mathematics

MDPI

Article

An Improved Grey Wolf Optimizer for a Supplier Selection and Order Quantity Allocation Problem

Avelina Alejo-Reyes [1], Erik Cuevas [2], Alma Rodríguez [1,2], Abraham Mendoza [1] and Elias Olivares-Benitez [1,*]

1 Facultad de Ingeniería, Universidad Panamericana, Álvaro del Portillo 49, Zapopan 45010, Mexico; aalejo@up.edu.mx (A.A.-R.); alma.rvazquez@academicos.udg.mx (A.R.); amendoza@up.edu.mx (A.M.)
2 Departamento de Electrónica, Universidad de Guadalajara, CUCEI. Av. Revolución 1500, Guadalajara 44430, Mexico; erik.cuevas@cucei.udg.mx
* Correspondence: eolivaresb@up.edu.mx; Tel.: +52-33-1373-4821

Received: 25 July 2020; Accepted: 23 August 2020; Published: 31 August 2020

Abstract: Supplier selection and order quantity allocation have a strong influence on a company's profitability and the total cost of finished products. From an optimization perspective, the processes of selecting the right suppliers and allocating orders are modeled through a cost function that considers different elements, such as the price of raw materials, ordering costs, and holding costs. Obtaining the optimal solution for these models represents a complex problem due to their discontinuity, non-linearity, and high multi-modality. Under such conditions, it is not possible to use classical optimization methods. On the other hand, metaheuristic schemes have been extensively employed as alternative optimization techniques to solve difficult problems. Among the metaheuristic computation algorithms, the Grey Wolf Optimization (GWO) algorithm corresponds to a relatively new technique based on the hunting behavior of wolves. Even though GWO allows obtaining satisfying results, its limited exploration reduces its performance significantly when it faces high multi-modal and discontinuous cost functions. In this paper, a modified version of the GWO scheme is introduced to solve the complex optimization problems of supplier selection and order quantity allocation. The improved GWO method called iGWO includes weighted factors and a displacement vector to promote the exploration of the search strategy, avoiding the use of unfeasible solutions. In order to evaluate its performance, the proposed algorithm has been tested on a number of instances of a difficult problem found in the literature. The results show that the proposed algorithm not only obtains the optimal cost solutions, but also maintains a better search strategy, finding feasible solutions in all instances.

Keywords: metaheuristic algorithms; grey wolf optimizer; supply chain management; supplier selection; order quantity allocation

1. Introduction

The purchase of raw materials for industrial manufacturing is an important task. Materials must be purchased at the right times and quantities since a shortage (an interruption of the production due to the lack of raw materials) causes large monetary losses. In these activities, one of the main challenges is determining the optimal purchasing parameters, the supplier, or the suppliers to order the raw material from, and how many items must be ordered from each supplier. This also involves the average inventory (and then, the size of the storage facility) and the monthly demand of items. A cost is calculated for each aspect of the purchasing, such as the setup cost, holding cost, and the cost of the items.

The research field related to this problem started with the so-called Economic Order Quantity (EOQ) model, a theory developed by Harris in 1923 [1]. It is the simplest form of order quantity allocation. The main objective is to minimize the total cost, where the mathematical model determines the optimal order quantity of an item [2].

Considering the importance of the EOQ model, in [3] the authors presented a survey describing the main results of the purchasing problem. It shows the extensions of Harris' model that have been developed over the years, such as purchasing models, including multi-stage inventory systems and scheduling or productivity issues. The survey concentrates on the modeling of complex inventory systems such as multiple production stages, parallel machines, or capacity constraints.

One important activity in the purchasing problem is the selection of the supplier or suppliers. Suppliers can offer different characteristics, prices, and quantity discounts (in several types). The interaction among these elements becomes complex. For instance, one supplier can offer a high percentage of non-defective items (which is a desirable feature), but at a higher cost per item. On the other hand, one supplier can offer an attractive purchasing cost, with a low percentage of non-defective items. The mathematical model of those real aspects usually leads to non-linear and high multi-modal cost functions where the optimal global solution is difficult to find. Supplier selection (SS) is the process of evaluating these criteria and selecting the best supplier or suppliers.

Supplier selection and the impact of the influence of purchasing strategies over the supply management activities have been studied in [4]. They developed a supplier performance evaluation tool based on operational and strategic criteria, with the aim of ensuring better purchasing, quality, delivery, flexibility, and innovation. Other authors have also examined the different applications of supplier selection, such as [5,6].

The difficulty of handling supplier selection depends on the criteria and aspects considered by the process [7,8]. As mentioned, the simplest formulation of the purchasing problem consists of considering: a single item, a single supplier, a constant demand, a single time period, and not considering quantity discounts (EOQ model). However, the problem complexity increases when other aspects are considered, such as multi-period [9], different types of discounts (all-unit cost, incremental discount, and total business volume discount), or multi-objective conditions [10].

The single-item complexity can also increase depending on several criteria. In [11], the authors presented four different mathematical programming formulations of the lot-sizing classical problem. It discusses different extensions for real-world applications of this problem. Other works—for example, [12–14]—have analyzed the lot-sizing problem and inventory costs for supplier selection considering larger-size problems.

In the last decade, with the aim to make this problem more realistic, the complexity of purchasing problems has evolved, and numerous models and solutions strategies have emerged. The consideration of multiple items increases the model complexity considerably. For example, in [15] the authors presented a mixed-integer programming model based on a piecewise linear approximation for the solution of multiple items. This work considered a multi-product, multi-constraint inventory system from suppliers and incremental quantity discounts. Another example is the work proposed in [16], where the supplier selection and order quantity allocation problems for multiple products have also been analyzed. In this work, a mixed-integer linear programming model for finding the total cost is presented. In the model, the suppliers also offer quantity discounts (all-units and incremental quantity discounts).

As a result of the purchasing problem complexity, especially for large instances, mathematical models usually have a large number of possible solutions. The number of possible solutions can be even infinite. This fact makes it sometimes impossible to evaluate all feasible solutions, even with a digital computer. In some cases, the number of solutions is not infinite, but so large that evaluating all the solutions may be impractical. Furthermore, these models are characterized by their non-linearity, discontinuity, and high multi-modality.

On the other hand, metaheuristic methods are optimization schemes inspired by our scientific understanding of biological or social systems, which at some abstraction level can be considered as search strategies [17]. Some examples of popular metaheuristic methods include Particle Swarm Optimization (PSO) [18], Genetic Algorithms [19], the Artificial Bee Colony (ABC) algorithm [20], the Differential Evolution (DE) method [21], the Harmony Search (HS) strategy [22], the Gravitational Search Algorithm (GSA) [23], and the Flower Pollination Algorithm (FPA) [24]. Metaheuristic schemes do not need convexity, continuity differentiability, or certain initial conditions, which corresponds to an important advantage considering classical techniques.

Alternatively to linear programming techniques, the problems of purchasing have also been conducted through metaheuristic schemes. In the literature, metaheuristic methods have demonstrated to obtain a better performance than those based on classical techniques in terms of accuracy and robustness. As a result, some approaches have been proposed considering different metaheuristic schemes. Some examples include techniques such as Genetic Algorithm (GA) [25–29] and PSO [30–33]. Although these schemes present interesting results, they have a critical problem—their low premature convergence. This fact generates that such methods frequently obtain sub-optimal solutions, mainly in multi-modal objective functions.

The GWO algorithm [34] is a recent metaheuristic technique based on the hunting behavior of grey wolves. It mimics the leadership, hierarchy, and hunting mechanism of grey wolves. They considered four types of wolves (alpha, beta, delta, and omega) for simulating the leadership hierarchy. Furthermore, they implemented the four main steps of hunting (searching for prey, hunting, encircling prey, and attacking the prey). Its interesting characteristics have motivated its use in several engineering problems, such as sustainable manufacturing [35] and supply chain [36]. In spite of its interesting results, the limited exploration of GWO presents great difficulties in its search strategy when it solves highly multi-modal optimization problems.

In this paper, an improved version of the GWO scheme is introduced to solve the highly multi-modal problem of purchasing. In the enhanced method, two additional elements have been included: (I) weighted factors and (II) a displacement vector. With such inclusions, the new method maintains its important characteristics, increasing its explorative properties so that the algorithm can converge to difficult high multi-modal optima. Different from linear programming techniques, the proposed method can solve supplier selection and purchasing problems under very complex and realistic scenarios, since it does not assume linearity and unimodality in its operation. On the other hand, in comparison to the original GWO and other metaheuristic schemes, our approach is capable of obtaining global optimal solutions due to the improved capacity to explore the search space extensively.

