Publié par : Published by : Publicación de la :	Faculté des sciences de l'administration Université Laval Québec (Québec) Canada G1K 7P4 Tél. Ph. Tel. : (418) 656-3644 Fax : (418) 656-7047
Édition électronique : Electronic publishing : Edición electrónica :	Aline Guimont Vice-décanat - Recherche et partenariats Faculté des sciences de l'administration
Disponible sur Internet : Available on Internet Disponible por Internet :	http ://www.fsa.ulaval.ca/rd rd@fsa.ulaval.ca

DOCUMENT DE TRAVAIL 2003-021

THE DYNAMIC-DEMAND JOINT REPLENISHMENT PROBLEM REVISITED: NEW SOLUTION PROCEDURES AND COMPARATIVE EVALUATION

Fayez F. Boctor
Gilbert Laporte
Jacques Renaud

Version originale : Original manuscript : Version original :	ISBN – 2-89524-171-6
Série électronique mise à jour : On-line publication updated : Seria electrónica, puesta al dia	05-2003

THE DYNAMIC-DEMAND JOINT REPLENISHMENT PROBLEM REVISITED: NEW SOLUTION PROCEDURES AND COMPARATIVE EVALUATION

FAYEZ F. BOCTOR[1,2], GILBERT LAPORTE[3] AND JACQUES RENAUD[1,2]
[1]Network Organization Technology Research Center (CENTOR),Université Laval, Canada, G1K 7P4
[2]Faculté des Sciences de l'Administration, Université Laval, Québec, Canada, G1K 7P4
[3] Canada Research Chair in Distribution Management, HEC Montréal, 3000 chemin de la Côte Sainte-Catherine, Montréal, Canada, H3T 2A7

ABSTRACT

This article considers the problem of coordinated ordering of items having deterministic but time varying demands where there is a common ordering cost if one or more of these items are ordered in addition to individual items ordering costs. Two new integer linear programming formulations are presented and compared to the classical formulation. Several well-known heuristics are described and a new improvement procedure is proposed. The relative performance of these heuristics is assessed. Results show the superiority of the new formulations and of the new improvement procedure.

Key words: Joint replenishment, Ordering costs, Dynamic demand, Heuristics.

1. INTRODUCTION

In multi-item inventory replenishment contexts, cost savings can be achieved by coordinating the replenishment of some items. In the *Dynamic-demand Joint Replenishment Problem* (DJRP), n item types must be replenished to satisfy the demand at T different time periods. Each order is made up of a *subset* of item types that are jointly replenished and generates two types of fixed costs: the *common ordering cost* associated with the order itself, and the *individual ordering cost* associated with each item type. The common ordering cost at period t is denoted S_t, while the individual ordering cost of item type i at period t is denoted by s_{it}. The *demand* d_{it} of item type i at period t is known for each of the T periods of the planning horizon. The *inventory holding cost* per unit of item type i during period t is denoted by h_{it}. It is assumed that replenishments are made at the beginning of each period, that items consumed during period t generate no holding cost during this period, and that an initial inventory I_{i0} is available for each item type i. The DJRP consists of determining the order quantities x_{it} for each item type i and for each period t in order to minimize the sum of replenishment and inventory holding costs over the whole planning horizon. In this article we assume that there are no quantity discounts, that no backlogging is allowed and that no limit is imposed on order sizes and on inventory levels. As shown by Arkin, Joneja and Roundy (1989), the DJRP is NP-hard.

Exact algorithms for the DJRP can be classified into four main categories: 1) dynamic programming (Zangwill, 1966; Veinott, 1969; Kao, 1979; Silver, 1979); 2) branch-and-bound (Erenguc, 1988; Federgruen and Tzur, 1994; Kirca, 1995; Robinson and Gao, 1996); 3) branch-and-cut (Raghavan and Rao, 1991; Raghavan, 1993); and 4) Dantzig-Wolfe decomposition (Raghavan and Rao, 1992). Only modest size instances can be solved be means of exact algorithms. Thus Erenguc (1988) reported results for $n = 12$ and $T = 20$; Raghavan and Rao (1991) have solved instances with $n = 30$ and $T = 20$; Raghavan and Rao (1992) were able to solve instances with $n = 100$ and $T = 40$. Finally, the largest instances solved by Federgruen and Tzur (1994) have size $n = 30$ and $T = 30$. Several heuristics capable of handling larger instances have been proposed. Fogarty and Barringer (1987) have used a dynamic programming approach to solve a simplified version of the problem. Silver and Kelle (1988) have described an improvement procedure applicable to any feasible solution. It successively considers each item type ordered at any given period and determines whether a cost saving could be achieved by ordering this item type within the previous order. Finally Atkins and Iyogun (1988) have extended the Silver-Meal (1973) heuristic to the DJRP, while Iyogun (1991) has proposed an extension of the well known part-period balancing method.

