Ranking Zero-Inventory Ordering Policies

Antonio Sedeño-Noda[1]
Departamento de Estadística, Investigación Operativa y Computación (DEIOC).
Universidad de La Laguna, CP 38271- La Laguna, Tenerife (España).

We address the problem of finding the K best zero-inventory ordering policies of an economic lot sizing problem considering n periods. The main result in this paper is the proof that the second best zero-inventory ordering policy can be obtained in O(n^2) time from the best zero-inventory ordering policy. Consequently, we design an O(Kn^2) time and O(Kn) space algorithm to determine the K best zero-inventory ordering policies. Also, we address the resolution of some problems in relation with the K best zero-inventory ordering policies problem.

Keywords: The Economic Lot sizing problem; Zero-Inventory Ordering Policies; K best solutions;

The *Economic Lot Sizing* (ELS) problem consist in to satisfy at minimum cost the known nonnegative demands for a specific commodity in a number of consecutive periods denominated *planning horizon*. For that, it is possible to store units of the commodity to satisfy future demands, but backlogging is not allowed. This problem was first addressed by Wagner and Within (1958). Several efficient methods to solve it exist (see Aggarwal and Park (1990), Federgruen and Tzur (1991), Wagelmans aet al. (1992)). Moreover, an optimal solution of ELS problem satisfies the *zero- inventory ordering* (ZIO) *property* (see Wagner and Whitin (1958), Wagner (1960) and Zangwill (1968)).

In many optimization problems, the calculation of alternative solutions for a given optimal solution has very significance. Knowledge of these alternative solutions can contribute to efficiently solve problems in which different criteria are considered or problems with additional constraints and/or conditions; where the best previous solution already cannot be used (for example it becomes unfeasible).

In this paper, we consider the K best ZIO policies problem as the problem to determine the K best ZIO solutions of the classical mathematical formulation of the ELS problem. We are unaware of any previous references to this problem in the literature. We establish a recursion that leads to obtain the best ZIO policy for a given optimal policy in O(n^2). From this result, we derive an algorithm to find the K best ZIO policies in O(Kn^2) time using O(Kn) space.

[1] Corresponding author. Tel.: +34922845054; fax: +34922318170
E-mail addresses: asedeno@ull.es

The structure of the paper follows. Section 1 presents the mathematical formulation of the ELS problem and the *K* best ZIO policies problem. In section 2, we introduce the method to obtain the second best ZIO policy. Section 3 contains a detailed pseudo code and an explanation of the *K* best ZIO policies algorithm and some procedures. Moreover, the worse case time and space theoretical complexity of the algorithm is proven. Finally, in section 4, we offer our conclusions and address other problem that can be solved by the introduced approach.

1. The Economic Lot Sizing problem and the *K* best ZIO policies problem.

In order to introduce a classical mathematical formulation of ELS problem, let us to consider the following attributes. Let *n* be the length of the planning horizon and d_i, p_i, f_i, h_i be, respectively, the demand, marginal production cost, setup cost and unit holding cost in period *i*, $i = 1,...,n$. We assume without lost of generality that $f_i \geq 0$ for any $i = 1,...,n$ (see Wagelmans et al. (1992), Van Hoesel and Wagelmans (1993)). In similar way, we assume without lost of generality that $d_n > 0$. Denote by $d_{ij} = \sum_{k=i}^{j} d_k$, $1 \leq i \leq j \leq n$, and define the following variables: (1) x_i being the number of units produced/ordered in period *i*; (2) I_i being the number of units in stock (inventory level) at the end of the period *i* and (3) y_i being a 0-1 variable that takes the value 1 when a setup occurs in period *i*; otherwise is 0. Then, a mathematical formulation of the ELS problem is:

$$\text{Minimize} \quad C(x,I) = \sum_{i=1}^{n} (p_i x_i + h_i I_i + f_i y_i) \qquad (1)$$

subject to

$$x_i + I_{i-1} - I_i = d_i, \qquad \text{for } i = 1,...,n \qquad (2)$$

$$d_{in} y_i - x_i \geq 0, \qquad \text{for } i = 1,...,n \qquad (3)$$

$$I_0 = I_n = 0 \qquad (4)$$

$$x_i \geq 0,\ I_i \geq 0,\ y_i \in \{0,1\} \quad \text{for } i = 1,...,n \qquad (5)$$

It is well-know that at least one optimal solution of the ELS problem satisfies the ZIO property (see Wagner and Whitin (1958), Wagner (1960) and Zangwill (1968)). The ZIO property establishes that $x_i I_{i-1} = 0,\ i = 1,...,n$, that is, production in period *i* equals 0 or d_{ik} for some $k \geq i$. From this property, the ELS problem is solved by several dynamic programming approaches.

As in Zangwill (1968), the ELS problem can be formulated as a Minimum Cost Flow problem. The underlying network for this problem is as follows. Let $G(V,A)$ be a directed network, where V is the set of $n+1$ (there are as many nodes as periods plus 1) nodes and A is the set of $2n-1$ arcs. Each node i ($i=1,...,n$) has a demand equal to $-d_i$, whereas the node 0 (source node) has to fulfill the demand in each node with an amount equal to $d_0 = \sum_{i=1}^{n} d_i$.

