ELSEVIER

European Journal of Operational Research 99 (1997) 433–444

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Theory and Methodology

An efficient optimal solution method for the joint replenishment problem

R.E. Wildeman *, J.B.G. Frenk, R. Dekker

Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands

Received Decmber 1995; accepted March 1996

Abstract

During the last two decades, many heuristic procedures for the joint replenishment problem have appeared in the literature. The only available optimal solution procedure was based on an enumerative approach and was computationally prohibitive. In this paper we present an alternative optimal approach based on global optimisation theory. By applying Lipschitz optimisation one can find a solution with an arbitrarily small deviation from an optimal value. An efficient procedure is presented which uses a dynamic Lipschitz constant and generates a solution in little time. The running time of this procedure grows only linearly in the number of items. © 1997 Elsevier Science B.V.

Keywords: Inventory; Joint replenishment; Multi-item; Global optimisation; Lipschitz optimisation

1. Introduction

In an inventory system with multiple items, cost savings can be obtained when the replenishment of several items are coordinated. 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 so 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.

Over the last two decades, the joint replenishment problem has received much attention. An optimal solution approach was presented by Goyal [3]. However, this approach is based on enumeration and the running time of the procedure grows exponentially with the number of items. Furthermore, as Van Eijs [15] has pointed out, the lower bound on an optimal cycle time as used by Goyal [3] does not guarantee a global optimal solution; Van Eijs [15] proposes a modified lower bound which guarantees an optimal solution. However, the resulting algorithm often requires even more computation time than the procedure of Goyal [3], except for large values of the major ordering cost. The enumeration approach by Goyal [3] and Van Eijs [15] is therefore only practical for few items. Klein and Ventura [12] consider the problem in discrete time, and they derive a lower and an upper bound on an optimal (discrete) cycle time. Subsequently, the authors apply enumeration on all intermediate discrete values between these bounds. This method, however, does not guarantee an optimal solution, since the authors apply

* Corresponding author.

PII S0377-2217(96)00072-0

the same wrong lower bound as Goyal [3]. Furthermore, the enumeration causes the running time to be dependent on the discretisation; when a more refined discretisation is applied, the running time increases proportionally.

Because of these enumeration problems, the attention has been focused on heuristic procedures. We refer the reader to the review article by Goyal and Satir [6] for an overview of the many heuristic procedures that have appeared in the literature up to 1989. After that, Kaspi and Rosenblatt [11] proposed an approach based on trying several values of the basic cycle time between a minimum and a maximum value, and then applying for each value a heuristic of Kaspi and Rosenblatt [10], which is a modified version of the algorithm of Silver [14]. The authors show in an extensive simulation experiment that their procedure outperforms all available heuristics. Later, Goyal and Deshmukh [5] proposed an improvement of the lower bound used by Kaspi and Rosenblatt [11]. The idea of trying multiple cycle times is arbitrary; indeed, how many values should be tried to obtain a good solution? Hariga [8] proposed a heuristic based on a relaxation of the problem. In Ben-Daya and Hariga [1] it is shown that the approach of Hariga [8] mostly outperforms the algorithm of Goyal and Deshmukh [5], though the differences are small.

As can be understood from the above, research into heuristic procedures for the joint replenishment problem has been a lively field and much improvement can hardly be expected. However, as with all heuristics, one cannot guarantee the quality of the generated solutions, however good the performance seems to be from simulation experiments. Consequently, a significant contribution to the rich literature on the problem can only come from a solution procedure that generates a solution in little time and that guarantees the quality of this solution. Therefore, we will not concentrate on new heuristics for the joint replenishment problem. Instead, we will focus in this paper on the global optimisation of the problem and we will show that by using Lipschitz optimisation the problem can efficiently be solved to optimality. We present a solution procedure that generates in little time a solution with an arbitrarily small deviation from an optimal value. The procedure requires a running time that grows only linearly in the number of items.

This paper is organised as follows. In the next section we give the problem formulation and in Section 3 we analyse the problem in more detail. Subsequently, in Section 4, we introduce a relaxation and present a solution procedure using Lipschitz optimisation. In Section 5 we present numerical results and in Section 6 we draw conclusions.

2. Problem formulation

In the formulation of the joint replenishment problem the following assumptions are made:

1. The demand rate for each item is known and constant;
2. Shortages are not allowed;
3. The replenishment rate is infinite (zero lead time);
4. There is an infinite horizon.

Moreover, to discuss the joint replenishment problem, we introduce the following notation:

- S major ordering cost,
- s_i minor ordering cost of item i,
- n number of items,
- D_i demand of item i per unit of time,
- h_i holding cost per unit of item i per unit of time,
- T basic cycle time,
- k_i frequency of ordering item i.

According to the above definitions it is clear that k_iT is the cycle time of item i.

Let now $\Phi_i(k_iT)$ denote the individual average costs of item i, when it is ordered each k_iT time units. The average costs consist of the average (minor) ordering cost and the average holding cost of item i and so we have

$$\Phi_i(k_iT) = \frac{s_i}{k_iT} + \frac{h_iD_i}{2}k_iT. \tag{1}$$

Observe that the individual average cost function $\Phi_i(\cdot)$ of item i is equal to the objective function of the simple economic order quantity model (see Porteus [13]). In the formulation of the joint replenishment problem the objective is the minimisation of the total average costs. The total average costs are the sum of the average major ordering cost and the individual average costs of each item.