With the purpose of testing our approach, a representative model popular in the literature have been selected. The model [37] considers multiple suppliers with limited capacity. It assumes that suppliers do not have 100% non-defective parts. The model considers a known demand over a finite planning horizon. Additionally, the maximum storage space for the buyer is considered to maximize the total profit. The decision variables are the order quantity for each product, selected suppliers, and purchasing order cycle; the formulation models a problem of supplier selection and lot-sizing inventory. The results show that the proposed algorithm does not just obtain the optimal cost solutions, but also maintains a better search strategy in all instances of the problem, finding feasible solutions in all instances.

The remainder of this article is organized as follows. In Section 2, the problem description and model formulation are presented. Section 3 describes the GWO algorithm. Section 4 describes the proposed modifications to the algorithm. Section 5 presents an illustrative example, along with numerical results and a statistical analysis. Finally, some important conclusions are summarized in Section 6.

2. Problem Description and Model Formulation

This section introduces the problem under study [37]. It consists of solving the supplier selection and order quantity allocation problem incorporating the total income, which considers the income not only of perfect items but also of imperfect items. The model considers several costs, such as the purchasing, ordering, screening, and holding costs. The model under study has been selected for two main reasons: (i) it provides a complex formulation considering several costs in the optimization, constraints, and decision variables; (ii) this model uses several parameters than can be changed in the design of experiments for comparison purposes.

The model characterizes the management of a supply chain where multiple products and multiple suppliers are considered. All the suppliers have a limited capacity. The model implements the scenario of receiving items that may not meet the requirements for the percentage of non-defective parts—a percentage of parts are not of perfect quality. The non-perfect items are sold as a single batch, prior to receiving the next shipment. These items are sold at a lower cost than the non-defective items. The demand is known along the finite planning horizon. The items can be purchased from potential suppliers. A holding cost applies to each item that must be stored. Maximum storage space is considered. With the aim to maximize the total profit, the company needs to determine who are the best suppliers for assigning an order to and how much order quantity must be placed for each product and in which period.

2.1. Assumptions of the Model

1. The ordering cost O_j for each supplier j (if an order is assigned) does not depend on the variety and order quantity of the items involved.
2. The holding cost h_i of the product i represents the cost of maintaining an item in stock.
3. Demand d_{it} represents the amount of the product i that is required in period t, and it is known along the planning horizon.
4. It is possible that suppliers do not offer perfect quality; the purchased items can contain a percentage P_{ij} of defective products; the percentage of perfect products would be $(1-P_{ij})$.
5. The purchased imperfect items are stored apart and sold prior to the next purchasing period as a single batch.
6. The purchasing price (of item i) from supplier j is defined as b_{ij}. The perfect quality items are sold at a price S_{gi} per unit, and the defective items are sold as a single batch at a lower cost S_{di}.
7. The 100% of the screening process of the order is made, which is defined with a unit screening cost v_i of item i.
8. Each supplier has a limited capacity for providing items per period.
9. The requirements of the items must be fulfilled in each period. Shortage or back-ordering is not allowed.
10. Each product requires a storage space w_i, and it considers the total available storage space W.

2.2. Variables and Parameters

Table 1 summarizes the description of the parameters that will be used along with the model.

Table 1. Problem notation of the input parameters.

	Data
n	Total available types of products.
r	The number of available suppliers.
t	The number of available periods.
d_{it}	Demand for the product i in period k (units).
b_{ij}	The purchasing cost of item i from supplier j.
h_i	Inventory holding cost per item i and time.
O_j	The setup cost of the j^{th} supplier.
P_{ij}	Percentage of defective items of product i from supplier j.
S_{gi}	The selling price of non-defective items i per unit.
S_{di}	The selling price of defective items i per unit.
v_i	Screening cost of item i.
c_{ij}	The capacity of supplier j for item i (units per period).
w_i	Storage space for item i.
W	Total available storage space.

2.3. Objective Function

The objective function is composed of two elements which will be described in this subsection. The first element is the total income of the company (R). It is computed through the transactions of good quality items plus the income of selling the imperfect quality items.

$$R = \sum_i \sum_j \sum_t X_{ijt}\left(1 - P_{ij}\right)S_{gi} + \sum_i \sum_j \sum_t X_{ijt}P_{ij}S_{di}, \tag{1}$$

where X_{ijt} symbolizes the ordered quantity (in units) for item i from supplier j in period t.

The processes of generating an order and purchasing the materials have an impact on several costs, such as the purchasing cost, ordering cost, screening cost, and holding cost. The sum of these costs represents the total expenditure of the company (E), which represents the second element. E is calculated as follows:

$$\begin{aligned} E = \sum_i \sum_j \sum_t \quad & X_{ijt}b_{ij} + \sum_j \sum_t O_j Y_{jt} + \sum_i \sum_j \sum_t X_{ijt}v_i \\ & + \sum_i \sum_t h_i\left(\sum_{k=1}^{t} \sum_j X_{ijk}\left(1 - P_{ij}\right) - \sum_{k=1}^{t} d_{ik}\right), \end{aligned} \tag{2}$$

where the first term represents the purchasing cost, which is calculated by the total items of certain types of products ordered at each supplier in any period, multiplied by the price of the item from the supplier. The second term determines the transaction cost for the suppliers, which does not depend on the variety of the ordered items nor on the order quantity. Ordering cost is calculated for each period in which an order is assigned at a supplier. The third term represents the total screening cost, which is calculated as the product of the total ordered items of each type of product and the respective screening cost per type of item. The last term represents the holding cost of maintaining each item that should be stored.

Therefore, the objective function corresponds to the total profit (Z) of the company, represented by the total income minus the total expenses.

$$Z = R - E. \tag{3}$$

As mentioned before, the objective is to find the ordered quantity for the product i from supplier j in period t, so as to maximize the total profit function. The formulation is summarized below:

Maximize:

$$Z = (\sum_i \sum_j \sum_t X_{ijt}(1 - P_{ij})S_{gi} + \sum_i \sum_j \sum_t X_{ijt}P_{ij}S_{di}) - (\sum_i \sum_j \sum_t X_{ijt}b_{ij} + \sum_j \sum_t O_j Y_{jt} + \sum_i \sum_j \sum_t X_{ijt}v_i + \sum_i \sum_t h_i(\sum_{k=1}^{t} \sum_j X_{ijk}(1 - P_{ij}) - \sum_{k=1}^{t} d_{ik})). \tag{4}$$

Subject to,

$$\left(\sum_{k=1}^{t} \sum_{j} X_{ijk}\left(1 - P_{ij}\right) - \sum_{k=1}^{t} d_{ik}\right) \geq 0, \quad \begin{array}{l} \forall i = 1, \ldots, n \\ \forall i = 1, \ldots, n \end{array} \tag{5}$$

$$\left(\sum_{k=1}^{t} d_{ik}\right) Y_{jt} - X_{ijt}\left(1 - P_{ij}\right) \geq 0, \quad \begin{array}{l} \forall i = 1, \ldots, n, \\ \forall j = 1, \ldots, r \\ \forall k = 1, \ldots, t, \end{array} \tag{6}$$

$$\sum_i w_i \left(\sum_{k=1}^{t} \sum_j X_{ijk}\left(1 - P_{ij}\right) - \sum_{k=1}^{t} d_{ik}\right) \leq W, \tag{7}$$

$$0 \leq X_{ijt} \leq c_{ij}, \quad \begin{array}{l} \forall i = 1, \ldots, n, \\ \forall j = 1, \ldots, r, \\ \forall k = 1, \ldots, t. \end{array} \tag{8}$$

The first constraint, represented by Equation (5), ensures that the demand for each type of item in each period is covered with the purchased items. The second constraint in Equation (6) ensures that all orders are accompanied by a transaction cost; if an order is assigned to supplier j in period t, then Y_{jt} is equal to *1*; otherwise, it is equal to *0*. The third constraint, Equation (7), determines that the total storage space is limited by W. Finally, the constraint represented by Equation (8) ensures that the order quantity per supplier does not exceed their capacity per period c_{ij}.

Deterministic methods usually find a global solution when the complexity of the problem is low. The complexity of this model can be determined by the number of constraints, as follows:

$$(n \cdot t) + (n \cdot r \cdot t) + 1 + 2(n \cdot r \cdot t), \tag{9}$$

where n is the total number of different products, r determines the number of available suppliers, and t represents the number of periods. When the size of the problem is large, it is extremely difficult to obtain a global solution in a reasonable time, and other strategies such as metaheuristics can be used to solve this type of problem. Table 2 shows how the number of constraints grows considerably when the type of items, the available suppliers, and the number of periods increase.

The size of the problem (dimension) is also determined by the number of decision variables. In this problem, the total number of decision variables is equal to:

$$(r \cdot t) + (n \cdot r \cdot t). \tag{10}$$

If we consider the use of metaheuristic algorithms, this number of variables can be reduced. Therefore, the model is simplified because there is a dependence between the variable Y_{it} (if an order was assigned at supplier j in the period t) and X_{ijt}. If $X_{ijt} > 0$, then $Y_{it} = 1$; otherwise, $Y_{it} = 0$. The total number of variables using this simplification is as follows:

$$(n \cdot r \cdot t). \tag{11}$$

Obtaining a global solution by commercial software, based on classical techniques, can take too long. For this reason, it is necessary to explore other strategies such as metaheuristics for solving this type of problem. Some metaheuristic methods, such as PSO and GA, have been used to obtain a good solution in a lower computational time [38]. However, a disadvantage of these methods is that they present a premature convergence, producing frequently suboptimal solutions.