We distinguish two types of arcs: *order/production arcs* which are related to the decision variables x_i with $i=1,...,n$ and the *inventory arcs* associated to the state variables I_i with $i=1,...,n$. Each order/production arc (0, *i*) in the network has a unit cost equals c_i, a setup cost f_i and an infinite capacity. On the other hand, each inventory arc (*i*, *i*+1) has a unit cost equal to h_i and an infinite capacity. The network for the ELS problem is depicted in Figure 1.

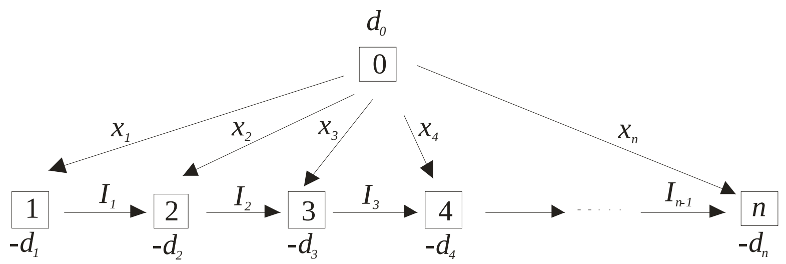

Figure 1. The Network for the ELS *problem*.

Given a pair (x_i, I_i) for $i=1,...,n$ defining a ZIO policy, we build the subset of arcs $T \subseteq A$ of *G* as follows: (1) if $x_i > 0$ then add the production arc (0, *i*) to *T*; (2) if $I_i > 0$ then add the inventory arc (*i*-1, *i*) to *T* and (3) if $x_i = 0$ and $I_i = 0$ then add (*i*-1, *i*) to *T* when $i \neq 1$ or add (0, 1) to *T* when $i=1$. By construction, if the arc $(0,i) \in T$ with $i>1$, then $x_i > 0$. Note that the arc (0, 1) always belongs to any tree *T* obtained from any ZIO policy. In any case, the obtained subset of arcs *T* is a spanning tree of *G*. Under this construction, any ZIO policy determines a unique spanning tree rooted at node 0 such that the unique path in the tree from root node 0 to every other node is a directed path. We refer to these spanning trees as *directed out-spanning trees*. Note that in this kind of tree, each node $i \in V \setminus \{0\}$ has only one predecessor node in the tree $(pred_i(T))$, that is, its in-degree is one. Moreover, any each node $i \in V \setminus \{0,n\}$ has only one successor node in the tree $(succ_i(T))$ since any node $i \in V \setminus \{0,n\}$ in *G* has out-degree one. Given a tree *T*, define $pt_i(T)$ as the first node posterior to node *i* such that the production arc $(0, pt_i(T)) \in T$. In the case that this arc does not exist, then

$pt_i(T) = n+1$. In similar way, define $th_i(T)$ as the node such that the production arc $(0, th_i(T)) \in T$ and this arc belong to the path from node 0 to node *i* in *T*. In particular, $th_i(T) = i$ for all $i \in V \setminus \{0\}$ such that $(0,i) \in T$. Inversely, any spanning tree of *G* determines univocally a ZIO policy. From this point, we denote by $x_i(T)$ and $I_i(T)$, the production and inventory at period *i* determined by a tree *T*, for all $i = 1,...,n$. For example,

$$x_i(T) = \begin{cases} d_{ipt_i(T)-1} & \text{if } (0,i) \in T \\ 0 & \text{if } (0,i) \notin T \end{cases} \text{ and } I_i(T) = \begin{cases} d_{i+1\,pt_i(T)-1} & \text{if } (i-1,i) \in T \\ 0 & \text{if } (i-1,i) \notin T \end{cases} \text{ for all } i = 1,...,n$$

Therefore, a ZIO policy for the ELS problem is completely determined by a tree *T* and the accumulated demands. Distance labels of the nodes corresponding to a tree *T* are also obtained by setting $\pi_0(T) = 0$ and $\pi_i(T) = p_{th_i(T)} + \sum_{j=th_i(T)}^{i-1} h_i$, $\forall i \in V - \{0\}$ and can be calculated in O(*n*) time. Thus, given a tree *T*, we define the reduced cost $\bar{c}_{ij}(T) = c_{ij} + \pi_i(T) - \pi_j(T)$, $\forall (i,j) \in A$. In the rest of the paper, we refer to a directed out-spanning tree as tree. Denote by $C(T) = C(x(T), I(T))$ the value of the objective function of the ELS problem associated with the tree *T*.

The *K* best ZIO policies problem consists in determining the *K* best ZIO solutions of the ELS problem. In other words, identifying the *K* best trees T^k with $k \in \{1,...,K\}$ such that $C(T^1) \le C(T^2) \le ... \le C(T^K)$ and for any other tree $T^p \ne T^k$ with $k \in \{1,...,K\}$ holds $C(T^p) \ge C(T^K)$. Note that we are restricted to trees *T* determined by the previously introduce process of building, that is, any production arc $(0,i) \in T$ with $i > 1$ implies $x_i(T) > 0$.

2. Determining a second best ZIO policy.

In this section, we introduce and prove the basic results to the efficient resolution of the second best ZIO problem. First, we introduce the following definitions.