In the literature two formulations of the joint replenishment problem have appeared. The first formulation takes into account the possibility of so-called *empty replenishment occasions*, which occur when the smallest frequency k_i is larger than one. For example, suppose that we have two items and that $k_1 = 2$ and $k_2 = 3$. If this happens, then two out of six replenishment occasions will not be used and so the average major ordering cost equals $4S/6T$ if the basic cycle time is given by T. This implies that a correction factor $\Delta(k)$, $k = (k_1, \dots, k_n)$, has to be used. Based on the principle of inclusion and exclusion (see Grimaldi [7]) the following expression for $\Delta(k)$ is given by Dagpunar [2]:

$$\Delta(k) = \sum_{i=1}^{n}(-1)^{i+1} \sum_{\{\alpha \subset \{1,\dots,n\}:\ |\alpha|=i\}} (\operatorname{lcm}(k_{\alpha_1}, \dots, k_{\alpha_i}))^{-1}, \tag{2}$$

where $\operatorname{lcm}(k_{\alpha_1}, \dots, k_{\alpha_i})$ denotes the least common multiple of the integers $k_{\alpha_1}, \dots, k_{\alpha_i}$. By this observation the joint replenishment problem with a correction factor is given by

$$(P_c) \quad \inf\left\{ \frac{S\Delta(k)}{T} + \sum_{i=1}^{n} \Phi_i(k_iT):\ k_i \in \mathbb{N},\ T > 0 \right\}.$$

Goyal [4] criticises the formulation of Dagpunar [2] and proposes to set the correction factor equal to one. In that case the joint replenishment problem without correction factor becomes

$$(P) \quad \inf\left\{ \frac{S}{T} + \sum_{i=1}^{n} \Phi_i(k_iT):\ k_i \in \mathbb{N},\ T > 0 \right\}.$$

Both problems are mixed continuous-integer programming problems and in general such problems are difficult to solve. Moreover, since the correction factor is hard to evaluate for large values of n (evaluation of (2) requires $\mathcal{O}(2^n)$ time) and the objective function of (P_c) is not separable in the vector k, it follows that (P_c) is more difficult to solve than (P).

In the literature very little attention is paid to problem (P_c); the attention is primarily focused on problem (P). However, despite the fact that (P) is easier to solve than (P_c), the optimal solution procedures for (P) that are known till now involve enumeration, which -in continuous time- becomes computationally prohibitive for a large number of items (see Goyal [3] and Van Eijs [15]). In discrete time we do not have this problem, but then the enumeration takes place among all discrete values between some bounds (see Klein and Ventura [12]), which may yield other problems. Also in this paper we will focus on problem (P), and we will show that by using Lipschitz optimisation the problem can efficiently be solved to optimality. Moreover, we will present some results for problem (P_c). The optimal solution of (P) obtained after Lipschitz optimisation can also be used as a feasible solution of (P_c). Furthermore, we will show that the solution of a relaxation of problem (P) yields a lower bound on the optimal objective value of problem (P_c), so that we can decide whether this feasible solution is good enough. In Section 5 we will show by numerical experiments that the gap between the lower bound and the generated feasible solution is usually quite small. To start our analysis, we will first consider problem (P) in more detail.

3. Analysis of problem (*P*)

To simplify the objective function of problem (P), observe that (P) is equivalent to

$$\inf_{T>0}\left\{ \frac{S}{T} + \sum_{i=1}^{n} \inf\{\Phi_i(k_iT):\ k_i \in \mathbb{N}\} \right\}. \tag{3}$$

Introduce now the functions $g_i(\cdot)$ given by

$$g_i(t) := \inf\{\Phi_i(k_it):\ k_i \in \mathbb{N}\}, \tag{4}$$

and observe that (3) (and consequently (P)) can be written as

$$(P) \quad \inf_{T>0}\left\{ \frac{S}{T} + \sum_{i=1}^{n} g_i(T) \right\}.$$

Denote by $v(P)$ the optimal objective value of (P) and by $T(P)$ an optimal T. We will first show that the functions $g_i(\cdot)$ can be evaluated easily.

It is not difficult to verify that for each $i \in \{1, \dots, n\}$ and $k \in \mathbb{N}$ the function $t \to \Phi_i(kt)$ on $(0, \infty)$ satisfies:

- $t \to \Phi_i(kt)$ is strictly convex;
- $t \to \Phi_i(kt)$ has a minimum for $t = x_i^*/k$ with x_i^* given by:

$$x_i^* = \sqrt{\frac{2s_i}{h_iD_i}}; \tag{5}$$

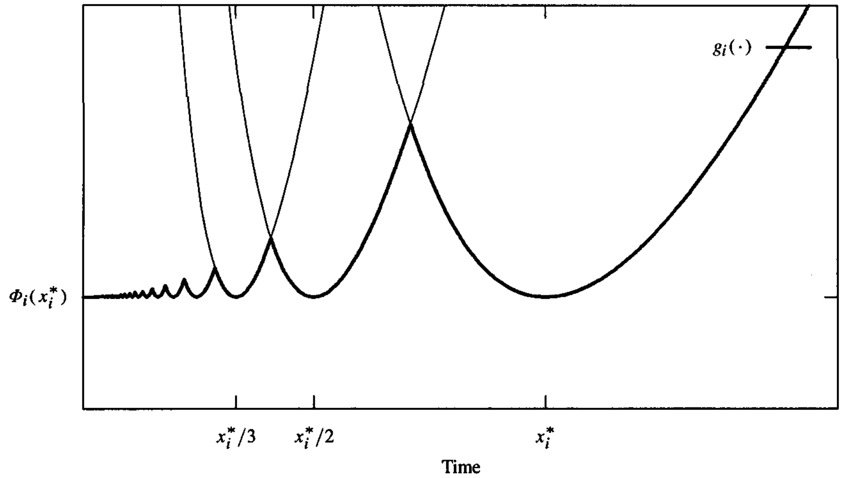

Fig. 1. An example of the function $t \to \Phi_i(kt)$ for $k = 1,2,3$. The thin lines are the graphs of the functions $\Phi_i(t), \Phi_i(2t), \Phi_i(3t)$. The (bold) graph of $g_i(\cdot)$ is the lower envelope of the functions $t \to \Phi_i(kt)$, $k \in \mathbb{N}$.

- $t \to \Phi_i(kt)$ is strictly decreasing on $(0, x_i^*/k)$ and strictly increasing on $(x_i^*/k, \infty)$.

In Fig. 1 the function $t \to \Phi_i(kt)$ for different values of the integer k is plotted. It is easy to verify that the intersection point of the functions $\Phi_i(kt)$ and $\Phi_i((k+1)t)$ is given by $(2s_i/h_iD_ik(k+1))^{1/2}$.

