Les Cahiers du GERAD ISSN: 0711–2440

Scheduling Multi-Buyer Multi-Item Replenishments: A Constraint Programming Approach

C.K. Chan, A.I. Correa,
B. K-S. Cheung, J.H.W. Lee,
A. Langevin

G–2008–27

March 2008

Les textes publiés dans la série des rapports de recherche HEC n'engagent que la responsabilité de leurs auteurs. La publication de ces rapports de recherche bénéficie d'une subvention du Fonds québécois de la recherche sur la nature et les technologies.

Scheduling Multi-Buyer Multi-Item Replenishments: A Constraint Programming Approach

Chi Kin Chan
Joseph H.W. Lee

Department of Applied Mathematics
The Hong Kong Polytechnic University
Kowloon, Hong Kong
machanck@inet.polyu.edu.hk
joseph.lee@inet.polyu.edu.hk

A.I. Correa
Bernard K-S. Cheung
André Langevin

GERAD and Department de mathématiques et génie industriel
École Polytechnique de Montréal
Montréal (Québec) Canada, H3C 3A7
iacorrea@polymtl.ca
bernard.cheung@gerad.ca
andre.langevin@polymtl.ca

March 2008

Les Cahiers du GERAD

G–2008–27

Abstract

This article addresses the problem of scheduling the replenishment of items to be delivered to multiple buyers from a single provider in order to minimize the total cost including the inventory holding cost, a major ordering cost for each period of replenishment, a minor ordering cost for each item included in a replenishment, and the routing cost. This problem can be modelled as an integer non linear program. The solution methodology is based on Constraint Programming. Experimentation of the proposed model and the solution approach is reported.

Key Words: Constraint programming; Logistics; Joint replenishment; Distribution; Scheduling.

Résumé

Cet article traite du problème de la planification de l'approvisionnement de produits multiples pour plusieurs détaillants à partir d'un fournisseur unique. L'objectif est de minimiser l'ensemble des coûts incluant les coûts des stocks, les coûts de commande et les coûts de transport. Ce problème est modélisé comme un programme non linéaire en nombres entiers qui est résolu par programmation par contraintes. L'article présente des résultats de tests expérimentaux.

1 Introduction

It is a common practice for a multi-branch firm to have its branches, each of which is independent and has a different demand pattern, ordering a group of items from a single supplier. An example is a large bank with many branches that may be required to replenish a great variety of forms from a single supplier. This is a multi-buyer multi-item problem. Obviously, coordinating the replenishments among the branches would reduce the firm's ordering and inventory costs.

The Joint Replenishment Problem (JRP) is the multi-item inventory problem of coordinating the replenishment of a group of items that may be jointly ordered from a single supplier. Each time an order is placed, a major ordering cost is incurred, independent of the number of items ordered. Furthermore, a minor ordering cost is incurred whenever an item is included in a replenishment order. Joint replenishment of a group of items reduces the number of times that the major ordering cost is charged, and therefore this saves costs. In the deterministic joint replenishment problem it is assumed that the major ordering cost is charged at a basic cycle time and that the ordering cycle of each item is some integer multiple of this basic cycle.

The JRP was first modelled as a single-buyer one. As early as 1974, Goyal proposed an enumerative procedure to obtain an optimal solution. Substantial computation is required and the running time of this procedure grows exponentially with the number of items. Subsequently many heuristics were developed to generate a near-optimal replenishment policy with less computation. Aksoy and Erenguk (1988) and Goyal and Satir (1989) conducted a survey of these procedures. Kaspi and Rosenblatt (1991), Goyal and Deshmukh (1993), and Hariga (1994) later further supplemented them. Viswanathan (1996) proposed an optimal algorithm by which tighter bounds for the optimal value of the basic period can be found, and hence the computational effort for finding the optimal solution is reduced. Wildeman et al. (1997) presented another optimal approach based on global optimisation theory. Musalem and Dekker (2005) presented a case study on joint replenishment policies in which they analyzed the effect of minimum order quantities and a complex transportation structure on inventories. For a review of the literature on JRP, see Khouja and Goyal (2008).