Definition 1. Two tree *T* and T' are adjacent if and only if both have $n-2$ arcs in common, that is, both trees differ in only one arc.

The above definition implies that a tree T' is obtained from a tree T by a T-exchange where the entering arc is just the arc $(i, j) \in T' \setminus T$ and $(p,q) \in T \setminus T'$ is the leaving arc. The kinds of possible T-exchange are:

(1) A production arc (0, *i*) with $i > 1$ is added to *T*. Then, the inventory arc (*i*-1, *i*) is deleted from *T* to obtain a valid tree. In this case, the production and inventory values of the arcs in the cycle $T \cup (0,i)$ are updated as follows $x_i(T') = I_{i-1}(T)$, $x_{th_i(T)}(T') = x_{th_i(T)}(T) - I_{i-1}(T)$ and $I_j(T') = I_j(T) - I_{i-1}(T)$ for $j = th_i(T),...,i-1$. Moreover, $C(T') = C(T) + \overline{c}_{oi}(T)I_{i-1}(T) + f_i$. We only consider this T-exchange when $I_{i-1}(T) > 0$. When $I_{i-1}(T) = 0$, we will obtain a tree T' with $x_i(T') = 0$, that is, the values of variables do not change.

(2) An inventory arc (*i*-1, *i*) with $i > 1$ is added to *T*. Then, the production arc (0, *i*) is deleted from *T* to obtain a valid tree. In this case, the production and inventory values of the arcs in the cycle $T \cup (i-1,i)$ are updated as follows $x_i(T') = 0$, $x_{th_{i-1}(T)}(T') = x_{th_{i-1}(T)}(T) + x_i(T)$ and $I_j(T') = I_j(T) + x_i(T)$ for $j = th_{i-1}(T),...,i-1$. Moreover, $C(T') = C(T) + \overline{c}_{i\text{-}1i}(T)x_i(T) - f_i$ (note that in the referred trees always $x_i(T) > 0$ when $i > 1$).

Once a T-exchange is performed, the distance labels in T' are updated in the following way: $\pi_k(T') = \pi_k(T) + \overline{c}_{ij}(T)$, for any node *k* that is a descendant node of node *j* in *T*. Moreover, let *T* and T' be two trees that differ in $p < n$ arcs. We use the term *multiple* T-*exchange* for the operation where $p < n$ arcs are entered simultaneously in tree *T* (see Sedeño-Noda and González-Martín (2006)).

We now need to obtain the tree T' with the smallest objective value $C(T') \geq C(T)$, from an optimal tree *T*. Therefore, we must investigate which multiple T-exchange leads to the smallest increase in the objective function of the ELS problem.

First, we consider a single T-exchange with an arc $(u,l) \in A \setminus T$ is add to *T*. If (u,l) is a production arc $(0,i)$ and $I_{i-1}(T) > 0$, then the increase $C(T') - C(T) = \overline{c}_{oi}(T)I_{i-1}(T) + f_i \geq 0$ since T is optimal. If (u,l) is an inventory arc $(i-1,i)$, then the increase $C(T') - C(T) = \overline{c}_{i\text{-}1i}(T)x_i(T) - f_i \geq 0$ since *T* is optimal.

Lemma 1. *Let T be a tree representing an optimal ZIO policy, then* $\overline{c}_{i-1i}(T) \geq 0$ *for all inventory arc* $(i-1,i) \in A$.

Proof. Since T is optimal, $\overline{c}_{i\text{-}1i}(T)x_i(T) - f_i \geq 0$ for all inventory arc $(i-1,i) \in A \setminus T$. Therefore, $\overline{c}_{i-1i}(T) \geq 0$ since $x_i(T) > 0$ and $f_i \geq 0$. Clearly, $\overline{c}_{i-1i}(T) = 0$ for all inventory arc $(i-1,i) \in T$. □

Now, we consider a multiple T-exchange with only two production arcs then, that is, we have $T \cup \{(0,i),(0,j)\}$ with $1 < i < j \leq n$. We suppose without lost of generality that $I_{i-1}(T) > 0$ and $I_{j-1}(T) > 0$. Then, theses sub-cases must be considered: A) if $th_i(T) \neq th_j(T)$ (that is, the path from node 0 to node *i* in *T* is not included in the path from node 0 to *j* in *T*), then $C(T') = C(T) + \overline{c}_{0i}(T)I_{i-1}(T) + f_i + \overline{c}_{0j}(T)I_{j-1}(T) + f_j$; B) if $th_i(T) = th_j(T)$ then $C(T') = C(T) + \overline{c}_{0i}(T)(I_{i-1}(T) - I_{j-1}(T)) + f_i + \overline{c}_{0j}(T)I_{j-1}(T) + f_j$. Thus, we can conclude that:

Lemma 2. *Let T be a tree representing an ZIO policy where $\overline{c}_{oi}(T)I_{i-1}(T) + f_i \geq 0$ for all production arc $(0,i) \in A \setminus T$. Then any tree obtained from T by a multiple T-exchange with two or more production arcs has an objective function value greater than or equal to the objective function value of at least one of the trees obtained by a single T-exchange with only one of these production arcs.*

