4	0,9	0	100000	2
Supplier 5	0,96	-1	180000	2,3
Supplier 6	0,82	0	170000	1,85

The objective is to allocate the quantity Q = 300000 units to suppliers able to ensure this order in compliance with the required conditions.

First of all, it is necessary to select the suppliers who can meet this order in terms of quality. In our example, with the exception of supplier 2, who does not deliver medium quality products, all suppliers are able to ensure this order.

4.4.1 Coding

Designing a genetic algorithm requires a first step, which consists in adequately representing the data of the solutions to the problem being addressed. The adequate representation of the data will allow us to be consistent with the various characteristics of the GA operators. The representation of a solution must be in the form of a gene chain. These indicate the values of an important parameter in solving the problem being addressed.

To facilitate the development and optimization of our algorithm, we have adopted real coding to represent the quantities of orders distributed to the selected suppliers to ensure a Cj order.

Our algorithm uses a chromosome with a length of k genes where k is the number of suppliers able to provide a Cj command. Genes are represented by an integer that represents the value Ai to be allocated to each supplier. The fig. 5 shows an example of the encoded shape of a chromosome.

30000	30000	30000	30000	30000

Fig. 5. Example of a chromosome used by the GA containing five genes

In this example the size of the chromosome is k= 5 which represents the number of suppliers able to handle the order. The genes contain values of 30000 units. This value is explained by the minimal affection that the company must have with these suppliers. The remainder of the order is R = 150000 units. This quantity must be allocated optimally.

4.4.2 Initial population

This step will allow us to generate a population of individuals that will serve as a basis for future generations. The rapidity

of convergence towards an optimal solution depends mainly on the choice of the initial population. For our problem, the solution consists in making a better allocation of the order quantity while respecting the required conditions. In this perspective, in order to reduce the time needed to find an optimal solution, it would be desirable to introduce good solutions in the initial population. To satisfy this requirement, we propose to build the initial population, a heuristic that generates a set of good solutions to accelerate the convergence of the algorithm. This heuristic gradually builds each solution as follows: first of all, we allocate a quantity of α% of the Cj order to the k selected suppliers. Then, we proceed by calculating the remainder (R) of the order given by the following formula:

$$R = Q - \sum_{i=1}^{k} \alpha\% * Q \quad (2)$$

Then, we will apply an algorithm for the allocation of the remainder of the order. The algorithm is designed as follows:

Start: allocation of the remainder of the order
If R=0 and the deadline is respected
Adopt this allocation
Else If R=0 and the deadline is not respected

Minimize the α%
Else If
/insertion/
Take a random percentage β% of the remainder (R)
Finding the set of positions where inserting β% may be possible
If the set of positions are not empty
Choose a random position
Insert this percentage
Else If all positions are empty
Minimize the β%
End If
As long as R > 2%.
Return to insertion
End As long as
End If
End

We present an example of an individual belonging to the initial population:

30000	30000	70000	140000	30000

Fig. 6. Example of chromosome used by the GA in the initial population

In this example, after the allocation of the 10% order, the genetic algorithm and randomly allocated to gene number 4 which represents supplier number 5 a percentage of 73% of the remainder of the order and 27% to gene number 3 which presents supplier number 4. The values became respectively for the two genes 3 and 4, 70000 and 140000 units. Thus, this allocation meets the requested deadline since the service level value for this allocation is 0.91.

4.3 Evaluation

A relevant evaluation of each solution requires the incorporation of a solution evaluation function called fitness. The latter is presented as a numerical value proportional to the quality of the solution. In our approach, we associate to each solution a fitness function that corresponds to the opposite of the total cost associated with an allocation that presents a solution to the problem. The fitness associated with a solution S is as follows:

$$F(S) = \frac{1}{\text{total cost}} \quad (6)$$

Such as:

$$\text{Total cost} = \sum_{i=1}^{k} Cui * Ai \quad (7)$$

4.4.4 Selection

The selection will allow us to constitute the first step in the generation of the population of descendants. To ensure the selection mechanism of parent chromosomes that will have the opportunity to propagate their genetic characteristics to children of the next generation. We adopt a random selection approach, which consists of random selection to ensure a well-diversified population.