Define for $k = 0, 1, \ldots$ the value $T_i^{(k)}$ given by

$$T_i^{(k)} := \begin{cases} \sqrt{\dfrac{2s_i}{h_iD_ik(k+1)}} & \text{if } k = 1,2,\ldots \\ \infty & \text{if } k = 0, \end{cases} \tag{6}$$

and introduce for $k \in \mathbb{N}$ the interval $I_i^{(k)} := [T_i^{(k)}, T_i^{(k-1)})$. Then clearly by (5) and (6) it follows for every $k \in \mathbb{N}$ that x_i^*/k belongs to $I_i^{(k)}$ and by this observation it is easy to verify that the optimal k_i in $\inf\{\Phi_i(k_it)\colon k_i \in \mathbb{N}\}$ is given by k if t belongs to $I_i^{(k)}$. Using this, an easy explicit formula for the optimal solution $k_i(t)$ as a function of t can be found, which is presented in the next lemma. Although the derivation of this formula is almost trivial, we could not find such an explicit formula in the literature.

Lemma 1. *An optimal value $k_i(t) \in \mathbb{N}$ given a value of $t > 0$ is given by*

$$k_i(t) = \left\lceil -\tfrac{1}{2} + \tfrac{1}{2}\sqrt{1 + \frac{8s_i}{h_iD_it^2}} \right\rceil, \tag{7}$$

with $\lceil\cdot\rceil$ denoting the upper-entier function.

Proof. As observed, given a value of $t > 0$, an optimal value of $k \in \mathbb{N}$ is such that $T_i^{(k)} \leqslant t < T_i^{(k-1)}$. Equivalently, using (6), the value of k must satisfy

$$\sqrt{\frac{2s_i}{h_iD_ik(k+1)}} \leqslant t \tag{8}$$

and

$$t < \sqrt{\frac{2s_i}{h_iD_i(k-1)k}}. \tag{9}$$

Inequality (8) is equivalent to

$$k^2 + k - \frac{2s_i}{h_iD_it^2} \geqslant 0$$

and since k must be positive we obtain

$$k \geqslant -\tfrac{1}{2} + \tfrac{1}{2}\sqrt{1 + \frac{8s_i}{h_iD_it^2}}. \tag{10}$$

Analogously, working out (9) yields that

$$0 < k < \tfrac{1}{2} + \tfrac{1}{2}\sqrt{1 + \frac{8s_i}{h_iD_it^2}}, \tag{11}$$

and combining (10) and (11) implies

$$-\tfrac{1}{2} + \tfrac{1}{2}\sqrt{1 + \frac{8s_i}{h_iD_it^2}} \leqslant k < \tfrac{1}{2} + \tfrac{1}{2}\sqrt{1 + \frac{8s_i}{h_iD_it^2}}. \tag{12}$$

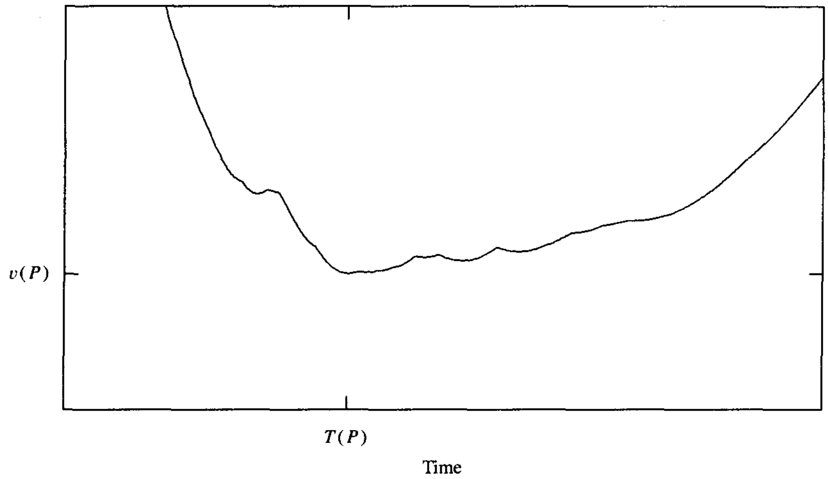

Fig. 2. An example of the objective function of problem (P); there are many local minima.

Since the square roots in these two inequalities are equal, the two expressions differ exactly 1 from each other, implying that either one integer is between them or that they are both integer. In either case, taking the upper entier of the expression in the left-hand side yields the integer k satisfying (12). Consequently, given a value of t, a corresponding optimal value $k_i(t)$ is given by (7), which completes the proof. □

By Lemma 1 it now follows that the functions $g_i(\cdot)$ defined in (4) are given by

$$g_i(t) = \Phi_i(k_i(t)t), \tag{13}$$

and consequently the optimisation problem (P) reduces to

$$\inf_{T>0}\left\{\frac{S}{T} + \sum_{i=1}^{n} \Phi_i(k_i(T)T)\right\}.$$

In Fig. 2 an example of the objective function of problem (P) is given. This objective function has in general multiple local minima and so (P) is a global-optimisation problem. In the following section we show that when the integrality constraints in (P) are relaxed, a convex-programming problem is obtained, which can be solved analytically. Using this optimal solution, we subsequently determine a feasible solution of (P), and since the relaxation yields a lower bound on the optimal objective value $v(P)$, we can decide whether this feasible solution is good enough. If it is not good enough, we apply a global-optimisation procedure to identify in little time a solution to (P) with a corresponding objective value that has an arbitrarily small deviation from the optimal value $v(P)$.

Such a global-optimisation procedure will not be obtained with respect to problem (P_c), which is in general much more difficult to solve than problem (P), because of the correction factor $\Delta(k)$. It can be shown that for some values of T there are no empty replenishments, in which case the correction factor is equal to one; in the report version of this paper [16] the authors show that $\Delta(k(T)) = 1$ for $T \geqslant \min_i\{((S+s_i)/h_iD_i)^{1/2}\}$. Consequently, for these values of T the objective functions of the problems (P) and (P_c) are equal. However, we could not prove a nice result for values of T smaller than $\min_i\{((S+s_i)/h_iD_i)^{1/2}\}$ and the authors believe that, given such a T, there does not exist an easy analytical formula for the optimal value $k_i(T)$, $i = 1,\ldots,n$. Therefore, solving (P_c) to optimality in a reasonable time using a global-optimisation procedure seems impossible and the only thing one can hope for is a fast procedure that generates a feasible solution and at the same time yields an upper bound on the deviation from the optimal objective value $v(P_c)$ of (P_c). Such a procedure will be discussed in the next section.