Table 2. Number of total constraints to the problem.

	$_i\backslash^j$	5	10	15	20	25	50	100	150	200
	5	321	621	921	1221	1521	3021	6021	9021	12,021
	10	641	1241	1841	2441	3041	6041	12,041	18,041	24,041
	15	961	1861	2761	3661	4561	9061	18,061	27,061	36,061
	20	1281	2481	3681	4881	6081	12,081	24,081	36,081	48,081
$t = 4$	25	1601	3101	4601	6101	7601	15,101	30,101	45,101	60,101
	50	3201	6201	9201	12,201	15,201	30,201	60,201	90,201	120,201
	100	6401	12,401	18,401	24,401	30,401	60,401	120,401	180,401	240,401
	150	9601	18,601	27,601	36,601	45,601	90,601	180,601	270,601	360,601
	200	12,801	24,801	36,801	48,801	60,801	120,801	240,801	360,801	480,801
	$_i\backslash^j$	5	10	15	20	25	50	100	150	200
	5	641	1241	1841	2441	3041	6041	12,041	18,041	24,041
	10	1281	2481	3681	4881	6081	12,081	24,081	36,081	48,081
	15	1921	3721	5521	7321	9121	18,121	36,121	54,121	72,121
	20	2561	4961	7361	9761	12,161	24,161	48,161	72,161	96,161
$t = 8$	25	3201	6201	9201	12,201	15,201	30,201	60,201	90,201	120,201
	50	6401	12,401	18,401	24,401	30,401	60,401	120,401	180,401	240,401
	100	12,801	24,801	36,801	48,801	60,801	120,801	240,801	360,801	480,801
	150	19,201	37,201	55,201	73,201	91,201	181,201	361,201	541,201	721,201
	200	25,601	49,601	73,601	97,601	121,601	241,601	481,601	721,601	961,601

3. Original Grey Wolf Optimizer

The Grey Wolf Optimizer (GWO) [34] algorithm is a new metaheuristic method inspired by the hunting behavior of the grey wolf in nature. Generally, they live in groups of 5–12 grey wolves and form a pack. The algorithm is based on the social hierarchy behavior of the wolves and their mechanism of obtaining prey (hunting). The wolf pack has several hierarchical levels: the alpha wolf (α) is responsible for making decisions about sleeping or hunting. They lead the herd, and the members follow the decisions of alpha wolves. The beta wolf (β) helps the alpha wolf, coordinating and collaborating with the management of the herd. They are subordinate to the alpha wolves. They represent the second level within a hierarchy. The other hierarchical level is fulfilled by delta wolves (δ). They complement the alpha and beta wolves in managing the herd. The omega wolves (Ω) are the lowest level of the hierarchy. They must obey the alpha, beta, and delta wolves.

GWO algorithm emulates the position of the prey as the optimal solution to an optimization problem. Then, using operators based on the wolves hunting process, the algorithm tries to obtain the position of the prey. The algorithm considers four stages in their structure:

- Encircling prey,
- Hunting,
- Attacking prey,
- Searching for prey.

3.1. Encircling Prey

The grey wolves begin the hunting process by encircling (surrounding) the prey. This action is determined using the following formulations (12), (13) to update the position of the wolves in the encircling action:

$$\vec{D} = |\vec{C}\vec{X}_p(t) - \vec{X}(t)|, \tag{12}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}\vec{D}, \tag{13}$$

where $\vec{X}_p$ is the position of the prey, $\vec{X}$ indicates the position of the wolves, t represents the current iteration, and $\vec{C}$ and $\vec{A}$ are the coefficients. The coefficient $\vec{A}$ determines the search radius of the hunting. The $\vec{C}$ and $\vec{A}$ coefficients are calculated as follows:

$$\vec{A} = 2\vec{a}\,\vec{r}_1 - \vec{a}, \tag{14}$$

$$\vec{C} = 2\vec{r}_2, \tag{15}$$

where $\vec{a}$ is linearly decreased from 2 to 0 along the course of iterations, and $\vec{r}_1$ and $\vec{r}_2$ are random values in the range [0, 1].

3.2. Hunting

In the real process of hunting, the alpha wolf determines the position of the prey, and the beta and delta wolves follow the alpha wolf and participate in the hunting. The positions of alpha (best candidate solution), beta, and delta have a better understanding of the potential location of prey. The method saves the first three best solutions obtained so far and forces the other search agents (including omegas) to update their positions according to the position of the best search agents.

$$\vec{D}_\propto = |\vec{C}_1\vec{X}_\propto - \vec{X}|, \quad \vec{D}_\beta = |\vec{C}_2\vec{X}_\beta - \vec{X}|, \quad \vec{D}_\delta = |\vec{C}_3\vec{X}_\delta - \vec{X}|, \tag{16}$$

$$\vec{X}_1 = \vec{X}_\propto - \vec{A}_1\vec{D}_\propto, \quad \vec{X}_2 = \vec{X}_\beta - \vec{A}_2\vec{D}_\beta, \quad \vec{X}_3 = \vec{X}_\delta - \vec{A}_3\vec{D}_\delta, \tag{17}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \tag{18}$$

3.3. Attacking Prey

Wolves capture the prey when it stops moving. This action is modeled decreasing the value of $\vec{a}$ over the course of iterations from *2* to *0*, then $\vec{A}$ is also decreased. $\vec{A}$ is a random value in [−2*a*, 2*a*]. If random values $\vec{A}$ are in [−1, 1], the next position of a search agent may be in any position between the position of the prey and its position, when $|A| < 1$, then the grey wolves are forced to attack the prey. With the use of these operators, the algorithm allows its search agents to update their position based on the position of the alpha, beta, and delta. Only using these operators, the algorithm is susceptible to stay in local solutions; for this reason, more operators are needed.

3.4. Search for Prey

The search is done according to the position of the wolves (alpha, beta, delta). The wolves diverge from each other with the purpose of searching for prey and converge to attack it. The divergence is reached using random values $\vec{A} > 1$ or $\vec{A} < 1$ to force the search agent to diverge from the prey. This process helps in exploration and allows finding a global solution.

4. Improved Grey Wolf Optimizer

The problem of supplier selection is discrete and can become extremely complex when the number of suppliers and items increases. These conditions and their numerous constraints produce objective functions with a high multi-modality. In spite of its interesting results, the limited exploration of GWO presents great difficulties in the search strategy when it solves highly multi-modal optimization problems. Likewise, the GWO has been designed to operate in continuous spaces. For this reason, it experiences inconsistencies when it is used in problems of a discrete nature. Under such conditions, an improved version of GWO is necessary in order to overcome this issue. In this work, an improved version of the GWO method, called iGWO, has been introduced to solve the problem under study.

The enhanced version incorporates two new elements: (1) weighted factors and (2) a displacement vector. With such inclusions, the new method increases and improves the explorative properties so that the algorithm can converge to difficult high multi-modal optima.

4.1. Weighted Factors

In the original GWO, particles are updated by considering the average combination of the alpha, beta, and delta wolves (Equation (18)). This mechanism guides individuals in the same proportion towards the best elements. However, it has been proved that this is not the best strategy [39], since that mechanism produces a limited exploration of the search space. Therefore, in the improved version of GWO, particles are updated using the following formulation:

$$\vec{X}(t+1) = w_1\vec{X}_1 + w_2\vec{X}_2 + w_3\vec{X}_3 + \vec{r}_3\vec{b}, \tag{19}$$

where w_1, w_2, and w_3 are the weighted factors that determine the contribution of each alpha, beta, and delta wolf. These weights are used to guide the search process towards the best elements but considering different proportions according to the hierarchy of grey wolves.

4.2. Displacement Vector

In the new iGWO, a displacement vector $\vec{r}_3\vec{b}_3$ (see Equation (19)) has been included in order to increase the exploration and prevent the consideration of unfeasible solutions. Here, $\vec{r}_3$ is a random value in the interval [−1, 1] that controls the direction of the search. The element $\vec{b}$ is included to promote exploration and prevent stagnation in local optima. This element is considered a tuning parameter that must be set with an initial value. To ensure convergence, $\vec{b}$ is non-linearly decreased throughout iterations. The definition of $\vec{b}$ is given by the following formulation:

$$\vec{b}(t+1) = \vec{b}(t)\left(1 - \frac{t^2}{t_{max}^2}\right), \tag{20}$$

where $t_{\max}$ is the maximum number of iterations.