Most of the literature has considered the JRP as a problem of coordinating the replenishment of a group of items of an individual buyer. In real-life situations, however, it is very usual for a multi-branch firm to order the same group of items for all branches from the same supplier. An example is a large bank with many branches that may be required to replenish a great variety of forms from a single supplier. The problem becomes a multi-buyer multi-item JRP. One may expect a cut in costs if the joint replenishments among the buyers are well coordinated. This is an area that has not received much attention until recently.

Chan et al. (2003) have proposed a modified genetic algorithm (GA) which is capable of solving the multi-buyer multi-item JRP efficiently and accurately. The GA has been shown to be a good method for the JRP. Khouja et al. (2000) experimented on 1600 randomly generated problems, comparing GA favorably with the best available heuristic, known as RAND, for solving the JRP. Their results showed that the GA can provide better solutions than the RAND for some problems, and never worse than it by more than 0.08%. But more important is that the GA has the advantages of being easy to implement, understand and modify, and being able to deal with constrained JRPs.

When the replenishment costs of multi-buyer multi-item replenishments contain a transportation cost component, cost saving may also be achieved due to the ability to share truck and loading equipment between the items and buyers. In addition, the design of a vehicle route for visiting a group of buyers may have significant effect on the magnitude of the replenishment costs. Hence, it is desirable to design an efficient joint replenishment strategy that co-ordinates inventory control and transportation planning. Viswanathan and Mathur (1997) considered the integration of vehicle routing and inventory decisions for a single warehouse, nulti-retailer, nulti-product distribution system with deterministic demands. Qu et al. (1999) considered multi-item joint replenishment in an inbound material-collection system with a central warehouse using uncapacitated vehicles, where geographically dispersed suppliers face stochastic demands. For a review of multi-item inventory and vehicle routing systems up to 1999, we refer the reader to Buffa and Munn (1990), Ben-Khedher and Yano (1994), and Bertazzi and Speranza (1999). More recently, Sindhuchao et al. (2005) developed a mathematical formulation model that integrates the inventory replenishment and transportation decisions for an inbound commodity collection system with one warehouse, multiple supplier, and multiple items. The authors considered the problem in a deterministic setting, and assumed that the items are replenished according to an economic order quantity policy. Chan et al. (2006) developed a solution approach to scheduling deliveries from a supplier to many buyers who use the JRP in replenishing their inventory from the supplier. Sarmah et al. (2008) studied the interaction between transportation and inventory costs in the context of supply chain co-ordination of a single-manufacturer multiple heterogeneous buyers situation.

The problem addressed in this article is defined as follows. The **Multi-buyer Multi-item Replenishment Scheduling Problem (MMRSP)** consists of finding the cycle of replenishment and each period of replenishment of each item for each buyer in order to minimize the sum of the inventory holding cost, the major set-up cost for each period with a replenishment, the minor ordering cost of each item for each buyer, and the routing cost to visit the buyers scheduled on each period. It is considered that there is at most one delivery per period to each buyer. The planning horizon consists of an integer number of periods (a period is typically a day). Hence the replenishment cycle lengths are expressed as integers.

The paper is organised as follows. In the second section we present an integer linear programming formulation of the MMRSP. In the third section a constraint programming formulation of the problem is proposed. This second model contains many logical constraints and binary variables. Some preliminary tests displayed time consuming computation efforts. In the fourth section we propose a reformulation of the constraint programming model which presents features like: compactness (concise mathematical model with fewer constraints and variables), aggregated cost-driven variables (counter variables), absence of binary variables, a very efficient search strategy based on priority on variables and no logical constraints. This integer non linear model is solved using the constraint programming tool ILOG OPL 5.2. Section 5 presents the experimentation. Section 6 describes a potential avenue for future research. A conclusion follows.

2 A mathematical programming formulation of the MMRSP

In this section we present a mathematical programming formulation of the MMRSP. This formulation is used to build in Section 3 a CP formulation.

Sets and parameters of the MMRSP model

L: Planning horizon.
$Buyers$: Set of buyers.
$nbBuyers$: Number of buyers.
$Items$: Set of items.
$nbItems$: Number of items.
$d_{i,j}$: Demand per unit time of item i for buyer j.
S: Major set-up cost.
$s_{i,j}$: Small set-up cost for ordering item i for buyer j.
R: Routing cost.
r: Inventory holding charge, in percent per unit time.
v_i: Unit value of item i.
$Cycles$: Set of all possible values of the basic cycle time interval between two replenishments of item i for buyer j ($k_{i,j}$).

