RAIRO-Oper. Res. 53 (2019) 887–901
https://doi.org/10.1051/ro/2017075

RAIRO Operations Research
www.rairo-ro.org

SUPPLIERS SELECTION PROBLEM WITH QUANTITY DISCOUNTS AND PRICE CHANGES: A HEURISTIC APPROACH

Filipe Rodrigues[1,*] and Cristina Requejo[1]

Abstract. This paper addresses a complex suppliers selection problem with multiple products, considering minimum package quantities, minimum order values related to delivery costs and discounted pricing schemes. Its main contribution is to present an integer linear programming (ILP) model for this suppliers selection problem as well as a model to analyse the impact of prices change. Furthermore, a hybrid heuristic and a genetic algorithm to obtain feasible solutions for this problem are presented. Several randomly generated examples are solved by using the above two models and the heuristic approaches. Experimental results demonstrate the robustness of the genetic algorithm and allow to realize which are the most important decisions in the suppliers selection problem.

Mathematics Subject Classification. 90B50, 90C11, 90C59.

Received April 18, 2017. Accepted October 8, 2017.

1. Introduction

The Suppliers Selection (SS) problem in Supply Chain Management represents one of the most critical tasks to be performed by the purchasing department of a business organization. The effective suppliers evaluation and purchasing processes are of vital importance as the business organizations are becoming increasingly dependent on their suppliers. With the market globalization, the number of potential suppliers and features to consider when selecting suppliers increases, thus the selection process becomes a more arduous task. The suppliers selection problem with multiple products is much more complex than the single product problem. For this reason, the majority of the published studies about this issue are very recent.

Since the products price, the delivery date and the quality of products and services are the most commonly used criteria in SS, a common approach to the SS problem, with multiple products, is a multi-objective approach. In [1, 19, 24, 29, 30] multi-objective (mixed integer) programming approaches are proposed, frequently integrated with other approaches.

Several approaches to the SS problem include, among others, the Analytic Hierarchy Process (AHP) [27], the use of fuzzy concepts [11, 16, 18, 23, 29], including the Fuzzy TOPSIS method [12, 13] and the use of genetic algorithms [2, 10, 14, 17, 28]. Together with these approaches several authors consider stochastic demands [12, 28, 30]. The SS problem under prices change has also received some attention over the last years. The most common approaches followed to deal with these kind of problems are based on the use of stochastic optimization [7, 20]

Keywords. Suppliers selection, multi-product, heuristics, genetic algorithm.

[1] Department of Mathematics, University of Aveiro.
*Corresponding author: fmgrodrigues@ua.pt

however other approaches based on game theory [9] and fuzzy theory [22] can be found in the literature. Furthermore, suppliers capacity constraints, quantity discounts (frequently with cost level conditions), delivery costs and budget constraints are conditions generally considered. In [5, 19] lot-sizing and stock constraints are considered. A literature review of approaches for the SS problem is given in [8, 25].

In this paper a Suppliers Selection (SS) problem with multiple products, minimum order values (MOV) related with delivery costs, minimum package quantities (MPQ) and discounted pricing schemes are considered. The motivation for our study is a suppliers selection problem arising at an electronic equipment assembly company. Here the problem is modeled as a single objective optimization problem assuming costs as the most important criteria. Since we are dealing with small electronic components which can be compared with perishable products whose lifetime is very limited, stocking policies are not desirable. For this reason, lot sizing and stock constraints are not applied here. However, mainly due to fluctuations of the currency exchange rates, there may be a change in the prices during the time gap between the suppliers selection and the actual purchase of products. Thus, the impact of prices change is analysed through the use of stochastic optimization.

From the supply chain management point of view, we consider a multi-product problem where several warehouses supply one plant. It is assumed that there is at least one supplier for each product and not all suppliers can supply all products.

Product package quantities are established by each supplier for every product defining minimum package quantities (MPQ) which is the supplier-predefined minimum number of product packets that can be ordered. Only supplier-predefined product packet quantities can be ordered.

Delivery costs are payable by the buyer only in certain conditions when the buyer order value does not reach a predefined minimum order value (MOV). In this case, if the total purchasing cost of the order to a supplier is less than its MOV, then the delivery cost has to be paid by the buyer. The delivery cost is assumed to be fixed and not depending on the order value.

For each product, each supplier defines a set of product quantity levels dealing with pricing schemes. Each level is associated with a cost according to some discount facility, in such a way that discounts are applied when certain product order quantities (a level) are attained.

The paper has the following contributions.

(i) It presents an integer linear programming (ILP) model to this complex SS problem with multiple products where minimum order values (MOV) related with delivery costs, minimum package quantities (MPQ) and discounted pricing schemes are taken into account.
(ii) It introduces a model to analyse the impact of prices change in this SS problem.
(iii) It develops a hybrid heuristic to obtain feasible solutions in a short time as well as a robust genetic algorithm for which several different strategies are tested.

The remainder of the paper is organized as follows. In Section 2 we present an ILP model to the problem and in Section 3 a model to analyse the impact of prices change. In Section 4 a hybrid heuristic is presented. In Section 5 a genetic algorithm is described. In Section 6 some numerical examples are presented and the obtained computational results are discussed. Conclusions are drawn in Section 7.

2. Mathematical model

In this section we present an ILP model to the SS problem with multiple products, minimum order values related with delivery costs, discounted pricing schemes and minimum package quantities. This model is presented in [3] and has some similarities to the multi-objective model proposed in [24]. However, we impose conditions on the MPQ and on the MOV related to delivery costs which are not considered in [24]. Furthermore, the model described herein has a single objective function minimizing the total purchasing cost.

Consider the set $S = \{1, \ldots, m\}$ of m suppliers and the set $P = \{1, \ldots, r\}$ of r products. For each product $p \in P$ there is a demand quantity Q_p. Define the set S_p of the suppliers $s \in S$ where product $p \in P$ is available, and for each supplier $s \in S$ define the set P_s of available products. Let $pack_{\text{sp}}$ be the minimum package quantity

(MPQ) of product p from supplier s for $p \in P_s$. To define discounted pricing schemes, for all $s \in S$ and all $p \in P$, consider the set $N_{\text{sp}} = \{1, \ldots, \lambda_{\text{sp}}\}$ of cost level conditions. To each cost level condition $n \in N_{\text{sp}}$ corresponds a product quantity level q_{spn} and a product unit cost c_{spn} associated to a discount facility offered by supplier s to product p when the ordered quantity ranges from the product quantity level q_{spn} up to the product quantity level $q_{sp(n+1)}$ of the next cost level condition (infinity, in case of last cost level condition *i.e.* when $n = \lambda_{\text{sp}}$ then $q_{sp(n+1)} = \infty$). The λ_{sp} represents the last cost level condition and is the total number of cost level conditions that supplier s offers to product p. Thus the total number of supply conditions is $SC = \sum_{s\in S,\ p\in P_s} \lambda_{\text{sp}}$ and this number corresponds to the overall total number of possible prices choices. Let mov_s be the minimum order value (MOV) of supplier $s \in S$ related to its delivery cost d_s.