We are not aware of any systematic survey or computational comparison of the various algorithms proposed for the DJRP. The purpose of this article is to fill this gap. We will first recall the classical integer linear programming formulation for the DJRP and we will propose two new formulations. We will then provide a systematic and unified description of the available DJRP heuristics. In addition we will propose a new improvement procedure that can be combined with any heuristic. The various formulations and heuristics will be compared on a set of randomly generated instances.

The remainder of the paper is organized as follows. In Section 2 we present three mathematical formulations for the DJRP. In Section 3 we describe several basic DJRP heuristics, while a new improvement algorithm is presented in Section 4. Section 5 contains a comparative assessment of these formulations and heuristics, and conclusions are given in Section 6.

2. MATHEMATICAL PROGRAMMING FORMULATIONS

We first present a classical mathematical programming formulation of the DJRP. It uses the following notation:

- S_t — common ordering cost at period t
- s_{it} — individual ordering cost for item type i at period t
- h_{it} — unit inventory holding cost for item type i during period t
- d_{it} — demand for item type i for period t
- I_{it} — inventory level of item type i at the end of period t (I_{i0} represents the initial inventory level)
- M — a sufficiently large number
- x_{it} — replenishment quantity of item type i at the beginning of period t
- y_{it} — binary variables equal to 1 if and only if item type i is replenished at the beginning of period t; i.e., $y_{it} = 1$ if $x_{it} > 0$
- z_t — binary variables taking the value 1 if an order is placed for period t
- B_{it} — the sum of demands for item i from period t to last period T
- c_{itq} — the cost of ordering for period t a quantity of item i that covers its demand from t to q; thus $c_{itq} = s_{it} + \sum_{r=t+1}^{q}(\sum_{k=t}^{r-1} h_{ik})d_{ir}$
- w_{itq} — binary variables taking the value 1 if and only if the quantity of item i ordered for period t covers its demand from t to q; i.e., $w_{it} = 1$ if $x_{it} = \sum_{r=t}^{q} d_{ir}$
- u_{itq} — binary variables taking the value 1 if and only if the demand of item i for period q is included in the order placed for period t.

The classical DJRP formulation is then:

(DJRP1) Minimize
$$\sum_{t=1}^{T}[S_t z_t + \sum_{i=1}^{n}\{s_{it}y_{it} + h_{it}I_{it}\}] \qquad (1)$$
subject to
$$I_{i,t-1} + x_{it} - I_{it} = d_{it} \qquad (i = 1,...,n,\ t = 1,...,T) \qquad (2)$$
$$x_{it} \le My_{it} \qquad (i = 1,...,n,\ t = 1,...,T) \qquad (3)$$
$$\sum_{i=1}^{n} y_{it} \le nz_t \qquad (t = 1,...,T) \qquad (4)$$
$$I_{it} \ge 0 \qquad (i = 1,...,n,\ t = 1,...,T) \qquad (5)$$
$$x_{it} \ge 0 \qquad (i = 1,...,n,\ t = 1,...,T) \qquad (6)$$
$$y_{it} = 0 \text{ or } 1 \qquad (i = 1,...,n,\ t = 1,...,T) \qquad (7)$$
$$z_t = 0 \text{ or } 1 \qquad (t = 1,...,T)\ . \qquad (8)$$

In this formulation the objective function computes the sum of orders and holding costs. Constraints (2) ensure demand satisfaction. Constraints (3) state that the replenishment quantity of an item type can be positive only if that item type is replenished, while constraints (4) mean that individual item types can only be included in a joint replenishment if that replenishment is made. Note that constraints (3) can be tightened if M is replaced with B_{it}.

The following properties of optimal DJRP solutions are valid:

Property 1: Any optimal DJRP solution is such that: $x_{it}^{*} \times I_{it-1}^{*} = 0$ ($i = 1,\ldots,n$; $t = 1,\ldots,T$).