In the following, we make the population evolve, through genetic operators who are the crossover and mutation.

4.4.5 Crossover

Through the crossover operator we will be able to increase the diversity of individuals by generating a new population of them. Through this operator we will try to converge towards a solution that seems better. This is done by producing two children using two parents randomly chosen from the selected population of individuals.

Using a crossover probability pc = 0.7, we adopted a single-point crossing algorithm as follows:

Start: Crossover
Choose two random parents
Choose a random crossover point
Switch the two parts located after the crossover point
Calculate the new quantity Q' after switching
If Q = Q' and the service level service is respected
Adopt this allocation
Else If Q = Q' and the service level service is not respected
Ignore this individual
Else If Q > Q'
/insertion/
Take a random percentage β% of the remainder (R)
Finding the set of positions where inserting β% may be possible
If the set of positions are not empty
Choose a random position
Insert this percentage
Else If the set of positions are empty
Minimize the β%
End If
As long as R > 2%.
Return to insertion
End as long as
/Subtraction
Else If Q < Q'
Take a random percentage β% of the rest (R)
Finding the set of positions or subtracting β% may be feasible
If all positions are not empty
Choose a random position
Subtract this percentage
Else If the set of positions are empty

Minimize the β%
End If
As long as R > 2%.
Back to Subtraction
End as long as
End If
End

An example of the crossover we have adopted is illustrated as follows:

We consider two individuals (parents) from the initial population to whom we will apply the crossover operator to produce new individuals (childrens).

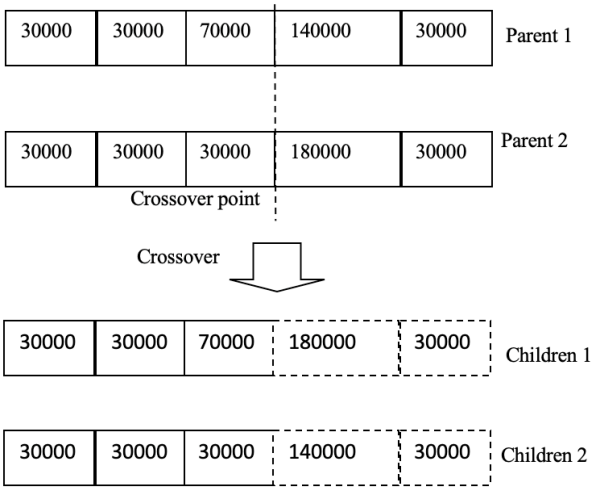

Fig. 7. Example of the crossover adopted in our algorithm

After the crossover of both parents and the birth of the two new individuals, we proceed by calculating the new quantities for the two children. For child 1 the new quantity Q' = 340000 units. This value is greater than Q per 40000 units that must be removed from a randomly selected gene in accordance with the requirements of the problem to be solved. For the second child it is necessary to add a value of 40000 units because Q' in this case is 260000 units.

The correction is made taking into account the capacity of the suppliers and the level of deadline required. If the algorithm fails to make this correction it rejects the new individual (children) and keeps the old individual (parent). The algorithm we adopted gave a correction for children 1 by removing the quantity 40000 units that exceeds the value of Q and for children 2 by adding the quantity 40000 units that is missing to the value of Q. The values of the level of service required for children 1 and 2 are 0.88 and 0.91 respectively. Both values respect the requirement of the order to be satisfied.