4. Solving problem (P)

In this section we develop a solution procedure for problem (P). To this end we consider first a relaxation.

4.1. A relaxation for problem (P)

By relaxing the constraints $k_i \in \mathbb{N}$ by $k_i \geqslant 1$, we obtain the following relaxation (R) of problem (P):

$$(R) \quad \inf\left\{\frac{S}{T} + \sum_{i=1}^{n} \Phi_i(k_i T):\ k_i \geqslant 1,\ T > 0\right\}.$$

Denote by $v(R)$ the optimal objective value of (R) and by $T(R)$ the optimal T. (Below we will show that a $T(R)$ exists and that it is unique.) Since (R) is a relaxation of (P) it clearly follows that $v(P) \geqslant v(R)$.

A similar relaxation was presented by Hariga [8] and to write down an analytical formula for the optimal solution $T(R)$ of (R), the author analysed the necessary Karush–Kuhn–Tucker conditions for a global minimum. However, by making use of the separability of the objective function of (R) in the vector $k = (k_1, \dots, k_n)$, a much easier proof of the validity of the formula for $T(R)$ can be obtained.

To start with the analysis of (R) we observe that (R) is equivalent to

$$\inf_{T>0}\left\{\frac{S}{T} + \sum_{i=1}^{n} g_i^{(R)}(T)\right\},$$

where $g_i^{(R)}(t) := \inf\{\Phi_i(k_i t):\ k_i \geqslant 1\}$. Since the function $\Phi_i(\cdot)$ is strictly decreasing on $(0, x_i^*)$ and strictly increasing on (x_i^*, ∞) it follows that

$$g_i^{(R)}(t) = \begin{cases} \Phi_i(x_i^*) & \text{if } t \leqslant x_i^* \\ \Phi_i(t) & \text{if } t \geqslant x_i^*. \end{cases} \tag{14}$$

By the definition of $\Phi_i(\cdot)$ (see (1)) and using the fact that $\Phi_i(\cdot)$ has a unique minimum x_i^* given by (5), we obtain the result that the function $g_i^{(R)}(\cdot)$ is continuously differentiable with derivative

$$\frac{\mathrm{d}}{\mathrm{d}t} g_i^{(R)}(t) = \begin{cases} 0 & \text{if } t \leqslant x_i^* \\ \dfrac{-s_i}{t^2} + \dfrac{h_i D_i}{2} & \text{if } t \geqslant x_i^*. \end{cases}$$

Since the derivative of $g_i^{(R)}(\cdot)$ is nonnegative, continuous and increasing on $(0, \infty)$, if follows that $g_i^{(R)}(\cdot)$ is convex, increasing, and continuously differentiable on $(0, \infty)$. Consequently, by the strict convexity and continuous differentiability of the function $t \to S/t$ on $(0, \infty)$, the objective function $h(\cdot)$ of (R), given by $h(t) = S/t + \sum_{i=1}^n g_i^{(R)}(t)$, is strictly convex and continuously differentiable on $(0, \infty)$. Because $\lim_{t\uparrow\infty} h(t) = \infty$ and $\lim_{t\downarrow 0} h(t) = \infty$ this implies that there exists a unique optimal solution $T(R) \in (0, \infty)$ and this value is the unique solution of the equation $h'(t) := \mathrm{d}h(t)/\mathrm{d}t = 0$.

To derive an analytical expression for $T(R)$, we first assume without loss of generality that $x_1^* \leqslant x_2^* \leqslant \cdots \leqslant x_n^*$. Under this assumption, the strictly increasing derivative $h'(\cdot)$ of the function $h(\cdot)$ is given by

$$h'(t) = \begin{cases} \dfrac{-S}{t^2} & \text{if } t \leqslant x_1^* \\ \dfrac{\sum_{i=1}^{l} h_i D_i}{2} - \dfrac{S + \sum_{i=1}^{l} s_i}{t^2} & \text{if } x_l^* \leqslant t \leqslant x_{l+1}^*,\ 1 \leqslant l \leqslant n-1 \\ \dfrac{\sum_{i=1}^{n} h_i D_i}{2} - \dfrac{S + \sum_{i=1}^{n} s_i}{t^2} & \text{if } t \geqslant x_n^*. \end{cases} \tag{15}$$

It follows from (15) that $h'(x_1^*) = -S/(x_1^*)^2 < 0$ and so $i^* := \max\{1 \leqslant i \leqslant n:\ h'(x_i^*) < 0\}$ exists. If $i^* < n$, we obtain $h'(x_{i^*+1}^*) \geqslant 0$, and so by the strict convexity of $h(\cdot)$ the optimal $T(R)$ belongs to the interval $[x_{i^*}^*, x_{i^*+1}^*]$. Analogously, if $i^* = n$, we have $T(R) \in [x_n^*, \infty)$. By setting the derivative of $h(\cdot)$ (see (15)) to zero, we now obtain the following result.

Lemma 2. *Assume without loss of generality that $x_1^* \leqslant x_2^* \leqslant \cdots \leqslant x_n^*$. If $i^* := \max\{1 \leqslant i \leqslant n:\ h'(x_i^*) < 0\}$, then the optimal solution $T(R)$ of (R) is given by*

$$T(R) = \sqrt{\frac{2(S + \sum_{i=1}^{i^*} s_i)}{\sum_{i=1}^{i^*} h_i D_i}}. \tag{16}$$

We already observed that $v(R) \leqslant v(P)$, since (R) is a relaxation of (P). In the next lemma we prove that also $v(R) \leqslant v(P_c)$, so that by solving problem (R) we also have a lower bound on the optimal objective value of problem (P_c).

