



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

European Journal of Operational Research xxx (2003) xxx–xxx

EUROPEAN
JOURNAL
OF OPERATIONAL
RESEARCHwww.elsevier.com/locate/dsw

The multiscenario lot size problem with concave costs

J. Gutiérrez^{a,*}, J. Puerto^b, J. Sicilia^a

^a *Dpto. de Estadística, Investigación Operativa y Computación, Universidad de La Laguna, Tenerife, La Laguna 38271, Spain*

^b *Dpto. de Estadística e Investigación Operativa, Universidad de Sevilla, Spain*

Received 10 October 2002; accepted 11 December 2002

Abstract

The dynamic single-facility single-item lot size problem is addressed. The finite planning horizon is divided into several time periods. Although the total demand is assumed to be a fixed value, the distribution of this demand among the different periods is unknown. Therefore, for each period the demand can be chosen from a discrete set of values. For this reason, all the combinations of the demand vector yield a set of different scenarios. Moreover, we assume that the production/reorder and holding cost vectors can vary from one scenario to another. For each scenario, we consider as the objective function the sum of the production/reorder and the holding costs. The problem consists of determining all the Pareto-optimal or non-dominated production plans with respect to all scenarios. We propose a solution method based on a multiobjective branch and bound approach. Depending on whether shortages are considered or not, different upper bound sets are provided. Computational results on several randomly generated problems are reported.

© 2002 Published by Elsevier Science B.V.

Keywords: Scenarios; Inventory; Multiple objective programming

1. Introduction

Since the late 1950s, special attention has been paid to the dynamic lot sizing problems. The interest lies in the fact that these models fit a great number of real world problems. Wagner and Whitin [24], and independently, Manne [9] pioneered this field. They assumed a multiperiod planning horizon with known demand, and proposed a procedure which is based on both the dynamic programming approach and the zero inventory order (ZIO) property. This property states that, among all those optimal plans, there exists at least one, in which for each period, the product between the stock level and the production/reorder must be equal to zero. This cost-minimizing production/reorder schedule has interesting qualitative features. The extension to backlogging was studied by Zangwill [25,27] and Manne and Veinott [10]. Also, Veinott [20] introduced the case with convex costs.

Unlike the original dynamic lot size problem [24], where the demands through the whole horizon are known, in this paper we consider that the demand vector is unknown rather than the total demand, which is

* Corresponding author. Tel.: +34-922-31-91-89; fax: +34-922-31-92-02.
E-mail address: jmgrrrez@ull.es (J. Gutiérrez).

31 assumed to be a fixed value. Furthermore, for each period, the demand can be chosen from a discrete finite
32 set. As a result, different scenarios can arise combining the different admissible values of the demand per
33 period. One of the most common examples for this problem are the promotions to clear stock. In this case,
34 although we know in advance the total number of items to be sold we can not determine an optimal reorder
35 plan because it is impossible to know with certainty how the demand is to occur period per period. Another
36 instance happens when a wholesaler of bricks should satisfy the demands for distinct builders. Despite the
37 wholesaler may know in advance the total demand of bricks needed to carry out the different constructions,
38 he does not know how this total demand is distributed through the planning horizon. However, the decision
39 maker can assume that the demand per period is taken from a discrete finite set. Besides, we allow in our
40 model that the production/reorder and holding cost vectors change from one scenario to another. Taking
41 into account these assumptions, the decision maker can not predict what scenario is to occur. Therefore,
42 this problem concerns with the optimization under uncertainty and, it takes place when a firm has to make
43 a decision under variable market conditions. In fact, the uncertainty is present up to a point in almost all
44 the decisions made in the real world.

45 How to handle the uncertainty in the scenario occurrence is not easy at all. One may want to come up
46 with a unique solution using conservative techniques or the principle of incomplete reason (utilities). On the
47 other hand, one may want to obtain the whole range of solutions that are non-dominated component-wise,
48 as a first step in the analysis of the problem, in order to shed light on the decision process. This set can be
49 seen as a sensitivity analysis of the admissible solutions of the scenario problem for any 'a priori' infor-
50 mation on the occurrence of the scenarios. The former analysis is normative: it prescribes a concrete course
51 of action (based on a utility), the latter is descriptive: it informs on the variability of the solution space.
52 Both analyses have advantages and disadvantages. The final decision should be made according to the goals
53 of the decision-maker. Notice that our goal in this paper is to study the second approach. It is worth re-
54 marking that similar analysis has been followed for other scenario problems in the recent literature of
55 operations research (see for instance [4,5,13,16]).

56 Dantzig [7] mentions the importance of considering uncertainty in the systems. In this sense, the so-
57 called scenario analysis has been developed to deal with the problem of the uncertainty. Assuming that all
58 the different situations of the system can be identified, this approach calculates the non-dominated solu-
59 tions. These solutions are robust with respect to any possible occurrence because they are non-dominated,
60 component-wise, by any other. Therefore, the approach consists of obtaining the Pareto-optimal solution
61 set.

62 This article is devoted to the problem of determining the Pareto-optimal policies for the multiscenario
63 dynamic lot sizing problem. For each scenario, we assume a planning horizon split into N periods. Three N -
64 tuple vectors represent the input data for each scenario: a deterministic demand vector, the carrying cost
65 vector and the replenishment cost vector. Also, in the backlogging case, a shortage cost vector is consid-
66 ered. As usual, the overall cost function consists of the sum of carrying and replenishment costs. The goal is
67 to schedule production/reorder in the various periods of each scenario so as to satisfy demand at minimal
68 cost simultaneously in all the scenarios.

69 The problem introduced in this paper fits into the multiobjective combinatorial optimization (MOCO).
70 MOCO problems are an emergent area of research in many fields of operations research (see e.g. [6,19]).
71 Nowadays, MOCO (see [3,19]) provides an adequate framework to tackle various types of discrete mul-
72 ticriteria problems. Within this research area, several methods are known to handle different problems. Two
73 of them are dynamic programming enumeration (see [22] for a methodological description and Klamroth
74 and Wiecek [8] for a recent application to knapsack problems) and implicit enumeration [15,28,29]. In
75 particular, the branch and bound scheme corresponds to an implicit enumeration method and, although it
76 is widely used in the single objective case, only a few papers apply this technique for MOCO since bounds
77 may be difficult to compute (see, e.g. [1,14,21]). The reader is referred to [3] for a complete survey of MOCO
78 methods).

79 It is worth noting that most of MOCO problems are *NP*-hard and intractable. In most cases, even if the
 80 single objective problem is polynomially solvable the multiobjective version becomes *NP*-hard. This is the
 81 case of spanning tree problems and min-cost flow problems, among others. As we have mentioned, an
 82 important tool to deal with these problems is the multicriteria dynamic programming (MDP) [3]. In the
 83 single objective case Morin and Esoboque [11] exploited the embedded-state recursive equations to over-
 84 come many of the problems caused by the curse of the dimensionality (see, for example, [2,12]). As an
 85 extension of the previous result, Villarreal and Karwan [22] introduced a procedure based on the dynamic
 86 multicriteria discrete mathematical programming (DMDMP) to generate the Pareto-optimal solution set
 87 for problems with more than one objective function. We will make use of these techniques to resolve our
 88 model. In this context, when time and efficiency become a real issue, different alternatives can be used to
 89 approximate the Pareto-optimal set. One of them is the use of general-purpose MOCO heuristics [6].
 90 Another possibility is the design of ‘ad hoc’ methods based on computing the extreme non-dominated
 91 solutions. Obviously, this last strategy does not guarantee that we obtain the whole set of non-dominated
 92 solutions. Nevertheless the reduction in computation time can be remarkable.

93 The rest of this paper is organized as follows. Section 2 introduces the notation and the model. In Section
 94 3, we show that when the objective function is concave and shortages are not allowed, the extreme points of
 95 the region of feasible production plans satisfy a modified version of ZIO property, and that the Pareto-
 96 optimal set will always contain modified ZIO solutions. Therefore, we propose an algorithm to compute
 97 this approximated solution set: the non-dominated modified ZIO policies. A subset of such policies will be
 98 used later as initial upper bound set in the general algorithm. Furthermore, in Section 4, when shortages are
 99 allowed, we show that the polyhedron extreme points hold a modified version of the property for the single
 100 scenario case. Again, a subset of the non-dominated policies satisfying the latter property are proposed as
 101 the initial upper bound set for the algorithm when shortages are allowed. In Section 5, we propose a MDP
 102 that solves the problem and a branch and bound scheme to reduce the computational burden of the above
 103 MDP. Also, in Section 6, computational results are reported for a set of dynamic multisenario lot size
 104 problems. Finally, Section 7 contains conclusions and some further remarks.

105 2. Notation and statement of the problem

106 We consider a dynamic production/inventory system with a finite planning horizon of N periods where
 107 an external known demand must be met at minimal cost. It is assumed that M scenarios or replications of
 108 that system are to be considered simultaneously and a unique (robust) policy belonging to the Pareto-
 109 optimal set is to be implemented. These replications model uncertainty in the parameter estimation, since
 110 neither the true values of the parameters of the system nor a probability distribution over them are known
 111 before hand. Therefore, we look for compromise solutions which must behave acceptably well in any of the
 112 admissible scenarios. This sort of system represents a multiple/serial decision process, since each scenario
 113 behaves as a serial multiperiod decision system and each production/reorder decision implies a parallel
 114 decision process. A graphical representation of this process is shown in Fig. 1.

115 Throughout we use the following notation:

- 116 $h_i^j(\cdot)$ holding cost for the j th period in the i th scenario.
 117 $c_i^j(\cdot)$ production/reorder cost for the j th period in the i th scenario.
 118 I_i^j inventory on hand at the end of the j th period in the i th scenario.
 119 d_i^j the demand for the j th period in the i th scenario.
 120 D the total demand ($\sum_{j=1}^N d_i^j = \sum_{j=1}^N d_s^j$ for any i and s in $\{1, \dots, M\}$).
 121 x_j the production/reorder quantity for the j th period.