The set *Cycles* may contain all integers from 1 to $L/2$ or be restricted to the factors of L if one wants an integer number of cycles for each item during the planning horizon.

Variables of the MMRSP model

$k_{i,j}$: Basic cycle time interval between two replenishments of item i for buyer j.
$F_{i,j}$: First period of replenishment of buyer j with item i.
$Y_p^{i,j}$: Binary variable equal to 1 if buyer j is replenished with item i at period $p \in \{0, 1, ..., L-1\}$, 0 otherwise.
Z_p^j: Binary variable equal to 1 if buyer j is replenished with at least one item at period $p \in \{0, 1, ..., L-1\}$, 0 otherwise.
X_p: Binary variable equal to 1 if there is at least one replenishment at period $p \in \{0, 1, ..., L-1\}$, 0 otherwise.

The objective function consists in the sum of the inventory holding, major and minor ordering, and routing costs.

$$\begin{aligned} Min \quad & L * \sum_{i \in Items, j \in Buyers} ((r/365) * d_{i,j} * v_i * k_{i,j})/2 + \sum_{i \in Items, j \in Buyers, p \in [0..L-1]} Y_p^{i,j} * s_{i,j} \quad + \\ & S * \sum_{p \in [0..L-1]} X_p + R * \sum_{j \in Buyers, p \in [0..L-1]} Z_p^j \end{aligned}$$

s.t.

$$F_{i,j} \leq k_{i,j} - 1 \qquad \forall i \in Items, j \in Buyers \tag{1}$$

$$Y_{F_{i,j}}^{i,j} = 1 \qquad \forall i \in Items, j \in Buyers \tag{2}$$

$$(p < F_{i,j}) \Rightarrow \left(Y_p^{i,j} = 0\right) \qquad \forall i \in Items, j \in Buyers, p \in [0..L-1] \tag{3}$$

$$Y_p^{i,j} = Y_{pmodk_{i,j}}^{i,j} \qquad \forall i \in Items, j \in Buyers, p \in [1..L-1] \tag{4}$$

$$Y_p^{i,j} \leq Z_p^j \qquad \forall i \in Items, j \in Buyers, p \in [0..L-1] \tag{5}$$

$$Z_p^j \leq X_p \qquad \forall j \in Buyers, p \in [0..L-1] \tag{6}$$

$$(k_{i,j} = k_{m,j}) \Rightarrow (F_{i,j} = F_{m,j}) \qquad \forall j \in Buyers, i, m \in Items : i < m \tag{7}$$

$$k_{i,j} \in Cycles \qquad \forall i \in Items, j \in Buyers \tag{8}$$

$$F_{i,j} \geq 0 \quad integer \qquad \forall i \in Items, j \in Buyers \tag{9}$$

$$X_p, Y_p^{i,j}, Z_p^j \quad binary \qquad \forall i \in Items, j \in Buyers, p \in [0..L-1] \tag{10}$$

Constraints (1) ensure that the first replenishment of item i for buyer j is before period $k_{i,j}$. Constraints (2) ensure that at the first period of replenishment of buyer j with item i (that is at $F_{i,j}$), variable $Y_{F_{i,j}}^{i,j}$ is set to 1. Constraints (3) ensure that, before the first period of replenishment of any buyer with any item the corresponding Y variable is set to 0, (there are no replenishment,). Constraints (4) ensure that the replenishments of buyer j with product i are performed in a cyclic way (the length of the cycle is given by k_{ij}). Constraints (5) enforce that if buyer j is replenished with at least one item then $Z_p^j = 1$. Constraints (6) ensure that if at least one buyer is replenished with any item at period p then $X_p = 1$. Constraints (7) are *synchronizing* constraints. They ensure that if for a given buyer, two items have the same replenishment cycle, they should be synchronized, i.e be replenished at the same periods. This would minimize the major set cost and the routing costs. Constraints (8), (9), and (10) define the variables.

The preceding formulation is computationally intractable because of non-linearities (constraints (3) and (7)) and the size of the model. Moreover, in constraints (2) and (4) a subscript of Y is a variable. To circumvent these difficulties, a CP formulation is developed.