Proof. Note that the conditions are that $\overline{c}_{oi}(T)I_{i-1}(T) + f_i \geq 0$ for all production arcs $(0,i) \in A \setminus T$. First, consider the case of two productions arcs $\{(0,i),(0,j)\}$ are added to *T*. If $th_i(T) \neq th_j(T)$, then $C(T') = C(T) + \overline{c}_{0i}(T)I_{i-1}(T) + f_i + \overline{c}_{0j}(T)I_{j-1}(T) + f_j$. It is clear that $C(T') \geq C(T) + \min\{\overline{c}_{0i}(T)I_{i-1}(T) + f_i, \overline{c}_{0j}(T)I_{j-1}(T) + f_j\}$. If $th_i(T) = th_j(T)$, then $C(T') = C(T) + \overline{c}_{0i}(T)(I_{i-1}(T) - I_{j-1}(T)) + f_i + \overline{c}_{0j}(T)I_{j-1}(T) + f_j$. In this case, we have $C(T') \geq C(T) + \overline{c}_{0j}(T)I_{j-1}(T) + f_j$ when $\overline{c}_{oi}(T) \geq 0$. When $\overline{c}_{oi}(T) < 0$, we have $(\overline{c}_{oj}(T) - \overline{c}_{oi}(T)) \geq \overline{c}_{oj}(T)$ and therefore, $C(T') \geq C(T) + \overline{c}_{0i}(T)I_{i-1}(T) + f_i + \overline{c}_{0j}(T)I_{j-1}(T) + f_j$, that is, $C(T') \geq C(T) + \min\{\overline{c}_{0i}(T)I_{i-1}(T) + f_i, \overline{c}_{0j}(T)I_{j-1}(T) + f_j\}$. In any case, $C(T')$ is greater than or equal to the function objective value of a tree obtained by a single T-exchange with the production arc determining the minimum value in $\{\overline{c}_{0i}(T)I_{i-1}(T) + f_i, \overline{c}_{0j}(T)I_{j-1}(T) + f_j\}$. It is clear that by induction can be proved that the lemma holds. □

Lemma 2 implies that a tree obtained from T by a sequence of consecutive T-exchanges with entering production arcs has an objective function value greater than or equal to the tree obtained from T by a single T-exchange with the minimum increase production arc. Now, we consider a multiple T-exchange with only two inventory arcs, that is, we have $T \cup \{(i-1, i), (j-1, j)\}$ with $1 < i < j \le n$. Then, theses sub-cases must be considered: C) if $th_{j-1}(T) \ne i$ (the cycles $T \cup (i-1, i)$ and $T \cup (j-1, j)$ do not contain arcs in common), then $C(T') = C(T) + \overline{c}_{i-1i}(T)x_i(T) - f_i + \overline{c}_{j-1j}(T)x_j(T) - f_j$; D) if $th_i(T) = th_j(T)$ then $C(T') = C(T) + \overline{c}_{i-1i}(T)(x_i(T) + x_j(T)) - f_i + \overline{c}_{j-1j}(T)x_j(T) - f_j$. Thus, we can conclude that:

Lemma 3. *Let T be a tree representing an ZIO policy where* $\overline{c}_{i-1}(T)x_i(T) - f_i \ge 0$ and $\overline{c}_{i-1i}(T) \ge 0$ *for all inventory arc* $(i-1, i) \in A \setminus T$. *Then any tree obtained from T by a multiple T-exchange with two or more inventory arcs has an objective function value greater than or equal to the objective function value of at least one of the trees obtained by a single T-exchange with only one of these inventory arcs*.

Proof. In similar way as proof of lemma 2.

Lemma 3 implies that a tree obtained from T by a sequence of consecutive T-exchanges with entering inventory arcs has an objective function value greater than or equal to the tree obtained from T by a single T-exchange with the minimum increase inventory arc.

Now, we consider a multiple T-exchange with a production arc $(0, i)$ and inventory arc $(j-1, j)$ with $1 < i < j \le n$, that is, we have $T \cup \{(0, i), (j-1, j)\}$. Then, theses sub-cases must be considered: E) if $pt_i(T) \ne j$ (that is, the path from node 0 to node i in T is not included in the path from node 0 to j-1 in T), then $C(T') = C(T) + \overline{c}_{0i}(T)I_{i-1}(T) + f_i + \overline{c}_{j-1j}(T)x_j(T) - f_j$; F) if $pt_i(T) = j$ then $C(T') = C(T) + \overline{c}_{0i}(T)(I_{i-1}(T) + x_j(T)) + f_i + \overline{c}_{j-1j}(T)x_j(T) - f_j$. Clearly, for case E) when T is an optimal ZIO policy, we have $C(T') \ge C(T) + \min\{\overline{c}_{0i}(T)I_{i-1}(T) + f_i, \overline{c}_{j-1j}(T)x_j(T) - f_j\}$. However, for case F) when T is an optimal ZIO policy, we have $C(T') \ge C(T) + \overline{c}_{0i}(T)I_{i-1}(T) + f_i + \overline{c}_{j-1j}(T)x_j(T) - f_j$ when $\overline{c}_{oi}(T) \ge 0$ and, therefore, $C(T') \ge C(T) + \min\{\overline{c}_{0i}(T)I_{i-1}(T) + f_i, \overline{c}_{j-1j}(T)x_j(T) - f_j\}$. When

$\overline{c}_{oi}(T) < 0$, we cannot conclude nothing a priori. In other words, it is possible that a tree obtained from *T* by a multiple T-exchange with a production arc $(0,i)$ and an inventory arc $(j-1,j)$ with $pt_i(T) = j$ and $\overline{c}_{oi}(T) < 0$ has a function objective value that is less than the objective function value of a tree obtained from *T* by a single T-exchange with the minimum increase arc among them.