Define the following variables:

x_{spn} : integer variables indicating the number of packages of product p ordered to supplier s with cost level condition n, for all $s \in S$, $p \in P_s$, $n \in N_{\text{sp}}$;

w_{spn} : integer variables indicating the number of units of product p ordered to supplier s with cost level condition n, for all $s \in S$, $p \in P_s$, $n \in N_{\text{sp}}$; notice that $w_{\text{spn}} = x_{\text{spn}} \times pack_{\text{sp}}$;

y_{spn} : binary variables with value 1 if cost level condition n associated with product p and supplier s is used and value 0 otherwise, for all $s \in S$, $p \in P_s$, $n \in N_{\text{sp}}$;

z_s : binary variables with value 1 if supplier s is used and value 0 otherwise, for all $s \in S$;

t_s : binary variables with value 1 if the delivery cost associated to supplier s is supported by the buyer and 0 otherwise, for all $s \in S$.

Considering these parameters and variables, the ILP model for the SS problem is as follows:

$$\min \sum_{s\in S,\ p\in Ps,\ n\in N_{\text{sp}}} c_{\text{spn}} \times w_{\text{spn}} + \sum_{s\in S} d_s \times t_s \tag{2.1}$$

$$\text{s.t.} \quad \sum_{s\in Sp,\ n\in N_{\text{sp}}} w_{\text{spn}} \geq Q_p, \qquad p \in P \tag{2.2}$$

$$\sum_{s\in Sp,\ n\in N_{\text{sp}}} y_{\text{spn}} = 1, \qquad p \in P \tag{2.3}$$

$$\sum_{p\in Ps,\ n\in N_{\text{sp}}} (c_{\text{spn}} \times w_{\text{spn}}) + mov_s \times t_s \geq mov_s \times z_s, \qquad s \in S \tag{2.4}$$

$$\sum_{p\in Ps,\ n\in N_{\text{sp}}} y_{\text{spn}} \geq z_s, \qquad s \in S \tag{2.5}$$

$$\sum_{p\in Ps,\ n\in N_{\text{sp}}} y_{\text{spn}} \leq SC \times z_s, \qquad s \in S \tag{2.6}$$

$$w_{\text{spn}} \geq q_{\text{spn}} \times y_{\text{spn}}, \qquad s \in S_p,\ p \in P,\ n \in N_{\text{sp}} \tag{2.7}$$

$$y_{\text{spn}} \leq x_{\text{spn}}, \qquad s \in S_p,\ p \in P,\ n \in N_{\text{sp}} \tag{2.8}$$

$$M \times y_{\text{spn}} \geq x_{\text{spn}}, \qquad s \in S_p,\ p \in P,\ n \in N_{\text{sp}} \tag{2.9}$$

$$w_{\text{spn}} = pack_{\text{sp}} \times x_{\text{spn}}, \qquad s \in S_p,\ p \in P,\ n \in N_{\text{sp}} \tag{2.10}$$

$$w_{\text{spn}}, x_{\text{spn}} \in \mathbb{N}, \qquad s \in S_p,\ p \in P,\ n \in N_{\text{sp}} \tag{2.11}$$

$$y_{\text{spn}}, z_s, t_s \in \{0, 1\}, \qquad s \in S_p,\ p \in P,\ n \in N_{\text{sp}} \tag{2.12}$$

With the objective function (2.1) the overall costs are minimized, including the total products cost and any delivery costs associated to the suppliers.

Constraints (2.2) aim to satisfy all the products demand. Constraints (2.3) guarantee that each product is supplied by only one supplier and use only one cost level condition. Constraints (2.4) relate the delivery costs with the MOV. The value $\sum_{p\in Ps,n\in N_{\text{sp}}}(c_{\text{spn}}\times w_{\text{spn}})$ is the total order value for supplier $s\in S$. If, for a selected supplier s, it holds $z_s=1$, then we have the following: when delivery costs have to be paid by the buyer, $t_s=1$, constraints (2.4) are redundant, and no constraints exists for the corresponding order value; when there are no delivery costs, $t_s=0$, the corresponding order value (if it exists) must be greater or equal than its MOV. Constraints (2.5) guarantee that if no cost level condition from a supplier is used, then this supplier is not in the solution. Constraints (2.6) establish that if a cost level condition from a supplier is used, then the corresponding supplier must also be used. Constraints (2.7) guarantee that when a product is supplied by a supplier using some cost level condition, the order quantity should be superior to the minimum order quantity of that product necessary to qualify it for the conditions established by the corresponding cost level condition. Constraints (2.8) guarantee that when the ordered number of packages of a product from a supplier for a certain cost level condition is zero, then the corresponding cost level condition is not used; when the corresponding cost level condition is used, then the ordered number of packages is non null. On the other hand, constraints (2.9) establish that when the ordered number of packages of a product from a supplier for a certain cost level condition is non null, then the corresponding cost level condition is used; when the corresponding cost level condition is not used, then the ordered number of packages is null. The value M used in these constraints can be the maximum number of packages that can be ordered. We use $M=\max\{\lceil\frac{\max\{Q_p,\frac{mov_s}{c_{\text{spn}}},q_{\text{spn}}\}}{pack_{\text{sp}}}\rceil,\ s\in S,p\in P_s,n\in N_{\text{sp}}\}$. Constraints (2.10) relate variables w_{spn} with variables x_{spn}. As most of the products are provided in packets, the quantity order of each product should be multiple of the packet size. Variables w_{spn} could be eliminated from the model, however for simplicity of model description and writing we use both variables w_{spn} and x_{spn}. Finally, constrains (2.11) and (2.12) are the constraints on the variables values. The first set of constrains is for the integer variables and the second is for the binary variables.

The number of constraints and variables in the model can be expressed according to the number SC and the cardinality of sets S and P. The model uses $2|P|+3|S|+4SC$ constraints and $2|S|+3SC$ variables.

3. Model to analyze the impact of prices change

Nominal prices used in the suppliers selection problem frequently change due to market changes or exchange rate changes. As a result a proportional change of the products price occurs. Therefore the prices at the time of the actual purchase of the products may be different from the ones considered at the time the suppliers selection is done. These changes may have an impact on the overall suppliers selection process. In order to analyse the impact of these prices change we develop a model that considers uncertainty in the prices values.

It was observed by the company that the most common prices change is such that all the prices either increase or decrease. Therefore we make this assumption in our analysis and consider three possible prices change that we will call scenarios: one in which the prices are unchanged, other in which all the prices increase in the same proportion, and another one where all the prices decrease in the same proportion. The first scenario is the most likely scenario when the time gap is short, the other two scenarios are equally likely scenarios.

Notice that when there is a proportional change of the products prices, the overall cost of the products of a solution will change accordingly. However, since we consider delivery costs, and these costs are dependent on the MOV, it means that the new cost value of the solution will be modified. Eventually some delivery costs must be paid, when its MOV value was not attained. Eventually some other delivery costs are no longer paid, when its MOV value was attained. Straightaway, to avoid paying these new delivery costs a better option could be to increase the order value. Consequently, when the prices change of all the products is proportional, the solution cost is not straightforward proportional to the prices change. Therefore it is important to realize that in the presence of delivery costs depending on the MOV, the optimal solution obtained with the nominal prices may no longer be valid and the new optimal solution must be obtained.