3 A first constraint programming formulation

Constraint Programming is a solution paradigm that differs from the mathematical programming approaches in many aspects. Originally CP was used essentially for satisfiability problems. A CP model defines a domain for each variable and a set of constraints (called *constraints store*). The constraints are not considered globally as in mathematical programming where they are treated as a set of constraints (the constraint matrix), but one at a time. Each constraint is used in turn to eliminate infeasible values for each variable. This is called domain reduction and continues until there is no more possible domain reduction. Either a feasible solution is found or not. When a feasible solution is found, the corresponding value of the objective function is evaluated. A new constraint is added to the constraint store (objective < value of the best solution found so far) to search for another feasible solution with a better objective function value. Then an exploration tree is developed similarly to the Branch and Bound exploration in mathematical programming, except that a new constraint is added to the constraint store whenever a better solution is found.

The exploration of the search tree (search strategy) is crucial in CP and has to be designed according to the problem at hand. The types of constraints that can be used in CP are much richer than the ones in mathematical programming. This richness is exploited in the next formulation. The CP formulation uses the same parameters and variables as the MIP formulation. In addition, we use the following upper bounds.

o_1 : Objective function value when there is a daily replenishment for each item.

o_2 : Objective function value when there is exactly one replenishment for each item in the horizon.

A variable *obj* is defined for the objective function value:

$$obj = L * \sum_{i \in Items, j \in Buyers} ((r/365) * d_{i,j} * v_i * k_{i,j})/2 + \sum_{i \in Items, j \in Buyers, p \in [0..L-1]} Y_p^{i,j} * s_{i,j} + S * \sum_{p \in [0..L-1]} X_p + R * \sum_{j \in Buyers, p \in [0..L-1]} Z_p^j$$

Then the constraint programming model is defined as follows:

CP1

$$Min \quad obj$$

s.t.

$$F_{i,j} \leq k_{i,j} - 1 \qquad \forall i \in Items, j \in Buyers \qquad (1)$$

$$(k_{i,j} = k_{m,j}) \Rightarrow (F_{i,j} = F_{m,j}) \qquad \forall j \in Buyers, i, m \in Items : i < m \qquad (7)$$

$$obj \leq min(o_1, o_2) \qquad (11)$$

$$Y_p^{i,j} = ((p \geq F_{i,j}) \text{ and } ((p - F_{i,j}) \bmod k_{i,j} = 0)) \qquad \forall i \in Items, j \in Buyers, p \in [0..L-1] \qquad (12)$$

$$Z_p^j = \bigcup_{i \in Items} \left(Y_p^{i,j} = 1\right) \qquad \forall j \in Buyers, p \in [0..L-1] \qquad (13)$$

$$X_p = \bigcup_{j \in Buyers} \left(Z_p^j = 1\right) \qquad \forall p \in [0..L-1] \qquad (14)$$

with variable definitions (8), (9), (10).

Constraint (11) sets an upper bound on the objective function. These are used as a (weak) way of starting from known feasible solutions to prune the search tree. Constraints (12,13,14) replace constraints (4,5,6). From a constraint programming point of view, (12,13,14) are computationally more efficient than (4,5,6). With constraints (12) the $Y_p^{i,j}$ variables equals 1 when there is a replenishment at period i. Each constraint of this type is a logical constraint whose right-hand side is a conjunction of two constraints. Any of these two right-hand side constraints takes the truth values of 1 or 0 depending on whether it is satisfied or not. Constraints (13) enforce that Z variables are equal to 1 whenever there is one item delivered to buyer j at period p, while constraints (14) ensure that X variables are equal to 1 whenever a delivery is done on period p.

Here, the transportation costs, as calculated in the objective function, are considered proportional to the number of visited customers. However, one could model the transportation costs if they are considered proportional to the distance travelled using the approximation formula for the Travelling Salesman Problem of Beardwood et al. (1959):

$$Length = K\sqrt{nA} = K\sqrt{A}\sqrt{n}.$$

In this formula, n is the number of points to visit and A is the area containing all the points. K is a constant that depends on the metric used. In our case, this formula could be written as follows:

$$K' \sqrt{\sum_{j \in Buyers, p \in [0..L-1]} Z_p^j}$$

Here K' corresponds to $K\sqrt{A}$.