In other words if item type i is replenished at the beginning of period t, it does not pay to hold this item type in stock during period $t - 1$ (Wagner and Within, 1958). An exception may occur if $I_{i0} > 0$.

The next two properties are also due to Wagner and Within (1958):

Property 2: If replenishment policies are restricted to those satisfying Property 1, then the optimal order x_{it}^{*} takes one of the values: d_{it}, $d_{it} + d_{i,t+1}$, …, $\sum_{q=t}^{T} d_{iq}$.

Property 3: If replenishment policies are restricted to those satisfying Property 1, then the optimal inventory level I_{it-1}^{*} takes one of the values: $0, d_{it}$, $d_{it} + d_{i,t+1}$, …, $\sum_{q=t}^{T} d_{iq}$.

The next property is due to Silver (1979):

Property 4: If $d_{iq} \sum_{r=t}^{q-1} h_{ir} > S_q + s_{iq}$ for $q > t$, then it is not optimal to replenish item type i at the beginning of period t.

We now present a new and more compact DJRP formulation based on Property 2. It assigns to the binary variable w_{itq} the value 1 if and only if the replenishment order of item type i at the beginning of period t covers the demand for this item type for all periods until period q, i.e., $w_{itq} = 1$ if and only if $x_{it} = \sum_{r=t}^{q} d_{ir}$. Also, define c_{itq} as the sum of individual ordering and holding costs if item type i replenished at the beginning of period t and covering its demand for periods t through q, i.e., $c_{itq} = s_{it} + \sum_{r=t+1}^{q} \left(\sum_{k=t}^{r-1} h_{ik} \right) d_{ir}$.

(DJRP2) Minimize
$$\sum_{t=1}^{T}[S_t z_t + \sum_{i=1}^{n}\sum_{q=t}^{T} c_{itq} w_{itq}] \qquad (9)$$

subject to
$$\sum_{q=1}^{t}\sum_{r=t}^{T} w_{iqr} = 1 \qquad (i=1,...,n;\ t=1,...,T) \qquad (10)$$

$$\sum_{i=1}^{n}\sum_{q=t}^{T} w_{itq} \le nz_t \qquad (t=1,...,T) \qquad (11)$$

$$w_{itq} = 0 \text{ or } 1 \qquad (i=1,...,n; t=1,...,T; q=1,...,T) \qquad (12)$$

$$z_t = 0 \text{ or } 1 \qquad (t=1,...,T). \qquad (13)$$

In this formulation, Property 4 can also be used to reduce the number of w_{itq} variables. Finally, we introduce yet another new formulation using binary variables u_{itq} equal to 1 if and only if the demand of item i for period q is included in the replenishment at the beginning of period t.

(DJRP3) Minimize
$$\sum_{t=1}^{T} S_t z_t + \sum_{i=1}^{n}\sum_{t=1}^{T} s_{it} u_{itt} + \sum_{i=1}^{n}\sum_{t=2}^{T}\sum_{q=1}^{t-1}\left(\sum_{r=q}^{t-1} h_{ir}\right) d_{it} u_{iqt} \qquad (14)$$

subject to
$$\sum_{q=1}^{t} u_{iqt} = 1 \qquad (i=1,...,n;\ t=1,...,p) \qquad (15)$$

$$\sum_{i=1}^{n} u_{itt} \le nz_t \qquad (t=1,...,T) \qquad (16)$$

$$u_{itq} \le u_{itt} \qquad (i=1,...,n; t=1,...,T-1; q=t+1,...,T) \qquad (17)$$

$$u_{itq} = 0 \text{ or } 1 \qquad (i=1,...,n; t=1,...,T; q=1,...,T) \qquad (18)$$

$$z_t = 0 \text{ or } 1 \qquad (t=1,...,T). \qquad (19)$$

Again, Property 4 can be used to reduce the number of u_{iqt} variables.

3. SUMMARY OF SOME CLASSICAL HEURISTICS

We now describe six basic heuristics for the DJRP, one of which is new.