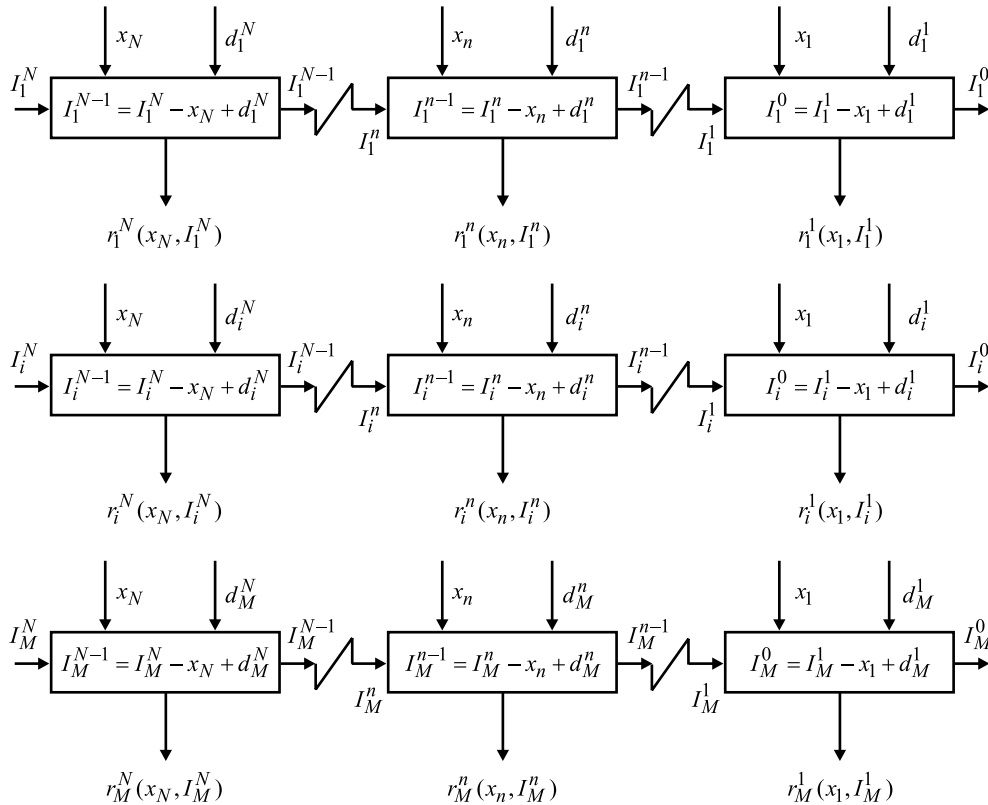


Fig. 1. The multisenario lot size problem scheme.

122 We assume, without loss of generality, that $I_i^0 = I_i^N = 0$ for $i = 1, \dots, M$.

123 The following definitions are required to simplify the formulation of the problem. Given a production/
 124 reorder vector $\mathbf{x} = (x_1, \dots, x_N) \in \mathbb{N}_0^N$, the inventory level vector for a scenario i is denoted by
 125 $I_i(\mathbf{x}) = (I_i^1, \dots, I_i^N)$, where

$$I_i^j = I_i^{j-1} + x_j - d_i^j, \quad j = 1, \dots, N. \tag{1}$$

127 In addition, the cumulative cost from period j to period k in scenario i is given by

$$R_i^{j,k}(\mathbf{x}) = \sum_{t=j}^k r_i^t(x_t, I_i^t), \tag{2}$$

129 where $r_i^t(x_t, I_i^t) = c_i^t(x_t) + h_i^t(I_i^t)$.

130 Therefore, the total cost vector $R(\mathbf{x})$ in all the scenarios for a production/reorder vector $\mathbf{x} \in \mathbb{N}_0^N$ is as
 131 follows

$$R(\mathbf{x}) = (R_1^{1,N}(\mathbf{x}), \dots, R_M^{1,N}(\mathbf{x})). \tag{3}$$

133 Then, the Pareto-optimal or non-dominated production/reorder plans set \mathcal{P} can be stated as

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{N}_0^N : \text{there is no other } \mathbf{y} \in \mathbb{N}_0^N : R(\mathbf{y}) \leq R(\mathbf{x}), \\ \text{with at least one of the inequalities being strict}\}, \tag{4}$$

135 where $R(\mathbf{y}) \leq R(\mathbf{x})$ means that $R_i^{1,N}(\mathbf{y}) \leq R_i^{1,N}(\mathbf{x})$ for $i = 1, \dots, M$.

136 Using the previous definitions, we can state the dynamic multiscenario lot size problem (DMLSP), or P
 137 for short, as follows:

$$\begin{aligned}
 (P) \quad & v - \min(R_1^{1,N}(\mathbf{x}), \dots, R_M^{1,N}(\mathbf{x})) \\
 \text{s.t. :} \quad & \\
 & I_i^0 = I_i^N = 0, & i = 1, \dots, M, \\
 & I_i^{j-1} + x_j - I_i^j = d_i^j, & j = 1, \dots, N, \quad i = 1, \dots, M, \\
 & x_j \geq 0, \quad \text{integer}, & j = 1, \dots, N, \\
 & I_i^j \geq 0, & j = 1, \dots, N, \quad i = 1, \dots, M,
 \end{aligned} \tag{5}$$

139 where $v - \min$ stands for finding the Pareto-optimal set. Thus, the goal consists of determining the Pareto-
 140 optimal solutions with respect to the M objective functions. The first constraint in P forces both the initial
 141 and the final inventory level to be zero in all the scenarios. The second constraint set concerns the well
 142 known material balance equation, and hence it states the flow conservation among periods in all the sce-
 143 narios. Production/reorder quantity must be always a non-negative integer. Finally, the last constraints set
 144 in P disallows shortages.

145 Since the single objective version for this problem can be solved using a dynamic programming algo-
 146 rithm, it seems reasonable to apply MDP for problem P . Accordingly, let $F(j, I_1^{j-1}, \dots, I_M^{j-1})$ be the set of the
 147 reachable non-dominated values, which correspond to production/reorder subplans (subpolicies) from the
 148 state $(I_1^{j-1}, \dots, I_M^{j-1})$ at period j . Since there are finitely many non-negative integers x_j that satisfy (1), the
 149 principle of optimality gives rise to the following functional equation:

$$F(j, (I_1^{j-1}, \dots, I_M^{j-1})) = v - \min_{x_j \in \mathbb{N}_0} \left\{ \begin{bmatrix} c_1^j(x_j) \\ \vdots \\ c_M^j(x_j) \end{bmatrix} + \begin{bmatrix} h_1^j(I_1^{j-1} + x_j - d_1^j) \\ \vdots \\ h_M^j(I_M^{j-1} + x_j - d_M^j) \end{bmatrix} \oplus F(j+1, (I_1^j, \dots, I_M^j)) \right\}, \tag{6}$$

151 where $A \oplus B = \{a + b : a \in A, b \in B\}$ for any two sets A, B .

152 Therefore, the set of Pareto-optimal production/reorder plans of problem P is given by the policies
 153 associated with the vectors in the set $F(1, 0, \dots, 0)$, and hence MDP algorithms give a solution for our
 154 problem. However, due to the inherent curse of the dimensionality of the MDP approach, we introduce a
 155 branch and bound scheme to decrease the running times of the solution method. For this reason, before
 156 introducing our procedure, we propose two upper bound sets to be applied in the branch and bound al-
 157 gorithm. According to Villarreal and Karwan [22], a set of upper bounds is a set of vectors such that each
 158 element is either efficient or is dominated by at least one efficient solution. Thus, the first upper bound set
 159 concerns the case without shortages and the second one represents the upper bound set for when stockouts
 160 are allowed.

161 In the next section, we propose an initial upper bound set assuming that both the carrying and the
 162 production/reorder costs are concave and stockouts are not permitted.

163 3. Case without shortages

164 In this section we assume that the cost function $R_i^{j,k}(\mathbf{x})$ is concave in \mathbf{x} for $i = 1, \dots, M, j = 1, \dots, N$ and
 165 $k \geq j$. Therefore, the following inequality holds:

$$R_i^{1,N}(\mathbf{x} + 1) - R_i^{1,N}(\mathbf{x}) \leq R_i^{1,N}(\mathbf{x}) - R_i^{1,N}(\mathbf{x} - 1), \tag{7}$$

167 where the plan $\mathbf{x} \pm 1$ differs from plan \mathbf{x} only in two periods where one unit of production/reorder is added
 168 or subtracted. In other words, let j and k be the periods (components) where the plan \mathbf{x} is to be modified,

169 then $\mathbf{x} + 1$ equals to \mathbf{x} excepting in period j where one more production/reorder unit is added and in period
 170 k where one production/reorder unit is subtracted. On the other hand, the plan $\mathbf{x} - 1$ equals to \mathbf{x} excepting
 171 in the period j in which one production/reorder unit is subtracted and in period k where one production/
 172 reorder unit is added.

173 Notice that the single objective model [24] can be formulated as a network flow problem [26]. Consid-
 174 ering concave costs, the solutions for the single objective version of this problem lie on extreme points of the
 175 feasible polyhedron. Furthermore, for each partition over the state set, there is always a representative plan
 176 satisfying that $I^{j-1}x_j = 0$ for any period j . This property is commonly known as zero inventory ordering
 177 (ZIO). Therefore, we can use a $O(N^2)$ algorithm [24] to determine the minimum cost plan via pairwise
 178 comparison.

179 We define now the ZIO property for the multiscenario case as follows: a plan \mathbf{x} is said to be ZIO for P if
 180 and only if

$$x_j \min\{I_1^{j-1}, \dots, I_M^{j-1}\} = 0 \quad \text{for } j = 1, \dots, N. \tag{8}$$

182 It is worth noting that this modification is the natural extension of the corresponding property in the
 183 scalar case. As it will be shown subsequently, efficient ZIO policies play an important role in the deter-
 184 mination of the Pareto set because they represent the set of basic solutions, namely, extreme solutions of P .
 185 For the sake of simplicity, we formulate problem P as a multicriteria network flow problem since efficient
 186 ZIO plans correspond to acyclic flows in the network as well. Accordingly, assuming non-negative concave
 187 costs, the underlying network for this problem, depicted in Fig. 2, is as follows. Let $G = (V, E)$ be a directed
 188 network, where V stands for the set of $n = (N + 2)M + 1$ nodes, and E represents the set of $m = 3MN$ edges.
 189 The nodes are classified in: production/reorder node (node 0), demand per scenario nodes $nd_s, s = 1, \dots, M$,