Finally, we consider a multiple T-exchange with an inventory arc $(i-1,i)$ and production arc $(0,j)$ with $1 < i < j \le n$, that is, we have $T \cup \{(i-1,i),(0,j)\}$. Then, theses sub-cases must be considered: G) if $th_j(T) \ne i$ (that is, the path from node 0 to node *i* in *T* is not included in the path from node 0 to *j* in *T*), then $C(T') = C(T) + \overline{c}_{0j}(T)I_{j-1}(T) + f_j + \overline{c}_{i-1i}(T)x_i(T) - f_i$; H) if $th_j(T) = i$ then $C(T') = C(T) + \overline{c}_{i-1i}(T)(x_i(T) - I_{j-1}(T)) - f_i + \overline{c}_{0j}(T)I_{j-1}(T) + f_j$. Clearly, for case G) when *T* is an optimal ZIO policy, we have $C(T') \ge C(T) + \min\{\overline{c}_{0j}(T)I_{j-1}(T) + f_j, \overline{c}_{oi}(T)x_i(T) - f_i\}$. However, for case H) when *T* is an optimal ZIO policy, we have $C(T') \ge C(T) + \overline{c}_{i-1i}(T)x_i(T) - f_i$ when $\overline{c}_{oj}(T) \ge \overline{c}_{i-1i}(T)$ and, therefore, $C(T') \ge C(T) + \min\{\overline{c}_{0j}(T)I_{j-1}(T) + f_j, \overline{c}_{i-1i}(T)x_i(T) - f_i\}$. When $\overline{c}_{oj}(T) < \overline{c}_{i-1i}(T)$ and $I_{j-1}(T) > 0$, we cannot conclude nothing a priori. In other words, it is possible that a tree obtained from *T* by a multiple T-exchange with an inventory arc $(i-1,i)$ and a production arc $(0,j)$ with $th_j(T) = i$ and $\overline{c}_{oj}(T) < \overline{c}_{i-1i}(T)$ has a function objective value that is less than the objective function value of a tree obtained from *T* by a single T-exchange with the minimum increase arc among them.

In summary, the second best ZIO policy can be obtained from the optimal tree *T* associated with the best ZIO policy by a single T-exchange with a production arc or an inventory arc or by a multiple T-exchange that alternates production arcs with inventory arcs keeping the above relations. In order to determine this special minimum T-exchange, we define $G_i(T)$ to be the minimum increase in the optimal ZIO policy determined by *T* that can be incurred when the arc $(pred_i(T),i)$ is replaced by an entering arc considering the planning horizon from *i* to *n*. From the above comments, the following recursion holds for any *i* with $1 < i \le n$:

$$G_i(T)=\begin{cases}\min\left\{\begin{array}{l}\overline{c}_{i-1i}(T)x_i(T)-f_i,\\ \min\limits_{\substack{i<j<pt_i(T)\\ eligible_j(T)=TRUE}}\left\{\begin{array}{l}\overline{c}_{i-1i}(T)(x_i(T)-I_{j-1}(T))-f_i+G_j(T)\\ :\overline{c}_{i-1i}(T)>\overline{c}_{oj}(T)\text{ and }I_{j-1}(T)>0\end{array}\right\}\end{array}\right\} & \text{if }(0,i)\in T\\ \min\left\{\begin{array}{l}\overline{c}_{0i}(T)I_{i-1}(T)+f_i:I_{i-1}(T)>0,\\ \min\limits_{\substack{\overline{c}_{0i}(T)<0\\ eligible_{pt_i(T)}(T)=TRUE}}\left\{\begin{array}{l}\overline{c}_{0i}(T)(I_{i-1}(T)+x_{pt_i(T)}(T))+f_i+\overline{c}_{pt_i(T)-1pt_i(T)}(T)x_{pt_i(T)}(T)-f_{pt_i(T)},\\ \min\limits_{\substack{pt_i(T)<j<pt_{pt_i(T)}(T)\\ eligible_j(T)=TRUE}}\left\{\begin{array}{l}\overline{c}_{0i}(T)(I_{i-1}(T)+x_{pt_i(T)}(T)-I_{j-1}(T))+f_i+\\ \overline{c}_{pt_i(T)-1pt_i(T)}(T)(x_{pt_i(T)}(T)-I_{j-1}(T))-f_{pt_i(T)}+G_j(T)\\ :\overline{c}_{pt_i(T)-1pt_i(T)}(T)>\overline{c}_{oj}(T)\text{ and }I_{j-1}(T)>0\end{array}\right\}\end{array}\right\}\end{array}\right\} & \\ & \text{if }(i-1,i)\in T\end{cases}$$