To develop the model the three sets of variables x_{spn}, w_{spn}, and y_{spn} (for all $s \in S$, $p \in P_s$, $n \in N_{\text{sp}}$,) are adapted to accomplish with information about each scenario. Let K be the set of scenarios. Thus, for all $s \in S$, $p \in P_s$, $n \in N_{\text{sp}}$ and $k \in K$, consider the integer variables x_{spnk} indicating the number of packages of product p ordered to supplier s with cost level condition n when scenario k occurs; the integer variables w_{spnk} indicating the number of units of product p ordered to supplier s with cost level condition n when scenario k occurs; and the binary variables y_{spnk} with value 1 if cost level condition n associated with product p and supplier s is used when scenario k occurs.

Let c_{spnk} (for all $s \in S$, $p \in P_s$, $n \in N_{\text{sp}}$, $k \in K$) be the unit cost of product p from supplier s at cost level condition n when scenario k occurs, and consider that v_k is the probability associated to scenario $k \in K$. By replacing in constraints (2.2)-(2.12) variables x_{spn}, w_{spn}, and y_{spn} by the corresponding variables x_{spnk}, w_{spnk}, and y_{spnk} and c_{spn} by c_{spnk}, we obtain the following model that reflects the prices change:

$$\min \sum_{s\in S,\ p\in Ps,\ n\in N_{\text{sp}},\ k\in K} v_k \times c_{\text{spnk}} \times w_{\text{spnk}} + \sum_{s\in S} d_s \times t_s \tag{3.1}$$

$$\text{s.t.} \qquad (2.2) - (2.12), \quad k \in K. \tag{3.2}$$

This model, named PC (Prices Change), induces a two-stage decision structure into the problem. In the first stage, the set of the required suppliers composing the solution is constructed. In the second stage, prices change are considered under some probability and all the suppliers conditions are consequently adapted to these changes. In this second stage, all the remainder decisions, as the choice of the best supply conditions, are taken. In Section 6 we present and analyze the obtained results by using this model and the ILP model. It was observed that the most important decision corresponds to the decisions taken at the first stage. Thus the strategy used to build the heuristics is based on these observations. In the heuristics the first decision is to select the suppliers and after adapt all the remaining decisions of the overall suppliers selection process to that first decision.

4. Hybrid heuristic

The hybrid heuristic Short Price Selection (SPS) we propose is divided in two parts. In the first part an initial feasible solution is constructed. In the second part two distinct strategies are used to improve the feasible solution.

To obtain a feasible solution we use what we call the Initial Feasible Solution (IFS) algorithm. We start by obtaining, for each cost level condition, the minimum quantity allowing the use of that cost level condition. This quantity is the maximum value between the quantity that establishes the cost level condition and the demand quantity of the product associated with this condition. Furthermore this quantity must be multiple of the MPQ value and its cost is obtained by multiplying this value by its corresponding unit cost, corresponding to the cost level condition in use. For each product choose the cost level condition with the minimum cost value of the order. With the selected conditions we obtain a feasible solution and to obtain its cost we still need to add, to the total purchase cost of the products, the delivery costs associated to suppliers for which its MOV is not achieved. The IFS algorithm is specified below. To make the description easier, each cost level condition is identified by its supplier $s \in S$, product $p \in P$ and discount level $n \in N_{\text{sp}}$. Denote by j_{spn} the cost level condition associated with supplier s, product p and level n. Let r_{spn} be the quantity of the product p ordered to supplier s associated to level condition n. Define the set Sol of the selected cost level conditions and the set $Quant$ of the corresponding quantities of the ordered products associated with these cost level conditions. The Sol and $Quant$ sets define the current solution. Additionally denote by $j_{s^*pn^*}$ the cost level condition used in a current solution for product p.

Initial Feasible Solution (IFS) Algorithm:

Input: data (S, P_s, N_{sp}, Q_p, q_{spn}, $pack_{\text{sp}}$, c_{spn}, mov_s, d_s)
Initialize: $Sol := \{\}$; $Quant := \{\}$.

$$
\begin{aligned}
&\textbf{For all } s \in S,\ p \in P_s,\ n \in N_{\text{sp}} \textbf{ do}\\
&\quad a_{\text{spn}} := \max\{Q_p, q_{\text{spn}}\};\\
&\quad b_{\text{spn}} := \left\lceil \frac{a_{\text{spn}}}{pack_{\text{sp}}} \right\rceil \times pack_{\text{sp}}.\\
&\textbf{For all } p \in P \textbf{ do}\\
&\quad j_{s^*pn^*} := \underset{s\in S_p,\ n\in N_{\text{sp}}}{\arg\min} \{b_{\text{spn}} \times c_{\text{spn}}\};\\
&\quad r_{j_{s^*pn^*}} := b_{j_{s^*pn^*}};\\
&\quad Sol := Sol \cup \{j_{s^*pn^*}\};\\
&\quad Quant := Quant \cup \{r_{j_{s^*pn^*}}\}.\\
&\textbf{Calculate } C_T := \text{Cost}(Sol, Quant).
\end{aligned}
$$

Output: feasible solution (Sol, $Quant$, C_T) of the problem.

The procedure *Cost, Cost(Sol,Quant)*, computes the cost of the solution identified by parameters *Sol* and *Quant*, and corresponds to the value of the objective function defined by expression (2.1).

The first strategy to improve a feasible solution for the SS problem is to eliminate some delivery costs and is described in the Delivery Costs Elimination (DCE) algorithm. The main idea is to increase the order quantity of one of the products supplied by one supplier for which the order value does not reaches its MOV. The increased amount should be sufficient so that the order value of the supplier reaches its MOV and avoids the delivery cost. Taking into account that the increased order quantity needs to be multiple of the MPQ value it is possible that for one supplier the new order value exceeds its MOV. Therefore the ordered quantity should be increased for the product that reaches the MOV at the lowest cost.

Delivery Costs Elimination (DCE) Algorithm:

Input: data ((Sol, $Quant$, C_T), S, P_s, N_{sp}, Q_p, q_{spn}, $pack_{\text{sp}}$, c_{spn}, mov_s, d_s).