The analysis of preliminary results of this version of the MMRSP yields interesting information:

- It takes 10 seconds to generate a first feasible solution.
- The size of the model is a major concern : there is a large number of logical constraints (and a lot of additional variables are generated behind the scene during the preprocessing phase).
- The upper bounds on the objective function value are not good.
- The domain reduction is poor.

To circumvent these difficulties, an enhanced model is developed. This new formulation avoids the use of binary variables which are usually not efficient in CP.

4 A second Constraint Programming formulation of the MMRSP

The main features of the reformulated MMRSP are the following:

1. The binary variables are replaced by *counter variables* to aggregate the information. X, Y and Z are replaced by four integer variables based on the cost structure.
2. The logical constraints are dropped.
3. The bounds on the objective function are dropped since they are not tight enough.
4. A very concise model is finally obtained (the period index disappears from the variables but the the first period of replenishment variables F_{ij} are still present).

Let the variables $C_{i,j}$ be the total number of times the buyer j is replenished with item i within the horizon $[0..L-1]$:

$$C_{i,j} = \sum_{p \in [0..L-1]} Y_p^{i,j}$$

Then:

$$C_{i,j} = \left\lfloor \frac{L}{k_{i,j}} \right\rfloor + 1 \qquad \forall\, i \in Items, j \in Buyers \tag{15}$$

The preceding equation is valid only if the only possible values for $k_{i,j}$ are factors of L. However if it is not the case, then the number of replenishments of item i for buyer j depends also on $F_{i,j}$ and the following equation should be used.

$$C_{i,j} = \left\lfloor \frac{L - 1 - F_{i,j}}{k_{i,j}} \right\rfloor + 1 \qquad \forall\, i \in Items, j \in Buyers \tag{16}$$

Recall constraints (5) of the MP formulation of the MMRSP. By summing twice on both sides of (5) on the range [0 .. L-1], and on the set of Buyers and combining them with (15), we obtain:

$$\sum_{j \in Buyers} C_{i,j} \leq \sum_{j \in Buyers} \sum_{p \in [0..L-1]} Z^j_p \qquad \forall\, i \in Items, j \in Buyers, p \in [0..L-1] \tag{17}$$

Let the variable B be the total number of visits during the horizon.

$$B = \sum_{j \in Buyers} \sum_{p \in [0..L-1]} Z^j_p.$$

Then (17) implies:

$$\sum_{j \in Buyers} C_{i,j} \leq B \qquad \forall\, i \in Items$$

This last set of inequalities can be written as the following single one.

$$B = max_{i \in Items} \sum_{j \in Buyers} C_{i,j} \tag{18}$$

Now recall constraints (6) of the MP formulation of the MMRSP. The combination of (5) and (6) implies:

$$Y^{i,j}_p \leq X_p \qquad \forall\, i \in Items, j \in Buyers, p \in [0..L-1]$$

By summing on both sides on the range [0 .. L-1], we obtain:

$$C_{i,j} \leq \sum_{p \in [0..L-1]} X_p \qquad \forall\, i \in Items, j \in Buyers$$

Let the variable A be the total number of periods with replenishments:

$$rA = \sum_{p \in [0..L-1]} X_p$$

Then:

$$C_{i,j} \leq A \qquad \forall\, i \in Items, j \in Buyers$$

Again, this set of inequalities can be rewritten as

$$A = max_{i \in Items, j \in Buyers} C_{i,j} \tag{19}$$

In summary, we use the following variables to reformulate the MMRSP model:

$k_{i,j} \in [1..L]$	Basic cycle time interval between two replenishments of item i of buyer j
$F_{i,j} \in [1..L-1]$	First period of replenishment of buyer j with item i.
$C_{i,j} \in [1..L]$	The total number of replenishments of buyer j with item i.
$A \in [1..L]$	The total number of replenishments periods.
$B \in [1..L * nbBuyers]$	The total number of visits during the horizon.

Then the revised constraint programming model is as follows.