The above recursion uses a flags *eligible*(T) that will be used later. For the moment, $eligible_i(T)$ = TRUE for all $i\in\{1,...,n\}$. Thus, the recursion formula states that if $(0,i)\in T$, we replace this arc with an inventory arc $(i-1,i)\notin T$. Moreover, note that in a sequence of T-exchanges alternating inventory arcs with production arcs, if the level of an inventory arc reaches a value with expression like $x_i(T)-I_{j-1}(T)$ (with $th_j(T)=i$), the next decision is to produce in the period j. Note that in this case, the arc $(j-1,j)$ belongs to *T*. For this reason in the above recursion formula, $G_i(T)$ has relation with $G_j(T)$ where $(j-1,j)\in T$, that is, the first decision for $G_j(T)$ is to produce in period *j*. If $(i-1,i)\in T$, then we replace this arc with an production arc $(0,i)\notin T$ or adding the arcs $\{(0,i),(pt_i(T)-1,pt_i(T))\}$ to T or, additionally we produce in a period *j* such that $th_j(T)=pt_i(T)$ (the relation with $G_j(T)$). Clearly, from the above recursion formula, the next result holds:

Theorem 1. *Given a best ZIO policy by a tree T, the second best ZIO policy is obtained from T by a multiple T-exchange that is the argument of the minimum of the values of* $G_i(T)$ *with* $1<i\le n$.

Denote by $pa(T)$ the number of production arcs in *T*. In similar way, $ia(T)$ is the number of inventory arcs in *T*. Note that to compute $G_i(T)$ involves $pt_i(T)-i$ comparisons when $(0,i)\in T$ and $1+pt_{pt_i(T)}(T)-pt_i(T)$ when $(i-1,i)\in T$ with $1<i\le n$. That is,

$\sum_{(0,i)\in T} pt_i(T) - i \le n-1$ and $\sum_{(i-1,i)\in T} 1 + pt_{pt_i(T)}(T) - pt_i(T) \le ia^2(T) / pa(T) \le n^2$. In other words, the second best ZIO policy is computed in the very worst case in O(n^2) time.

In order to identify the multiple T-exchange, we associated the period $next_i$ with each period $i \in \{2,...,n\}$ where $next_i = j$ when $G_i(T) = cte + G_{next_i}(T)$; otherwise $next_i = NULL$. Let $i^* = \arg\min_{1<i<n} \{G_i(T)\}$ then, the second best ZIO policy is determine by calling the next procedure with $k = i^*$.

Procedure (BSZ) BuildingSecondZIO(k, $next$, var $pred(T)$);	
(1)	**While** ($k \neq$ NULL) **do**
(2)	**If** ($pred_k(T) == 0$) **then** $pred_k(T) = k-1$;
(3)	**Else**
(4)	$pred_k(T) = 0$;
(5)	**If** ($next_k \neq$ NULL) **then** $pred_{pt_k(T)}(T) = pt_k(T) - 1$;
(6)	$k = next_k$;

Note that the BSZ procedure uses *pred*(*T*) labels instead a set of arcs *T* to determine the second best ZIO policy. Given a tree *T*, let $A(T)$ be a subset of arcs of *A**T* such that *T* is a best ZIO policy considering the set of production and inventory arcs in $T \cup A(T)$. In particular, the inventory arcs in $T \cup A(T)$ satisfy lemma 1. Then, we obtain the next result.

Lemma 4. *Given a tree T and a set of non-tree arcs* $A(T)$ *such that T is a best ZIO policy in* $T \cup A(T)$. *Let* T' *be the second best ZIO policy obtained from T making a multiple T-exchange with the set of arcs determined by* $G_{i^*}(T)$. *Then any inventory arc in the set* $A(T') = A(T) \cup T \setminus T' - \{(pred_{i^*}(T), i^*)\}$ *has a non-negative reduced cost with respect to the tree* T'.

Proof. Since T' is a second best ZIO policy in $T \cup A(T)$ then, T' is a best ZIO policy in $T' \cup A(T')$ where $A(T') = A(T) \cup T \setminus T' - \{(pred_{i^*}(T), i^*)\}$ and therefore, lemma 1 holds. □

3. A *K* best ZIO policies algorithm.

We are interested in generating the *K* best ZIO policies in order without repeating the calculation of the same best solution. In this case, the problem consist in determine de *K* trees associated with *K* best ZIO policies. To do so, given a tree *T* representing the best ZIO policy instead to store the set of arcs $A(T)$ for a given tree *T*, we maintain an Boolean label named $eligible_i(T)$ for each node $i \in V$. $eligible_i(T)$ is FALSE if and only if the arc $(pred_i(T), i)$ cannot be chosen to leave the tree *T* (equivalently, no arc arriving at node *i* can be selected to enter into the tree *T*). Otherwise, $eligible_i(T)$ is TRUE. For example, once the second best ZIO policy T' is determined from *T* and by $G_{i^*}(T)$, we set $eligible_{i^*}(T)=$ FALSE and $eligible_{i^*}(T')=$ FALSE. Therefore, any tree obtained from T' subsequently contains the arc $(pred_{i^*}(T'), i^*)$. In addition, any tree obtained from *T* contains the arc $(pred_{i^*}(T), i^*)$ (binary partition strategy). From these comments, it is clear that the determination of the same tree is not performed twice or more times. Thus, using the above notation, the determination of the second best ZIO policy for a given tree *T* and the labels $eligible(T)$ consists in applying the recursion given in previous section for any node *i* with $1 < i \le n$ and $eligible_i(T) = TRUE$.