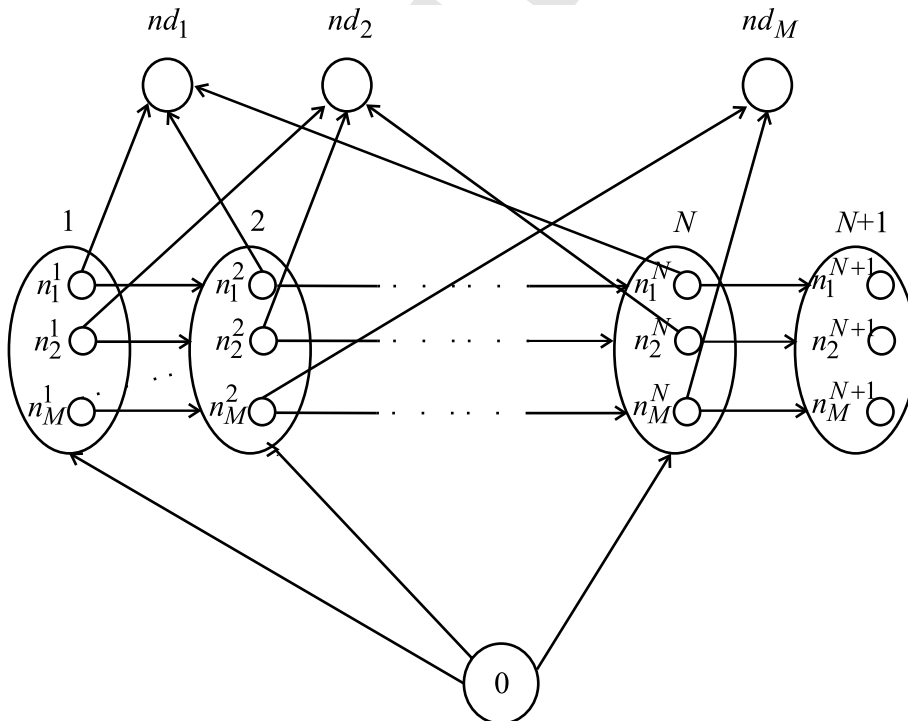


Fig. 2. The network of problem P .

	x_1	x_2	\dots	x_N	I_1^1	\dots	I_1^{N-1}	I_1^N	\dots	I_M^1	\dots	I_M^{N-1}	I_M^N
	(0,1)	(0,2)	\dots	(0,N)	(1,2)	\dots	(N-1,N)	(N,N+1)	\dots	(1,2)	\dots	(N-1,N)	(N,N+1)
0	1	1	\dots	1	0	\dots	0	0	\dots	0	\dots	0	0
1	-1	0	\dots	0	1	\dots	0	0	\dots	0	\dots	0	0
2	0	-1	\dots	0	-1	\dots	0	0	\dots	0	\dots	0	0
\dots			\dots			\dots			\dots		\dots		
N	0	0	\dots	-1	0	\dots	-1	1	\dots	0	\dots	0	0
N+1	0	0	\dots	0	0	\dots	0	-1	\dots	0	\dots	0	0
\dots			\dots			\dots			\dots		\dots		
0	1	1	\dots	1	0	\dots	0	0	\dots	0	\dots	0	0
1	-1	0	\dots	0	0	\dots	0	0	\dots	1	\dots	0	0
2	0	-1	\dots	0	0	\dots	0	0	\dots	-1	\dots	0	0
\dots			\dots			\dots			\dots		\dots		
N	0	0	\dots	-1	0	\dots	0	0	\dots	0	\dots	-1	1
N+1	0	0	\dots	0	0	\dots	0	0	\dots	0	\dots	0	-1

190 and intermediate nodes. The intermediate nodes are organized per layers. Thus, in layer j , there are M
 191 nodes denoted by $n_s^j, s = 1, \dots, M, j = 1, \dots, N + 1$.

192 There are M arcs from node 0 to each layer. The flow entering these arcs is equal. It can be seen as a
 193 single flow that is virtually multiplied M times so that the same amount is directed to each one of the nodes
 194 in this layer. These arcs can be considered as a pipeline that at a certain point is transformed into M
 195 branches. Each one of these branches receives exactly the same flow that the one that enters through the
 196 initial node of the arc. The arc from production/reorder node 0 to layer j is related to the production/re-
 197 order variable x_j in period j . The virtual multiplication of the production/reorder is because the different
 198 scenarios do not occur simultaneously in reality. Actually, only one of them is to occur, and we are con-
 199 sidering simultaneous (parallel) network flow problems with the same kind of input. The arc from 0 to n_s^j
 200 has a cost $c_s^j(\cdot), s = 1, \dots, M$ and $j = 1, \dots, N$.

201 In addition, there are also arcs from n_s^j to $n_s^{j+1}, s = 1, \dots, M$ and $j = 1, \dots, N$. Each arc in this category is
 202 an inventory arc associated to the state variables I_s^j and its cost is $h_s^j(\cdot)$. Finally, there are arcs leaving each
 203 node n_s^j towards nd_s with values $d_s^j, s = 1, \dots, M$ and $j = 1, \dots, N$.

204 We proceed now to show that non-dominated ZIO policies represent the set of extreme solutions of
 205 problem P . Previously, let us consider first the explicit representation of the multicriteria node-arc incidence
 206 matrix A of the network:

207 Notice that each block of $N + 2$ rows represents a scenario and the columns are divided in two groups:
 208 the first N columns are related to the arcs from the producer node to the N periods, and the rest of columns
 209 concern the inventory holding between two consecutive periods for each scenario. Using the above matrix A
 210 and denoting by $\mathbf{x} = (x_1, \dots, x_N)$ and $\mathbf{I} = (I_1^1, \dots, I_1^N, \dots, I_M^1, \dots, I_M^N)$ it is straightforward that we get the
 211 constraints set of problem P as follows:

$$(\mathbf{x}, \mathbf{I})A^t = -(-D, d_1^1, \dots, d_1^N, 0, \dots, -D, d_M^1, \dots, d_M^N, 0).$$

Proposition 1. *The constraint matrix A for problem P has rank $MN + 1$.*

214 **Proof.** Indeed, each block of $N + 2$ rows has one row (e.g. the last one) being linearly dependent since the
 215 sum by blocks equals zero. According to this argument, the rank is, at most, $M(N + 1)$. In addition, in the
 216 remaining matrix the row corresponding to node 0 appears M times (one per block), hence $(M - 1)$ of them
 217 could be removed resulting in a matrix with $MN + 1$ rows.

218 Now, removing the last constraint in each block and using the columns corresponding to
 219 $x_N, I_1^1, \dots, I_1^N, \dots, I_M^1, \dots, I_M^N$, a triangular matrix is obtained with elements in the diagonal equal to one.

	$(0, N)$	$(1, 2)$	\dots	$(N - 1, N)$	$(N, N + 1)$	\dots	$(1, 2)$	\dots	$(N - 1, N)$	$(N, N + 1)$	
0	1	0	\dots	0	0	\dots	0	\dots	0	0	
1	0	1	\dots	0	0	\dots	0	\dots	0	0	
2	0	-1	\dots	0	0	\dots	0	\dots	0	0	
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\ddots	\vdots	\ddots	\vdots	\vdots	
N	-1	0	\dots	-1	1	\dots	0	\dots	0	0	(9)
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\ddots	\vdots	\ddots	\vdots	\vdots	
1	0	0	\dots	0		\dots	1	\dots	0	0	
2	0	0	\dots	0		\dots	-1	\dots	0	0	
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\ddots	\vdots	\ddots	\vdots	\vdots	
N	-1	0	\dots	0	\dots	\dots	0	\dots	-1	1	

221 Therefore, since a submatrix with rank $MN + 1$ exists the result follows. \square

222 The following theorem states that the basic solutions for our problem fulfill that the demand in each
 223 period is satisfied from either the production/reorder in that period or the units carried in the inventory, but
 224 not by both simultaneously. Thus, in the underlying network of the problem, each node (excepting the
 225 production/reorder node) is attainable either from the production/reorder node or from the predecessor
 226 holding node, but never from both. Hence, the graph associated to the non-null variables of any feasible
 227 basic solution verifies for any period j : $x_j \min\{I_1^{j-1}, \dots, I_M^{j-1}\} = 0$.

228 **Theorem 2.** Any basic solution of problem P fulfills that $x_j \min\{I_1^{j-1}, \dots, I_M^{j-1}\} = 0$ for any period j ,
 229 $j = 1, \dots, N$.

230 **Proof.** Assume without loss of generality that the variables x_1, x_2 are non-null. Let us consider the columns
 231 that correspond with these variables and the inventory carrying variables from period 1 to 2, i.e. I_1^1, \dots, I_M^1 .
 232 The matrix has two columns $(0, 1)$ and $(0, 2)$, for the variables x_1 and x_2 ; and M columns, one per scenario
 233 for the I_s^1 variables $s = 1, \dots, M$.

$$\begin{bmatrix} x_1 & x_2 & I_1^1 & I_2^1 & \dots & I_M^1 \\ (0, 1) & (0, 2) & (1, 2) & (1, 2) & \dots & (1, 2) \\ + & - & + & + & \dots & + \\ 1 & 1 & 0 & 0 & \dots & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 \\ 0 & -1 & -1 & 0 & \dots & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & \dots & 0 \\ -1 & 0 & 0 & 1 & \dots & 0 \\ 0 & -1 & 0 & -1 & \dots & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & \dots & 0 \\ -1 & 0 & 0 & 0 & \dots & 1 \\ 0 & -1 & 0 & 0 & \dots & -1 \\ \dots & & & & & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

235 It is easy to see that the linear combination of columns with coefficients $+1, -1, +1, \dots, +1$ gives the null
 236 vector. Therefore, all the considered variables can not be part of any basic solution. Hence, the condition
 237 holds. \square

238 For linear cost problems this results implies that there is always a non-dominated ZIO policy. However,
 239 for general concave cost problems this results must be proven.

240 **Proposition 3.** The Pareto-optimal solution set of problem P contains, at least, one ZIO policy.

241 **Proof.** Assume that all ZIO policies are dominated. Let \mathbf{z} be a non-extreme efficient point such that \mathbf{z} makes
 242 the function $R_i^{1,N}(\cdot)$ minimal. That is, \mathbf{z} is a plan with cost smaller than or equal to the rest of non-domi-

243 nated policies in the i th scenario. We can assert that \mathbf{z} exists, otherwise, the efficient point that minimizes
 244 $R_i^{1,N}(\cdot)$ would be an extreme point and the theorem would follow. Furthermore, assume \mathbf{x} being a feasible
 245 extreme point such that the following inequality holds:

$$R_i^{1,N}(\mathbf{z}) < R_i^{1,N}(\mathbf{x}).$$

247 We can also guarantee that \mathbf{x} always can be found, otherwise, $R_i^{1,N}(\mathbf{z}) = R_i^{1,N}(\mathbf{x})$ for all the extreme points
 248 \mathbf{x} , that is, the i th component of the cost vector of \mathbf{x} equals to the minimal value for this component and \mathbf{z}
 249 could have been taken an extreme point.

