TI 2001-102/4
Tinbergen Institute Discussion Paper

Optimal Continuous Order Quantity (*s, S)* Policies

Emöke Bázsa[1,2]
Peter den Iseger[1]

[1] *Econometric Institute, Faculty of Economics, Erasmus University Rotterdam,* [2] *Tinbergen Institute*

OPTIMAL CONTINUOUS ORDER QUANTITY (s, S) POLICIES

THE 45-DEGREES ALGORITHM

EMŐKE BÁZSA
TINBERGEN INSTITUTE AND ECONOMETRIC INSTITUTE,
ERASMUS UNIVERSITY ROTTERDAM, P.O. BOX 1738,
3000 DR ROTTERDAM, THE NETHERLANDS,
E-MAIL: BAZSA@FEW.EUR.NL,
AND PETER DEN ISEGER
ECONOMETRIC INSTITUTE, ERASMUS UNIVERSITY ROTTERDAM,
P.O. BOX 1738, 3000 DR ROTTERDAM, THE NETHERLANDS,
E-MAIL: ISEGER@FEW.EUR.NL

Abstract

The most recent optimization algorithm for (s, S) order policies with continuous demand was developed by Federgruen and Zipkin (1985). This was also the first efficient algorithm, which uses policy iteration instead of discretization. Zheng and Federgruen (1991) developed an even more efficient algorithm for computing discrete order quantity (s, S) inventory policies. Since the continuous case prohibits enumeration, this algorithm does not apply to continuous order quantity systems. In this paper an efficient algorithm for continuous order quantity (s, S) policies is developed. A marginal cost approach is used for determining the optimal s. Furthermore, we construct two aid functions (generated by the optimality conditions for s and S), and exploiting their special properties a simple and efficient algorithm is obtained. The algorithm converges monotonically, such that at every iteration a policy improvement is obtained. Since every iteration finds a local minimum of the expected average cost, the number of iterations is at most N, where $N < \infty$ represents the number of local minimums. The algorithm also applies to discrete order quantity systems, in which case it basically reduces to the algorithm of Zheng and Federgruen (with the difference that in general our algorithm will take larger than unit steps, since we are not using enumeration.)

1. Introduction

In the present article we give a simple and efficient algorithm for finding optimal (s, S) policies for inventory systems with continuous demand. Ongoing research on supply chains has increased the interest in continuous-demand models. Clark and Scarf (1960) were the first to show that the optimal policy for a two-echelon inventory system, with finite horizon, can be computed by decomposing the problem into two separate single-location problems. For the depot an (s, S) policy solves the problem, and constitutes an optimal order policy for the whole system. However, the lack of an efficient optimization algorithm for such single location problems results in suboptimality of the overall solution. Although recent research has extended the results of Clark and Scarf (see, for instance, Eppen and Schrage (1981), Federgruen and Zipkin (1984), Rosling (1989)), a truly efficient algorithm for the continuous order quantity (s, S) policy was missing until now.

To our knowledge, the only successful attempt to tackle the continuous demand case directly, without previous discretization, is the optimization algorithm of Federgruen and Zipkin(1985). Although Zheng and Federgruen(1991) present an improvement in the complexity of the calculation of an optimal (s, S) policy, their algorithm does not apply to continuous order quantity systems (since the continuous case prohibits enumeration). Some of the ideas of the Federgruen-Zipkin algorithm translate nicely into our algorithm (for comparisons of the two algorithms see Section 4.2.2). However, our algorithm has a different approach and it is more efficient. The algorithm converges monotonically, such that at every iteration a policy improvement is obtained. Since every iteration finds a local minimum of the expected average cost, the number of iterations is at most N, where $N < \infty$ represents the number of local minimums. Besides, the algorithm is also easy to understand, every step can be followed using a simple graphical representation.

The idea is the following: the lower and upper bounds for the optimal s and S, found in Section 3, define a feasibility interval for our search. We start the search for local minimums of the expected average cost function $C(s, S)$ alternately from the left and from the right of the feasibility interval, which eventually reduces to zero. It is vital not to leave out any relevant local minimum while make the search as efficient and fast as possible. Therefore, when a local minimum is found, it defines a "relevance level", such that the next local search will find only those local minimums which represent an improvement with respect to this level. Obviously, every improvement updates the relevance level. In this way, the last found local minimum will be the global minimum for $C(s, S)$.