$$
\begin{aligned}
&\textbf{For all } s \in S \textbf{ do}\\
&\quad V_s := \textstyle\sum_{p\in P_s,\ n\in N_{\text{sp}},\ j_{\text{spn}}\in Sol} r_{j_{\text{spn}}} \times c_{j_{\text{spn}}};\\
&\quad \textbf{If } 0 < V_s < mov_s, \textbf{ then}\\
&\quad\quad V_{\text{inc}} := mov_s - V_s;\\
&\quad\quad \textbf{If } V_{\text{inc}} < d_s \textbf{ then}\\
&\quad\quad\quad \min := +\infty;\ qt := -1;\\
&\quad\quad\quad \textbf{For all } p \in P_s \text{ such that } j_{\text{spn}} \in Sol, \text{ for any } n \in N_{\text{sp}} \textbf{ do}\\
&\quad\quad\quad\quad \textbf{For all } n \in N_{\text{sp}} \textbf{ do}\\
&\quad\quad\quad\quad\quad \textbf{If } j_{\text{spn}} \in Sol \textbf{ then}\\
&\quad\quad\quad\quad\quad\quad acr := \left\lceil \frac{\frac{V_{\text{inc}}}{c_{\text{spn}}}}{pack_{\text{sp}}} \right\rceil \times pack_{\text{sp}};\\
&\quad\quad\quad\quad\quad\quad \textbf{If } V_s + acr \times c_{\text{spn}} < \min \textbf{ then}\\
&\quad\quad\quad\quad\quad\quad\quad \min := V_s + acr \times c_{\text{spn}};\\
&\quad\quad\quad\quad\quad\quad\quad j_{\min} := j_{\text{spn}};\\
&\quad\quad\quad\quad\quad\quad\quad qt := r_{j_{s^*pn^*}} + acr.\\
&\quad\quad\quad\quad\quad \textbf{Otherwise } (\text{if } j_{\text{spn}} \notin Sol)\\
&\quad\quad\quad\quad\quad\quad acr := \left\lceil \frac{V_{\text{inc}} + r_{j_{s^*pn^*}} \times c_{j_{s^*pn^*}}}{c_{\text{spn}} \times pack_{\text{sp}}} \right\rceil \times pack_{\text{sp}};\\
&\quad\quad\quad\quad\quad\quad \textbf{If } V_s - r_{j_{s^*pn^*}} \times c_{j_{s^*pn^*}} + acr \times c_{\text{spn}} < \min \textbf{ then}\\
&\quad\quad\quad\quad\quad\quad\quad \min := V_s - r_{j_{s^*pn^*}} \times c_{j_{s^*pn^*}} + acr \times c_{\text{spn}};\\
&\quad\quad\quad\quad\quad\quad\quad j_{\min} := j_{\text{spn}};
\end{aligned}
$$

$qt := acr$.
If $\min - V_s < d_s$ **then**
$r_{j_{\min}} := qt$;
$Sol := Sol \backslash \{j_{s^*pn^*}\} \cup \{j_{\min}\}$;
$Quant := Quant \backslash \{r_{s^*pn^*}\} \cup \{r_{j_{\min}}\}$.
Calculate $C_T :=$Cost(Sol, $Quant$).
Output: feasible solution (Sol, $Quant$, C_T) of the problem.

The second strategy to improve a feasible solution is the Suppliers Reduction Number (SRN) algorithm. The main idea is to eliminate suppliers from the solution and increase the order value of the remaining suppliers by simultaneously avoiding the payment of the associated delivery costs. The first step is to sort the suppliers in the solution having order values not reaching its MOV and having hight delivery costs. Then, successively, select one such supplier and distribute its order by the remaining suppliers such that, for each product, the selected supplier to receive the order should has the lowest price for each product. If the total cost of the new solution is lower than the total cost of the previous solution, then remove the supplier from the solution and replace the previous solution by this new solution. Repeat the process for the next supplier in the sorted list.

Suppliers Reduction Number (SRN) Algorithm:

Input: data ((Sol, $Quant$, C_T), S, P_s, $N_{\rm sp}$, $c_{\rm spn}$, mov_s, V_s, $b_{\rm spn}$).
Step 1: Sort the suppliers in the solution such that if
$\min\{(mov_{s_1} - V_{s_1}), d_{s_1}\} > \min\{(mov_{s_2} - V_{s_2}), d_{s_2}\}$,
then the supplier s_1 appears first in the list than the supplier s_2.
Let T be the sorted list of the suppliers.
Step 2: **For all** $s \in T$ **do**
$Sol_{\rm prov} := Sol$; $Quant_{\rm prov} := Quant$;
If $V_s < mov_s$ **then**
For all $p \in P_s$ such that $j_{\rm spn} \in Sol$, for any $n \in N_{\rm sp}$, **do**
$\min := +\infty$;
For all $\bar{s} \in T \backslash \{s\}$ **do**
If $p \in P_{\bar{s}}$ **then**
For all $n \in N_{\bar{s}p}$, **do**
If $b_{\bar{s}pn} \times c_{\bar{s}pn} < \min$ **then**
$\min := b_{\bar{s}pn} \times c_{\bar{s}pn}$;
$j_{\min} := j_{\bar{s}pn}$.
If $\min < \infty$ **then**
$Sol_{\rm prov} := Sol_{\rm prov} \backslash \{j_{spn^*}\} \cup \{j_{\min}\}$;
$Quant_{\rm prov} := Quant_{\rm prov} \backslash \{r_{spn^*}\} \cup \{b_{j_{\min}}\}$.
Calculate $U_{\rm prov} :=$Cost ($Sol_{\rm prov}, Quant_{\rm prov}$).
If $U_{\rm prov} < C_T$ **then**
$C_T := U_{\rm prov}$;
$Sol := Sol_{\rm prov}$;
$Quant := Quant_{\rm prov}$.
Output: feasible solution (Sol, $Quant$, C_T).

The SPS heuristic consists in (i) obtaining a feasible solution using the IFS algorithm and (ii) improving the feasible solution as follow: (1) apply the DCE algorithm to the feasible solution; (2) apply the SNR algorithm to the feasible solution obtained; and (3) apply the DCE algorithm again to the feasible solution.

5. Genetic algorithm

In this section we describe a genetic algorithm for the SS problem having six variants. A genetic algorithm [4, 15, 21] is a heuristic based on the theory of natural evolution and has the following generic steps: generation of an initial population, evaluation, selection, crossover and mutation. First, the initial population is formed by generating a set of initial feasible solutions of the problem (individuals). Afterwards, this population is repeatedly modified by the genetic algorithm: a pair of individuals is chosen under some selection rules and is combined under crossover rules giving rise to a new individual (offspring); mutations can be applied to increase the population diversity and the new generation is selected. The convergence to the optimal solution of a genetic algorithm is dependent on the specific rules and operators used. In this work, we compare three selection methods for crossover and two selection rules for the new generation. Therefore obtaining six variants.

To begin we need to obtain the initial population, *i.e.* a set of initial feasible solutions to the SS problem. A feasible solution to the SS problem is formed by a set of suppliers supplying all the demanded products. The main idea of the construction of these solutions is to associate a main supplier to each solution. After, obtain the set of products supplied by this main supplier and the best cost level condition offered by this supplier for each product. When the main supplier can not supply all the required products, to complete the solution, we use a greedy cover algorithm that selects from the remaining ones the supplier that can supply the highest number of products not in the solution at the global lowest cost. Repeat this selection until all the required products are supplied. As a solution is built by associating a main supplier to each solution, the population dimension was fixed equal to m, the number of the available suppliers.

The evaluation of each solution in the population is obtained by calculating the solution cost as given by expression in (2.1) and computed using the procedure *Cost*.