250 Also, by concavity of the cost functions, the following expression must be fulfilled:

$$R_i^{1,N}(\theta\mathbf{z} + (1 - \theta)\mathbf{x}) \geq \theta R_i^{1,N}(\mathbf{z}) + (1 - \theta)R_i^{1,N}(\mathbf{x}),$$

252 where θ is a scalar that ranges in $[0, 1]$.

253 In addition, let \mathbf{p} be a point on a facet of the feasible set such that \mathbf{p} is aligned with \mathbf{z} and \mathbf{x} , and \mathbf{z} can be
 254 expressed as a convex combination of \mathbf{p} and \mathbf{x} . Hence, the following inequality holds:

$$R_i^{1,N}(\theta\mathbf{x} + (1 - \theta)\mathbf{p}) \geq \theta R_i^{1,N}(\mathbf{x}) + (1 - \theta)R_i^{1,N}(\mathbf{p}).$$

256 Since \mathbf{z} is minimal for $R_i^{1,N}(\cdot)$

$$R_i^{1,N}(\mathbf{z}) \leq R_i^{1,N}(\mathbf{p}).$$

258 Taking $\hat{\theta}$ such that $\mathbf{z} = \hat{\theta}\mathbf{x} + (1 - \hat{\theta})\mathbf{p}$, the following contradiction occurs

$$R_i^{1,N}(\hat{\theta}\mathbf{x} + (1 - \hat{\theta})\mathbf{p}) = R_i^{1,N}(\mathbf{z}) \geq \hat{\theta}R_i^{1,N}(\mathbf{x}) + (1 - \hat{\theta})R_i^{1,N}(\mathbf{p}).$$

260 Notice that $R_i^{1,N}(\mathbf{z}) < R_i^{1,N}(\mathbf{x})$ and $R_i^{1,N}(\mathbf{z}) \leq R_i^{1,N}(\mathbf{p})$, then we have that

$$R_i^{1,N}(\mathbf{z}) \geq \hat{\theta}R_i^{1,N}(\mathbf{x}) + (1 - \hat{\theta})R_i^{1,N}(\mathbf{p}) > \hat{\theta}R_i^{1,N}(\mathbf{z}) + (1 - \hat{\theta})R_i^{1,N}(\mathbf{z}) = R_i^{1,N}(\mathbf{z}).$$

262 That is, $R_i^{1,N}(\mathbf{z}) > R_i^{1,N}(\mathbf{z})$. \square

263 Since we know that there exist Pareto policies satisfying the ZIO property and the procedure in (6) that
 264 computes the complete Pareto set has a large complexity, we are now interested in determining the Pareto
 265 policies within the ZIO plans. This may be considered in some cases as an approximation to the actual
 266 Pareto set (indeed, ZIO plans coincide with extreme solutions as Theorem 2 shows). The fact is that the
 267 non-dominated ZIO policies represent an initial upper bound set to be used in the branch and bound al-
 268 gorithm.

269 In order to compute the Pareto ZIO plans, we need to introduce some notation. Let $I(j)$ denote the set of
 270 state vectors at the beginning of period j . Notice that $I(0) = I(N + 1) = (0, \dots, 0)$. In addition, let
 271 $D_i^{j,k} = \sum_{t=j}^{k-1} d_t^i$ be the accumulated demand from period j to k in scenario i and let $(I_1^{j-1}, \dots, I_M^{j-1}) \in I(j)$ be a
 272 given state vector in period j . Moreover, let us admit that there is a null component in $(I_1^{j-1}, \dots, I_M^{j-1})$, hence
 273 the decision variable x_j should be distinct to zero to prevent shortages. Thus, the feasible decisions set
 274 corresponding to a state vector $(I_1^{j-1}, \dots, I_M^{j-1})$ in period j is given by

$$\Psi(j, (I_1^{j-1}, \dots, I_M^{j-1})) = \begin{cases} 0, & \text{if } I_i^{j-1} > 0 \text{ for all } i, \\ \max_{1 \leq i \leq M} \{0, D_i^{j,k} - I_i^{j-1}\}, & \text{otherwise.} \end{cases} \quad k = j + 1, \dots, N + 1,$$

276 Assuming that $(I_1^{j-1}, \dots, I_M^{j-1})$ contains a component equal to zero, it can be easily proved that any decision
 277 $x_j \neq \max_{1 \leq i \leq M} \{0, D_i^{j,j+1} - I_i^{j-1}\}$, $l = 1, \dots, N + 1 - j$, results in a non-ZIO policy.

278 Accordingly, given a period j and an inventory vector $(I_1^{j-1}, \dots, I_M^{j-1}) \in I(j)$, the set $F(j, (I_1^{j-1}, \dots, I_M^{j-1}))$
 279 of cost vectors corresponding to Pareto ZIO subpolicies for the subproblem with initial inventory vector
 280 $(I_1^{j-1}, \dots, I_M^{j-1})$ is as follows:

$$\begin{aligned}
 F(j, (I_1^{j-1}, \dots, I_M^{j-1})) = & \min_{x_j \in \Psi(j, (I_1^{j-1}, \dots, I_M^{j-1}))} \left\{ v - \min_{x_j \in \Psi(j, (I_1^{j-1}, \dots, I_M^{j-1}))} \left[\begin{array}{l} c_1^j(x_j) \\ \vdots \\ c_M^j(x_j) \end{array} \right] + \left[\begin{array}{l} h_1^j(I_1^{j-1} + x_j - D_1^{j,j+1}) \\ \vdots \\ h_M^j(I_M^{j-1} + x_j - D_M^{j,j+1}) \end{array} \right] \oplus F(j+1, (I_1^{j-1} \right. \\
 & \left. + x_j - D_1^{j,j+1}, \dots, I_M^{j-1} + x_j - D_M^{j,j+1})) \right\}. \tag{10}
 \end{aligned}$$

282 Notice that the whole set of Pareto ZIO policies for P is determined when $F(1, (0, \dots, 0))$ is achieved.

283 **Proposition 4.** *The MDP algorithm for problem (10) runs in $O(4^N M^2)$.*

284 **Proof.** Given an initial inventory vector $(I_1^{j-1}, \dots, I_M^{j-1}) \in I(j)$, it is clear that x_j can only take values in
 285 $\Psi(j, (I_1^{j-1}, \dots, I_M^{j-1}))$ to satisfy property (8). Thus, if $I_i^{j-1} \neq 0$ for all i , the number of decisions for state
 286 $(I_1^{j-1}, \dots, I_M^{j-1})$ is at most $N - j + 1$, otherwise the unique decision is $x_j = 0$. Each different decision leads to a
 287 new state vector in the following period, hence the maximum number of states at the beginning of stage $j + 1$
 288 is $N - j + 1$ as well. Remark that the computational effort to make up the accumulated demands matrix
 289 $\bar{D}_{M \times N} = \{\bar{d}_{i,j} = D_i^{j,N+1}\}$ is $O(MN)$, and also $O(M(N - j) + 1)$ comparisons must be carried out to obtain the
 290 maximum values. Hence, the determination of $\Psi(j, (I_1^{j-1}, \dots, I_M^{j-1}))$ requires of $O(M(N - j) + 1)$ operations.

291 By virtue of the ZIO property, there are at most two vectors reaching one state in period 2 and, at most,
 292 four vectors can achieve any state in period 3. In general, in one state of period j there are at most 2^{j-1}
 293 vectors to be evaluated via pairwise comparisons. Therefore, the number of comparisons for one state of
 294 period j is given by $O((2^{j-1}(2^{j-1} - 1)/2)M)$. Accordingly, the number of comparisons in period j is
 295 $O(((2^{j-1}(2^{j-1} - 1)/2)M)(M(N - j) + 1))$. Thus, the procedure carries out $O(M \sum_{j=2}^N 2^{j-2}(2^{j-1} - 1)$
 296 $(M(N - j) + 1))$ comparisons, and hence the complexity is $O(4^N M^2)$. \square

297 As Proposition 4 states, the implicit enumeration process of the whole set of efficient ZIO policies for P
 298 requires a number of operations which grows exponentially with the input size. This is not a surprising
 299 result since the multicriteria network flow problem, which is in general NP -hard (Ruhe [17]), can be reduced
 300 to the problem we deal with.

301 From the computational point of view, the algorithm based on (10) is inefficient, hence we propose a
 302 different approach to obtain an approximated solution set. This method consists of obtaining the optimal
 303 solution for each scenario in $O(N^2)$. Notice that, as a consequence of disallowing shortages, some of these
 304 solutions could be infeasible for problem P . In this case, all the scenarios with infeasible solutions are solved
 305 again using a demand vector where each component corresponds to the marginal maximum demand,
 306 namely, the j th value in this vector coincides with $(\max_{1 \leq i \leq M} \{D_i^{1,j+1}\} - \max_{1 \leq i \leq M} \{D_i^{1,j}\})$. Remark that
 307 the demand vector obtained in this way is a ZIO plan and, hence, is feasible for P . Moreover, the com-
 308 putational effort to determine this set of policies is $O(MN^2)$. In addition, these plans can also be used as the
 309 starting upper bound set of the branch and bound scheme when shortages are not permitted.

310 We proceed below to analyze the case when both the carrying and the production/reorder costs are
 311 concave and shortages are permitted.

312 4. Case with shortages

313 This section is devoted to the case in which inventories on hand are not restricted to be positive. When I_i^j
 314 is negative, it now represents a shortage of $-I_i^j$ units of unfilled (backlogged) demand that must be satisfied
 315 by production/reorder during periods j through N .

316 We assume, for simplicity, that $h_i^j(I_i^j)$ represents the holding/shortage unit cost function for period j in
 317 scenario i . When I_i^j is non-negative, $h_i^j(I_i^j)$ remains equal to the cost of having I_i^j units of inventory on hand
 318 at the end of period j in scenario i . When I_i^j is negative, $h_i^j(I_i^j)$ becomes the cost of having a shortage of $-I_i^j$
 319 units of unfilled demand on hand at the end of period j in scenario i .

320 In the single scenario version, there exists at least one period with inventory on hand equal to zero
 321 between two consecutive periods with production/reorder different from zero [25,27]. That is, if $x_j > 0$ and
 322 $x_l > 0$ for $j < l$, then $I^k = 0$ for at least one k so that $j \leq k < l$. This idea is exploited to develop an $O(N^3)$
 323 algorithm to determine an optimal policy [27].