In order to minimize a function with two variables, one can write down two optimality equations (the derivatives w.r.t. these variables equal 0). Based on these optimality relations we construct two aid functions, $s = \phi(S)$ and $s = \psi(S)$. It turns out that these aid functions have two simple but crucial properties: (1) both of them increase slower than 45 degrees; and (2) ψ always intersects ϕ in a maximum or minimum of ϕ, and these are the only stationary points for ϕ.

Having obtained these results, now the algorithm builds on two simple observations:

(a) For a fixed order-up-to level S_0 one can always find a unique $\bar{s}$, which minimizes $C(s, S_0)$. Hence, $\bar{s}$ can be determined with no effort.

(b) The local minimum points $(\bar{s}, \bar{S})$ of $C(s, S)$ coincide with the local maximum points of ϕ, such that $\bar{s} = \phi(\bar{S})$. Moreover, the global minimum of C coincides with the global maximum of ϕ.

Thus the problem reduces to finding the global maximum of ϕ. First we construct a local search, $LM(S_0)$, which finds the closest maximum of ϕ, starting at S_0. The subroutine LM converges monotonically to this maximum point, such that between the starting point S_0, and the found local maximum there will be no other stationary points for ϕ (thus also not for

C). LM is only based on properties (1), (2), and observations (a) and (b). This maximum point of ϕ defines the "relevance level" $\hat{s}_k$ (k is the actual number of iterations). Now solely using property (1) we construct an other subroutine, which finds the first point where ϕ increases above the relevance level $\hat{s}_k$. At this point we restart the local search LM, finding the next maximum, which determines the next relevance level, $\hat{s}_{k+1}$. Obviously, due to this construction, $\hat{s}_k < \hat{s}_{k+1}$, that is, $C(\hat{s}_k, S_k) > C(\hat{s}_{k+1}, S_{k+1})$, thus the algorithm converges monotonically to the global optimum. All the subroutines converge linearly. Moreover, since every iteration finds a local minimum of the expected average cost, the number of iterations is at most N, where $N < \infty$ represents the number of local minimums.

2. The model and the marginal cost

For the sake of generality this paper focuses on continuous time, continuous order quantity inventory systems, governed by an (s, S) policy. However, all the results, including the algorithm, remain valid for discrete time, discrete order quantity models. The demand process is a compound renewal process $\mathbf{D}(t) := \sum_{k=1}^{\mathbf{N}(t)} \mathbf{Y}_k$, where $\mathbf{Y}_k, k \in I\!N \cup \{0\}$ are the i.i.d. individual demands. $U(x)$ denotes the renewal function related to the sequence of individual demands with a renewal in 0, while $m(x)$ denotes its density. K is the fixed cost to place an order. The long run expected average cost of a system associated with an (s, S) policy is given by (cf. Bázsa and Iseger (2000)).

$$C(s, S) = \frac{K/I\!E\mathbf{t} + \int_0^{S-s} I\!E f(S - t - \mathbf{D}_\infty(0, L])U(dt)}{U(S - s)}, \tag{2.1}$$

where f is a given cost-rate function (its form is irrelevant for the further analysis), $\mathbf{D}_\infty(0, L]$ is the limiting distribution of the lead time demand, and $I\!E\mathbf{t}$ is the expected interarrival time. If we denote with $c(s)$ the long run expected average cost of an $(s - 1, s)$ model[1] without ordering costs,

[1] This model is in fact the generalized form of the classical $(s-1, s)$ model: an order is placed as soon as a demand occurs (thus not necessarily of size 1!)

knowing that this is given by $\mathbb{E}f(s - \mathbf{D}(0, L])$, $C(s, S)$ can be written in terms of $c(s)$:

$$C(s, S) = \frac{K + \int_0^{S-s} c(S - t)U(dt)}{U(S - s)}, \tag{2.2}$$

where K is normalized as $K := K/\mathbb{E}\mathbf{t}$. If there is no ordering cost, that is, $K = 0$ the optimal policy satisfies $S = s$ and $C(s, S) = c(s)$. Let s^* be the optimal order-up-to level for an $(s - 1, s)$ policy without ordering cost $(K = 0)$, that is, $s^* = \arg\min c(s)$. From these observations it also follows for any given s and S, that

$$c(s^*) < C(s, S). \tag{2.3}$$

Let us assume that $-c(s)$ is unimodal such that

$$c_s(s) < 0 \text{ for all } s < s^*, \text{ and } \lim_{s \downarrow -\infty} c(s) = +\infty. \tag{2.4}$$

The following lemma gives a marginal cost - condition for the optimality of the reorder level for a given order-up-to level. This lemma can be interpreted as the continuous version of Lemma 1 of Zheng and Federgruen (1991).