In the selection phase, pairs of individuals are chosen for crossover. We have implemented three different selection methods [15] that differ in the selection pressure, the random method, the tournament method and the roulette wheel method. In the random method the costs of the solutions are not considered. In the case that m is even, $m/2$ pairs are randomly selected. In the case that m is odd, one solution for the next generation is randomly selected and $(m-1)/2$ pairs are randomly selected for crossover. In the tournament method [15] two solutions from the population are randomly selected and the solution having the best cost (lowest cost) is the first element of the pair. Then other two solutions from the population are randomly selected and the solution having the best cost (lowest cost) is the second element of the pair. This process is repeated until we obtain a set of $2m$ pairs of solutions for crossover. Contrary to the random method, in this method a solution can be repeatedly selected or never selected. The roulette wheel method [15] is as follows. Firstly, obtain the fitness of each solution which is the inverse value of its cost. Therefore, the solutions with lower cost will have higher fitness than the solutions with higher cost. Secondly, obtain the fitness of the population which is the sum of the fitness of all the solutions in the population. Next, generate a random value r between 0 and the fitness of the population. After randomly select solutions until the sum of their fitness is greater than or equal to r. The last selected solution for which this sum exceeds the value r is selected for crossover. This selection is repeated to obtain the second element of the pair. The process is repeated until a set of $2m$ pairs of solutions is obtained for crossover. Note that in this method a solution can be repeatedly selected or never selected.

In the crossover operation pairs of individuals, previously selected, are combined giving rise to a new individual. One crossover operator is used: the ISF algorithm. In this operator consider as available suppliers only the suppliers from the pair of solutions selected for crossover and use the ISF algorithm to obtain an offspring.

The mutation operation consists in applying the DCE algorithm and the SRN algorithm, just once, and in this order. All the new individuals suffer a mutation.

After the individuals are evaluated, selected, reproduced and mutated the next generation is created. The size of the population in each generation is constant and equal to m. The selection of the individuals to form the next generation is made from the present generation and the generated offsprings using two different rules. The first rule is named Descendant and Best Progenitor Selection (DBPS). All the generated offsprings are selected for the new generation, and from each pair of progenitors the one having the best cost (lowest cost) is selected for the new generation. The second rule is named Two Best Solutions Selection (TBSS) and it selects among the progenitors and the offspring the two with the best cost (lowest cost).

6. Computational results

In this section the computational results obtained using the hybrid heuristic, the genetic algorithm, the ILP model and the PC model are reported.

All the computational results are obtained for the 25 instances of the SS problem randomly generated. We consider the number r of required products as well as the number m of available suppliers to be in $\{10, 20, 50, 70, 100\}$. For each one of the r products its demand quantity Q_p, $p \in \{1, \ldots, r\}$, is uniformly generated in the interval $(0, 1000]$. For each one of the m suppliers its MOV, mov_s, $s \in \{1, \ldots, m\}$, is uniformly generated in the interval $[0, 100]$ and its associated delivery cost, d_s, is uniformly generated in the interval $[0, 50]$. For each supplier $s \in \{1, \ldots, m\}$ the set P_s of available products is randomly defined and accordingly is defined the set S_p of available suppliers for product p. For each pair (s, p) of a supplier s and a required product p the MPQ, $pack_{\text{sp}}$, is randomly generated as a multiple quantity of 10 in the interval $(0, 100]$. The cost level conditions are established by generating product quantity levels, q_{spn}, in increasing order in the interval $[0, 1000]$, such that the first one is always 0, and the corresponding unit cost of the product, c_{spn}, also in an increasing order, has values in the interval $(0, 2]$. Once the level conditions are established one can define the number λ_{sp} for each pair (s, p) and the total number SC. The costs c_{spnk} in the PC model are taken as $(0.8c_{\text{spn}}, c_{\text{spn}}, 1.2c_{\text{spn}})$.

The computational results are obtained using a processor Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz with 6GB of RAM. The hybrid heuristic and the genetic algorithm are implemented in *Java*. Using the hybrid heuristic and the genetic algorithm feasible solutions corresponding to upper bounds for the optimal solution value are obtained. The ILP model is used to obtain the optimal solution of the SS problem and the PC model is used to obtain the optimal solution, when prices change occur under some probability. Both models are solved using the branch-and-bound method from the optimization software Xpress IVE 7.3.

The computational results obtained for the 25 instances of the SS problem are displayed in Tables 1, 2, 3, and 4. In all the tables in the first column we display the example number having results in that line. The dimension of the examples are displayed in columns two to four: the number of suppliers, the number of products and the number of supply conditions, respectively.

To evaluate the impact of the prices change ten realizations of the probability vector (v_1, v_2, v_3) were considered. Based on historical data, and taking into account that usually the time gap between the suppliers selection process and the purchase of the products is short, the probability of unchanged prices during that time is the highest and was estimated to vary between 0.5 and 0.8. The probability of decreasing prices is not greater than the probability of increasing prices. Therefore, we assume that in five of the ten instances $v_1 = 0.4 \times (1 - v_2)$ and $v_3 = 0.6 \times (1 - v_2)$. In the remaining five instances we assume that $v_1 = v_2 = 0.5 \times (1 - v_2)$. In Table 1 are displayed the average results for the ten realizations of the probability vector (v_1, v_2, v_3) for each one of the tested instances. In column five, named Opt_{PC}, is displayed the average optimal value of the solution obtained using the PC model when considering the three scenarios simultaneously. In column six, named EEV, is displayed the average expected value of the solution obtained for the problem with nominal prices (without price changes) computed when price changes are allowed. To obtain this value we start by solving model ILP considering the nominal prices, and then the optimal value of the variables z_s and t_s, for all $s \in S$, is fixed. By doing so we consider that in a first decision level the suppliers and the corresponding delivery cost conditions are selected. Afterwards, with the first stage decisions fixed, the ILP model is solved for each one of the three scenario prices and the mean value, EEV value, is computed. In the last column, we compute the average Value of the Stochastic Solution (VSS), which is the difference between the values EEV and Opt_{PC}. The VSS is a quality value indicating how good is the solution obtained for the problem with nominal prices, in terms of the first level decision variables, when prices change can occur. Thus, the lower the VSS, the higher the quality of the solution.

TABLE 1. Computational results obtained to evaluate the influence of the prices change.

Ex	$\|S\|$	$\|P\|$	SC	Opt_{PC}	EEV	VSS
1	10	10	495	191.8	191.8	0
2	10	20	1177	198.4	198.4	0
3	10	50	3578	479.7	479.7	0
4	10	70	1942	787.1	787.1	0
5	10	100	2731	1330.1	1330.1	0
6	20	10	802	159.2	159.2	0
7	20	20	1445	177.2	177.2	0
8	20	50	1813	495.2	495.2	0
9	20	70	1615	949.5	950.1	0.6
10	20	100	1432	1539.8	1540.2	0.4
11	50	10	661	124.2	124.2	0
12	50	20	1035	254.3	254.3	0
13	50	50	2613	545.5	545.5	0
14	50	70	1682	1081.2	1081.8	0.6
15	50	100	1447	1709.9	1709.9	0
16	70	10	516	148.8	148.8	0
17	70	20	1868	188.0	188.1	0.1
18	70	50	1601	739.5	739.8	0.3
19	70	70	3468	717.4	717.4	0
20	70	100	1367	2082.9	2082.9	0
21	100	10	750	117.7	117.7	0
22	100	20	3113	157.2	157.2	0
23	100	50	1902	656.5	656.5	0
24	100	70	1170	1430.0	1430.0	0
25	100	100	1197	2597.8	2597.8	0

Observing Table 1, the values VSS either are, almost always, equal to zero or very small. Such VSS values suggest that the solutions obtained for the nominal price scenario are also good solutions for the case when prices can change, in terms of the first level decisions. This means that a right choice for the set of suppliers (first level decisions) when the problem with nominal prices is considered leads to solutions close to the optimal solutions obtained by the stochastic model. Hence, among the several decisions within the suppliers selection process, the most important one is the choice of the set of suppliers involved in the overall decision. These results support the strategy used for the construction of the genetic algorithm in which the first step is the selection of the suppliers and afterwards all the other decisions are taken.