324 Assuming that inventory levels are unconstrained, we can adapt the previous property to the multi-
 325 scenario case as follows:

$$\text{If } x_j > 0 \text{ and } x_l > 0 \text{ for } j < l, \text{ then } I_i^k = 0, \text{ for some } i \text{ and } k, j \leq k < l. \tag{11}$$

327 Unlike the ZIO property for the multiscenario case, the above expression allow us to obtain all the plans
 328 satisfying (11) independently. In other words, any plan satisfying (11) for one scenario is to be feasible for
 329 the rest of scenarios, hence a straightforward approach to generate the whole plans set is to determine each
 330 set (one per scenario) separately. Again, these plans play a relevant role for obtaining the Pareto set of
 331 problem P with stockouts, since, as Theorem 5 shows, they represent the extreme points of the feasible set.

332 We can use again the network introduced in Section 3 to characterize the extreme solutions of P with
 333 shortages. Accordingly, the following theorem states that such extreme points represent acyclic policies.
 334 That is, demand in a period k is satisfied from the production/reorder either in a previous period ($j \leq k$) or
 335 in a successor period ($l \geq k$). Therefore, in the underlying network of the problem, each node (excepting the
 336 production/reorder node) is attainable from only one of the following nodes: the production/reorder node,
 337 the predecessor holding node or the successor backlogging node.

338 **Theorem 5.** *Any basic solution for problem P with shortages is acyclic.*

339 **Proof.** Following a similar reasoning to that in Theorem 2, let us select, for each block (scenario), any two
 340 columns corresponding to production/reorder arcs in (9), e.g., columns j and l . Moreover, we select, for
 341 each scenario, the columns related to periods j up to l . It is easy to see that a linear combination of these
 342 columns with coefficients $+1, -1, +1, \dots, +1$ respectively, gives the null vector. Therefore, any basic so-
 343 lution is acyclic. \square

344 **Proposition 6.** *The Pareto-optimal set of problem P with shortages contains, at least, one plan satisfying*
 345 *property (11).*

346 **Proof.** Similar to that in Proposition 3. \square

347 Notice that not all the basic plans belong to the Pareto-optimal set and, the solution time required to
 348 determine the whole non-dominated solutions set increases with the input data. Therefore, obtaining the
 349 efficient plans among the extreme plans seems to be a reasonable approach, not only as approximation to
 350 the real Pareto-optimal set but also as an upper bound set to be used in the branch and bound scheme.
 351 Thus, taking into account that the feasible decisions set verifying (11) for one state $(I_1^{j-1}, \dots, I_M^{j-1}) \in I(j)$ is
 352 as follows

$$\Phi(j, (I_1^{j-1}, \dots, I_M^{j-1})) = \begin{cases} 0, & \text{if } I_i^{j-1} > 0 \text{ for all } i, \\ \{0\} \cup \{-I_i^{j-1} + D_i^{j,k}\}, & \text{otherwise.} \\ i = 1, \dots, M, \end{cases}$$

354 we can now determine the non-dominated cost vectors set for the state $(I_1^{j-1}, \dots, I_M^{j-1})$ in period j according
 355 to the following functional equation:

$$F(j, (I_1^{j-1}, \dots, I_M^{j-1})) = \underset{x_j \in \Phi(j, (I_1^{j-1}, \dots, I_M^{j-1}))}{v - \min} \left\{ \begin{array}{l} \left[\begin{array}{c} c_1^j(x_j) \\ \vdots \\ c_M^j(x_j) \end{array} \right] + \left[\begin{array}{c} h_1^j(I_1^{j-1} + x_j - D_1^{j,j+1}) \\ \vdots \\ h_M^j(I_M^{j-1} + x_j - D_M^{j,j+1}) \end{array} \right] \\ \oplus F(j+1, (I_1^{j-1} + x_j - D_1^{j,j+1}, \dots, I_M^{j-1} + x_j - D_M^{j,j+1})) \end{array} \right\}. \quad (12)$$

357 Remark that when $F(1, (0, \dots, 0))$ is evaluated, the non-dominated solutions set satisfying (11) is achieved.

358 **Proposition 7.** *The MDP algorithm for the problem (12) runs in $O((M(MN+1)^{2N})/(2(MN)^2))$.*

359 **Proof.** In period j , x_j can take values from $\Phi(j, (I_1^{j-1}, \dots, I_M^{j-1}))$. Accordingly, the maximum number of
 360 states in any period is $M(N-1)+1$. Also, in one state of period j there are, at most, $(MN+1)^{j-1}$ vectors.
 361 Therefore, at most, $(M(MN+1)^{j-1}((MN+1)^{j-1}-1))/2$ comparisons have to be made. Consequently, the
 362 total number of comparisons is $O\left(M \sum_{j=2}^N ((MN+1)^{j-1}((MN+1)^{j-1}-1))/2\right)$, and hence the procedure
 363 runs in $O((M(MN+1)^{2N})/(2(MN)^2))$. \square

364 Since the implementation of the algorithm based on (10) involves a number of operations, which in-
 365 creases exponentially with the input size, we propose a different approach to obtain an approximated so-
 366 lution set. This method consists of obtaining the optimal solution for each scenario in $O(N^3)$. Unlike the
 367 case without shortages, all the single scenario solutions are to be feasible for problem P . Therefore, the
 368 computational effort to determine the set of optimal solutions for each scenario is $O(MN^3)$, and these plans
 369 are proposed as the starting upper bound set of the branch and bound scheme when shortages are allowed.

370 Once the initial upper bound sets for both shortages and not shortages situations have been introduced,
 371 we present in the following section the branch and bound scheme, as well as an initial lower bound set to
 372 determine the Pareto-optimal set.

373 5. The Pareto-optimal Set for the dynamic multiscenario lot size problem

374 Before introducing the solution method, we need some additional notation. Let $\mathbf{D}_j \in \mathbb{N}_0^M$ be a vector
 375 where each component $i = 1, \dots, M$ corresponds to $D_i^{1,j}$ and, also, let $T(j+1, (I_1^j, \dots, I_M^j))$ denote the set of
 376 cost vectors associated to subplans that attain the state vector $(I_1^j, \dots, I_M^j) \in I(j+1)$. That is,

$$T(j+1, (I_1^j, \dots, I_M^j)) = \{T(j, (I_1^{j-1}, \dots, I_M^{j-1})) \oplus (r_1^j(x, I_1^j), \dots, r_M^j(x, I_M^j)) : x \in \mathbb{N}_0, \\ I_i^{j-1} + x - D_i^{j,j+1} = I_i^j, \text{ for all } i \text{ and } (I_1^{j-1}, \dots, I_M^{j-1}) \in I(j)\}.$$

378 Since we are interested in calculating the non-dominated policies that reach the state $(0, \dots, 0) \in$
 379 $I(N+1)$, we must determine the efficient plans among those in $T(N+1, (0, \dots, 0))$ via pairwise compar-
 380 ison. As Villarreal and Karwan [22] pointed out, a necessary condition for a Pareto-optimal point is that it
 381 must contain, as its first $n-1$ components, an efficient solution to an $(n-1)$ -stage problem, hence the
 382 previous process must be applied in all the attainable states. Thus, the efficient subplans should be selected
 383 in every attainable state. Therefore, we define $T^*(j+1, (I_1^j, \dots, I_M^j))$ to be the set of non-dominated
 384 subplans that attain the state (I_1^j, \dots, I_M^j) .

385 Moreover, the interval for the decision variable x can be calculated according to the following argument:
 386 the lot size for the state (I_1^j, \dots, I_M^j) must be at least equal to zero or $\max_{1 \leq i \leq M} \{0, D_i^{j+1, j+2} - I_i^j\}$, respec-
 387 tively, depending on whether shortages are permitted or not. On the other hand, the upper bound for the
 388 interval corresponds to the remaining quantity to reach the total demand, hence x ranges in
 389 $[0, \max_{1 \leq i \leq M} \{0, D_i^{j+1, N+1} - I_i^j\}]$ in case of allowing shortages or in $[\max_{1 \leq i \leq M} \{0, D_i^{j+1, j+2} - I_i^j\},$
 390 $\max_{1 \leq i \leq M} \{0, D_i^{j+1, N+1} - I_i^j\}]$, otherwise. In addition, given a period j , let s be the scenario so that
 391 $D_s^{1, j+1} = \max_{1 \leq i \leq M} \{D_i^{1, j+1}\}$. Then, we consider as initial state vector in $I(j)$ either vector
 392 $(D_s^{1, j+1} - D_1^{1, j+1}, \dots, D_s^{1, j+1} - D_M^{1, j+1})$, if shortages are not allowed, or vector $(-D_1^{1, j+1}, \dots, -D_M^{1, j+1})$ otherwise.
 393 Thus, the rest of vectors in $I(j)$ are obtained just augmenting one unit each component as many times as
 394 $D - (D_s^{1, j+1} - D_i^{1, j+1})$ or $D - (-D_i^{1, j+1})$ for any i , respectively.

395 Taking into account that $I(1) = I(N + 1) = (0, \dots, 0)$, we can now outline the MDP algorithm.

396 **Algorithm 1.** Determine the Pareto-optimal set for problem P

397 **DATA:** matrices d_i^j, c_i^j, h_i^j , numbers M and N , and sets $I(j), j = 1, \dots, N + 1$

398 1: **for** $j \leftarrow N$ **downto** 1 **do**

399 2: **for all** state $(I_1^j, \dots, I_M^j) \in I(j + 1)$ **do**

400 3: **for all** state $(I_1^{j-1}, \dots, I_M^{j-1}) \in I(j)$ **do**

401 4: **if** $I_i^j - I_i^{j-1} + d_i^j \geq 0$ and $I_i^j - I_i^{j-1} + d_i^j = I_s^j - I_s^{j-1} + d_s^j$ for $i \neq s$ **then**

402 5: $x_j = I_i^j - I_i^{j-1} + d_i^j$

403 6: insert x_j and its cost vector in state $(I_1^{j-1}, \dots, I_M^{j-1})$ and update $T^*(j, (I_1^{j-1}, \dots, I_M^{j-1}))$

404 7: **end if**

405 8: **end for**

406 9: **end for**

407 10: **end for**

408 11: return $T^*(1, (0, \dots, 0))$

409 **Example 1.** For the sake of completeness, we present the following numerical example to illustrate the
 410 previous results for the case without shortages.

	d_i^j			c_i^j			h_i^j		
	$j = 1$	$j = 2$	$j = 3$	$j = 1$	$j = 2$	$j = 3$	$j = 1$	$j = 2$	$j = 3$
$i = 1$	5	10	5	5	5	5	1	1	0
$i = 2$	10	6	4	10	2	5	20	1	0
$i = 3$	15	2	3	5	5	5	100	100	0

411 As you can see, all possible plans are collected in the graph depicted in Fig. 3. In this graph, each node
 412 represents one state that is identified by its inventory level vector (in parenthesis). Also, within each node,
 413 the partial cost vectors (in brackets) associated to subplans that attain this node are shown. Those subplans
 414 which are dominated by any other subplan in the same node are marked with an asterisk. For each node,
 415 the leaving arcs (arrows) represent the possible decisions for this node. The right-most node contains the
 416 non-dominated solution set.