3.1. The Fogarty and Barringer heuristic

The Fogarty and Barringer (1987) method is one of the earliest heuristics for the DJRP. As will be seen in Section 5, it is also one of the best. It simplifies the problem by adding the requirement that when a replenishment is made, it should cover exactly all demands until the next replenishment. Thus, a solution can be characterized by its replenishment periods. The optimal solution of the simplified problem can be obtained by solving the following dynamic program:

$$f_t = \min_{q \le t}\{f_{q-1} + c_{qt}\} \qquad (t = 2,...,T)$$

subject to
$$f_1 = S_1 z_1 + \sum_{i=1}^{n} s_{i1} y_{i1}$$

$$c_{qt} = S_q z_q + \sum_{i=1}^{n}\{s_{iq} y_{iq} + \sum_{r=q+1}^{t} d_{ir} (\sum_{k=q}^{r-1} h_{ik})\}$$

$$z_q = 1 \text{ if } \sum_{i=1}^{n}\sum_{r=q}^{t} d_{ir} > 0, \text{and } z_q = 0 \text{ otherwise}$$

$$y_{iq} = 1 \text{ if} \sum_{r=q}^{t} d_{ir} > 0, \text{ and } y_{iq} = 0 \text{ otherwise.}$$

3.2. The greedy add heuristic

Federgruen and Tzur (1994) have proposed a greedy add heuristic for the DJRP. A simple implementation of this heuristic is as follows. Initially make a single replenishment at period 1, sufficient to cover the entire demand for each item type for all periods. Then consider in turn each period having no replenishment and compute the saving associated with making an extra replenishment at that period while reducing the previous replenishment accordingly. Repeat until either there are *T* replenishment or until no more savings can be achieved. In details, the greedy add heuristic can be described as follows:

Step 1. Initialization

Set $x_{i1} = \sum_{t=1}^{T} d_{it}$ (i = 1, …, n), and P = {2, …, T}.

Step 2. Determination of best saving

For each $t \in P$, let q_t^- be the latest replenishment period before t and q_t^+ be the period of the first replenishment after t. If there is no replenishment after t, set $q_t^+ = T + 1$. Compute:

$$g_t = \{\sum_{i=1}^{n} (\sum_{r=q_t^-}^{t} h_{ir})(\sum_{r=t}^{q_t^+ - 1} d_{ir})\} - \{S_t + \sum_{i=1}^{n} s_{it} y_{it}\},$$

where
$$y_{it} = 1 \text{ if } \sum_{r=t}^{q_t^+ - 1} d_{ir} > 0, \text{ and } y_{it} = 0 \text{ otherwise.}$$

Step 3. Add replenishment or stop

Remove from P all values of t for which $g_t < 0$. If $P = \varnothing$, stop. Otherwise make a replenishment at period t^* yielding the maximum saving g_t and accordingly reduce the replenishment made at period $q_{t^*}^-$. Go to step 2.

It is worth noting at this point that the reduction of P in Step 3 is justified by the fact that g_t cannot increase in the course of the algorithm. Thus if a period t yields a negative saving g_t at a certain stage it is never interesting to make a replenishment at that period thereafter.

3.3. The greedy drop heuristic

This new heuristic is the natural counterpart of the greedy add heuristic. It starts with a replenishment at each period and iteratively removes replenishments as long as savings can be generated. The notation used is identical to that of the previous heuristic.

Step 1. Initialization

Set $x_{it} = d_{it}$ for $i = 1, \ldots, n$ and $t = 1, \ldots, T$. Set $R = \{1, \ldots, T\}$.

Step 2. Determination of best saving

For each $t \in R$, compute the potential savings:

$$g_t = \{S_t + \sum_{i=1}^{n} s_{it} y_{it}\} - \{\sum_{i=1}^{n} (\sum_{r=q_t^-}^{t} h_{ir})(\sum_{r=t}^{q_t^+-1} d_{ir})\},$$

where $y_{it} = 1$ if $\sum_{r=t}^{q_t^+-1} d_{ir} > 0$, and $y_{it} = 0$ otherwise.

Step 3. Drop replenishment or stop

Remove from R all values of t for such that $g_t<0$. If $R=\emptyset$, stop. Otherwise cancel the replenishment at period t^* yielding the maximum saving g_t and accordingly increase the replenishment made at $q_{t^*}^-$. Go to Step 2. Again, reducing R in Step 3 is justified by the fact that g_t cannot increase during the subsequent iterations of the algorithm.