Lemma 2.1. *For any fixed order-up-to level S_0, the cost function $-C(s, S_0)$ is unimodal in $s \in (-\infty, s^*)$ and reaches its minimum in $\bar{s}$. Moreover, $\bar{s}$ is the unique solution of the equality*

$$C(s, S_0) = c(s), \tag{2.5}$$

and the following inequalities hold

$$C(s, S_0) < c(s) \quad \text{if and only if} \quad s < \bar{s}, \tag{2.6}$$

$$C(s, S_0) > c(s) \quad \text{if and only if} \quad \bar{s} < s \leq s^*. \tag{2.7}$$

Proof. Let us start from relation (2.2), and take the derivative of $C(s, S_0)$ w.r.t. s. This yields

$$C_s(s, S_0) = (C(s, S_0) - c(s))\frac{m(S_0 - s)}{U(S_0 - s)}. \tag{2.8}$$

Having observed relation (2.3) it follows that

$$C_s(s^*, S_0) > 0 \quad \text{for all } S_0. \tag{2.9}$$

Let us suppose now that there exists a stationary point $s_0 < s^*$ of $C(s, S_0)$ (that is, $C_s(s_0, S_0) = 0$). The second order derivative of $C(s, S_0)$ in this point is given by

$$C_{ss}(s_0, S_0) = -c_s(s_0). \tag{2.10}$$

Since $s_0 < s^*$, it is clear through relation (2.4) that $C_{ss}(s_0, S_0) > 0$, which implies that s_0 is a local minimum for $C(s, S_0)$. This means, that any stationary point $s_0 < s^*$ must be a local minimum, which is impossible. We can conclude therefore, that there is only one minimum: $\bar{s} < s^*$. Furthermore, $\bar{s}$ is a minimum for $C(s, S_0)$ if and only if $C_s(\bar{s}, S_0) = 0$, that is, if and only if $C(\bar{s}, S_0) = c(\bar{s})$, proving thus (2.5).

Furthermore, if $\bar{s}$ is a global minimum, and $-C(s, S_0)$ is unimodal on $(-\infty, \bar{s}]$, then for $s < \bar{s}$ $C_s(s, S_0) < 0$. This implies directly that $C(s, S_0) > c(s)$ for all $s < \bar{s}$. On the other hand, if $\bar{s} < s < s^*$ then $C_s(s, S_0) > 0$, which means that $C(s, S_0) < c(s)$ for all $\bar{s} < s < s^*$. It only remains to prove that there exists a stationary point for $C(s, S_0)$, S_0 fixed. Splitting the expression for C with respect to s^* yields:

$$C(s, S_0) \leq c(s)\frac{U(S_0 - s) - U(S_0 - s^*)}{U(S_0 - s)} + \int_0^{S_0 - s^*} c(S_0 - t)\frac{U(dt)}{U(S_0 - s)}.$$

Taking $s \longrightarrow -\infty$ yields $C_s(-\infty, S_0) < 0$, on the other hand $C_s(s^*, S) > 0$, which implies that a stationary point for C does exist. □

3. Bounds for the optimal reorder and order-up-to levels

The optimal order-up-to level s^* of the $(s-1, s)$ policy (with $K = 0$) represents an upper bound for the optimal reorder level s, and a lower bound for the optimal order-up-to level S of an (s, S) policy. This lower, respectively upper bound were first discovered by Veinott and Wagner (1965).

Lemma 3.1. *If $(\bar{s}, \bar{S})$ is an optimal policy, that is, $(\bar{s}, \bar{S})$ is a global minimum for $C(s, S)$, then*

$$\bar{s} < s^* \quad \text{and} \quad s^* < \bar{S}. \tag{3.1}$$

Moreover, the following assertions hold:

$$\text{if} \quad \min_{S>s^*} C(s,S) > c(s) \quad \text{then} \quad \bar{s} < s; \tag{3.2}$$

$$\text{if} \quad \min_{S>s^*} C(s,S) < c(s) \quad \text{then} \quad s < \bar{s}. \tag{3.3}$$

Proof. Suppose that the contrary of (3.1) is true, that is $\bar{S} \leq s^*$. This means that there exists a $\delta > 0$ such that $\bar{S} + \delta = s^*$. The cost in these points is given by

$$C(\bar{s}+\delta, \bar{S}+\delta) = \frac{K + c(\bar{S}+\delta) + \int_0^{\bar{S}-\bar{s}} c(\bar{S}+\delta-t)U(dt)}{U(\bar{S}-\bar{s})}.$$