Lemma 3. *It follows that $v(P) \geqslant v(P_c) \geqslant v(R)$.*

Proof. Since for every vector $k = (k_1, \ldots, k_n) \in \mathbb{N}^n$ it holds that $\Delta(k) \leqslant 1$, the first inequality follows immediately. To prove the second inequality, we observe that for every $\varepsilon > 0$ there exists a vector $(T_\varepsilon, k_1(T_\varepsilon), \ldots, k_n(T_\varepsilon))$ satisfying

$$v(P_c) \geqslant \frac{S\Delta(k(T_\varepsilon))}{T_\varepsilon} + \sum_{i=1}^{n} \Phi_i(k_i(T_\varepsilon)T_\varepsilon) - \varepsilon$$
$$= \frac{S\Delta(k(T_\varepsilon))}{T_\varepsilon} + \sum_{i=1}^{n} \Phi_i\left(\frac{k_i(T_\varepsilon)\Delta(k(T_\varepsilon))T_\varepsilon}{\Delta(k(T_\varepsilon))}\right) - \varepsilon.$$

Using $\Delta(k(T_\varepsilon)) \geqslant (\min_i\{k_i(T_\varepsilon)\})^{-1}$, we have $k_i(T_\varepsilon)\Delta(k(T_\varepsilon)) \geqslant 1$ for every i, and consequently

$$v(P_c) \geqslant \frac{S\Delta(k(T_\varepsilon))}{T_\varepsilon} + \sum_{i=1}^{n} \inf\left\{\Phi_i\left(\frac{k_iT_\varepsilon}{\Delta(k(T_\varepsilon))}\right) : k_i \geqslant 1\right\} - \varepsilon$$
$$\geqslant \inf_{T>0}\left\{\frac{S}{T} + \sum_{i=1}^{n} \inf\{\Phi_i(k_iT) : k_i \geqslant 1\}\right\} - \varepsilon$$
$$= v(R) - \varepsilon.$$

Since $\varepsilon > 0$ is arbitrary, the desired result follows. □

Once (R) is solved, we have an optimal $T(R)$ for (R). Sometimes this $T(R)$ is also optimal for problems (P) and (P_c), as is shown by the following lemma.

Lemma 4. *Assume without loss of generality that $x_1^* \leqslant x_2^* \leqslant \cdots \leqslant x_n^*$. If for the optimal $T(R)$ of problem (R) it holds that $T(R) \geqslant x_n^*$, then $(T(R), 1, \ldots, 1)$ is an optimal solution of (P) and (P_c).*

Proof. Since $T(R) \geqslant x_n^*$ is an optimal solution of problem (R) it follows by (14) that the corresponding scalars k_i, $i = 1, \ldots, n$, equal 1 and so $(T(R), 1, \ldots, 1)$ is also a feasible solution of problems (P_c) and (P). Hence we obtain $v(R) = S/T(R) + \sum_{i=1}^n \Phi_i(T(R)) \geqslant v(P)$, yielding by Lemma 3 $v(R) = v(P_c) = v(P)$, implying that $(T(R), 1, \ldots, 1)$ is also an optimal solution of (P_c) and (P). □

Observe that from (15) it follows that $T(R) \geqslant x_n^*$ if and only if $S \geqslant (x_n^*)^2 \sum_{i=1}^n h_iD_i/2 - \sum_{i=1}^n s_i$. Consequently, the condition $T(R) \geqslant x_n^*$ in Lemma 4 is easy to check.

Finding a Feasible Solution of (P) and (P_c)

Observe for $T(R) < x_n^*$ that $T(R)$ may not be an optimal solution of problems (P) and (P_c). This is due to the fact that the values of k_i corresponding with $T(R)$ (see (14)) are not necessarily integer, and this implies that the optimal solution of (R) is in general not feasible for (P) and (P_c). However, the vector $(T(R), k_1(T(R)), \ldots, k_n(T(R)))$, with the scalars $k_i(T(R))$, $i = 1, \ldots, n$, obtained by substitution of $T(R)$ in (7), is a feasible solution of (P) and (P_c).

Let $v(FP)$ be the objective-function value of (P) evaluated at this feasible solution. Then we have

$$v(FP) = \frac{S}{T(R)} + \sum_{i=1}^{n} g_i(T(R))$$
$$\geqslant v(P) \geqslant v(P_c) \geqslant v(R).$$

Hence we can check the quality of the constructed feasible solution; if $v(FP)$ is close to $v(R)$ then we have found a feasible solution that is good enough for both (P) and (P_c).

If it is not close enough, we will apply a fast global-optimisation procedure for solving (P). To this end, we need an interval that contains an optimal $T(P)$.

4.2. Lower and upper bounds on $T(P)$

Goyal [3] states that an optimal value of $T(P)$ is within the interval $[\min_i x_i^*, T(P_1)]$, with $T(P_1)$ the optimal solution of the optimisation problem

$$(P_1) \quad \inf_{T>0}\left\{\frac{S}{T} + \sum_{i=1}^{n} \Phi_i(T)\right\}.$$

Observe that $T(P_1)$ is the optimal value of T when all items are jointly replenished. It is easily verified that

$$T(P_1) = \sqrt{\frac{2(S + \sum_{i=1}^n s_i)}{\sum_{i=1}^n h_iD_i}}. \tag{17}$$

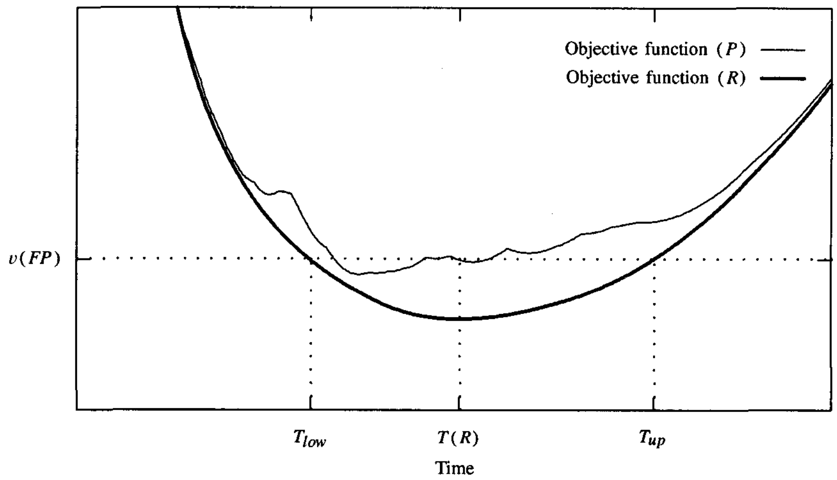

Fig. 3. A lower bound $T_{\rm low}$ and an upper bound $T_{\rm up}$ on an optimal $T(P)$ are found where the objective function of relaxation (R) equals $v(FP)$, the value of the objective function of problem (P) in $T(R)$.