3.4. The extended Silver-Meal heuristic

We present two versions of an extension of the Silver-Meal (1973) heuristic to the DJRP. According to this heuristic the first replenishment should occur at the first period of the horizon where there is a positive demand. Recall that this heuristic, designed to handle the case of a single item type, computes for each period t the cost SM_t per unit of time of including the demand of period t in the last replenishment. Assuming that it was decided that this last replenishment should take place at period q, then SM_t can be expressed as a function of individual ordering cost s_q as follows:

$$SM_t(s_q) = [s_q + \sum_{r=q+1}^{t} (d_r \sum_{u=q}^{r-1} h_u)]/(t-q+1), \qquad \text{if } t > q,$$

and $$SM_t(s_q) = s_q, \qquad \text{if } t = q.$$

A replenishment at q should cover up to the period $t^* \leq T$ corresponding to the local minimum of SM_t, and if $t^* < T$, the subsequent replenishment should take place at $t^* + 1$.

In the presence of several item types, this procedure must be modified in order to account for the common ordering cost S_t. To do so, this heuristic compute Δ_i, the part of S_t to be attributed to item type i. The Atkins and Iyogun (1988) heuristic makes a replenishment at period 1. Then, assuming that the last replenishment of i is made at q_i, for each subsequent period t the heuristic determines, for each product type i included in a set A_t called the *increasing set*, whether the demand d_{it} should be replenished at t or at q_i. The first version of the extended Silver-Meal heuristic defines A_t *as* $A_t = \{i : SM_{i,t-1}(s_{i,q_i}) < SM_{it}(s_{iq_i})\}$. The steps of this first version are.

Step 1. Initialization

Set $t = 1$,

$q_i = 1$ $(i = 1,\ldots,n)$ (period of the last replenishment for item type i)

$q = 1$ (period of the last replenishment)

$B_q = \{1,\ldots,n\}$ (set of item types included in the replenishment made at period q)

$\overline{B}_q = \{1,\ldots,n\} \setminus B_q$

$SM_{it}(s_{iq_i}) = s_{it}$ $(i = 1,\ldots,n)$.

Step 2. Incrementation

Set $t = t+1$. Compute $SM_{it}(s_{iq_i}) = [s_{iq_i} + \sum_{r=q_i+1}^{t} (d_{ir} \sum_{u=q_i}^{r-1} h_{iu})]/(t-q_i+1)$ for i=1, …, n, and define A_t. For all $i \in A_t \cap \overline{B}q$, check whether $(\sum_{r=q}^{t} d_{ir})(\sum_{r=q_i}^{q-1} h_{ir}) > s_{iq}$. If this inequality holds, then replenish item type i at period q. Add i to B_q.

Step 3. Test for replenishment at period t

For all $i \in A$ compute Δ_i such that $SM_{i,t-1}(s_{iq_i} + \Delta_i) = SM_{it}(s_{iq_i} + \Delta_i)$. If $\sum_{i \in A_t} \Delta_i \geq S_t$, go to Step 4. Otherwise, go to Step 2.

Step 4. Replenishment at period t

Set $q = t$, $B_q = A_t$ and $\overline{B}_q = \{1,...,n\} \setminus B_q$. Also set $q_i = t$ and $SM_{it}(s_{iq_i}) = s_{iq_i}$ for all $i \in A_t$. If $t < T$, go to Step 2. Otherwise, stop.

A second version of the extended Silver-Meal heuristic was proposed by Iyogun (1991). It is similar to the first version, except that

$$A_t = \left\{ i : SM_{i,t-1}(s_{i,q_i}) < SM_{it}(s_{iq_i}) \text{ and } \sum_{r=q_i+1}^{t-1} \left(d_{ir} \sum_{k=q_i}^{t-2} h_{ik} \right) > s_{iq_i} \right\}.$$

3.5. The generalized part-period balancing heuristic

Two generalizations of the part-period heuristic (De Matteis and Mendoza, 1968) were developed by Iyogun (1991). According to these heuristics a first replenishment is made at period 1. Then, assuming that the last replenishment is made at period q, for each subsequent period t, compute the set C_t of item of type i whose holding cost since their last replenishment exceeds s_{it}. Then determine the first period u at which the total holding cost exceeds the total replenishment cost (the sum of the common ordering and individual ordering costs). In the first version of the heuristic, two replenishment options are considered at periods $u - 1$ and u. In the second version, the replenishment takes place at period u. The first version of this heuristic can be described as follows.