Let $i^* = \arg\min_{1<i<n} \left\{ G_i(T) : eligible_i(T) = TRUE \right\}$ then, the second best ZIO policy is determined in O(n^2) time as already has been commented. Thus, let us assume that the first *k* trees $T^{k'}$, $k' \in \{1,...,k\}$, have been calculated. Clearly, the $(k'+1)th$ best ZIO policy is the best solution among the second best ZIO policies that can be obtained from each one of the trees $T^{k'}$ with $k' \in \{1,...,k\}$. For each calculated tree T^p in the algorithm, the node $i^* = \arg\min_{1<i<n} \left\{ G_i(T^p) : eligible_i(T^p) = TRUE \right\}$ is calculated and stored and the second best ZIO policy is stored together with the index *p* indicating the associated *p*th tree in a heap using as key the value $G_{i^*}(T) + C(T)$. We denote this heap by *H* in the algorithm.

In the algorithm, we store the *pred*(*T*) labels instead to store *T*. The main algorithm starts with an optimal tree $T^1 = T^*$ storing $pred(T^1)$ as the first best tree. The flags $eligible_i(T^1) = TRUE$ for each node $i \in \{1,...,n\}$. The index of the number of best solutions determined *k* is set to 1. Then, the algorithm computes all labels associated with the ZIO policy T^1. In other words, $C(T)$, $x_i(T)$, $I_i(T)$ and $\pi_i(T)$ for all node *i* are calculated instead

K Best ZIO Policies (KBZP) **Algorithm**;

/* Initialization */

Let $T^1 = T^*$ be an optimal tree (ZIO policy) and store $pred(T^1)$;

Set $k = 1$; $eligible_i(T^1) = TRUE \;\; \forall i \in \{1,\ldots,n\}$;

Compute $C(T^1)$, $\pi_i(T^1)$, $x_i(T^1)$ and $I_i(T^1)$ $\forall i \in \{1,\ldots,n\}$;

Create Heap H;

Let $i^* = \arg\min_{1<i<n} \{G_i(T^1) : eligible_i(T^1) = TRUE\}$;

If ($i^* \neq$ NULL) **then**

    $pred(T) = pred(T^1)$;

    BuildingSecondZIO(i^*, *next*, *pred*(*T*));

    Insert { i^*, *pred*(*T*), *k*, $G_{i^*}(T^1) + C(T^1)$ } in *H*;

While (($k < K$) **and** ($H \neq \emptyset$))

    Extract first { i^*, *pred*(*T*), *p* ,*C*} of *H;*

    Set $k = k + 1$; $pred(T^k) = pred(T)$;

    $eligible_{i^*}(T^p) = FALSE$; $eligible_i(T^k) = eligible_i(T^p)$ $\forall i \in \{1,\ldots,n\}$;

    Compute $C(T^p)$, $\pi_i(T^p)$, $x_i(T^p)$ and $I_i(T^p)$ $\forall i \in \{1,\ldots,n\}$;

    Let $i^* = \arg\min_{1<i<n} \{G_i(T^p) : eligible_i(T^p) = TRUE\}$;

    If ($i^* \neq$ NULL) **then**

        $pred(T) = pred(T^p)$;

        BuildingSecondZIO(i^*, *next*, *pred*(*T*));

        Insert { i^*, *pred*(*T*), *k*, $G_{i^*}(T^p) + C(T^p)$ } in *H*;

    Compute $C(T^k)$, $\pi_i(T^k)$, $x_i(T^k)$ and $I_i(T^k)$ $\forall i \in \{1,\ldots,n\}$;

    Let $i^* = \arg\min_{1<i<n} \{G_i(T^k) : eligible_i(T^k) = TRUE\}$;

    If ($i^* \neq$ NULL) **then**

        $pred(T) = pred(T^k)$;

        BuildingSecondZIO(i^*, *next*, *pred*(*T*));

        Insert { i^*, *pred*(*T*), *k*, $G_{i^*}(T^k) + C(T^k)$ } in *H*;

/* end of the loop */

to be stored for a given tree T . The heap H is created and the element { i^*, *pred*(*T*), 1, $G_{i^*}(T^1) + C(T^1)$ } is inserted in H wherever node i^* exists. Note that $i^* = \arg\min_{1<i<n} \{G_i(T^1) : eligible_i(T^1) = TRUE\}$ and the BuildingSecondZIO procedure determines the second best ZIO policy *pred*(*T*) for a given i^*. Then, the algorithm starts with a loop until the K best solutions are identified or no more feasible solutions are possible. Thus, in any

iteration in the algorithm, the first element in the heap is extracted. This element identifies the (k+1)th best solution by the *pred*(T) labels (line (12)). Now, the algorithm sets $eligible_{i^*}(T^p) = FALSE$ and $eligible_i(T^k) = eligible_i(T^p)$ for all $i \in \{1,...,n\}$ leading with the binary partition scheme. Then, the labels and variables associated with the tree T^p are calculated. The second best ZIO policy for T^p considering the *eligible*(T^p) flags is determined and stored (if it exists). The same operation is made for the tree T^k.