The following specifications of the genetic algorithm are settled. The size of the initial population is defined to be equal to the total number of available suppliers, which is also the number of individuals in each new generation. The number of child-bearing couples is half the size of the population for the random method and is double the size of the population for the other two methods. We apply mutation to each new offspring. This option was taken after preliminary computational results done without applying mutation revealed poor quality of the results. Each new generation consists of elements selected in two different ways among the individuals of the current generation and the offsprings generated. Let the number of generations, which is also the stopping criterion, be equal to 100. Thus, the algorithm stops at iteration 100, producing a list of feasible solutions, possibly not all distinct. The final SS solution is one of minimum cost among those feasible solutions.

To evaluate the quality of the obtained solutions the gap between the upper bound, B_u, and the optimal solution, Opt_{ILP}, is calculated as follows:

$$gap = 100\frac{(B_u - \text{Opt}_{ILP})}{\text{Opt}_{ILP}},$$

TABLE 2. Computational results obtained by the SPS heuristic and the six variants of the genetic algorithm.

							Random Method				**Tournament Method**				**Roulette Wheel Method**			
Ex	$\|S\|$	$\|P\|$	SC	Opt$_{ILP}$	SPSH		DBPS		TBSS		DBPS		TBSS		DBPS		TBSS	
1	10	10	495	190.8	209.6	9.9%	**190.8**	**0.0%**	**190.8**	**0.0%**	**190.8**	**0.0%**	**190.8**	**0.0%**	**190.8**	**0.0%**	**190.8**	**0.0%**
2	10	20	1177	198.2	204.6	3.2%	**200.2**	**1.0%**	**200.2**	**1.0%**	**200.2**	**1.0%**	**200.2**	**1.0%**	**200.2**	**1.0%**	**200.2**	**1.0%**
3	10	50	3578	476.6	501.6	5.2%	**476.6**	**0.0%**	**476.6**	**0.0%**	**476.6**	**0.0%**	**476.6**	**0.0%**	**476.6**	**0.0%**	**476.6**	**0.0%**
4	10	70	1942	782.0	795.0	1.7%	**795.0**	**1.7%**	**795.0**	**1.7%**	**795.0**	**1.7%**	**795.0**	**1.7%**	**795.0**	**1.7%**	**795.0**	**1.7%**
5	10	100	2731	1318.0	1334.0	1.2%	**1334.2**	**1.2%**	**1334.2**	**1.2%**	**1334.2**	**1.2%**	1342.2	1.8%	**1334.2**	**1.2%**	1342.2	1.8%
6	20	10	802	158.2	213.0	34.6%	158.6	0.3%	**158.2**	**0.0%**	**158.2**	**0.0%**	**158.2**	**0.0%**	**158.2**	**0.0%**	**158.2**	**0.0%**
7	20	20	1445	175.6	233.4	32.9%	**178.4**	**1.6%**	**178.4**	**1.6%**	**178.4**	**1.6%**	**178.4**	**1.6%**	**178.4**	**1.6%**	**178.4**	**1.6%**
8	20	50	1813	489.4	548.2	12.0%	**514.0**	**5.0%**	522.8	6.8%	515.8	5.4%	521.2	6.5%	515.8	5.4%	528.8	8.1%
9	20	70	1615	941.8	1046.4	11.1%	973.8	3.4%	973.8	3.4%	973.8	3.4%	973.8	3.4%	**972.4**	**3.2%**	973.8	3.4%
10	20	100	1432	1527.8	1488.1	3.9%	**1566.6**	**2.5%**	**1566.6**	**2.5%**	**1566.6**	**2.5%**	**1566.6**	**2.5%**	**1566.6**	**2.5%**	**1566.6**	**2.5%**
11	50	10	661	123.6	190.2	53.9%	**128.0**	**3.6%**	**128.0**	**3.6%**	**128.0**	**3.6%**	**128.0**	**3.6%**	**128.0**	**3.6%**	**128.0**	**3.6%**
12	50	20	1035	252.8	399.4	58.0%	**259.2**	**2.5%**	**259.2**	**2.5%**	**259.2**	**2.5%**	**259.2**	**2.5%**	262.4	3.8%	**259.2**	**2.5%**
13	50	50	2613	539.4	673.3	25.4%	601.7	11.5%	596.1	10.5%	**587.7**	**8.9%**	605.1	12.2%	595.1	10.3%	598.5	11.0%
14	50	70	1682	1075.6	1223.8	13.8%	**1132.2**	**5.3%**	**1132.2**	**5.3%**	1145.9	6.5%	1143.2	6.3%	1143.2	6.3%	1139.9	6.0%
15	50	100	1447	1700.4	1934.4	13.8%	1787.5	5.1%	1779.6	4.7%	**1775.9**	**4.4%**	1794.9	5.6%	1790.8	5.3%	1795.4	5.6%
16	70	10	516	146.5	198.7	33.8%	**149.5**	**2.0%**	**149.5**	**2.0%**	**149.5**	**2.0%**	**149.5**	**2.0%**	**149.5**	**2.0%**	**149.5**	**2.0%**
17	70	20	1868	186.2	323.2	73.6%	193.7	4.0%	193.8	4.1%	195.4	4.9%	194.5	4.5%	195.2	4.8%	**193.0**	**3.7%**
18	70	50	1601	733.4	956.0	30.4%	818.6	11.6%	820.6	11.9%	827.4	12.8%	827.8	12.9%	820.0	11.8%	**815.0**	**11.1%**
19	70	70	3468	710.0	930.4	31.0%	799.0	12.5%	799.5	12.6%	797.9	12.4%	**787.7**	**10.9%**	794.3	11.9%	795.1	12.0%
20	70	100	1367	2069.0	2217.0	7.2%	**2142.5**	**3.6%**	**2142.5**	**3.6%**	2149.0	3.9%	2153.0	4.1%	2151.0	4.0%	2178.5	5.3%
21	100	10	750	115.8	218.2	88.4%	**122.8**	**6.0%**	**122.8**	**6.0%**	**122.8**	**6.0%**	**122.8**	**6.0%**	**122.8**	**6.0%**	**122.8**	**6.0%**
22	100	20	3113	156.0	271.0	73.7%	171.6	10.0%	171.0	9.6%	171.6	10.0%	**165.4**	**6.0%**	170.4	9.2%	169.4	8.6%
23	100	50	1902	651.6	857.1	31.5%	719.5	10.4%	735.5	12.9%	731.5	12.3%	730.2	12.1%	720.1	10.5%	**716.3**	**9.9%**
24	100	70	1170	1419.0	1643.8	15.8%	1561.4	10.0%	1557.8	9.8%	1553.6	9.5%	**1550.2**	**9.2%**	1558.2	9.8%	1557.0	9.7%
25	100	100	1197	2478.2	2590.8	4.6%	2575.0	3.9%	2592.1	4.6%	2584.5	4.3%	**2564.0**	**3.5%**	2592.1	4.6%	2596.2	4.8%
			Smallest gap				0.0%		0.0%		0.0%		0.0%		0.0%		0.0%	
			Biggest gap				12.5%		12.9%		12.8%		12.9%		11.9%		12.0%	
			Number of best solutions				14		14		14		15		12		14	

Table 3. Some statistics for the computational results obtained by the six variants of the genetic algorithm.