Step 1. Initialization

Set: $t = 1$, $q = 1$ and $q_i = 1$ for $i = 1, \ldots, n$.

Step 2. Determination of the candidate replenishment periods

For all $t > q$ and $t \leq T$, determine the set $C_t = \left\{ i : H_{it} = \sum_{r=q_i+1}^{t} d_{ir} \left(\sum_{q=q_i}^{r-1} h_{iq} \right) > s_{it} \right\}$. Let u be the earliest period t for which $\sum_{i \in C_t} (H_{it} - s_{it}) > S_t$. If no such period exists, stop

Step 3. Determination of the next replenishment period

If $\sum_{i \in C_u} (H_{iu} - s_{iu}) - S_u < S_{u-1} - \sum_{i \in C_{u-1}} (H_{i,u-1} - s_{i,u-1})$, replenish all item type of C_u at period u and set $q = u$ and $q_i = u$ for all $i \in C_u$. Otherwise, replenish all item types of C_{u-1} at period $u - 1$ and set $q = u - 1$ and $q_i = u - 1$ for all $i \in C_{u-1}$. Go to step 2.

In the second version of Iyogun's generalization of the part-period balancing heuristic, Step 3 always consists of replenishing all item types of C_u at period u.

3.6. The Silver-Kelle improvement heuristic

Silver and Kelle (1988) described an improvement procedure applicable to any feasible solution. It successively considers each item type ordered at any given period and determines whether a cost saving could be achieved by adding the order quantity of this item type to the previous order.

4. A NEW IMPROVEMENT HEURISTIC BASED ON SOLUTION PERTURBATION

We now propose an improvement procedure that can be applied to any DJRP feasible solution. We have found it pays to apply it to a solution obtained by means of the Fogarty and Barringer (1987) heuristic followed by the Silver-Kelle (1988) improvement heuristic. The proposed procedure is based on the solution perturbation principle introduced by Storer, Wu and Vaccari (1992). It consists of first making a slight random modification to a feasible solution, while maintaining feasibility, and post-optimizing the perturbed solution. The idea is that by doing this, the probability of becoming trapped in a local optimum is reduced. Perturbation was successfully applied to other combinatorial optimization problems (see, e.g., Renaud, Boctor and Laporte, 2002). Our perturbation heuristic can be described as follows.

Step 1. Initial solution

Define the *current solution* as a DJRP feasible solution obtained by any heuristic.

Step 2. Solution perturbation

Repeat the following operations λ times. Choose t randomly in $[1, T]$. If the current solution contains a replenishment at period t, cancel it and combine it with the previous replenishment. Otherwise, add a replenishment at period t including all items contained in the previous replenishment which is then reduced accordingly.

Step 3. Solution improvement

Attempt to improve the perturbed solution by applying the greedy drop heuristic followed by the Silver-Kelle improvement heuristic. If the best known solution has improved, update it. Go to Step 2 until a stopping criterion has been reached.

Steps 2 and 3 are applied as long as the best known solution has not improved for a set number θ of consecutive iterations. In our implementation we used $\lambda = 3$ and $\theta = 6T$. Other values have been tried and these values seem to give better results.

5. COMPUTATIONAL RESULTS

The formulations and heuristics just described were tested on a total of 720 randomly generated instances. These were first divided into four equal subgroups of different instance sizes : $n = 10$, $T = 13$; $n = 10$, $T = 26$; $n = 20$, $T = 13$; $n = 20$, $T = 26$. Each group is made up of six equal subgroups called S1, S2, S3, S4, S5 and S6. For the sake of simplicity, the generated problems are such that the common ordering cost, the individual ordering costs and the inventory holding costs are constant through time; consequently we will drop the index t from the corresponding variables. In all subgroups, we set $S = 1000$ for all t. We then generated the h_i values accordingly to a continuous uniform distribution in [0.1 , 0.6]. Individual ordering costs s_i are randomly generated such that $\sum_{i=1}^{n} s_i / S = \alpha$. The demands d_{it} were obtained by first generating a mean value $\mu_{it} = \left(s_i + 2\tilde{X}S/n\right)/\beta\, h_i$, where $\tilde{X}$ is a random number in [0, 1], and by then setting $d_{it} = 5\lfloor 2\tilde{X}\mu_{it}/5 \rfloor$, where $\lfloor . \rfloor$ is the integer part function. The values of α and β are provided in Table 1 for each subgroup.