30000	30000	~~70000~~ 50000	~~180000~~ 160000	30000

Children 1

Fig. 8. Example of a correction after the crossing phase in the case of suppression

~~30000~~ 70000	30000	30000	140000	30000

Children 2

Fig. 9. Example of a correction after the crossing phase in the case of insertion

4.4.6 Mutation

Using a mutation probability pm= 0.1, this operator will allow us to act on the chromosome in a general way by running through its genes. The operator applies a small mutation to a randomly selected gene of an individual, using the principle of randomly modifying the value assigned to that gene. The mutation will make it possible to avoid states of premature convergence caused by local optimums in the research space by diversifying the individuals in the new population. For this reason, we have gradually added new individuals with new characteristics to the population. In the figure 10, we show an example of the mutation adopted.

In the example below the genetic algorithm randomly took child 2 corrected, and it made a modification of gene number 2. This modification changes the value 30000 to 70000.

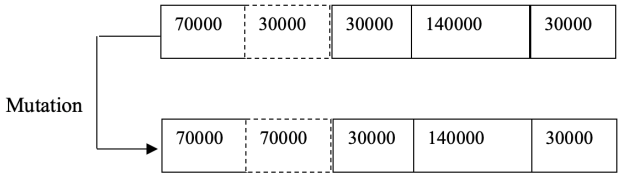

Fig. 10. Example of a mutation

Once the mutation is made we proceed in the same way as in the case of the crossover. Indeed, it is necessary to calculate the new quantity Q' after the mutation. If Q' is different from quantity Q, proceed by inserting or deleting the quantity according to the case study. If Q = Q' the service level condition must be checked. In this example the algorithm to remove 40000 units from gene number 4. For the service level, the individual generated after the mutation respects the deadline level with a value of 0.9.

70000	70000	30000	~~140000~~ 100000	30000

Fig. 11. Example of correction after the transfer

4.4.7 Replacement

In this phase, it is necessary to reintroduce the descendants obtained by the respective application of the selection, crossover and mutation operators into the population of their parents indicated by the initial population. However, the creation of the new population may cause the risk that the best solutions will be lost after crossover and mutation operations. To overcome this problem, we have adapted the replacement procedure chosen by [20] study. This replacement consists in creating an intermediate Pitr population, which consists of the current Pn population and new solutions from genetic operators, then it is necessary to opt for the principle of the elitism method which consists in copying one or more of the best chromosomes in the new generation. 50% of the new population Pn+1 will contain the best Pitr solutions, then the other part of Pn+1 will be supplemented by solutions randomly chosen in Pitr, and which are not yet part of the population Pn+1. This replacement considerably improves the genetic algorithm, as it avoids losing the best solutions.

4.4.8 Shutdown criteria

For the stop criteria that have been chosen in our algorithm. We have tried to increase the number of iterations until we

have a degree of uniformity of individual solutions to our problem. The number of iterations adopted is 200 iterations.

For the example we treated. After a number of tests equal to 10, we had the following results:

Table 2. Result of the problem addressed by the proposed model

	Supplier 1	Supplier 3	Supplier 4	Supplier 5	Supplier 6	Deadline satisfaction rate	Total cost
Test 1	124500	42210	30000	30000	73290	0.87	581335.50
Test 2	132801	30000	30000	30000	77199	0.87	581140.00
Test 3	133500	63015	30000	30000	43485	0.87	582825.75
Test 4	101280	57720	30000	30000	81000	0.87	580950.00
Test 5	99318	59427	30000	30000	81255	0.87	580937.00
Test 6	104292	56910	32298	30000	68250	0.87	581404.81
Test 7	132600	30000	30000	35400	72000	0.87	583560.00
Test 8	135000	30000	36750	30000	68250	0.87	582262.50
Test 9	119280	30000	50202	30000	70518	0.87	583494.31
Test 10	132960	30000	34066	30000	72974	0.87	581757.87

According to the results indicated on the table, we opt for the allocation of test number 5. As it meets the required level of deadline and has the lowest cost.