417 Fig. 3 illustrates also the case where a non-ZIO plan dominates a ZIO plan, namely, the ZIO plan
 418 $(17, 0, 3)$ with cost vector $\{114, 326, 300\}$ is dominated by the non-ZIO plan $(15, 3, 2)$ with cost vector
 419 $\{113, 268, 200\}$.

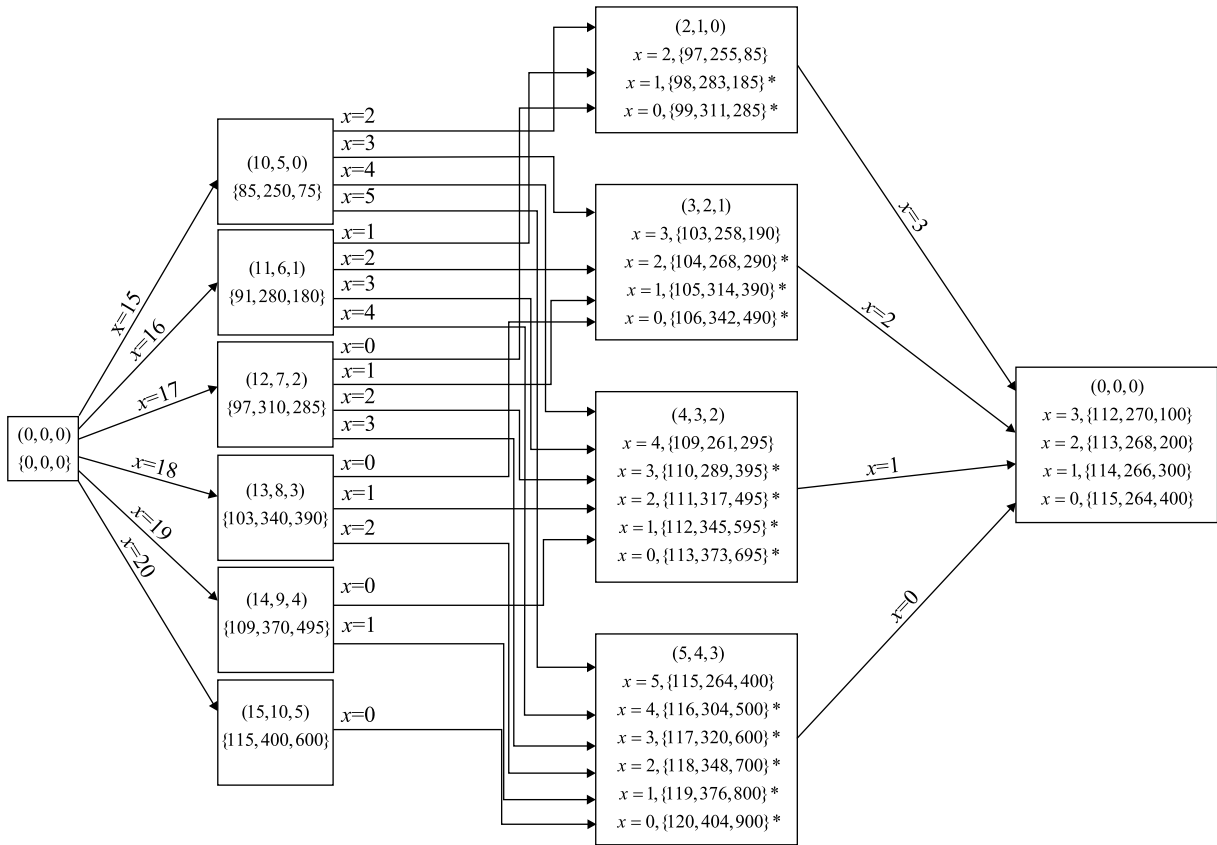


Fig. 3. Complete description of Pareto-optimal plans of Example 1.

420 Since Algorithm 1 becomes untractable as the difference $(D - \max_{1 \leq i \leq M} \{d_i^1\})$ increases, a branch and
 421 bound approach is proposed. We first focus our attention on the case without shortages. The other case is
 422 commented later on. We should reformulate problem P without shortages in a more appropriate way.
 423 Accordingly, we denote by $(I_1^n, \dots, I_M^n) \in I(n+1)$ a state vector at the beginning of period $n+1$, and let
 424 $P(n, (I_1^n, \dots, I_M^n))$ be the set of Pareto-values of the subproblem consisting of periods 1 to n with final in-
 425 ventory vector (I_1^n, \dots, I_M^n) . Therefore, we can now state the problem as follows

$$P(n, (I_1^n, \dots, I_M^n)) = v - \min \left[\sum_{j=1}^n c_1^j(x_j) + \sum_{j=1}^{n-1} h_1^j \left(\sum_{k=1}^j x_k - D_1^{1,j+1} \right) + h_1^n(I_1^n), \dots, \right. \\ \left. \sum_{j=1}^n c_M^j(x_j) + \sum_{j=1}^{n-1} h_M^j \left(\sum_{k=1}^j x_k - D_M^{1,j+1} \right) + h_M^n(I_M^n) \right]$$

$$\text{s.t. : } \sum_{j=1}^k x_j \geq D_i^{1,k+1}, \quad k = 1, \dots, n-1; \quad i = 1 \dots, M$$

$$\sum_{j=1}^n x_j = D_i^{1,n+1} + I_i^n, \quad i = 1 \dots, M,$$

427 It is worth noting that $P(n, (I_1^n, \dots, I_M^n)) = T^*(n + 1, (I_1^n, \dots, I_M^n))$. Now, it can be determined the Pareto
 428 values of the complementary problem $\bar{P}(n + 1, (I_1^n, \dots, I_M^n))$, i.e., the problem consisting of periods $n + 1$ to
 429 N with initial inventory vector (I_1^n, \dots, I_M^n) , as follows

$$\begin{aligned} \bar{P}(n + 1, (I_1^n, \dots, I_M^n)) = v - \min & \left[\sum_{j=n+1}^N c_1^j(x_j) + \sum_{j=n+1}^{N-1} h_1^j \left(I_1^n + \sum_{k=n+1}^j x_k - D_1^{n+1,j+1} \right) \right. \\ & + h_1^N \left(I_1^n + \sum_{k=n+1}^N x_k - D_1^{n+1,N+1} \right), \dots, \sum_{j=n+1}^N c_M^j(x_j) \\ & + \sum_{j=n+1}^{N-1} h_M^j \left(I_M^n + \sum_{k=n+1}^j x_k - D_M^{n+1,j+1} \right) \\ & \left. + h_M^N \left(I_M^n + \sum_{k=n+1}^N x_k - D_M^{n+1,N+1} \right) \right] \\ \text{s.t.: } & \sum_{j=n+1}^k x_j \geq D_i^{n+1,k+1} - I_i^n, \quad k = n + 1, \dots, N; \quad i = 1 \dots, M \\ & \sum_{j=n+1}^N x_j = D_i^{n+1,N+1} - I_i^n, \quad i = 1 \dots, M, \end{aligned}$$

431 Remark that when shortages are allowed, the first set of constraints in both formulations P and \bar{P} should
 432 be removed. Again, the optimality principle gives rise to the following recursive equation which provides
 433 the Pareto-optimal set for P .

$$F(1, (0, \dots, 0)) = v - \min_{\substack{(I_1^n, \dots, I_M^n) \in I^{(n+1)} \\ n=1, \dots, N}} (P(n, (I_1^n, \dots, I_M^n)) \oplus \bar{P}(n + 1, (I_1^n, \dots, I_M^n))).$$

435 These equations along with upper and lower bound sets allow us to introduce the branch and bound
 436 scheme into the dynamic programming heap. According to Villarreal and Karwan [22], a set LB of lower
 437 bounds for a vector-valued problem is a set of points that satisfy the following conditions: (i) each element
 438 is either efficient or dominates at least one of the efficient solutions of the problem, and (ii) each efficient
 439 solution is dominated by at least one member of the set, or it is indeed a member of the set. In addition,
 440 recall that a set UB of upper bounds is a set of points such that each element is either efficient or is
 441 dominated by at least one efficient solution.

442 Assume that we know both lower bounds $LB(n + 1, (I_1^n, \dots, I_M^n))$ for each subproblem $\bar{P}(n + 1,$
 443 $(I_1^n, \dots, I_M^n))$ and also global upper bounds UB for the original problem $F(1, (0, \dots, 0))$.

444 Consider $f \in P(n, (I_1^n, \dots, I_M^n))$ such that for any $lb \in LB(n + 1, (I_1^n, \dots, I_M^n)) : f + lb \geq u$ for some
 445 $u \in UB$. It is straightforward that the branch generated by f needs not being explored. Indeed, $u \in UB$ and,
 446 therefore, there exists \hat{f} efficient (it may occur that $lb = \hat{f}$) so that $\hat{f} \leq u$. Hence, $\hat{f} \leq f + lb \leq f +$ (any
 447 feasible completion). This implies that no completion of f can be efficient.