CP2

$$Min \quad L * \sum_{i \in Items, j \in Buyers} ((r/365) * d_{i,j} * v_i * k_{i,j})/2+$$
$$\sum_{i \in Items} \sum_{j \in Buyers} C_{i,j} * s_{i,j} + S * A + R * B$$

s.t.

$$F_{i,j} \leq k_{i,j} - 1 \qquad \forall\, i \in Items, j \in Buyers \qquad (1)$$
$$(k_{i,j} = k_{m,j}) \Rightarrow (F_{i,j} = F_{m,j}) \qquad \forall\, j \in Buyers, i, m \in Items : i < m \qquad (7)$$
$$C_{i,j} = \left\lfloor \frac{L}{k_{i,j}} \right\rfloor + 1 \qquad \forall\, i \in Items, j \in Buyers \qquad (15)$$
$$B = max_{i \in Items} \sum_{j \in Buyers} C_{i,j} \qquad (18)$$
$$A = max_{i \in Items, j \in Buyers} C_{i,j} \qquad (19)$$

with definitions of variables (9-10) and $A, B, C_{i,j} \geq 0$ and integer.

In this reformulated model the objective is still to minimize the sum of inventory holding, major ordering, minor ordering, and routing costs. Constraints (15 defines the variables $C_{i,j}$ in terms of $k_{i,j}$. Constraint (18) defines B the total number of visits during the horizon. Constraint (19) defines A the total number of replenishments periods.

It is interesting to see that the reformulated model does not contain any explicit reference to the period index. The reformulated model is thus very concise. From a constraint programming point of view, this model is more efficient since there are no explicit reference to any binary variables. It is often very recommended to avoid as much as possible the use of any binary variable since their presence hinders the performance of the model.

5 Experimentation

To solve the constraint programming models CP1 and CP2, the ILOG CP Optimizer 1.0 was used. Models CP1 and CP2 were tested on problems with the number of items ranging from 10 to 26 and the numbers of buyers from 7 to 26. The horizon length L is 120 periods (days). The Inventory carrying charge r was set at 10% per year. The major set-up cost S equals \$40 and the minor set-up costs $s_{i,j}$ fixed at 1. The routing cost R equals % 5. The values of the demand matrix d and the item value matrix range from 1 to 30 and from 1 to 20 respectively.

For the search strategy, various orders of variable instantiation have been tried. The most efficient one has been instantiation of C first, F second, and then k. For each set of variables, priority is set on the variable with the largest impact on the objective function (as proposed by Refalo, 2004). Selection of value is also based on the greatest impact on objective first. For the search type, we used the Restart method (Deep First Search with restart after a certain number of failures (20 in our case). The Restart method was introduced by Refalo (2004).

As discussed earlier, model CP1 presented many pitfalls: an astronomical number of constraints generated by the system during the solution process, a very large number of binary variables that are not efficient with constraint programming, and a slow convergence. On the other hand, CP2 did not present those pitfalls. We present hereafter the results of CP2.

Table 1 summarizes the results. The columns of that table shows for each problem the number of items, the number of buyers, the number of variables (note that additional variables are generated in the course of the solution process), the number of constraints, and the number of faisable solutions found. The last three columns shows the objective value after 5, 15, and 60 minutes of solution time respectively.

In all the problems, the constraint programming algorithm quickly improve the objective value, and in most problems the best solution is found after 15 minutes of computation. However the optimality is not proven even after one hour of computation. It should be reminded that the problem is a large scale integer non-linear mathematical program and that type of problem is very difficult to solve. Hence the results presented here are satisfactory. Figure 1 illustrates the typical solution process in terms of objective improvement.

Table 1: Results of tests.

					Objective	**Objective**	**Objective**
				Faisable	**value**	**value**	**value**
Items	**Buyers**	**Variables**	**Constraints**	**solutions**	**(5 min)**	**(15 min)**	**(60 min)**
10	16	482	4482	884	3916.2	3916.2	3916.2
10	22	662	6116	1407	5683.9	5661.6	5661.6
10	25	752	6946	1602	6469.9	6169.2	6169.2
15	7	317	3965	548	2185.2	2185.2	2185.2
16	10	482	6404	1105	3368.4	3368.4	3368.4
18	10	542	7924	1450	3449.9	3449.9	3449.9
26	14	9094	21786	588	13827.7	9977.3	9977.3
26	15	1172	23400	208	16648.5	13771.8	8341.7