If we want to solve the problem of order allocation according to the Moroccan company's policy we will opt for 3 scenarios:

- the first scenario: taking into account the capacities of suppliers, opt for less expensive suppliers,
- the second scenario: taking into account the suppliers' capacities, take into account the rate of satisfaction of deadlines, and whether there is still a quantity to be allocated to lower-cost suppliers,
- the third scenario: taking into account the capacities of suppliers, take into account the rate of satisfaction of deadlines.

The results of the three scenarios are shown in Table 3

Table 3. Result of the problem addressed by the three scenarios

	Supplier 1	Supplier 3	Supplier 4	Supplier 5	Supplier 6	deadline satisfaction rate	Total cost
Scenario 1	40000	30000	30000	30000	170000	0,846	576500
Scenario 2	140000	30000	30000	30000	70000	0,86	581500
Scenario 3	140000	70000	30000	30000	30000	0,87	583500

To compare the results of Tables 2 and 3, we illustrated the results on the graph shown in Fig. 12.

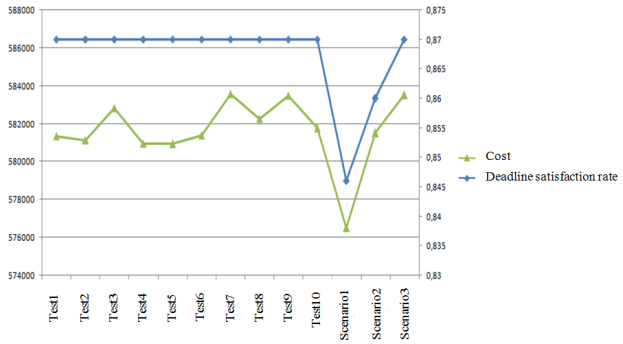

Fig. 12. Graphical representation of the results shown in Tables 2 and 3.

Fig. 12 show that both scenarios 1 and 2 give results of the deadline satisfaction rate below the requirement with values of 0.846 and 0.86. For scenario number 2, in addition to the rate of satisfaction of deadlines which is not respected, the total purchase cost is higher than several results which present solutions in line with our problem. Opting for one of these two scenarios will generate the risk of late delivery beyond a certain date, this can lead to requirements for payment of financial compensation by customers and problems that can go as far as cancellation of orders by customers. Scenario 3 presents a suitable solution to the problem to be solved, but it is not an optimal solution, as it could save in terms of total purchase cost. Indeed, we can choose other alternative solutions provided by the model we have developed.

Aware of the high risks associated with procurement, we proposed an order allocation model based on a genetic algorithm. The proposed model has a particularity that leads to better results for the allocation problem. This particularity lies in generating the right solutions while maintaining the same level of performance required. Thus, our model does not impose the quantity to be allocated, the model is free to try several combinations of allocation until it finds the best one. Our model will also be of significant value to companies that adopt the multi-supplier procurement approach, including companies that require significant numbers to ensure customer satisfaction. Indeed, in such situations where decisions regarding the order allocation become more and more difficult and can generate significant risks, our model can show a very effective solution to overcome this type of problem.

5. Conclusion

As mentioned above, little attention is paid in the literature to decisions on the order allocation quantities to selected suppliers in the case of multiple procurement. The increasing attention paid to partnering with suppliers not only increases the importance of the decision to select suppliers, but also of how orders are allocated to its supplier base.

In this article, we have proposed a model to solve the problem of order allocation quantities in a multi-supplier environment. Ensuring good quality, on-time delivery with minimal cost are the objectives of our model.

For the allocation phase of the order quantity, we used a genetic algorithm to find an allocation that represents an optimal solution.

The strength of the genetic algorithm adopted in this work is that it does not impose a specific quantity to allocate a supplier. Indeed, the algorithm can test several combinations of quantities of different values and then find the optimal solution. This is applied for the crossover and mutation steps.

Solving this problem can give companies an advantage in terms of risk minimization as it will allow them to find solutions that meet the required quality within the required time delivery. In some cases, companies may make decisions for which the penalties for delays will be more costly than sourcing at prices they consider high.