448 Once the branch and bound scheme has been outlined, the following step consists of determining how
 449 the UB and LB sets are initialized. We set the UB with the non-dominated ZIO policies which are obtained
 450 in previous sections. On the other hand, different LB sets can be determined depending on the cost functions
 451 type. In case of linear costs, we propose two sets. The first concerns with the continuous relaxation of the
 452 problem. The second approach consists of determining the optimal policies for each scenario using the
 453 Wagelmans et al. algorithm [23] and applying, for each pair of optimal plans, a procedure to calculate the
 454 lower envelope. Another case arises when the cost functions are concave. Under this assumption, Theorem

Table 1
Parameter values for ten randomly generated problems

	<i>d</i>			<i>c</i>			<i>h</i>								
	1	2	3	1	2	3	1	2	3						
P1															
S1	6	3	3	3	7	5	1	2	x						
S2	7	2	3	2	3	2	6	5	x						
P2															
S1	7	4	4	2	7	8	1	1	x						
S2	3	7	5	3	4	4	1	5	x						
S3	7	3	5	7	3	4	1	1	x						
P3															
S1	6	7	2	2	6	5	1	2	x						
S2	5	7	3	6	2	1	3	3	x						
S3	6	6	3	5	4	5	2	4	x						
S4	7	7	1	1	3	7	4	5	x						
	<i>d</i>				<i>c</i>				<i>h</i>						
	1	2	3	4	1	2	3	4	1	2	3	4			
P4															
S1	5	7	5	3	5	5	7	5	1	1	1	x			
S2	7	5	3	5	7	5	5	5	1	1	1	x			
P5															
S1	5	6	5	4	1	5	5	3	2	1	1	x			
S2	4	5	6	5	6	4	2	2	3	3	2	x			
S3	6	4	4	6	2	1	2	3	5	4	3	x			
P6															
S1	3	9	7	5	7	3	5	6	4	1	2	x			
S2	7	5	6	6	5	4	4	5	4	3	3	x			
S3	7	5	5	7	7	5	5	2	5	5	4	x			
S4	8	4	4	8	3	4	5	4	3	3	5	x			
P7															
S1	5	2	7	7	6	7	2	3	1	1	2	x			
S2	10	5	4	2	7	7	6	1	3	1	4	x			
S3	6	6	4	5	4	4	5	3	4	1	1	x			
S4	11	3	4	3	2	8	6	7	1	1	2	x			
S5	9	2	6	4	3	5	7	6	1	2	2	x			
	<i>d</i>					<i>c</i>					<i>h</i>				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
P8															
S1	8	2	6	5	4	8	7	5	7	6	4	3	3	1	x
S2	5	5	5	5	5	1	6	7	5	6	1	2	2	2	x
S3	4	4	5	6	6	2	2	3	2	1	5	6	7	6	x
P9															
S1	9	5	6	2	3	7	5	2	7	6	5	6	1	1	x
S2	10	3	5	3	4	8	3	6	4	2	2	1	4	3	x
S3	7	4	7	4	3	6	4	5	5	4	4	3	5	2	x
S4	8	5	4	3	5	5	6	4	6	5	1	2	7	5	x

Table 1 (continued)

	d			c			h								
	1	2	3	1	2	3	1	2	3						
P10															
S1	5	3	2	2	3	2	8	6	7	5	2	1	2	1	x
S2	7	3	2	1	2	6	3	5	5	2	5	3	2	4	x
S3	6	6	1	1	1	5	4	8	6	6	1	1	4	6	x
S4	8	1	3	1	2	4	8	7	6	5	4	2	5	3	x
S5	5	2	3	3	2	5	4	7	7	6	1	3	3	2	x

8 shows that a linear conversion of the cost functions reduces to the problem of finding a *LB* set for the original problem.

Theorem 8. The Pareto-optimal solution set obtained with any linear function $L(\mathbf{x}) = (L_1^{1,N}(\mathbf{x}), \dots, L_M^{1,N}(\mathbf{x}))$ such that for any feasible \mathbf{x} it holds $R_i^{1,N}(\mathbf{x}) \geq L_i^{1,N}(\mathbf{x}), i = 1 \dots, M$; is a *LB* set for problem *P*.

Proof. Let us assume that the cost functions $R_i^{1,j}$ defined in (2) are concave. Furthermore, let $L_i^{1,N}$ be a linear function such that for any feasible \mathbf{x} it holds $R_i^{1,N}(\mathbf{x}) \geq L_i^{1,N}(\mathbf{x}), i = 1 \dots, M$, and let $L(\mathbf{x}) = (L_1^{1,N}(\mathbf{x}), \dots, L_M^{1,N}(\mathbf{x}))$.

Let us denote $LB = L(E(L_1^{1,N}, \dots, L_M^{1,N}))$ where $E(L_1^{1,N}, \dots, L_M^{1,N})$ is the set of Pareto-optimal solutions of the problem

$$\begin{aligned}
 & v - \min(L_1^{1,N}(\mathbf{x}), \dots, L_M^{1,N}(\mathbf{x})) \\
 & \text{s.t. :} \\
 & I_i^0 = I_i^N = 0, \quad i = 1, \dots, M, \\
 & I_i^{j-1} + x_j - I_i^j = d_i^j, \quad j = 1, \dots, N, \quad i = 1, \dots, M, \\
 & I_i^j \geq 0, \quad x_j \text{ integer}, \quad j = 1, \dots, N, \quad i = 1, \dots, M,
 \end{aligned}$$

Moreover, we denote by $E(R_1^{1,N}, \dots, R_M^{1,N})$ the Pareto-optimal set of the original problem *P*. Accordingly, if $\mathbf{x} \in E(R_1^{1,N}, \dots, R_M^{1,N})$ then either $\mathbf{x} \in E(L_1^{1,N}, \dots, L_M^{1,N})$ or $\mathbf{x} \notin E(L_1^{1,N}, \dots, L_M^{1,N})$. In the first case, $L(\mathbf{x}) = (L_1^{1,N}(\mathbf{x}), \dots, L_M^{1,N}(\mathbf{x})) \in LB$ and hence $L(\mathbf{x}) \leq R(\mathbf{x})$, where $R(\mathbf{x})$ was defined in (3). In the second case, it must exist \mathbf{y} such that $\mathbf{y} \in E(L_1^{1,N}, \dots, L_M^{1,N})$ and $L(\mathbf{y}) \leq L(\mathbf{x})$. Thus, $L(\mathbf{y}) \in LB$ and $L(\mathbf{y}) \leq R(\mathbf{x})$. Therefore, *LB* is an actual lower bound for problem *P*. \square

6. Computational experience

This section is divided into two parts. In the first part, the Pareto-optimal set for ten randomly generated problems are reported. On the other hand, the second part is devoted to test the efficiency of the two algorithms, the MDP procedure and the Branch and Bound (B&B) approach, as a function of both the number of scenarios and the number of periods.

To simplify the computational experiment, we have chosen the cost functions to be linear and the inventory levels to be non-negative. Taking into account these assumptions, the problems have been solved using the procedure given in the previous section.

In this part, Tables 1 and 2 show the input data for ten problems and the non-dominated plans with their overall cost vectors respectively. Table 1 is organized as follows: the first column indicates the number of the problem, the rows represent the scenarios (*S_i* represents the *i*th scenario) and the rest of columns give for the different periods the values for the demand, unit holding cost and unit reorder cost respectively. This computational experience involves problems with two scenarios and four periods up to problems with five

Table 2

Pareto-optimal sets for the ten problems in Table 1

P1	{7, 2, 3} (51, 26)	{8, 1, 3} (48, 31)	{9, 0, 3} (45, 36)		
P2	{7, 4, 4} (74, 62, 78) {12, 0, 3} (54, 67, 103)	{8, 3, 4} (70, 62, 83) {13, 0, 2} (50, 72, 108)	{9, 2, 4} (66, 62, 88) {14, 0, 1} (46, 77, 113)	{10, 1, 4} (62, 62, 93) {15, 0, 0} (42, 82, 118)	{11, 0, 4} (58, 62, 98)
P3	{7, 7, 1} (64, 69, 78, 35) {12, 2, 1} (49, 104, 93, 45)	{8, 6, 1} (61, 76, 81, 37) {13, 1, 1} (46, 111, 96, 47)	{9, 5, 1} (58, 83, 84, 39) {14, 0, 1} (43, 118, 99, 49)	{10, 4, 1} (55, 90, 87, 41)	{11, 3, 1} (52, 97, 90, 43)
P4	{7, 5, 5, 3} (112, 116) {7, 10, 0, 3} (107, 121)	{7, 6, 4, 3} (111, 117)	{7, 7, 3, 3} (110, 118)	{7, 8, 2, 3} (109, 119)	{7, 9, 1, 3} (108, 120)
P5	{6, 5, 5, 4} (70, 88, 49) {11, 0, 5, 4} (60, 113, 79) {16, 0, 0, 4} (55, 163, 124)	{7, 4, 5, 4} (68, 93, 55) {12, 0, 4, 4} (59, 123, 88)	{8, 3, 5, 4} (66, 98, 61) {13, 0, 3, 4} (58, 133, 97)	{9, 2, 5, 4} (64, 103, 67) {14, 0, 2, 4} (57, 143, 106)	{10, 1, 5, 4} (62, 108, 73) {15, 0, 1, 4} (56, 153, 115)
P6	{8, 4, 7, 5} (153, 116, 134, 110) {8, 9, 2, 5} (148, 131, 159, 120)	{8, 5, 6, 5} (152, 119, 139, 112) {8, 10, 1, 5} (147, 134, 164, 122)	{8, 6, 5, 5} (151, 122, 144, 114) {8, 11, 0, 5} (146, 137, 169, 124)	{8, 7, 4, 5} (150, 125, 149, 116)	{8, 8, 3, 5} (149, 128, 154, 118)
P7	{11, 4, 4, 2} (132, 134, 112, 95, 107) {16, 0, 3, 2} (138, 151, 132, 73, 102) {21, 0, 0, 0} (170, 194, 158, 65, 103)	{12, 3, 4, 2} (132, 137, 116, 90, 106) {17, 0, 2, 2} (144, 156, 136, 71, 101)	{13, 2, 4, 2} (132, 140, 120, 85, 105) {18, 0, 1, 2} (150, 161, 140, 69, 100)	{14, 1, 4, 2} (132, 143, 124, 80, 104) {19, 0, 0, 2} (156, 166, 144, 67, 99)	{15, 0, 4, 2} (132, 146, 128, 75, 103) {20, 0, 0, 1} (163, 180, 151, 66, 101)
P8	{8, 2, 6, 5, 4} (167, 118, 117) {13, 0, 3, 5, 4} (207, 101, 157)	{9, 1, 6, 5, 4} (172, 114, 122) {14, 0, 2, 5, 4} (217, 98, 167)	{10, 0, 6, 5, 4} (177, 110, 127) {15, 0, 1, 5, 4} (227, 95, 177)	{11, 0, 5, 5, 4} (187, 107, 137) {16, 0, 0, 5, 4} (237, 92, 187)	{12, 0, 4, 5, 4} (197, 104, 147)
P9	{10, 4, 6, 2, 3} (139, 154, 159, 160) {10, 9, 1, 2, 3} (184, 144, 169, 180) {10, 4, 10, 0, 1} (127, 188, 185, 192)	{10, 5, 5, 2, 3} (148, 152, 161, 164) {10, 10, 0, 2, 3} (193, 142, 171, 184) {10, 4, 11, 0, 0} (125, 199, 193, 203)	{10, 6, 4, 2, 3} (157, 150, 163, 168) {10, 4, 7, 1, 3} (135, 160, 164, 165)	{10, 7, 3, 2, 3} (166, 148, 165, 172) {10, 4, 8, 0, 3} (131, 166, 169, 170)	{10, 8, 2, 2, 3} (175, 146, 167, 176) {10, 4, 9, 0, 2} (129, 177, 177, 181)
P10	{8, 4, 1, 1, 1} (84, 89, 78, 96, 105) {8, 5, 0, 1, 1} (87, 90, 75, 99, 105) {13, 0, 0, 1, 1} (67, 130, 85, 99, 115)	{9, 3, 1, 1, 1} (80, 97, 80, 96, 107) {9, 4, 0, 1, 1} (83, 98, 77, 99, 107)	{10, 2, 1, 1, 1} (76, 105, 82, 96, 109) {10, 3, 0, 1, 1} (79, 106, 79, 99, 109)	{11, 1, 1, 1, 1} (72, 113, 84, 96, 111) {11, 2, 0, 1, 1} (75, 114, 81, 99, 111)	{12, 0, 1, 1, 1} (68, 121, 86, 96, 113) {12, 1, 0, 1, 1} (71, 122, 83, 99, 113)