However, the lower bound $\min_i x_i^*$ is *not correct* for an optimal $T(P)$, as has also been pointed out by Van Eijs [15]. In this subsection we show that solving (R) yields both a lower and an upper bound on $T(P)$. The upper bound is often better than $T(P_1)$, as will be shown by numerical experiments in Section 5.

As in Section 4.1, let $v(FP)$ be the objective-function value of (P) in $T(R)$. We will show that a lower and an upper bound on $T(P)$ are given by the values of T where the objective function of (R) equals $v(FP)$. This is established by the following lemma.

Lemma 5. *Let $T_{\rm low}$ be the smallest and $T_{\rm up}$ be the largest T for which the objective function of (R) is equal to $v(FP)$. Then $T_{\rm low} \leqslant T(P) \leqslant T_{\rm up}$.*

Proof. Since the objective function of (R) is strictly convex, we clearly have the result that $T_{\rm low} \leqslant T(R) \leqslant T_{\rm up}$. Consequently, for values of $T < T_{\rm low}$ the objective function of (R) is larger than $v(FP)$. Since (R) is a relaxation of (P), the objective function of (P) is also larger than $v(FP)$ for values of $T < T_{\rm low}$, so that $T_{\rm low}$ is a lower bound on $T(P)$.

The proof that $T(P) \leqslant T_{\rm up}$ is analogous. □

In Fig. 3 it is illustrated how the bounds $T_{\rm low}$ and $T_{\rm up}$ are generated. Notice that the lower bound $T_{\rm low}$ can be found by bisection on the interval $(0, T(R)]$. With respect to the upper bound $T_{\rm up}$, it is easy to check whether it is better than $T(P_1)$ given by (17). To this end, evaluate the objective function of (R) in $T(P_1)$ and check whether it is smaller than or equal to $v(FP)$. If this is so, then $T(P_1)$ is at least as good as $T_{\rm up}$, otherwise $T_{\rm up}$ is better. In the latter case we can easily find $T_{\rm up}$ with a bisection on the interval $[T(R), T(P_1)]$.

Let now $T_l = T_{\rm low}$ and let $T_u = \min\{T_{\rm up}, T(P_1)\}$ be the smallest upper bound, then we have $T(P) \in [T_l, T_u]$. Consequently, it is sufficient to apply a global-optimisation technique on the interval $[T_l, T_u]$ to find a value for $T(P)$.

4.3. Lipschitz optimisation

Efficient global-optimisation techniques exist when the objective function is *Lipschitz*. A univariate function is said to be Lipschitz if for each pair x and y the absolute difference of the function values in these points is smaller than or equal to a constant (called the *Lipschitz constant*) multiplied by the absolute distance between x and y. More formally:

Definition 1. A function $f(\cdot)$ is said to be Lipschitz on the interval $[a,b]$ with Lipschitz constant L, if for all $x, y \in [a,b]$ it holds that $|f(x) - f(y)| \leqslant L|x - y|$.

Since the objective function of (P) is Lipschitz on the interval $[T_l, T_u]$ (see the appendix), global-optimisation techniques can be applied to this interval to obtain a solution with a corresponding objective value that is arbitrarily close to the optimal objective value $v(P)$ (see the chapter on Lipschitz optimisation in Horst and Pardalos [9]). In the appendix we derive a Lipschitz constant L for the objective function of (P) on $[T_l, T_u]$.

There are several Lipschitz-optimisation algorithms (see Horst and Pardalos [9]). The simplest one, called the *passive algorithm*, evaluates the function to be minimised at the points $a + \varepsilon/L$, $a + 3\varepsilon/L$, $a + 5\varepsilon/L, \ldots$, and takes that point at which the value of the function is minimal. The function value in this point does not differ more than ε from the *global* minimal value in $[a, b]$. The algorithm of Evtushenko is based on the passive algorithm, but takes the next step larger than $2\varepsilon/L$ if the current function value is larger than 2ε above the current *best* known value, which makes the algorithm faster than the passive algorithm. The best known value in this algorithm is initialised at the value $v(FP)$ that is found after solving the relaxation. For our problem this procedure can be improved, since the shape of the objective function of problem (P) is such that the Lipschitz constant is decreasing in T (this is shown in the appendix). Using this, we can extend the algorithm of Evtushenko to deal with a *dynamic* Lipschitz constant; after each function evaluation (going from left to right) the Lipschitz constant is recomputed so that larger steps can be taken. Due to the easy formula for the Lipschitz constant, this recomputation is fast.

There are many other Lipschitz-optimisation algorithms and these use more sophisticated techniques and may be faster than the algorithm of Evtushenko. However, we choose Evtushenko's algorithm (with a dynamic Lipschitz constant) because of its simplicity and because it is easy to implement. Furthermore, the resulting procedure is very fast; in the next section we show that even when a relative deviation of 0.001% is required, a solution is found in little time.

Relation with the approach of Kaspi and Rosenblatt

Kaspi and Rosenblatt [11] have proposed an approach in which the interval containing an optimal $T(P)$ is divided into a number of equidistant values, on each of which the authors apply a heuristic of Kaspi and Rosenblatt [10] (which is a modified version of the algorithm of Silver [14]). This idea of trying multiple cycle times is arbitrary, since it is not clear how many values should be tried to obtain a good solution. Furthermore, taking equidistant values is not always efficient.

In fact, our approach also tries multiple values of T, but it does so in an efficient way and a way that guarantees the quality of the solution. The choice of a next value of T is efficient, because in the modified algorithm of Evtushenko the length of the next step depends on the best known value of the objective function, on the Lipschitz constant, and on the precision that is required. For example, when the current value of the objective function is above the best known (minimal) value, then a larger step can be taken since the function is Lipschitz and must first bridge the difference between the current value and the best known minimum, before it can possibly come below this minimum; furthermore, when the objective function is peaked, then the Lipschitz constant will be larger and hence smaller steps are taken to guarantee a good solution; finally, larger steps are taken when less precision is required. Moreover, the application of a dynamic Lipschitz constant makes it possible to choose a Lipschitz constant which is as small as possible.