Under this update mechanism, occasionally random steps are permitted to jump into a feasible area in case the global best is stuck in an unfeasible solution. In the beginning, larger steps are allowed. However, the displacement vector is non-linearly decreased over time to balance the exploration-exploitation rate. Besides, since the supplier selection problem requires an integer solution, the updated positions given by Equation (19) are rounded to the nearest integer toward negative infinity.

5. Experimental Results

A representative formulation introduced in [37] has been considered as an illustrative problem to test the performance of the proposed method. It has been selected in order to maintain compatibility with other studies reported in the literature. The problem consists of three different products, three suppliers, and four-time periods. Assuming Equation (10) as a basis, we have 48 decision variables. They can be reduced to 36 decision variables (Equation (11)). The parameters for this problem are described in Tables 3–7.

Table 3. Demand for the three items over the planning horizon.

Items	Periods			
	1	**2**	**3**	**4**
1	170	155	160	140
2	85	90	80	105
3	280	255	290	300

Table 4. Purchasing price of items from the supplier.

Items	Supplier		
	1	**2**	**3**
1	25	27	24
2	30	32	33
3	54	50	49

Table 5. Percentage of defective items for each supplier.

Items	Supplier		
	1	**2**	**3**
1	0.03	0.02	0.03
2	0.02	0.03	0.05
3	0.04	0.04	0.01

Table 6. Ordering cost per supplier.

Supplier		
1	**2**	**3**
3000	2700	3500

Table 7. S_{gi}, S_{di}, w_i, h_i, and v_i costs for each product.

Items	S_{gi}	S_{di}	w_i	h_i	v_i
1	50	20	0.2	5	2
2	34	25	0.18	3.5	1.5
3	60	40	0.5	8	1.8

The capacity c_{ij} of product i from supplier j per period is 1000 units for all suppliers. The total available space W is limited to 200.

The popular software LINGO and the proposed Improved Grey Wolf Optimizer (iGWO) have been used for solving the model. The experiments have been implemented using MATLAB R2019a, in a computer with an intel(R) Core (TM)i7-8550u cpu@1.80 GHz 1.99 GHz processor.

The results are shown in Tables 8 and 9. Observe that iGWO presents a higher profit than the classical optimization tools. The algorithm obtains a result that is 60% better than the result obtained by LINGO.

Table 8. Order quantity for each product from each supplier and per period, X_{ijt}, using commercial software.

Period 1				Period 2			
Item/Supplier	**1**	**2**	**3**	**Item/Supplier**	**1**	**2**	**3**
1	175.2577	0	175.2577	1	0	158.1633	0
2	86.8	0	0	2	0	92.78351	0
3	0	0	282.8283	3	0	265.625	0
Period 3				**Period 4**			
Item/Supplier	**1**	**2**	**3**	**Item/Supplier**	**1**	**2**	**3**
1	164.9485	0	164.9485	1	0	142.89	0
2	81.7	0	0	2	0	108.25	0
3	0	0	292.9293	3	0	312.5	0
Objective function value				$11,364.93			

Table 9. Order quantity for each product for each supplier and per period, X_{ijt}, using the iGWO algorithm.

	Period 1				Period 2		
Item/Supplier	**1**	**2**	**3**	**Item/Supplier**	**1**	**2**	**3**
1	0	302	0	1	380	0	378
2	93	0	0	2	92	0	0
3	0	0	283	3	0	0	259
	Period 3				**Period 4**		
Item/Supplier	**1**	**2**	**3**	**Item/Supplier**	**1**	**2**	**3**
1	0	0	218	1	363	0	0
2	0	0	85	2	108	0	0
3	0	0	293	3	313	0	0
Objective function value				$18,433.30			

5.1. Weighted Factors

An experiment was performed with the purpose of analyzing the accuracy and consistency of the proposed algorithm (iGWO). In the experiment, several parameters of the model were changed to confirm the robustness of the algorithm. These parameters are the demand d_{it}, the total available space W, and the capacity of the supplier for each item c_{ij}. For each parameter, three levels were analyzed. The demand (d_{it}) of the problem instance presented in Table 3 was changed at 75% and 125% of the actual demand. Case 1 (for demand) corresponds to the original demand presented in Table 3; case 2 and case 3 correspond to the new demand considering 75% and 125%, respectively, of the original demand. The total available space (W) was considered for case 1, case 2, and case 3 at 200, 400, and 600, respectively. The capacity of suppliers (c_{ij}) was changed. Case 1 considers the original demand at 1000 units per item and per period; for case 2 and case 3, the demand is presented in Table 10.

Table 10. Supplier capacity for statistical analysis.

	Case 2				Case 3		
Item/Supplier	**1**	**2**	**3**	**Item/Supplier**	**1**	**2**	**3**
1	600	600	600	1	450	450	450
2	580	580	580	2	435	435	435
3	620	500	480	3	465	375	360

When modifying the parameters, 27 different scenarios were generated. All the scenarios have been solved considering the proposed iGWO method. The results have been compared with those produced by other methods such as LINGO, original Grey Wolf Optimizer (GWO) [34], Modified Grey Wolf Optimizer (mGWO) [39], Proportional-based Grey Wolf Optimizer (PGWO) [40], Tournament-based Grey Wolf Optimizer (TGWO) [40], Particle Swarm Optimization (PSO) [30], Differential Evolution (DE) [21], and Success-History based Adaptive DE with Linear population size reduction (L-SHADE) [41]. In the comparisons, the parameters of these methods have been configured according to the reported values provided by their own references. All these settings are summarized in Table 11.

Table 11. Parameter configurations of metaheuristic algorithms.

Settings Configuration	
iGWO	$b = 50$, a linearly decreased from 2 to 0, $w_1 = 0.4$, $w_2 = 0.2$, $w_3 = 0.4$
PSO	$c_1 = 2$, $c_2 = 2$
DE	$CO = 0.5$, $\boldsymbol{F} = 0.2$
L-SHADE	$r^{N^{init}} = 18$, $r^{arc} = 1.4$, $p = 0.11$, $H = 5$
GWO	a linearly decrease from 2 to 0
mGWO	a linearly decrease from 2 to 0
PGWO	a linearly decrease from 2 to 0
TGWO	a non-linearly decrease from 2 to 0

The 27 scenarios are identified as follows: instance (1,2,3) indicates that it considers case 1 of demand, case 2 of total available space, and case 3 for supplier capacity. The original instance is defined as (1,1,1). Since metaheuristic algorithms are stochastic methods, the optimization process is repeated in 10 independent executions for every metaheuristic algorithm (with 1000 iterations) to verify the consistency of the results. The population for the algorithms was 100 individuals, and the size dimension is 36. For each algorithm, 10 results are obtained, which represent the best-found solutions. With this information, the performance of the algorithms are statistically compared considering the following indicators: the average profit Z_a, the median of the results Z_m, the best profit Z_b, the worst profit Z_w, and the standard deviation S. Indicators Z_b, Z_w, Z_a, and Z_m evaluate the accuracy of the algorithms, and S evaluates the consistency of the solutions and, therefore, the robustness of the metaheuristic algorithms. First, the performance of the algorithms in the instances where only one parameter is changed is analyzed. These instances are: (1,1,1), (2,1,1), (3,1,1), (1,2,1), (1,3,1), (1,1,2), (1,1,3). Table 12 presents the statistical indicators of these instances for the 10 executions per method.

From all instances in Table 12, only the iGWO algorithm found a feasible solution in all the 10 executions of the seven instances. In the instances (1,1,1) and (2,1,1), the best result was presented by iGWO at $18,433.30 and $18,008.18, respectively. GWO and mGWO found only one solution. PGWO, TGWO, PSO, DE, and L-SHADE did not find a feasible solution. For the instance (3,1,1), the best result was presented by DE with $24,041.09; therefore, the algorithm only managed to find three solutions out of 10 feasible solutions. The profit of iGWO is only 7% lower than the best solution; also, the average profit and median of the profit of iGWO are better than those of DE. GWO and mGWO found seven and nine solutions out of 10, respectively; PGWO, TGWO, PSO, and L-SHADE did not find a feasible solution. For the instance (1,2,1), the best result was presented by iGWO with $33,842.24. iGWO, mGWO, and DE found a feasible solution for each execution. PSO found two feasible solutions out of 10. PGWO, TGWO, and L-SHADE did not find a feasible solution. For instance (1,3,1), the best result was presented by mGWO with $44,099.66; therefore, the average profit and median of the profit of iGWO is better than all algorithms. GWO, mGWO, and DE found a feasible solution for each execution. PGWO, TGWO, PSO, and L-SHADE found two, one, eight, and seven solutions out of 10, respectively.