Theorem 2. *The KBZP algorithm computes the K Best ZIO policies in* O(Kn^2) *time and O(*Kn*) space in a planning horizon of n periods*.

Proof. In the beginning of the algorithm, the determination of $T^1 = T^*$ requires O($n \log n$) time (see Aggarwal and Park (1990), Federgruen and Tzur (1991), Wagelmans aet al. (1992)). Storing $pred(T^1)$ and making all nodes eligible needs O(n) time. Lines (3) involves an O(n) time. The calculation of $i^* = \arg\min_{1<i<n}\{G_i(T^1) : eligible_i(T^1) = TRUE\}$ requires an effort O(n^2) and the determination of the best second solution (lines (6)-(8)) needs O(n) time. The operations of create (line (4)) and insert (line (9)) in the heap take O(1) time. Clearly, the algorithm makes at most K iterations. In each iteration of the algorithm, two second best solutions are determined in O(n^2) time overall. Also, in each iteration, the algorithm makes one extract first heap and two insert heap operations in O(log k + log k +log (k+1)) time using a binary heap. The operations relative to lines (13)-(14), (20) are made in O(n) time. Thus, the worst case complexity of the algorithm is O($Kn^2 + K\log K$ +) time and, since $K < 2^n$, then O(Kn^2) time. On the other hand, the space required by the algorithm is O(Kn), since for each calculated tree T , the labels *pred*(T) and *eligible*(T) are stored and there are in the *heap* at most K trees. □

4. Conclusions.

From this paper, we conclude that the second best ZIO policy can be determined in O(n^2) by dynamic programming. From this result, we design an algorithm to find the K best ZIO policies in O(Kn^2) time and O(Kn) space. This method has application for the ELS problem with additional constraints. For example, we can use this approach to solve the ELS problem with a limited number of production periods, that is, where we add to the mathematical formulation of the ELS problem a new constraint like (6):

$$\text{Minimize} \quad C(x,I) = \sum_{i=1}^{n} (p_i x_i + h_i I_i + f_i y_i) \qquad (1)$$

subject to

$$x_i + I_{i-1} - I_i = d_i, \qquad \text{for } i = 1,\ldots,n \qquad (2)$$

$$d_{in} y_i - x_i \geq 0, \qquad \text{for } i = 1,\ldots,n \qquad (3)$$

$$I_0 = I_n = 0 \qquad (4)$$

$$x_i \geq 0,\ I_i \geq 0,\ y_i \in \{0,1\} \quad \text{for } i = 1,\ldots,n \qquad (5)$$

$$\sum_{i=1}^{n} y_i \leq R \qquad (6)$$

It is easy to prove that an optimal solution of the above problem satisfying the ZIO property exists. Therefore, we can compute the best ZIO policies in order for the ELS problem without constraint (6) until a ZIO policy satisfying the added constraint is found.

In the case of the ELS problem with linear cost, that is, when the setup cost are negligible, the algorithm given in Sedeño-Noda and González-Martín (2010) solves the *K* best ZIO policies problem in O(*Kn*) time using O(*K*+ *n*) space.

Acknowledgments.

This work has been partially supported by Spanish Government Research Project MTM2009-08830.

References.

Ahuja, R., Magnanti, T., Orlin, J. B., 1993. *Network Flows*. Prentice-Hall, inc.

Aggarwal, A., Park, J. K., 1990. Improved algorithms for Economic Lot-Size problems. Working Paper, Laboratory for Computer Science, MIT, Cambridge Mass.

Federgruen, A., Tzur, M., 1990. A simple forward algorithm to solve general Dynamic Lot Sizing Models with n periods in O(*n*log*n*) or O(*n*) time. *Management Science 37, 909-925.*

Sedeño-Noda, A., González-Martín, C., 2006. Shortest Path Simplex Algorithm with a Multiple Pivot Rule. *Tecnival Report nº 2, Departamento de Estadística, Investigación Operativa y Computación* (to appear in *Asia-Pacific Journal of Operational Research*).

Sedeño-Noda, A., González-Martín, C., 2010. On the K shortest path trees. *European Journal of Operational Research* 202 (3), 628-635.

Van Hoesel, S. P. M., Wagelmans, A. P. M., 1993. Sensitivity analysis of the Economic Lot-Sizing problem. *Discrete Applied Mathematics*, 45, 291-312.

Wagelmans, A. P. M., Van Hoesel, S. P. M., Kolen, A., 1992. Economic Lot Sizing: an O(*n*log*n*) Algorithm That Runs in Linear Time in the Wagner-Whitin Case. *Operations Research* 40 (1), 145-156.

Wagner, H. M., Whitin, T., 1958. Dynamic version of the economic lot size model. *Management Science* 5, 89-96.

Wagner, H. M., 1960. A Postcript to Dynamic Problems in the Theory of the Firm. *Naval Research Logistics Quarterly* 7, 7-12.

Zangwill, W. I., 1966. A Deterministic Multi-period Production Scheduling Model with Backlogging. *Management Science* 13,105-119.