4.4. A solution procedure for (P)

We can summarise the results in this section in the formulation of the following solution procedure for problem (P):

1. Find the optimal $T(R)$ of problem (R) using (16).
2. If $T(R) \geqslant x_n^*$ then $T(P) = T(R)$ is optimal for (P) and $k_i = 1$, $i = 1, \ldots, n$; stop.
3. Otherwise, we first find a feasible solution for problem (P) by substitution of $T(R)$ in (7). If the corresponding objective value $v(FP)$ is close enough to $v(R)$, then it is also close to $v(P)$ and so we have a good solution; stop.
4. If the solution is not good enough, apply Lipschitz-optimisation with a dynamic Lipschitz constant on the interval $[T_l, T_u]$ to find a value for $T(P)$ and a corresponding objective value with arbitrarily small deviation from $v(P)$.

5. Numerical results

In this section we will show by numerical experiments that the solution procedure for (P) described in the previous section, using the modified algorithm of Evtushenko for the Lipschitz optimisation, generates a solution in little time. Furthermore, our experiments illustrate that the running time of the procedure grows approximately linearly in the number of items. The procedure is implemented in Borland Pascal version 7.0 on a standard 486 66 MHz personal computer.

Remember that the solution of problem (P) can be used as a feasible solution of problem (P_c). By computing the difference between $v(R)$ and $v(P)$ in the numerical examples, we are by Lemma 3 able to say something about the deviation of the objective value corresponding to this feasible solution from the optimal value $v(P_c)$. We will not further investigate problem (P_c), since incorporation of the correction factor $\Delta(k)$ in a solution procedure is too time consuming.

In the numerical experiments we generated 2400 problems as follows (the values are taken from Ben-Daya and Hariga [1]):

1. The demand D_i is taken randomly from $[100, 100000]$;
2. The ordering cost s_i is taken randomly from $[0.5, 5]$;
3. The holding cost h_i is taken randomly from $[0.2, 3]$;
4. The major ordering-cost values considered are $S = 5, 10, 15, 20$;
5. The number of items $n = 5, 10, 15, 20, 25, 30$.

Hence there are 24 combinations of n and S and for each combination we generated 100 examples.

The running time of the solution procedure depends on the number of items, on the set-up cost, and on the precision required. First, we will consider the influence of the number of items n and the set-up cost S, while the precision is kept fixed. We chose a relative precision instead of an absolute precision, since the optimal objective values may differ considerably in the examples and a relative precision enables a fair comparison. To this end, we took an *absolute* precision equal to $\varepsilon v(R)$. This implies that a solution generated by the algorithm does not deviate more from $v(P)$ than $\varepsilon v(R)$, and accordingly the *relative* deviation is smaller than or equal to $(v(P) + \varepsilon v(R) - v(P))/v(P) = \varepsilon v(R)/v(P)$, which by Lemma 3 is smaller than or equal to $\varepsilon v(P)/v(P) = \varepsilon$. We first took ε equal to 10^{-5} (i.e., $\varepsilon = 0.001\%$). In Table 1 the average running time is given for the hundred random examples that were generated for each of the 24 combinations of n and S. As can be seen from the table, the running time decreases in the set-up cost S. This is due to a steeper objective function for larger S; a larger S causes smaller upper bounds for $T(P)$ and, as a result, smaller intervals on which Lipschitz optimisation has to be applied. Furthermore, it can be seen from the table that the running time increases linearly in the number n of items; from the last row in Table 1 it follows that the average running time increases with approximately 0.14 seconds when n increases with 5 items (starting with 0.10 seconds for $n = 5$). The linear increment of speed is a nice result when it is considered that Lipschitz optimisation is an optimal solution procedure and that alternative optimal procedures in continuous time published so far in the literature involve only enumeration methods with exponentially growing running times.

Table 1
Average running time (sec) of the solution procedure as a function of n and S; $\varepsilon = 10^{-5}$

S	n					
	5	10	15	20	25	30
5	0.14	0.30	0.48	0.64	0.70	0.87
10	0.09	0.24	0.37	0.57	0.72	0.84
15	0.10	0.23	0.36	0.48	0.69	0.78
20	0.07	0.19	0.35	0.47	0.61	0.75
Average	0.10	0.24	0.39	0.54	0.68	0.81

The running time also depends on the precision that is required. For less precision Lipschitz optimisation becomes much faster. In Table 2 the average running time as a function of the relative precision ε and the number of items n are tabulated. Although the precision ε increases each time with a factor ten, the average running times increase only with a factor of approximately five. As can be seen from the table, the solution procedure requires little time, even for 30 items and a precision of 0.001%. Observe that from this table it follows that the linearly increasing running time also holds for $\varepsilon = 0.1\%$ and $\varepsilon = 0.01\%$.

Table 2
Average running time (sec) of the solution procedure as a function of n and ε

ε	n						Average
	5	10	15	20	25	30	
0.1%	0.00	0.00	0.00	0.02	0.05	0.06	0.02
0.01%	0.00	0.05	0.07	0.10	0.13	0.16	0.09
0.001%	0.10	0.24	0.39	0.54	0.68	0.81	0.46

In the 2400 examples the gap $(v(P)-v(R))/v(R)$ was maximally only 1.611% and on average only 0.608%. By Lemma 3 it follows that the gap $(v(P)-v(P_c))/v(P_c)$ is even smaller, and this implies that an optimal solution of (P) is on average a good feasible solution for problem (P_c). Consequently, to obtain a good solution of problem (P_c), it will often be sufficient to solve problem (P), which takes little time as already shown. Only if the gap $(v(P)-v(R))/v(R)$ is not small enough, one can try to find a better solution of problem (P_c). However, this implies the incorporation of the correction factor $\Delta(k)$, which requires much computation time.