For the instance (1,1,2), the best result was presented by iGWO, at $22,432.70. GWO and mGWO found three and two solutions out of 10, respectively. PGWO, TGWO, PSO, DE, and L-SHADE did not find a feasible solution. For the instance (1,1,3), the best result was presented by iGWO, at $22,432.70. GWO, mGWO, and PGWO found one, two, and one solution out of 10, respectively. TGWO, PSO, DE, and L-SHADE did not find a feasible solution. Figure 1 shows that the profit of the found the best solution by LINGO and the iGWO, GWO, mGWO algorithms. These metaheuristic algorithms were selected because they managed to find more feasible solutions than the others.

Table 12. Statistical indicator for seven instances.

Indicator	LINGO	iGWO	GWO	mGWO	PGWO	TGWO	PSO	DE	L-SHADE
Instance: (1,1,1)									
Z_a	$11,364.93	$15,734.66	-	-	*	*	*	*	*
Z_m		$16,626.29	-	-	*	*	*	*	*
S		$3476.07	-	-	*	*	*	*	*
Z_b		$18,433.30	$6238.92	$17,498.60	*	*	*	*	*
Z_w		$6481.26	-	-	*	*	*	*	*
% Feasible Solutions		100%	10%	10%	0%	0%	0%	0%	0%
Instance: (2,1,1)									
Z_a	$5525.58	$14,298.80	-	-	*	*	*	*	*
Z_m		$14,915.53	-	-	*	*	*	*	*
S		$3240.45	-	-	*	*	*	*	*
Z_b		$18,008.19	$3134.26	$3533.75	*	*	*	*	*
Z_w		$7770.92	-	-	*	*	*	*	*
% Feasible Solutions		100%	10%	10%	0%	0%	0%	0%	0%
Instance: (3,1,1)									
Z_a	$21,008.76	$18,602.24	$15,131.57	$16,589.93	*	*	*	$18,306.22	*
Z_m		$19,175.79	$12,937.30	$15,235.60	*	*	*	$19,330.20	*
S		$2546.07	$4232.78	$4114.77	*	*	*	$6309.49	*
Z_b		$22,262.90	$22,658.36	$22,653.24	*	*	*	$24,041.09	*
Z_w		$14,350.89	$9214.35	$11,099.60	*	*	*	$11,547.37	*
% Feasible Solutions		100%	70%	90%	0%	0%	0%	30%	0%

Table 12. *Cont.*

Instance: (1,2,1)									
Indicator	**Algorithm**								
	LINGO	**iGWO**	**GWO**	**mGWO**	**PGWO**	**TGWO**	**PSO**	**DE**	**L-SHADE**
Z_a		$29,966.42	$24,292.48	$25,704.28	*	*	$23,863.71	$23,681.45	*
Z_m		$30,387.98	$23,346.78	$25,093.52	*	*	-	$22,952.12	*
S	$11,364.93	$3227.18	$2559.25	$2426.86	*	*	$1133.05	$2405.00	*
Z_b		$33,842.24	$28,270.01	$29,811.53	*	*	$24,664.90	$28,276.88	*
Z_w		$23,216.43	$21,459.58	$22,912.52	*	*	$23,062.53	$21,058.27	*
% Feasible Solutions		100%	100%	100%	0%	0%	20%	100%	0%
Instance: (1,3,1)									
Indicator	**Algorithm**								
	LINGO	**iGWO**	**GWO**	**mGWO**	**PGWO**	**TGWO**	**PSO**	**DE**	**L-SHADE**
Z_a		$41,602.21	$36,379.10	$37,158.70	$30,443.51	*	$36,224.65	$35,347.98	$6410.08
Z_m		$42,000.14	$37,452.46	$36,717.96	-	*	$34,925.10	$35,512.86	$3571.7
S	$11,364.93	$1276.68	$3893.49	$4285.87	$1011.85	*	$2829.55	$914.68	$4105.24
Z_b		$43,068.69	$41,229.04	$44,099.66	$31,159.00	$25,268.91	$40,624.35	$36,771.34	$14,676.22
Z_w		$39,393.00	$29,735.00	$32,412.86	$29,728.03	*	$32,094.18	$33,807.33	$2611.68
% Feasible Solutions		100%	100%	100%	20%	10%	80%	100%	70%
Instance: (1,1,2)									
Indicator	**Algorithm**								
	LINGO	**iGWO**	**GWO**	**mGWO**	**PGWO**	**TGWO**	**PSO**	**DE**	**L-SHADE**
Z_a		$16,213.74	$15,676.17	$4969.52	*	*	*	*	*
Z_m		$15,214.86	$16,560.68	-	*	*	*	*	*
S	$11,364.93	$3628.17	$2457.38	$1932.81	*	*	*	*	*
Z_b		$22,432.70	$17,568.85	$6336.22	*	*	*	*	*
Z_w		$10,458.29	$12,898.98	$3602.81	*	*	*	*	*
% Feasible Solutions		100%	30%	20%	0%	0%	0%	0%	0%

Table 12. *Cont.*

Instance: (1,1,3)									
Indicator	**Algorithm**								
	LINGO	**iGWO**	**GWO**	**mGWO**	**PGWO**	**TGWO**	**PSO**	**DE**	**L-SHADE**
Z_a	$11,364.93	$17,104.16	-	$6257.03	*	*	*	*	*
Z_m		$17,778.35	-	-	*	*	*	*	*
S		$3606.26	-	$7251.80	*	*	*	*	*
Z_b		$22,318.83	$11,053.27	$11,384.83	$17,436.81	*	*	*	*
Z_w		$11,639.69	-	$1129.24	*	*	*	*	*
% Feasible Solutions		100%	10%	20%	10%	10%	80%	100%	70%

* No Solution was Found; - It is not Possible to Calculate the Indicator.

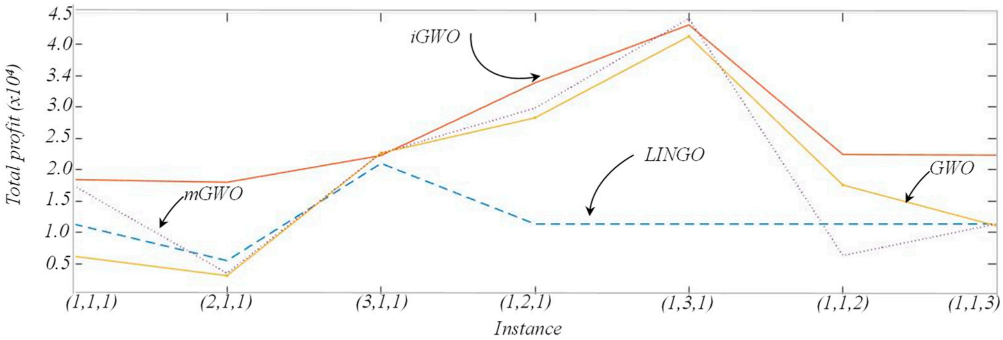


Figure 1. Best solutions found by LINGO, iGWO, GWO, and mGWO for some instances.

Table 13 summarizes the results of the best solution with profit Z_b for the seven instances presented previously.

Table 13. Statistical indicator for the seven instances.

	(1,1,1)	(2,1,1)	(3,1,1)	(1,2,1)	(1,3,1)	(1,1,2)	(1,1,3)
X_{111}	0	0	0	379	379	0	0
X_{121}	302	375	0	0	351	0	0
X_{131}	0	0	380	380	380	379	380
X_{211}	93	0	0	0	85	0	0
X_{221}	0	66	0	0	2	0	0
X_{231}	0	0	151	90	0	94	117
X_{311}	0	0	0	0	0	0	0
X_{321}	0	220	0	0	0	0	0
X_{331}	283	0	363	288	283	283	290
X_{112}	380	0	350	377	378	0	0
X_{122}	0	372	0	377	0	379	0
X_{132}	378	0	365	0	380	0	380
X_{212}	92	0	138	92	92	0	0
X_{222}	0	70	0	0	0	93	0
X_{232}	0	0	0	0	0	0	95
X_{312}	0	0	0	0	212	0	0
X_{322}	0	201	0	266	0	266	0
X_{332}	259	0	324	0	53	0	266
X_{113}	0	368	0	375	319	380	380
X_{123}	0	0	274	0	373	0	0
X_{133}	218	0	0	0	0	106	0
X_{213}	0	62	0	0	0	82	0
X_{223}	0	0	132	0	97	0	0
X_{233}	85	0	0	85	0	0	85
X_{313}	0	227	0	0	153	0	70
X_{323}	0	0	380	0	151	0	0
X_{333}	293	0	0	296	0	297	226
X_{114}	363	369	0	380	380	0	0
X_{124}	0	0	4	0	372	0	380
X_{134}	0	0	190	378	380	379	0
X_{214}	108	0	0	109	83	0	0
X_{224}	0	82	139	0	25	109	116
X_{234}	0	0	138	0	0	0	0
X_{314}	313	77	0	0	0	0	0
X_{324}	0	159	0	0	0	0	319
X_{334}	0	0	380	306	304	306	0
Total Profit	$18,433.31	$18,008.19	$24,041.09	$33,842.24	$44,099.66	$22,432.70	$22,318.83
Algorithm	*iGWO*	*iGWO*	*DE*	*iGWO*	*mGWO*	*iGWO*	*iGWO*
Purchasing Cost	$110,445.00	$92,846.00	$132,328.00	$134,861.00	$163,740.00	$109,209.00	$109,561.00
Ordering Cost	$22,200.00	14,100.00	$18,900.00	$25,200.00	$30,600.00	$18,900.00	$16,200.00
Screening Cost	$5915.40	$4979.20	$6777.60	$7936.80	$10,040.80	$5886.60	$5767.30
Holding Cost	$4893.61	$4937.89	$4586.55	$9829.94	$14,920.46	$4845.10	$4568.31

Observe in Table 13 the values for the decision variable X_{ijt}; the total profit for each solution; and the behavior of the purchasing, ordering, screening, and holding cost.