The results of the calculation example indicate that the model can help managers to allocate the quantities ordered among suppliers in an optimal way. In addition, it can be seen that the decision model is systematic and that the allocation of orders to suppliers can be done easily and quickly using the genetic algorithm.

As a perspective we will try to update our algorithm taking into account a dynamic environment that consider changing capacities, quality levels and deadlines over time.

References

1. F. Cheng and F. Ye, A two objective optimisation model for order splitting among parallel suppliers. International Journal of Production Research 49 (10): 2759-2769, (2011).
2. A. Pan, Allocation of order quantity among suppliers. Journal of Purchasing and Materials Management 25 (3): 36–39, (1989).
3. S. Ryu, A. Chen, X. Xu and k. Choi, A dual approach for solving the combined distribution and assignment problem with link capacity constraints. Networks and Spatial Economics 14 (2): 245–270, (2014).
4. M. Mehlawat and S. Kumar, A goal programming approach for a multi-objective multi-choice assignment problem. A Journal of Mathematical Programming and Operations Research 63 (10): 1549–1563, (2014).
5. B. Domenech and A. Lusa, A MILP model for the teacher assignment problem considering teachers preferences. European Journal of Operational Research. 249 (3): 1153-1160, (2016).
6. O. Liu, J. Wang, J. Ma and Y. Sun, An intelligent decision support approach for reviewer assignment in R&D project selection. Computers in Industry 76: 1-10, (2016).
7. I. Younas, F. Kamrani, M. Bashir and J. Schubert, Efficient Genetic Algorithms for Optimal Assignment of Tasks to Teams of Agents. Neurocomputing 314: 409-428, (2018).
8. R. Kawtummachain and NV. Hop, Order allocation in a multiple-supplier environment. International Journal of Production Economics**.** 93–94 (8): 231–238, (2005).
9. X. Qi, Order splitting with multiple capacitated suppliers. European Journal of Operational Research 178 (2): 421–432, (2005).
10. GJ. Burke, SS. Erenguc and A. J. Vakharia, Optimal requirement allocation among quantity-discount quoting suppliers. Operations Management Research 1 (1): 53–60, **(**2008).
11. FC. Yang, KT. Chen, MT. Wang, PY. Chang and KC. Sun, Mathematical modeling of multi- plant order allocation problem and solving by genetic algorithm with matrix representation. The International Journal of Advanced Manufacturing Technology 5 (9–12) : 1251–1259, (2010) .
12. W. C. Tsai and CH. Wang, Decision making of sourcing and order allocation with price discounts. Journal of Manufacturing Systems 27(11): 47–54, (2010).
13. H. Haleh and A. Hamidi, A fuzzy MCDM model for allocating orders to suppliers in a supply chain under uncertainty over a multiperiod time horizon. Expert Systems With Applications. 38 (8): 9076–9083, (2011).
14. W. Xiang, F. Song and F. Ye, Order allocation for multiple supply–demand networks within a cluster. Journal of Intelligent Manufacturing 25 (6): 1367–1376, (2014).
15. P. Renna and G. Perrone, Order allocation in a multiple suppliers-manufacturers environment within a dynamic cluster. The International Journal of Advanced Manufacturing Technology 80: 171-182, (2015).
16. C. Revitasari and I. N. Pujawan, Shifting orders among suppliers considering risk, price and transportation cost. IOP Conference Series: Materials Science and Engineering 337, 012032, (2018).
17. I. Abuiziah and N. Shakarneh, A review of genetic algorithm optimization: Operations and applications to water pipeline systems. International Journal of Mathematical, Computational, Physical and Quantum Engineering 7 (12): 1283-1289, (2013).
18. RL. Haupt and SE Haupt, Practical Genetic Algorithms. Wiley Interscience Publication, (1998).
19. JH. Holland, Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI, (1975) .
20. H. Messaoud, Contribution à la gestion du problème de transport dynamique et statique durable. Thèse de doctorat, (2015).