Finally, the experiments also showed that the upper bound T_{up} introduced in Section 4 is mostly better than the upper bound $T(P_1)$ introduced by Goyal [3]. From the 2400 numerical examples it turned out that in 1725 cases (that is, in 72% of the examples) the upper bound T_{up} was smaller (and consequently better) than $T(P_1)$.

6. Conclusions

We presented a new optimal solution approach for the joint replenishment problem. Until now, the available optimal solution approaches have involved enumeration, which required exponentially growing running times in the continuous-time case. We followed an alternative solution approach based on global optimisation theory, and we were able to apply Lipschitz optimisation to obtain a solution with an arbitrarily small deviation from the optimal value. We presented a solution procedure using a dynamic Lipschitz constant, which generates a solution in little time. A very important characteristic of the solution procedure is that the running time increases only linearly in the number of items. Now that a fast optimal solution procedure for the joint replenishment problem exists, with a linear time complexity, the development of (new) heuristics for the problem will be futile.

Appendix A. Determination of Lipschitz constant

We will prove here that the objective function of problem (P) is Lipschitz on the interval $[T_l, T_u]$ and we will derive an easy expression for the Lipschitz constant L.

It is obvious that if L_i is a Lipschitz constant for the function $g_i(\cdot)$ (see (4)) and L_0 is a Lipschitz constant for S/T, then a Lipschitz constant L for the objective function of (P) is given by

$$L = L_0 + \sum_{i=1}^{n} L_i. \tag{A.1}$$

For a differentiable function, a Lipschitz constant on some interval is given by the maximum of its derivative in absolute value on this interval. Therefore, since the derivative of S/T equals $-S/T^2$, and $|-S/T^2|$ is maximal on $[T_l, T_u]$ for T_l, we have:

$$L_0 = S/T_l^2. \tag{A.2}$$

A similar argument can be followed to obtain an expression for L_i. The function $g_i(\cdot)$ has maximum slopes (in absolute value) in the intersection points $T_i^{(k)}$, $k \in \mathbb{N}$ (see (6)). Although the function $g_i(\cdot)$ is not differentiable in these intersection points, the left and right-hand derivatives exist. Because of (13) it follows that the left-hand derivative of $g_i(\cdot)$ in the intersection point $T_i^{(k)}$ is given by $(k+1)\Phi_i'((k+1)T_i^{(k)})$, whereas the right-hand derivative in this intersection point is given by $k\Phi_i'(kT_i^{(k)})$. It is easy to verify that

$$(k+1)\Phi_i'((k+1)T_i^{(k)}) = -k\Phi_i'(kT_i^{(k)}) = \tfrac{1}{2}h_iD_i.$$

That is, for all $k \in \mathbb{N}$ the absolute values of both the left and right-hand derivative of $g_i(\cdot)$ in the intersection point $T_i^{(k)}$ equal $\frac{1}{2}h_iD_i$, and so we obtain

$$L_i = \tfrac{1}{2}h_iD_i, \quad i = 1, \ldots, n. \tag{A.3}$$

Combining (A.1), (A.2) and (A.3) yields the following expression for the Lipschitz constant L for the objective function of problem (P) on $[T_l, T_u]$:

$$L = \frac{S}{T_l^2} + \frac{1}{2}\sum_{i=1}^{n} h_i D_i. \tag{A.4}$$

Notice that the expression of L depends only on the interval $[T_l, T_u]$ by the value of T_l, and that a larger T_l yields a smaller L. Consequently, if L_1, L_2 are the Lipschitz constants for the objective function of (P) on the intervals $[T_1, T_u]$ and $[T_2, T_u]$ respectively, with $T_1 \leqslant T_2 \leqslant T_u$, then $L_1 \geqslant L_2$. This proves the correctness of applying a dynamic Lipschitz constant in the global-optimisation procedure described in Section 4.

References

[1] Ben-Daya, M. and Hariga, M., "Comparative study of heuristics for the joint replenishment problem", *Omega, International Journal of Management Science* 23 (1995) 341–344.

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

[3] Goyal, S.K., "Determination of optimum packaging frequency of items jointly replenished", *Management Science* 21 (1974) 436–443.

[4] Goyal, S.K., "A note on formulation of the multi-item single supplier inventory problem", *Journal of the Operational Research Society* 33 (1982) 287–288.,

[5] Goyal, S.K., and Deshmukh, S.G., "A note on 'the economic ordering quantity of jointly replenished items'", *International Journal of Production Research* 31 (1993) 2959–2962.

[6] Goyal, S.K., and Satir, A.T., "Joint replenishment inventory control: Deterministic and stochastic models", *European Journal of Operational Research* 38 (1989) 2–13.

[7] Grimaldi, R.P., *Discrete and Combinatorial Mathematic - An Applied Introduction*, Addison-Wesley, Reading, MA, 1989.

[8] Hariga, M., 'Two new heuristic procedures for the joint replenishment problem", *Journal of the Operational Research Society* 45 (1994) 463–471.

[9] Horst, R., and Pardalos, P.M., *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht, 1995.

[10] Kaspi, M., and Rosenblatt, M.J., "An improvement of Silver's algorithm for the joint replenishment problem", *IIE Transactions* 15 (1983) 264–269.

[11] Kaspi, M., and Rosenblatt, M.J., "On the economic ordering quantity for jointly replenished items", *International Journal of Production Research* 29 (1991) 107–114.

[12] Klein, C.M., and Ventura, J.A., "An optimal method for a deterministic joint replenishment inventory policy in discrete time", *Journal of the Operational Research Society* 46 (1995) 649–657.

[13] Porteus, E.L., "Stochastic inventory theory", chapter 12 in: *Handbooks in Operations Research and Management Science, Volume 2: Stochastic Models*, North-Holland, Amsterdam, 1990.

[14] Silver, E.A., "A simple method of determining order quantities in joint replenishments under deterministic demand", *Management Science* 22 (1976) 351–1361.

[15] Van Eijs, M.J.G., "A note on the joint replenishment problem under constant demand", *Journal of the Operational Research Society* 44 (1993) 185–191.

[16] Wildeman, R.E., Frenk, J.B.G., and Dekker, R., "An efficient optimal solution method for the joint replenishment problem", Technical Report 9542/A, Econometric Institute, Erasmus University Rotterdam, 1995.