As a second analysis, the best profit found for each instance (27 instances) is presented. See Table 14, and observe that iGWO achieved 21 best results out of the 27 instances (77%). There are three instances ((3,3,3), (3,2,3), (3,1,3)) in which only the LINGO and iGWO algorithms found a result, therefore the best results for these instances were generated by iGWO.

Table 14. Best results for each instance.

Instance	Total Profit	Algorithm	Instance	Total Profit	Algorithm
(2,2,1)	$27,900.68	*iGWO*	(3,3,1)	$49,834.26	*mGWO*
(2,1,2)	$21,167.70	*iGWO*	(3,2,2)	$40,994.51	*PSO*
(1,3,2)	$43,488.09	*iGWO*	(1,2,2)	$35,839.30	*iGWO*
(1,3,1)	$44,099.66	*mGWO*	(1,2,3)	$33,407.78	*iGWO*
(1,1,3)	$22,318.83	*iGWO*	(3,1,2)	$23,155.55	*mGWO*
(2,3,3)	$39,982.45	*iGWO*	(1,3,3)	$43,941.35	*iGWO*
(3,1,1)	$24,041.09	*DE*	(3,1,3)	$22,811.44	*iGWO*
(3,3,3)	$49,952.29	*iGWO*	(2,1,3)	$18,698.81	*iGWO*
(3,3,2)	$38,302.08	*iGWO*	(2,2,3)	$28,370.03	*iGWO*
(2,3,1)	$40,969.35	*GWO*	(2,3,2)	$40,031.26	*iGWO*
(2,1,1)	$18,008.19	*iGWO*	(1,1,1)	$18,433.30	*iGWO*
(3,2,1)	$36,706.60	*iGWO*	(1,1,2)	$22,432.70	*iGWO*
(3,2,3)	$39,901.01	*iGWO*	(1,2,1)	$33,842.24	*iGWO*
(2,2,2)	$30,960.60	*iGWO*			

Table 15 shows both the best results and the processing time for each instance using iGWO.

Table 15. Best results for each instance using iGWO.

Instance	Total Profit	Processing Time	Instance	Total Profit	Processing Time
(2,2,1)	$27,900.68	38.81	(3,3,1)	$48,323.10	37.44
(2,1,2)	$21,167.70	130.23	(3,2,2)	$36,578.07	40.00
(1,3,2)	$43,488.09	38.14	(1,2,2)	$35,839.30	38.41
(1,3,1)	$43,068.69	67.37	(1,2,3)	$33,407.78	37.32
(1,1,3)	$22,318.83	39.28	(3,1,2)	$21,261.06	37.61
(2,3,3)	$39,982.45	37.16	(1,3,3)	$43,941.35	35.54
(3,1,1)	$22,262.89	36.92	(3,1,3)	$22,811.44	40.96
(3,3,3)	$49,952.29	37.29	(2,1,3)	$18,698.81	39.66
(3,3,2)	$38,302.08	37.22	(2,2,3)	$28,370.03	97.64
(2,3,1)	$40,312.11	38.06	(2,3,2)	$40,031.26	38.49
(2,1,1)	$18,008.19	36.63	(1,1,1)	$18,433.30	89.98
(3,2,1)	$36,706.60	37.36	(1,1,2)	$22,432.70	67.68
(3,2,3)	$39,901.01	37.23	(1,2,1)	$33,842.24	94.17
(2,2,2)	$30,960.60	39.55			

Figure 2 shows the main effects of the best solutions for the 27 instances considering the iGWO algorithm. The best results are presented considering case 3 of demand, case 3 of the total available space, and case 3 of supplier capacity. There is a large difference in the profit when the total available space is increased.

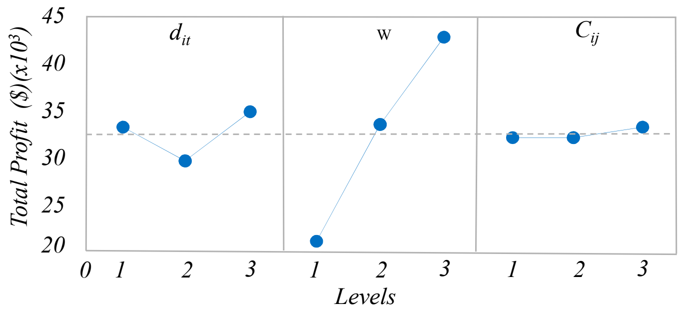


Figure 2. Main effects for the total profit.

Figure 3 presents the main effects of the processing time for the best results (27 instances) considering the iGWO algorithm. Observe that the lowest time is obtained considering case 3 of demand, case 3 of total available space, and case 3 of supplier capacity.

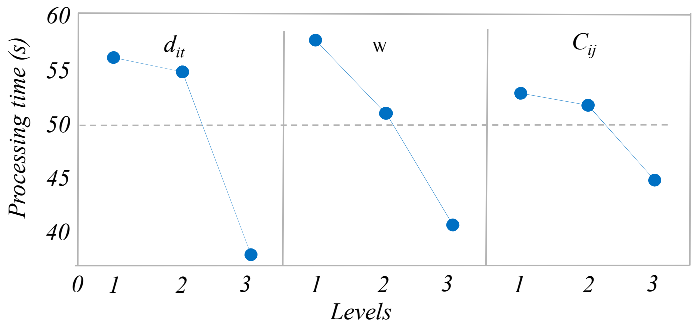


Figure 3. Main effects for the processing time.

From the numerical results, it can be stated that, different from the linear programming techniques, the proposed method is able to solve the supplier selection and purchasing problems under very complex and realistic scenarios, since it does not assume linearity and unimodality in its operation. On the other hand, in comparison to the original GWO and other metaheuristic schemes, our approach is capable of obtaining optimal solutions due to the improved capacity to avoid sub-optimal search locations. Despite its interesting performance properties, the proposed scheme maintains two disadvantages of very high computational cost and difficulty in implementation, as it is not incorporated within the suite of commercial software.

5.2. Statistical Analysis

In this section, we present a statistical analysis of the instances (1,1,1), (2,1,1), (3,1,1), (1,2,1), (1,3,1), (1,1,2), and (1,1,3), in order to show whether there is a significant difference between the profits obtained by LINGO and the metaheuristic methods (iGWO, mGWO, PGWO, TGWO, PSO, DE, and L-SHADE).

The instances were executed using LINGO and the metaheuristic algorithms, each algorithm for 10 independent times. Then, the non-parametric statistical technique, the Kruskal–Wallis test, was used to test for significance. Recall that this statistical test compares the medians among the nine methods used. Table 16 shows the p-values, which present evidence of a significant difference between the medians of the methods (LINGO, iGWO, mGWO, PGWO, TGWO, PSO, DE, and L-SHADE) around the total profit; also, it is possible to observe that the iGWO algorithm presents the best median in five out of seven instances.

Table 16. Kruskal–Wallis test for the total profit.