Ex	$\|S\|$	$\|P\|$	SC	Standard Deviation Value	Mean Value	Mean Gap	Lowest Gap
1	10	10	495	0.0	190.8	0.0%	0.0%
2	10	20	1177	0.0	200.2	1,0%	1.0%
3	10	50	3578	0.0	476.6	0.0%	0.0%
4	10	70	1942	0.0	795.0	1.7%	1.7%
5	10	100	2731	5.7	1336.9	1.4%	1.2%
6	20	10	802	0.3	158.3	0.0%	0.0%
7	20	20	1445	0.0	178.4	1.6%	1.6%
8	20	50	1813	10.5	519.7	6.2%	5.0%
9	20	70	1615	0.0	973.6	3.4%	3.2%
10	20	100	1432	0.0	1566.6	2.5%	2.5%
11	50	10	661	0.0	128.0	3.6%	3.6%
12	50	20	1035	0.0	259.7	2.7%	2.5%
13	50	50	2613	2.3	597.4	10.7%	8.9%
14	50	70	1682	5.4	1139.4	5.9%	5.3%
15	50	100	1447	5.6	1787.4	5.1%	4.4%
16	70	10	516	0.0	149.5	2.0%	2.0%
17	70	20	1868	0.5	194.3	4.3%	3.7%
18	70	50	1601	2.5	818.6	11.6%	11.1%
19	70	70	3468	2.8	797.0	12.2%	10.9%
20	70	100	1367	25.5	2152.8	4.0%	3.6%
21	100	10	750	0.0	122.8	6.0%	6.0%
22	100	20	3113	1.6	169.9	8.9%	6.0%
23	100	50	1902	2.3	725.5	11.4%	9.9%
24	100	70	1170	3.1	1556.4	9.7%	9.2%
25	100	100	1197	15.0	2584.0	4.3%	3.5%

where B_u is obtained using the hybrid heuristic or the genetic algorithm and Opt_{ILP} is obtained using the ILP model.

The Table 2 displays the computational results for the heuristics. In the fifth column of the table the optimal solution of the problem obtained using the ILP model is displayed. The obtained results by the SPS heuristic are displayed in the sixth and seventh columns. In the remaining columns the results obtained by the six genetic algorithm variants are shown. For each pair of columns the value of the obtained solution and its corresponding gap are reported. With bold we mark the best solution for each example.

For each pair of the columns, the last three lines present results reached in each column: the smallest gap, the biggest gap and the number of times the best solution is attained. The roulette wheel method together with the DBPS selection method is the variant having the smallest difference between the biggest gap and the smallest gap. In fact, no variant of the genetic algorithm is absolutely better than the others. For example, using the random method together with the selection methods DBPS and TBSS we obtained the best solution for the 8th and 20th examples. Using the tournament method together with the selection methods DBPS and TBSS we obtained the best solution for the 13th and 22th examples. Using the roulette wheel method together with the selection methods DBPS and TBSS we obtained the best solution for the 9th and 23th examples.

The gaps associated with the obtained solutions vary between 0.0 and 12.9 percent. In general, the gaps increase as the number of suppliers and the number of products increase.

The solutions obtained with the genetic algorithm are always better than the solutions obtained by the SPS heuristic. These results are expected because the SPS heuristic is totally included in all variants of the genetic algorithm.

TABLE 4. Execution time used by the six variants of the genetic algorithm to obtain the results displayed in Table 2.

Ex	$\|S\|$	$\|P\|$	SC	Random Method		Tournament Method		Roulette Wheel Method	
				DBPS	TBSS	DBPS	TBSS	DBPS	TBSS
1	10	10	495	1	1	1	1	1	1
2	10	20	1177	1	1	3	4	3	4
3	10	50	3578	15	15	65	65	67	67
4	10	70	1942	16	16	62	63	63	58
5	10	100	2731	39	41	166	168	171	171
6	20	10	802	1	1	2	3	3	3
7	20	20	1445	2	2	10	11	10	10
8	20	50	1813	16	16	68	68	69	66
9	20	70	1615	28	30	113	112	112	120
10	20	100	1432	47	48	189	198	195	191
11	50	10	661	1	1	3	2	3	3
12	50	20	1035	5	5	19	21	18	20
13	50	50	2613	65	65	266	257	270	294
14	50	70	1682	83	85	351	345	352	343
15	50	100	1447	150	145	586	591	586	579
16	70	10	516	1	1	3	3	3	3
17	70	20	1868	11	11	43	42	45	44
18	70	50	1601	64	65	257	269	258	267
19	70	70	3468	243	237	1058	936	937	972
20	70	100	1367	209	214	854	901	867	907
21	100	10	750	3	3	6	7	7	8
22	100	20	3113	27	27	101	117	92	89
23	100	50	1902	126	130	482	488	483	500
24	100	70	1170	134	133	543	519	538	533
25	100	100	1197	257	251	988	1030	1012	1009

For each example and for the six variants of the genetic algorithm, Table 3 displays the standard deviation value, the mean value, the gap associated to the mean value and the lowest gap of the obtained results. For the set of examples considered, the lowest gap values of the obtained results by the six variants of the genetic algorithm vary between 0.0 and 11.1 percent. Such results show that the proposed genetic algorithm is very robust and it can be applied to a large number of different instances, producing good feasible solutions.

The values displayed in the column of the standard deviation value allow us to say that the variants of the genetic algorithm sometimes produce very different results, as was the case for the 20th and the 25th examples.

In Table 4 the execution computational times, in seconds, used by the genetic algorithm are displayed. The computational times associated with the SPS heuristic are not displayed in this table because all of them are lower than one second. For our experiments, all execution times were lower than 1080 s (18 min). In general, the computational time increases with an increase in the number of suppliers and in the number of products. Further, the computational time also increases with an increase in the number of supply conditions. In fact, the highest execution time corresponds to the 19th example, the example having the highest number of supply conditions.

7. Conclusion

In this work we address the suppliers selection problem with several products, considering minimum package quantities (MPQ), discounted pricing schemes and minimum order values (MOV) related with delivery costs.