Table 1. Values of α and β

Subgroup	α	β
S1	0.5	6.0
S2	0.5	10.0
S3	1.0	6.0
S4	1.0	10.0
S5	2.0	6.0
S6	2.0	10.0

All instances were first solved to optimality by CPLEX 8.0 (on a Pentium III, 1.26GHz under Windows 2000 Server) using the DJRP1, DJRP2 and DJRP3 formulations presented in Section 2. Table 2 gives the average solution time for each of these three formulations for each group and subgroup of instances. The computation times associated with DJRP2 are three times smaller than those of DJRP1. Similarly, DJRP3 cuts the time of DJRP1 by half.

Table 2: Average computational times for the three formulations (in seconds)

	DJRP1	DJRP2	DJRP3
Instance size			
10x13	0.92	0.26	0.24
20x13	3.21	1.07	0.83
10x26	7.21	1.57	3.11
20x26	43.45	13.10	21.31
Subgroup			
S1	9.86	3.93	5.57
S2	18.20	3.85	6.63
S3	9.85	4.13	4.73
S4	15.88	3.99	8.05
S5	10.15	4.11	8.16
S6	18.24	3.97	5.09
General average	13.70	4.00	6.37

All instances were then solved by means of the following eight heuristics :

FB : The Fogarty and Barringer heuristic;

GA : The greedy add heuristic;

GD : The greedy drop heuristic;

SM1 : The extended Silver-Meal heuristic (version 1);

SM2 : The extended Silver-Meal heuristic (version 2);

PB1 : The extended part-period balancing heuristic (version 1);

PB2 : The extended part-period balancing heuristic (version 2);

PH : The perturbation heuristic proposed in Section 4.

The heuristics were coded in Basic and run on a IBM ThinkPad (Pentium III, 750 MHz). All these heuristics, except PH, are followed by the Silver-Kelle (SK) improvement procedure. Recall that PH includes FB and SK. We present in Table 3 the average deviation of each heuristic solution value with respect to the optimum for each group and each subgroup of instances.

This table clearly shows that PH applied after FB and SK produces the smallest average deviation and the largest number of optimal solutions. Otherwise, among the classical heuristics described in Section 3, the Fogarty and Barringer method followed by Silver-Kelle improvement procedure is clearly the best in spite of its relative simplicity. In contrast some of the worst results are produced by the most intricate methods. The effect of applying the Silver-Kelle heuristic is not shown in the table but is always marginal. In contrast, applying PH after FB reduces the average deviation from 0.028 to 0.014 on our test problems. It also produces the smallest maximal deviation from the optimum and the largest number of optimal solutions. All computational times were negligible (less than one second on average).

Table 3. Average deviation with respect to the optimum for tested heuristics

Heuristic	**FB**	**GA**	**GD**	**SM1**	**SM2**	**PB1**	**PB2**	**PH**
Instance size								
10 x 13	0.043	2.163	1.930	2.807	5.714	5.251	2.481	0.001
10 x 26	0.017	1.813	3.334	4.206	6.663	4.977	2.185	0.007
20 x 13	0.035	2.249	2.651	2.641	5.830	4.506	2.221	0.030
20 x 26	0.021	2.024	3.702	2.670	5.188	3.431	2.825	0.019
Subgroup								
S1	0.046	0.830	2.291	2.548	5.712	5.514	2.455	0.028
S2	0.023	3.212	3.225	4.023	5.985	3.738	2.514	0.012
S3	0.021	0.826	2.428	2.024	5.359	5.255	2.305	0.013
S4	0.017	3.453	3.349	3.657	5.545	3.654	2.456	0.011
S5	0.039	0.828	2.377	2.135	5.695	5.338	2.242	0.011
S6	0.028	3.223	3.756	4.099	6.795	3.749	2.596	0.010
Global results								
Average %	0.029	2.062	2.904	3.081	5.848	4.541	2.428	0.014
Min %	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Max %	1.876	9.229	9.556	18.100	20.000	14.718	9.097	0.778
Number of optimal solutions	619	138	41	119	37	26	114	646

6. CONCLUSIONS

The *Dynamic-Demand Joint Replenishment Problem* is central to inventory management. Introduced more than 30 years ago, it has been studied by several authors and a number of heuristics have been proposed for its solution. One contribution of this study has been to provide a unified description, new formulations and, for the first time, a systematic computational comparison of these heuristics. We have shown that our two new integer linear programming formulations for the DJRP are significantly faster than the classical formulation. In addition, all heuristic solution values were compared to each other and to the optimal values. We have shown that the Fogarty-Barringer heuristic, while being relatively simple, offers the best performance. Another contribution of this article has been to propose a new and efficient perturbation heuristic which outperforms all previous methods.