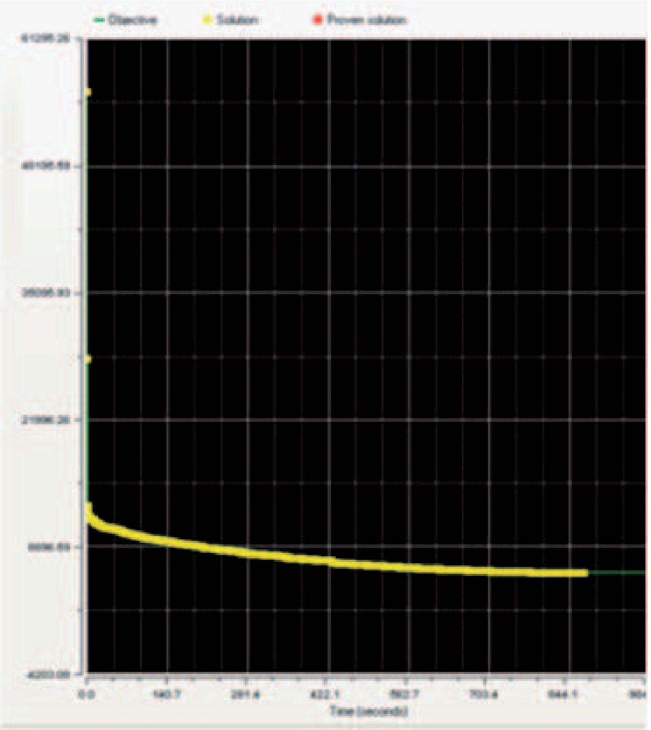

Figure 1: Sequence of objective function values for problem with 10 items and 25 buyers.

6 Avenue of further investigation

Constraint programming allows modelling constraints that are difficult if not impossible to model in standard mathematical programming. For instance complex logic constraints can be easily modelled. However, solving models with such complex constraints may be hard, especially if there are a large number of them. One reason is that many CP solvers will generate automatically a new 'logic' variable for each logic constraint which make convergence of the process more difficult. To cope with this increase difficulty, one has to design a specific search strategy. We present here another version of the MMMRSP model with the variable A and B of model CP2 expressed in terms of F and k.

CP3

$$\begin{aligned} Min \quad L * \sum_{i \in Items, j \in Buyers} & ((r/365) * d_{i,j} * v_i * k_{i,j})/2 + \\ & \sum_{i \in Items} \sum_{j \in Buyers} C_{i,j} * s_{i,j} + S * A + R * B \end{aligned}$$

s.t.

$$F_{i,j} \leq k_{i,j} - 1 \qquad \forall\, i \in Items, j \in Buyers \tag{1}$$

$$(k_{i,j} = k_{m,j}) \Rightarrow (F_{i,j} = F_{m,j}) \qquad \forall\, j \in Buyers, i, m \in Items : i < m \tag{7}$$

$$C_{i,j} = \left\lfloor \frac{L - 1 - F_{i,j}}{k_{i,j}} \right\rfloor + 1 \qquad \forall\, i \in Items, j \in Buyers \tag{16}$$

$$B = \sum_{p \in [0..L-1]} \sum_{j \in Buyers} \left(\left(\sum_{i \in Items} (F_{i,j} = p mod k_{i,j}) \right) \geq 1 \right)$$

$$A = \sum_{p \in [0..L-1]} \left(\left(\sum_{j \in Buyers} \sum_{i \in Items} (F_{i,j} = p mod k_{i,j}) \right) \geq 1 \right)$$

with definitions of variables (9–10) and $A, B, C_{i,j} \geq 0$ and integer.

Although the definition of A and B is very concise and based on logic expressions, because of the numerous variables generated when solving the problem the algorithm crashes after around 15 minutes of computer time. An interesting avenue of research would be to elaborate a specific search strategy for this model.

7 Conclusions

Most of the literature has only considered the joint replenishment of a group of items ordered by an individual buyer. Hence the buyer's joint replenishment has been treated in isolation. However, it is not uncommon that a multi-branch firm has all its branches, i.e. buyers, ordering the same group of items from the supplier. This becomes a multi-buyer multi-item problem. Clearly there are potential opportunities to reduce costs by coordinating the joint replenishments among the buyers. If the replenishment costs of the buyers contain a transportation cost component, it is desirable to design an efficient joint replenishment strategy that co-ordinates inventory control and transportation planning so as to further reduce the costs. This problem corresponds to an integer non linear program which is very difficult to address. This paper has successfully developed a constrained programming model for the problem and devised a very efficient search strategy for its solution. The results of our experimentation show that very good solution (optimal or nearly optimal) can be obtained very quickly. Future research could encompass the design of more efficient and particular search strategies.