Comparison of the Total Cost, Instance (1,1,1)					
Algorithm	**Median**	**Method**	**Degrees of Freedom**	**H-Value**	***p*-Value**
LINGO	$11,364.9				
iGWO	$16,626.3	Adjusted for ties	8	44.86	0.000
GWO	0.0				
mGWO	0.0				
PGWO	0.0				
TGWO	0.0	No ties	8	79.07	0.000
PSO	0.0				
DE	0.0				
L-SHADE	0.0				
Comparison of the Total Cost, Instance (2,1,1)					
Algorithm	**Median**	**Method**	**Degrees of Freedom**	**H-Value**	***p*-Value**
LINGO	$5525.6				
iGWO	$14,915.5	Adjusted for ties	8	47.14	0.000
GWO	0.0				
mGWO	0.0				
PGWO	0.0				
TGWO	0.0	No ties	8	83.09	0.000
PSO	0.0				
DE	0.0				
L-SHADE	0.0				
Comparison of the Total Cost, Instance (3,1,1)					
Algorithm	**Median**	**Method**	**Degrees of Freedom**	**H-Value**	***p*-Value**
LINGO	$21,008.8				
iGWO	$19,175.8	Adjusted for ties	8	54.51	0.000
GWO	$12,937.3				
mGWO	$15,235.6				
PGWO	0.0				
TGWO	0.0	No ties	8	66.74	0.000
PSO	0.0				
DE	$19,330.20				
L-SHADE	0.0				
Comparison of the Total Cost, Instance (1,2,1)					
Algorithm	**Median**	**Method**	**Degrees of Freedom**	**H-Value**	***p*-Value**
LINGO	$11,364.9				
iGWO	$30,388.0	Adjusted for ties	8	72.56	0.000
GWO	$23,346.8				
mGWO	$25,093.5				
PGWO	0.0				

Table 16. *Cont.*

TGWO	0.0	No ties	8	78.58	0.000
PSO	0.0				
DE	$22,952.1				
L-SHADE	0.0				
Comparison of the Total Cost, Instance (1,3,1)					
Algorithm	**Median**	**Method**	**Degrees of Freedom**	**H-Value**	***p*-Value**
LINGO	$11,364.9				
iGWO	$42,000.1	Adjusted for ties	8	68.65	0.000
GWO	$37,452.5				
mGWO	$36,718.0				
PGWO	0.0				
TGWO	0.0	No ties	8	69.76	0.000
PSO	$34,925.1				
DE	$35,512.9				
L-SHADE	$3571.7				
Comparison of the Total Cost, Instance (1,1,2)					
Algorithm	**Median**	**Method**	**Degrees of Freedom**	**H-Value**	***p*-Value**
LINGO	$11,364.9				
iGWO	$15,214.9	Adjusted for ties	8	44.35	0.000
GWO	$16,560.68				
mGWO	0.0				
PGWO	0.0				
TGWO	0.0	No ties	8	71.31	0.000
PSO	0.0				
DE	0.0				
L-SHADE	0.0				
Comparison of the Total Cost, Instance (1,1,3)					
Algorithm	**Median**	**Method**	**Degrees of Freedom**	**H-Value**	***p*-Value**
LINGO	$11,364.9				
iGWO	$17,778.3	Adjusted for ties	8	44.79	0.000
GWO	0.0				
mGWO	0.0				
PGWO	0.0				
TGWO	0.0	No ties	8	74.12	0.000
PSO	0.0				
DE	0.0				
L-SHADE	0.0				

5.3. Exploration-Exploitation Study

Exploration represents the ability of a metaheuristic scheme to produce solutions within different areas of the search space. Exploitation is the process in which the search process is intensified over promising areas of the space with the objective of refining the existing solutions [42]. A metaheuristic approach initially promotes exploration. However, as the generations progress, the exploitation should be intensified to improve existing solutions.

Schemes based on metaheuristic principles involve a set of solutions to exploit and explore the search space in order to obtain the optimal solutions for an optimization task. In their operation, the best quality solutions attract other agents conducting the search process towards their locations. As a result of this effect, the distance among individuals decreases while the results of the exploitation increase. Conversely, if the distance among solutions increases, the consequences of the exploration in the metaheuristic scheme are reinforced.

To evaluate the distance among search agents, a diversity index called the dimension-wise diversity assessment [43] is assumed. Under this index, the diversity is computed as follows:

$$Div_j = \frac{1}{n}\sum_{i=1}^{n}\left|median\left(x^j\right) - x_i^j\right|, Div = \frac{1}{m}\sum_{j=1}^{m} Div_j, \tag{21}$$

where $median\left(x^j\right)$ corresponds to the median value of the j-th dimension from the complete population. x^j symbolizes the j-th dimension corresponding to the i-th search agent. n represents the total number of individuals in the population, whereas m corresponds to the number of variables that involve the optimization formulation to be solved.

Under this procedure, the evaluation of the diversity in every dimension Div_j is formulated as the mean distance between the j-th dimension of each individual and the median value from that dimension. Therefore, the diversity of the complete population Div is evaluated by calculating the averaged value of Div_j for each dimension. Div is computed in each iteration during the complete evolution process.

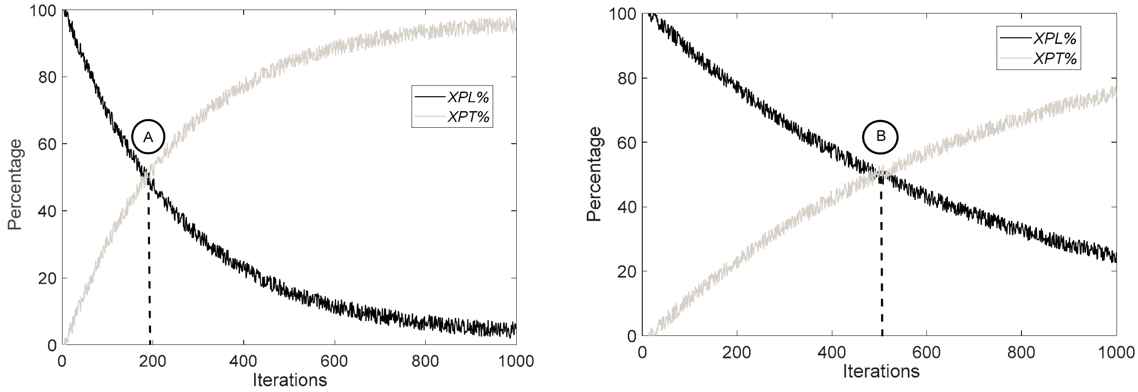
Once computed the value of Div, the exploration-exploitation balance can be computed as the percentage of the time that the processes of exploring or exploiting invest in terms of its diversity. Such values can be evaluated at every iteration by using the following models:

$$XPL\% = \left(\frac{Div}{Div_{max}}\right) \times 100, XPT\% = \left(\frac{|Div - Div_{max}|}{Div_{max}}\right) \times 100, \tag{22}$$

where Div_{max} corresponds to the maximum Div obtained in the complete optimization process.

$XPL\%$ represents the percentage of exploration, which corresponds to the level of exploration. It relates the diversity in each iteration with the maximal reached diversity. On the other hand, $XPT\%$ represents the percentage of exploitation that expresses the level of exploitation. It is computed as the complementary percentage of $XPL\%$, since the difference between the maximum diversity and the current diversity from a particular iteration is generated as a result of the attraction of search agents. Therefore, both indexes $XPL\%$ and $XPT\%$ are mutually complementary. Figure 4 shows the evolution of the balance between exploration and exploitation obtained by the original GWO (Figure 4a) scheme and the improved GWO (Figure 4b) method, considering as an optimization problem the instance (1,2,2). This instance corresponds to a representative optimization task that reflects the complexity of the purchasing problems from an optimization perspective. In the simulation, a total number of 100 iterations have been considered.

In order to compare their performance, the point in which both process exploration and exploitation maintain the same proportion $(XPL\% = 50,\ XPT\% = 50)$ is evaluated. This point represents the location at which the algorithm changes its behavior from the exploration (where the value of $XPL\% > XPT\%$) into exploitation ($XPL\% < XPT\%$).



(**a**) (**b**)

Figure 4. Evolution of the balance between exploration and exploitation obtained by (**a**) the original GWO scheme and (**b**) the improved GWO method considering as optimization problem the instance (1,2,2).

As can be seen from Figure 4, the improved GWO maintains a higher level of exploration, since the balance point (B) is reached in 500 generations. On the other hand, the original GWO method presents a lower exploration level, considering that its balance point (A) is located around the 200 generations. This fact demonstrated that the improved version of GWO is able to explore the search space extensively in order to obtain globally optimal solutions to the complex purchasing problems. This remarkable result is provoked by the inclusion of (I) weighted factors and (II) a displacement vector. These elements avoid the excessive concentration of the search agents in locations, allowing a better distribution within the search space.

6. Conclusions

Supply chain management requires that processes and models may be able to provide solutions in a fast and efficient manner. This paper addresses the supplier selection and order quantity allocation problem. This problem is characterized by its discontinuity, non-linearity, and high multi-modality. In this paper, a modified version of the GWO scheme is introduced to solve this type of complex optimization problem. The improved GWO method called iGWO includes weighted factors and a displacement vector to promote the exploration of the search strategy, avoiding the use of unfeasible solutions.

A representative difficult problem of the literature was selected with the purpose of testing the behavior of the proposed algorithm. Solutions were obtained using LINGO and the proposed iGWO scheme. After exhaustive experimentation, the results demonstrate that the proposed algorithm does not just lead to lower total cost solutions, but also performs a better search strategy in all the compared scenarios.