Acknowledgement

This research work was partially supported by grants OPG0036509, OPG0039682 and OPG0172633 from the Canadian National Science and Engineering Research Council. This support is gratefully acknowledged.

REFERENCES

Arkin, E., Joneja, D. and Roundy, R., 1989, Computational complexity of uncapacitated multi-echelon production planning problems, *Operations Research Letters*, 8, 61-68.

Atkins, D. and P. Iyogun, 1988, A Heuristic with lower bound performance guarantee for the multi-product lot-size problem, *IIE Transactions*, 20, 4, 369-373.

De Matteis, J.J. and A.G. Mendoza, 1968, An economic lot sizing technique: The part period balancing algorithm, *IBM Systems Journal*, 7, 1, 30-46.

Erenguc, S., 1988, Multiproduct dynamic lot-sizing model with coordinated replenishment, *Naval Research Logistics*, 35, 1-22.

Federgruen, A., and M. Tzur, 1994, The Joint Replenishment Problem with the time-varying costs and demands: Efficient, asymptotic and ε-optimal solutions, *Operations Research*, 42, 6, 1067-1086

Fogarty, D. and R. Barringer, 1987, Joint order release decisions under dependent demand, *Production and Inventory Management Journal*, 28, 1, 55-61.

Iyogun, P., 1991, Heuristic methods for the multi-product dynamic lot size problem, *Journal of the Operational Research Society*, 42, 10, 889-894.

Kao, E., 1979, A multi-product dynamic lot-size model with individual and joint set-up costs, *Operations Research*, 27, 2, 279-289.

Kirca, Ö., 1995, A primal-dual algorithm for the dynamic lot sizing problem with joint set-up costs, *Naval Research Logistics*, 42, 5, 791-806.

Raghavan, P., 1993, The multi-item lot sizing problem with joint replenishments, Ph. D. Dissertation, Leonard N. Stern School of Business, New York University.

Raghavan, P. and M. Rao, 1991, The multi-item lot sizing problem with joint replenishment: A Polyhedral approach, Technical report SOR-91-8 Department of Statistics and Operations Research, Leonard N. Stern School of Business, New York University.

Raghavan, P. and M. Rao, 1992, Formulations for multi-item lot sizing problem with joint replenishment, Technical report SOR-92-19 Department of Statistics and Operations Research, Leonard N. Stern School of Business, New York University, New York.

Renaud, J., F. F. Boctor and G. Laporte, 2002, Perturbation heuristics for the pickup and delivery traveling salesman problem, *Computers & Operations Research*, 29, 9, 1129-1141.

Robinson, E., and L. Gao, 1996, A dual ascent procedure for multi-product dynamic demand coordinated replenishment with backlogging, *Management Science*, 42, 11, 1556-1564.

Silver, E. and H. Meal, 1973, A heuristic selecting lot size requirements for the case of deterministic time varying demand rate and discrete opportunities for replenishment, *Production and Inventory Management Journal*, 14, 2, 64-74.

Silver, E., 1979, Coordinated replenishments of items under time varying demand: Dynamic programming formulation, *Naval Research Logistics Quarterly*, 26, 1, 141-151.

Silver, E. and P. Kelle, 1988, More on "Joint order release decisions under dependent demand", *Production and Inventory Management Journal*, 29, 2, 71-72.

Storer R., Wu S. D. and Vaccari R., 1992, New search spaces for sequencing problems with application to job shop scheduling. *Management Science*, 38, 10, 1495-1509.

Veinott, J., 1969, Minimum concave-cost solution of Leontief substitution models of multifacility inventory systems, *Operations Research*, 17, 2, 262-291.

Wagner, H. and T. Within, 1958, A dynamic version of the economic lot size model, *Management Science*, 5, 1, 86-96.

Zangwill, W., 1966, A Deterministic multiproduct multifacility production and inventory model, *Operations Research*, 14, 3, 486-507.