An integer linear programming model minimizing the total cost, the criteria elected as the most important, is presented. The impact of prices change is analyzed through the results obtained using an integer linear programming model considering scenarios. The obtained results suggested that the most important decision in the SS problem corresponds to the choice of the set of the required suppliers. Based on this finding, a genetic algorithm is built in which a hybrid heuristic is used to select the suppliers. Since minimum order values related with delivery costs has not been usually considered in the context of the suppliers selection problem, both the hybrid heuristic and the genetic algorithm proposed are new. Therefore, one of the most important contributions of this work is to present one exact formulation and two heuristic approaches for the SS problem with minimum order values.

Based on 25 randomly generated instances, the efficiency and efficacy of the implemented algorithms is tested. In particular, the results suggest that the genetic algorithm is robust in the obtention of good solutions for the SS problem, since the values of the obtained solutions are very close to the value of the optimal solution determined by the exact model. In fact, the relative gap for all the used instances is always lower than 13%, and lower than 6% in most of the cases.

As future work it is to extend the methodologies followed in this paper to the SS problem in which stocking policies are desirable (no perishable products). This assumption makes the SS problem much harder to solve since it is necessary to obtain the optimal solution over a specific time horizon for a set of projects/instances that are not independent.

Acknowledgements. The research of the first author has been supported by *Center for Research and Development in Mathematics and Applications (CIDMA)* and FCT, the Portuguese Foundation for Science and Technology under Grant PD/BD/114185/2016. The research of the second author has been partially supported by CIDMA and FCT, within project UID/MAT/04106/2013.

References

[1] F. Arikan, Multiple Objective Fuzzy Sourcing Problem with Multiple Items in Discount Environment. *Math. Problems Eng.* (2015) 1–14.

[2] Z.H. Che and H.S. Wang, Supplier selection and supply quantity allocation of common and non-common parts with multiple criteria under multiple products. *Comput. Industrial Eng.* **55** (2008) 110–133.

[3] M. Costa, C. Requejo and F. Rodrigues, The suppliers selection problem: a case study. Operations Research and Big Data, Vol. 15 of Studies in Big Data Springer International Publishing (2015) 175–184.

[4] J. Dréo, A. Pétrowski and P. Siarry and E. Taillard, Metaheuristics for Hard Optimization. Methods and Case Studies. Springer (2006).

[5] D. Esmaeili, A. Kaazemi and B. Pourghannad, A two-level GA to solve an integrated multi-item supplier selection model. *Appl. Math. Comput.* **219** (2013) 7600–7615.

[6] B. Feng, Z.P. Fan and Y. Li, A decision method for supplier selection in multiservice outsourcing. *Int. J. Prod. Econ.* **132** (2011) 240–250.

[7] R. Hammami, C. Temponi and Y. Frein, A scenario-based stochastic model for supplier selection in global context with multiple buyers, currency fluctuation uncertainties, and price discounts. *Eur. J. Oper. Res.* **233** (2014) 159–170.

[8] W. Ho, X. Xu and P.K. Dey, Multi-criteria decision making approaches for supplier evaluation and selection: A literature review. *Eur. J. Oper. Res.* **202** (2010) 16–24.

[9] C. Hsieh, Y.L. Chang and C.H. Wu, Competitive pricing and ordering decisions in a multiple-channel supply chain. *Int. J. Prod. Econ.* **154** (2014) 156–165.

[10] C. Hsieh and Y.T. Lu, Manufacturers return policy in a two-stage supply chain with two risk-averse retailers and random demand. *Eur. J. Oper. Res.* **207** (2010) 514–523.

[11] F. Jolai, S. Yazdian, K. Shahanaghi and M. Khojasteh, Integrating fuzzy TOPSIS and multi-period goal programming for purchasing multiple products from multiple suppliers. *J. Purchasing Supply Manag.* **17** (2011) 42–53.

[12] S. Kara, Supplier selection with an integrated methodology in unknown environment. *Expert Syst. Appl.* **38** (2011) 2133–2139.

[13] S. Kilic, An integrated approach for supplier selection in multi-item/multi-supplier environment. *Appl. Math. Model.* **37** (2013) 7752–7763.

[14] P.L. Meena and S.P. Sarmah, Multiple sourcing under supplier failure risk and quantity discount: A genetic algorithm approach. *Trans. Res. Part E: Logistics Trans. Rev.* **50** (2013) 84–97.

[15] Z. Michalewicz, Genetic algorithms + data structures = evolution programs. Springer Science and Business Media (2013).

[16] S. Nazari−Shirkouhi, H. Shakouri, B. Javadi and A. Keramati, Supplier selection and order allocation problem using a two-phase fuzzy multi-objective linear programming. *Appl. Math. Model.* **37** (2013) 9308–9323.

[17] M. Paydar and M. Saidi−Mehrabad, A hybrid genetic algorithm for dynamic virtual cellular manufacturing with supplier selection. *Int. J. Adv. Manuf. Technology* **92** (2017) 3001–3017.

[18] J. Razmi and E. Maghool, Multi-item supplier selection and lot-sizing planning under multiple price discounts using augmented ε-constraint and Tchebycheff method. *Int. J. Adv. Manuf. Technology* **49** (2010) 379–392.

[19] J. Rezaei and M. Davoodi,Multi-objective models for lot-sizing with supplier selection. *Int. J. Prod. Econ.* **130** (2011) 77–86.

[20] A. Sen, H. Yaman, K. Güler and E. Körpeoglu, Multi-period supplier selection under price uncertainty. *J. Oper. Res. Soc.* **65** (2014) 1636–1648.

[21] S.N. Sivanandam and S.N. Deepa, Introduction to Genetic Algorithms. Springer (2007).

[22] G. Sun, Y. Liu and Y. Lan, Fuzzy two-stage material procurement planning problem. *J. Intell. Manuf.* **22** (2011) 319–331.

[23] S. Torabi and E. Hassini, Multi-site production planning integrating procurement and distribution plans in multiechelon supply chains: an interactive fuzzy goal programming approach. *Int. J. Prod. Res.* **47** (2009) 5475–5499.

[24] V. Wadhwa and A.R. Ravindran,Vendor selection in outsourcing. *Comput. Oper. Res.* **34** (2007) 3725–3737.

[25] N. Ware, S. Singh and D. Banwet, Supplier selection problem: A state-of-the-art review. *Manag. Sci. Lett.* **2** (2012) 1465–1490.

[26] C.A. Weber, J.R. Current and A. Desai, An optimization approach to determining the number of vendors to employ. *Supply Chain Manag. Int. J.* **5**(2005) 90–98.

[27] W. Xia and Z. Wu, Supplier selection with multiple criteria in volume discount environments. *Omega* **35** (2007) 494–504.

[28] P.C. Yang, H.M. Wee, S. Pai and Y.F. Tseng, Solving a stochastic demand multiproduct supplier selection model with service level and budget constraints using Genetic Algorithm. *Expert Syst. Appl.* **38** (2011) 14773–14777.

[29] M.H.F. Zarandi and S. Saghiri, A comprehensive fuzzy multi-objective model for supplier selection process. In *FUZZ '03. The The 12th IEEE International Conference on Fuzzy Systems* (2003) 368–373

[30] G. Zhang and L. Ma, Optimal acquisition policy with quantity discounts and uncertain demands. *Int. J. Prod. Res.* **47** (2009) 2409–2425.