Author Contributions: Conceptualization, A.A.-R., A.M. and E.O.-B.; methodology, E.C.; software, A.A.-R. and A.R.; validation, A.A.-R. and A.R.; formal analysis, A.A.-R. and A.R.; investigation, A.A.-R. and A.R.; resources, E.C. and E.O.-B.; data curation, A.A.-R. and A.R.; writing—original draft preparation, A.A.-R. and A.R.; writing—review and editing, A.M., E.O.-B. and E.C.; visualization, A.A.-R. and A.R.; supervision, E.C., A.M. and E.O.-B.; project administration, A.A.-R. and A.R.; funding acquisition, E.O.-B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Universidad Panamericana (grant number UP-CI-2020-GDL-07-ING).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Harris, F.W. How many parts to make at once. *Mag. Manag.* **1913**, *10*, 135–136. [CrossRef]
2. Kundu, A.; Guchhait, P.; Pramanik, P.; Maiti, M.K.; Maiti, M. A production inventory model with price discounted fuzzy demand using an interval compared hybrid algorithm. *Swarm Evolut. Comput.* **2016**, *34*, 1–17. [CrossRef]
3. Glock, C.H.; Grosse, E.H.; Ries, J.M. The lot sizing problem: A tertiary study. *Int. J. Prod. Econ.* **2014**, *155*, 39–51. [CrossRef]
4. Nair, A.; Jayaram, J.; Das, A. Strategic purchasing participation, supplier selection, supplier evaluation and purchasing performance. *Int. J. Prod. Res.* **2015**, *53*, 6263–6278. [CrossRef]
5. Dobos, I.; Vörösmarty, G. Inventory-related costs in green supplier selection problems with Data Envelopment Analysis (DEA). *Int. J. Prod. Econ.* **2019**, *209*, 374–380. [CrossRef]
6. Mafakheri, F.; Breton, M.; Ghoniem, A. Supplier selection-order allocation: A two-stage multiple criteria dynamic programming approach. *Int. J. Prod. Econ.* **2011**, *132*, 52–57. [CrossRef]
7. Chang, C.T.; Chen, H.M.; Zhuang, Z.Y. Integrated multi-choice goal programming and multi-segment goal programming for supplier selection considering imperfect-quality and price-quantity discounts in a multiple sourcing environment. *Int. J. Syst. Sci.* **2014**, *45*, 1101–1111. [CrossRef]
8. Ayhan, M.B.; Kilic, H.S. A two stage approach for supplier selection problem in multi-item/multi-supplier environment with quantity discounts. *Comput. Ind. Eng.* **2015**, *85*, 1–12. [CrossRef]
9. Choudhary, D.; Shankar, R. Modeling and analysis of single item multi-period procurement lot-sizing problem considering rejections and late deliveries. *Comput. Ind. Eng.* **2011**, *61*, 1318–1323. [CrossRef]
10. Ebrahim, R.M.; Razmi, J.; Haleh, H. Scatter search algorithm for supplier selection and order lot sizing under multiple price discount environment. *Adv. Eng. Softw.* **2009**, *40*, 766–776. [CrossRef]
11. Brahimi, N.; Dauzere-Peres, S.; Najid, N.M.; Nordli, A. Single item lot sizing problems. *Eur. J. Oper. Res.* **2006**, *168*, 1–16. [CrossRef]
12. Chen, S.; Feng, Y.; Kumar, A.; Lin, B. An algorithm for single-item economic lot-sizing problem with general inventory cost, non-decreasing capacity, and non-increasing setup and production cost. *Oper. Res. Lett.* **2008**, *36*, 300–302. [CrossRef]
13. Massahian Tafti, M.P.; Godichaud, M.; Amodeo, L. Models for the Single Product Disassembly Lot Sizing Problem with Disposal. *IFAC-PapersOnLine* **2019**, *52*, 547–552. [CrossRef]
14. Ghaniabadi, M.; Mazinani, A. Dynamic lot sizing with multiple suppliers, backlogging and quantity discounts. *Comput. Ind. Eng.* **2017**, *110*, 67–74. [CrossRef]
15. Haksever, C.; Moussourakis, J. Determining order quantities in multi-product inventory systems subject to multiple constraints and incremental discounts. *Eur. J. Oper. Res.* **2008**, *184*, 930–945. [CrossRef]
16. Bohner, C.; Minner, S. Supplier selection under failure risk, quantity and business volume discounts. *Comput. Ind. Eng.* **2017**, *104*, 145–155. [CrossRef]
17. Nanda, S.J.; Panda, G. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm Evol. Comput.* **2014**, *16*, 1–18. [CrossRef]
18. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
19. Goldberg, D. Genetic Algorithms in Search. In *Optimization and Machine Learning*; Addison-Wesley Professional: Reading, MA, USA, 1989.
20. Karaboga, D. An Idea based on Honey Bee Swarm for Numerical Optimization. In *Technical Report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department*; Erciyes University: Kayseri, Turkey, 2005; p. 10.
21. Storn, R.; Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
22. Geem, Z.W. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [CrossRef]
23. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]

24. Yang, X.S. Flower pollination algorithm for global optimization. In Proceedings of the International Conference on Unconventional Computing and Natural Computation, Orléan, France, 3–7 September 2012; Volume 7445, pp. 240–249.
25. Lee, A.H.; Kang, H.Y.; Lai, C.M.; Hong, W.Y. An integrated model for lot sizing with supplier selection and quantity discounts. *Appl. Math. Model.* **2013**, *37*, 4733–4746. [CrossRef]
26. Sampson, J.R. *Adaptation in Natural and Artificial Systems (John H. Holland)*; The University of Michigan Press: Ann Arbor, MI, USA, 1975.
27. Lee, C.K.H. A review of applications of genetic algorithms in operations management. *Eng. Appl. Artif. Intell.* **2018**, *76*, 1–12. [CrossRef]
28. Mishra, P.; Talati, I. Optimizing Integrated Production-Inventory Model for Time-Dependent Deteriorating Items Using Analytical and Genetic Algorithm Approach. *Soft Comput. Probl. Solv.* **2019**, *816*, 535–546.
29. Woo, Y.B.; Kim, B.S. A genetic algorithm-based matheuristic for hydrogen supply chain network problem with two transportation modes and replenishment cycles. *Comput. Ind. Eng.* **2019**, *127*, 981–997. [CrossRef]
30. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
31. Mousavi, S.M.; Bahreininejad, A.; Musa, S.N.; Yusof, F. A modified particle swarm optimization for solving the integrated location and inventory control problems in a two-echelon supply chain network. *J. Intell. Manuf.* **2017**, *28*, 191–206. [CrossRef]
32. Kang, H.; Lee, A.H.I.; Wu, C.; Lee, C. An efficient method for dynamic-demand joint replenishment problem with multiple suppliers and multiple vehicles. *Int. J. Prod. Res.* **2017**, *55*, 1065–1084. [CrossRef]
33. Xiong, F.; Gong, P.; Jin, P.; Fan, J.F. Supply chain scheduling optimization based on genetic particle swarm optimization algorithm. *Cluster Comput.* **2019**, *22*, 14767–14775. [CrossRef]
34. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
35. Yang, Y.; Yang, B.; Wang, S. An Improved Grey Wolf Optimizer Algorithm for Energy-Aware Service Composition in Cloud Manufacturing. *Int. J. Adv. Manuf. Technol.* **2019**, *105*, 3079–3091. [CrossRef]
36. Khalilpourazari, S.; Pasandideh, S.H.R.; Niaki, S.T.A. Optimizing a multi-item economic order quantity problem with imperfect items, inspection errors, and back-orders. *Soft. Comput.* **2019**, *23*, 11671–11698. [CrossRef]
37. Rezaei, J.; Davoodi, M. A deterministic, multi-item inventory model with supplier selection and imperfect quality. *Appl. Math. Model.* **2008**, *32*, 2106–2116. [CrossRef]
38. Alejo-Reyes, A.; Olivares-Benitez, E.; Mendoza, A.; Rodriguez, A. Inventory replenishment decision model for the supplier selection problem using metaheuristic algorithms. *Math. Biosci. Eng.* **2020**, *17*, 2016–2036. [CrossRef] [PubMed]
39. Mittal, N.; Singh, U.; Balwinder, S.S. Modified grey wolf optimizer for global engineering optimization. *Appl. Comput. Intell. Soft Comput.* **2016**, *4598*, 1–16. [CrossRef]
40. Al-Betar, M.A.; Awadallah, M.A.; Faris, H.; Aljarah, I.; Hammouri, A.I. Natural selection methods for Grey Wolf Optimizer. *Expert Syst. Appl.* **2018**, *113*, 481–498. [CrossRef]
41. Tanabe, R.; Fukunaga, A.S. Improving the search performance of SHADE using linear population size reduction. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 1658–1665.
42. Črepinšek, M.; Liu, S.H.; Mernik, M. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Surv.* **2013**, *45*, 1–33. [CrossRef]
43. Morales-Castañeda, B.; Zaldívar, D.; Cuevas, E.; Fausto, F.; Rodríguez, A. A better balance in metaheuristic algorithms: Does it exist? *Swarm Evol. Comput.* **2020**, *54*, 100671. [CrossRef]