483 scenarios and five periods. In Table 2, for each problem, the efficient plans with their respective costs are
 484 allocated in consecutive cells of the same row.

Table 3

Comparison of running times (in sec.)

Scenarios (M)	Periods (N)	Average time (MDP)	Average time (B&B)
2	3	7.08	4.98
2	4	8.90	0.66
2	5	24.67	12.80
3	3	19.93	13.25
3	4	11.23	1.24
3	5	2.76	0.63
4	3	10.70	4.65
4	4	15.94	5.90
4	5	22.85	1.46
5	3	20.54	5.00
5	4	76.47	13.15
5	5	17.06	11.28

485 The MDP solution procedure was coded in C++ using LEDA libraries. The main difficulty to implement
 486 this code is the storage requirement which increases with the difference ($D - \max_{1 \leq i \leq M} \{d_i^1\}$). This diffi-
 487 culty, known as curse of dimensionality, was already discussed by Villarreal and Karwan [22]. These au-
 488 thors argued that as the number of objective functions increases so does the solution time. The problems
 489 proposed in Table 1 were solved in a workstation HP 9000-712/80. Another interesting aspect of the
 490 problem concerns its sensitivity. After several samples, we notice that slight changes in the input data make
 491 the Pareto-optimal set to vary drastically.

492 The B&B scheme has been incorporated to the MDP procedure as follows: for each subproblem
 493 $\bar{P}(n+1, I_1^n, \dots, I_M^n)$, the LB set is obtained from calls to the ADBASE code developed by Steuer [18]. This
 494 code gives the supported non-dominated solutions for continuous linear multicriteria problems. As a
 495 consequence of both the input to and the output from the ADBASE code is file typed, conversions of the
 496 form matrix(C++)-file(ADBASE) and file(ADBASE)-matrix(C++) are required. Moreover, since all the
 497 parameters are integer and the constraints matrix is unimodular, the extreme solutions given by ADBASE
 498 are integer-valued as well, i.e., feasible for P . Hence, as a result the non-dominated solutions associated to
 499 the first subproblem are also considered as the initial UB for the original problem $F(1, (0, \dots, 0))$.

500 Now, we provide, in Table 3, the average running times for different instances of this problem. For each
 501 pair (M, N) ten instances were run. The parameters have been generated according to the following values:
 502 the total demand D ranges in the interval $[1, 1000]$, the unit carrying and reorder costs vary between 1 and
 503 100. The troubles in the computational experience arise as a consequence of the ADBASE limitations. As
 504 the number of scenarios or periods increases so does the number of rows and columns in the constraint
 505 matrix of the linear multiobjective problem and the problem becomes intractable. Therefore, only some
 506 (M, N) combinations can be tested.

507 Our computational experiments show that the B&B scheme outperforms the MDP approach in all cases.
 508 The small difference in some instances between the average running times of both procedures is due to each
 509 subproblem in the B&B calls to the ADBASE code. Therefore, the bottleneck of the B&B procedure is just
 510 the time required to obtain the LB set for each subproblem. In spite of this difficulty, the B&B results in
 511 CPU times smaller than the MDP method.

512 7. Concluding remarks

513 In this article we introduce different algorithms to solve the multiscenario lot size problem. Throughout
 514 the paper, the case with concave costs is discussed. The solution procedures for this case have been im-
 515 plemented using the DMDMP approach and exploiting the dynamic lot size problem's properties. More-

516 over, a B&B procedure has been implemented with a reasonably good behavior in most cases. We are
 517 interested in improving this procedure by finding *LB* sets that are not obtained from external routine, which
 518 will decrease much more the running times of the B&B versus MDP.

519 References

- 520 [1] M.J. Alves, J. Climaco, An interactive reference point approach for multiobjective mixed-integer programming using branch-and-
 521 bound, *European Journal of Operational Research* 124 (2000) 478–494.
- 522 [2] R. Bellman, S.E. Dreyfus, *Applied Dynamic Programming*, Princeton University Press, Princeton, New Jersey, 1961.
- 523 [3] M. Ehrgott, X. Gandibleux, An annotated bibliography of multicriteria combinatorial optimization, *OR-Spektrum* 22 (2000) 425–
 524 460.
- 525 [4] E. Fernández, J. Puerto, The multiobjective solution of the uncapacitated plant location problem, *European Journal of*
 526 *Operational Research*, in press.
- 527 [5] F.R. Fernández, S. Nickel, J. Puerto, A. Rodriguez-Chia, Robustness of the Pareto-solution for the multicriteria minisum location
 528 problem, *Journal of Multicriteria Decision Analysis* 10 (2001) 191–203.
- 529 [6] X. Gandibleux, A. Jaszkiwicz, A. Freville, R. Slowiński (Eds.), Multi-objective Metaheuristics, *Journal of Heuristics* 6 (3) (2000)
 530 291–431 (special issue).
- 531 [7] P. Horner, Planning under Uncertainty, *Question & Answers with G. Dantzig*, *OR/MS Today*, October 1999, pp. 26–30.
- 532 [8] K. Klamroth, M. Wiecek, Dynamic programming approach to the multiple criteria Knapsack problem, *Naval Research Logistics*
 533 47 (2000) 57–76.
- 534 [9] A.S. Manne, Programming of economic lot sizes, *Management Science* 4 (1958) 115–135.
- 535 [10] A.S. Manne, A.F. Veinott Jr., in: A.S. Manne (Ed.), *Investments for Capacity Expansion: Size, Location and Time-Phasing*, MIT
 536 Press, Cambridge, Mass, 1967 (Chapter 11).
- 537 [11] T.L. Morin, A.M.O. Esoboque, The imbedded state space approach to reducing dimensionality in dynamic programs of higher
 538 dimensions, *Journal of Mathematical Analysis and Applications* 48 (1974) 801–810.
- 539 [12] G.L. Nemhauser, *Introduction to Dynamic Programming*, John Wiley and Sons, New York, 1966.
- 540 [13] J. Puerto, F.R. Fernández, Multicriteria Weber location problems, *Journal of the Multicriteria Decision Analysis* 18 (1999) 268–
 541 280.
- 542 [14] R. Ramesh, S. Zionts, M. Karwan, A class of practical interactive branch and bound algorithms for multicriteria integer
 543 programming, *European Journal of Operational Research* 26 (1986) 161–172.
- 544 [15] L.M. Rasmussen, Zero-one programming with multiple criteria, *European Journal of Operational Research* 26 (1986) 83–95.
- 545 [16] A.M. Rodriguez-Chia, J. Puerto, A multicriteria competitive Markov decision process, *Mathematical Methods of Operational*
 546 *Research* 55 (2002) 359–369.
- 547 [17] G. Ruhe, Complexity results for multicriterial and parametric network flows using a Pathological graph of Zadeh, *Zeitschrift für*
 548 *Operations Research* 32 (1988) 9–27.
- 549 [18] R.E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*, John Wiley and Sons, New York, 1986.
- 550 [19] E.L. Ulungu, J. Teghem, Multi-objective combinatorial optimization problems: A survey, *Journal of Multi-criteria Decision*
 551 *Analysis* 3 (1994) 83–104.
- 552 [20] A.F. Veinott Jr., Production planning with convex costs: A parametric study, *Management Science* 10 (1964) 441–460.
- 553 [21] M. Villarreal, H. Karwan, S. Zionts, An interactive branch and bound procedure for multicriterion integer linear programming,
 554 in: G. Fandel, T. Gal (Eds.), *Multiple Criteria Decision Making Theory and Application*, *Lecture Notes in Economics and*
 555 *Mathematical Systems* 177, Springer, Berlin, 1980, pp. 448–467.
- 556 [22] M. Villarreal, H. Karwan, Multicriteria dynamic programming with an application to the integer case, *Journal of Optimization*
 557 *Theory and Applications* 38 (1) (1982) 43–69.
- 558 [23] A. Wagelmans, S.V. Hoesel, A. Kolen, Economic lot sizing: An $O(n \log n)$ algorithm that runs in linear time in the wagner-whitin
 559 case, *Operations Research* 40 (1992) 145–155.
- 560 [24] H.M. Wagner, T. Whitin, *Dynamic Problems in the Theory of the Firm*, in: T. Whitin (Ed.), *Theory of Inventory Management*,
 561 second ed., Princeton University Press, Princeton, NJ, 1957.
- 562 [25] W.I. Zangwill, A deterministic multi-period production scheduling model with backlogging, *Management Science* 13 (1966) 105–
 563 119.
- 564 [26] W.I. Zangwill, Minimum concave cost flows in certain networks, *Management Science* 14 (1968) 429–450.
- 565 [27] W.I. Zangwill, A backlogging model and a multi-echelon model of a dynamic economic lot size production system—a network
 566 approach, *Management Science* 15 (1969) 506–527.
- 567 [28] S. Zionts, A survey of multiple criteria integer programming methods, *Annals of Discrete Mathematics* 5 (1979) 389–398.
- 568 [29] S. Zionts, J. Wallenius, Identifying efficient vectors: Some theory and computational results, *Operations Research* 23 (1980) 785–
 569 793.