References

Aksoy, Y., Erenguk, S., 1988. Multi-item inventory models with coordinated replenishments: A survey. International Journal of Production and Operations Management 8, 63–73.

Beardwood, J., Halton, J.H., Hammersley, J.M., 1959. The shortest path through many points. Proceedings of the Cambridge Philological Society 55, 299–327.

Ben-Khedher, N., Yano, C.A., 1994. The multi-item joint replenishment problem with transportation and container effects. Transportation Science 28 (1), 37–47.

Bertazzi, L., Speranza, M.G., 1999. Minimizing logistic costs in multistage supply chains. Naval Research Logistics 46, 399–417.

Buffa, F.P., Munn, J.R., 1990. Multi-item grouping algorithm yielding near-optimal logistics cost. Decision Sciences 21 (1), 14–34.

Chan, C.K., Cheung, B.K-S., Langevin A., 2003. Solving the multi-buyer joint replenishment problem with a modified genetic algorithm. Transportation Research Part B 37, 291–299.

Chan, C.K., Li, L.Y., Ng, C.T., Cheung, B.K-S., Langevin, A., 2006. Scheduling of multi-buyer joint replenishments. International Journal of Production Economics 102, 132–142.

Dagpunar, J.S., 1982. Formulation of a multi item single supplier inventory problem. Journal of the Operational Research Society 33 (3), 285–286.

Goyal, S.K., 1974. Determination of optimum packaging frequency of items joint replenished. Management Science 21, 436–443.

Goyal, S.K., Deshmukh, S.G., 1993. A note on the economic ordering quantity of jointly replenished items. International Journal of Production Research 31, 2959–2962.

Goyal, S.K., Satir, A.T., 1989. Joint replenishment inventory control: deterministic and stochastic models. European Journal of Operational Research 38, 2–13.

Kaspi, M., Rosenblatt, M.J., 1991. On the economic ordering quantity for jointly replenished items. International Journal of Production Research 29, 107–114.

Khouja, M., Goyal, S., 2008. A review of the joint replenishment problem literature: 1989–2005. European Journal of Operational Research 186, 1–16.

Khouja, M., Michaelwicz Z., Satoskar, S.S., 2000. A comparison between genetic algorithms and the RAND method for solving the joint replenishment problem. Production Planning and Control 11, 556–564.

Hariga, M., 1994. Two new heuristic procedures for the joint replenishment problem. Journal of the Operational Research Society 45, 463-471.

Musalem, E.P., Dekker, R., 2005. Controlling inventories in a supply chain: a case study. International Journal of Production Economics 93–94, 179–188.

Porras, E., Dekker, R., 2003. An efficient optimal solution method for the joint replenishment problem with minimum order quantities. Report Series Econometric Institute, Erasmus University, Rotterdam, EI2003-52.

Qu, W.W., Bookbinder, J.H., Iyogun, P., 1999. An integrated inventory-transportation system with modifried periodic policy for multiple products. European Journal of Operational Research 115, 254–269.

Refalo, P. 2004. Impact-based search strategies for constraint programming. 10th International Conference, CP 2004, Toronto, Canada, September 27–October 1, Principles and Practice of Constraint Programming–CP 2004, Volume 3258, Lecture Notes in Computer Science, Springer Berlin, 557–571.

Sarmah, S.P., Acharya, D., Goyal, S.K., 2008. Coordination of a single-manufacturer/multi-buyer supply chain with credit option. International Journal of Production Economics 111, 676–685.

Sindhuchao, S., Romeijn, H.E., Akcali, E., Boondiskulchok, R., 2005. An integrated inventory-routing system for multi-item joint replenishment with limited vehicle capacity. Journal of Global Optimization 32, 93–118.

Viswanathan, S., 1996. A new optimal algorithm for the joint replenishment problem. Journal of the Operational Research Society 47, 936–944.

Viswanathan, S., Mathur, K., 1997. Integrating routing and inventory decisions in one-warehouse multi-retailer multi-product distribution systems. Management Science 43 (3), 294–312.

Wildeman, R.E., Frenk, J.B.G., Dekker, R., 1997. An efficient optimal solution method for the joint replenishment problem. European Journal of Operational Research 99, 433–444.