Since $c(\bar{S}+\delta) = c(s^*) = \min_S c(S)$, and c decreases on $(-\infty, s^*]$, it follows that the former expression is smaller than

$$\frac{K + c(\bar{S}) + \int_0^{\bar{S}-\bar{s}} c(\bar{S}-t)U(dt)}{U(\bar{S}-\bar{s})} = C(\bar{s}, \bar{S}),$$

in conclusion, $C(\bar{s}+\delta, \bar{S}+\delta) < C(\bar{s}, \bar{S})$, which is a contradiction with the optimality of the policy $(\bar{s}, \bar{S})$, proving that $\bar{S} > s^*$. Suppose now that $\bar{s} > s^*$, then there exists a $\delta > 0$ such that $\bar{s} - \delta = s^*$. By a similar argument as before we obtain that $C(\bar{s}-\delta, \bar{S}-\delta) < C(\bar{s}, \bar{S})$, which is a contradiction, concluding thus $s^* > \bar{s}$.

If $(\bar{s}, \bar{S})$ is an optimal policy and the inequality in relation (3.3) holds, then

$$c(s) < \min_{S>s^*} C(s,S) \leq C(s, \bar{S}),$$

and this implies by Lemma 2.1, (2.6), that $\bar{s} < s < s^*$. Since c is non increasing, we obtain that relation (3.3), i.e.,

$$c(\bar{s}) = C(\bar{s}, \bar{S}) \leq \min_{S>s^*} C(s,S) < c(s)$$

implies $s < \bar{s}$. □

One can also derive bounds for the optimal cost, which will generate an upper bound for the optimal order-up-to level. This upper bound is tighter than the one presented by Zheng and Federgruen(1991, Lemma 2).

Lemma 3.2. *Let C^* denote the optimal cost achieved with the optimal policy $(\bar{s}, \bar{S})$, that is,*

$$C^* = C(\bar{s}, \bar{S}) = \min_{s,S} C(s, S).$$

The following inequality holds:

$$C^* \geq K(1 - F_Y(\bar{S} - \bar{s})) + c(\bar{S}). \tag{3.4}$$

This generates an upper bound $S^u := \sup\{S > s^* : C^* \geq K(1 - F_Y(S - \bar{s})) + c(S)\}$ for the optimal order-up-to level.

Proof. For any function f, define the shift operator ϕ_s, $s \in \mathbb{R}$ as $\phi_s f(x) := f(s + x)$ for all x, and define $\bar{C}$ as

$$\bar{C}(x) := K + (\phi_s c * U)(x).$$

Straightforward calculation of $\bar{C} * F_Y$ yields

$$\bar{C}(x) = \phi_s c(x) + K(1 - F_Y(x)) + (\bar{C} * F_Y)(x),$$

hence the cost function $C(s, S)$ can be written in terms of $\bar{C}$, and

$$C(s, S) = \frac{\bar{C}(S - s)}{U(S - s)} = \frac{c(S) + K(1 - F_Y(S - s)) + (\bar{C} * F_Y)(S - s)}{U(S - s)}.$$

Since C^* is the optimal cost, obviously $C^* \leq C(s, S)$ for all s, S, thus $C^*U(S-t-s) \leq C(s, S-t)U(S-t-s) = \bar{C}(S-t-s)$, for all $0 \leq t \leq S-s$. This implies that

$$C(s, S) \geq \frac{c(S) + K(1 - F_Y(S - s)) + ((C^*U) * F_Y)(S - s)}{U(S - s)},$$

and the last term is equal to

$$C^* + \frac{c(S) + K(1 - F_Y(S - s)) - C^*}{U(S - s)}.$$

This yields in particular that

$$0 \geq \frac{c(\bar{S}) + K(1 - F_Y(\bar{S} - \bar{s})) - C^*}{U(\bar{S} - \bar{s})},$$

which implies obviously the conclusion of the lemma. □

Remark 3.3. Suppose that $(\bar{s}, \bar{S})$ is the optimal policy. $\bar{s}$ is optimal if and only if $C(\bar{s}, \bar{S}) = c(\bar{s})$ (cf. Lemma 2.1). Substituting this into relation (3.4) (in Lemma 3.2) we obtain:

$$c(\bar{s}) \geq K(1 - F_Y(\bar{S} - \bar{s})) + c(\bar{S}).$$

The derivative of the cost function $C(s, S)$ with respect to S is given by

$$C_S(s, S) = h(s, S) - C_s(s, S), \tag{3.5}$$

where $C_s(s, S)$ is given by relation (2.8), and $h(s, S)$ is given by

$$h(s, S) := \frac{\int_0^{S-s} c_S(S - t)U(dt)}{U(S - s)}. \tag{3.6}$$

The higher order derivatives of h yield the higher order derivatives of the cost function C, and the former are given by

$$h_s(s, S) = (h(s, S) - c_s(s))\frac{m(S - s)}{U(S - s)}, \tag{3.7}$$

$$h_S(s, S) = -h_s(s, S) + \Omega(s, S), \tag{3.8}$$

where Ω is given by

$$\Omega(s, S) := \frac{\int_0^{S-s} c_{SS}(S - t)U(dt)}{U(S - s)} + (c_S(s^{*+}) - c_S(s^{*-}))\frac{m(S - s^*)}{U(S - s)}. \tag{3.9}$$

Assumption 3.4. We assume in the rest of the paper that c is convex, such that relation (2.4) holds and $c_s(s) \geq 0$ for all $s > s^*$.

Note, that c is not necessarily strictly convex.

Lemma 3.5. *The derivative of the function $h(s, S)$ w.r.t. S is given by relation (3.8). Moreover, the function $\Omega(s, S)$, defined by relation (3.9), is positive for every $s \leq s^* \leq S$.*

Proof. Consider now the decomposition of $h(s, S)$:

$$h(s,S) = \int_{0-}^{(S-s^*)^-} c_S(S-t)\frac{U(dt)}{U(S-s)} + \int_{(S-s^*)^+}^{S-s} c_S(S-t)\frac{U(dt)}{U(S-s)},$$

and take the derivative of the two terms with respect to S. The expression for h_S results immediately. We consider now two cases: when c_s is continuous in s^* and when c_s is not continuous in s^*.

If c_s is continuous in s^* then, since $c_s(s) < 0$ for $s < s^*$ and $c_s(s) > 0$ for $s > s^*$, it is not possible that $c_{ss}(s) = 0$ in a neighborhood of s^*. This yields that $\Omega(s, S) > 0$ and the term

$$(c_S(s^{*+}) - c_S(s^{*-}))\frac{m(S-s^*)}{U(S-s)} = 0.$$

If, on the other hand, c_s is not continuous in s^*, then we obtain

$$(c_S(s^{*+}) - c_S(s^{*-}))\frac{m(S-s^*)}{U(S-s)} > 0,$$

since $s^{*-} < s^*$ and $c_s(s) < 0$ for all $s < s^*$, while $s^{*+} > s^*$ and $c_s(s) \geq 0$ for all $s < s^*$. This yields again that $\Omega(s, S) > 0$. □

Lemma 3.6. *The function $h(s, S)$ is increasing in s.*

Proof. Let's decompose the expression (3.6) in the following way:

(3.10)
$$h(s,S) = \int_{0-}^{(S-s^*)^-} c_S(S-t)\frac{U(dt)}{U(S-s)} + \int_{(S-s^*)^+}^{S-s} c_S(S-t)\frac{U(dt)}{U(S-s)}.$$

Since c_S is a non decreasing function (c is convex), the previous term is greater or equal than

$$c_S(s^*)\frac{U(S-s^*)}{U(S-s)} + c_S(s)\frac{U(S-s) - U(S-s^*)}{U(S-s)} \;>\; c_S(s),$$

having $c_S(s) < 0$ and $c_S(s^*) \geq 0$. Summarizing these relations yields that for all $s < s^*$ and all S

$$h(s, S) > c_s(s). \tag{3.11}$$

This together with (3.7) implies immediately that

$$h_s(s, S) > 0, \tag{3.12}$$

that is, h is increasing with respect to s. □

While the marginal cost relation provides the iterations for the optimal s, finding each time a stationary point for a fixed S, we need to solve now $C_S(s, S) = 0$. Considering the form of C_S (see relation (3.5)), this does not promise an efficient search; instead we can make use of the following remark.

Remark 3.7. If for a fixed order-up-to level S_0 the reorder level s_0 represents a local minimum for $C(s, S)$, then $h(s_0, S_0) = 0$ if and only if S_0 is a stationary point for $C(s, S)$.

4. A fast algorithm for the continuous case

4.1. **Looking for the global minimum.** Consider the following optimality and pseudo-optimality equations (see relation (2.5) and Remark 3.7)

$$C(s, S) = c(s) \quad \text{and} \tag{4.1}$$

$$h(s, S) = 0, \tag{4.2}$$

and define $\phi(S)$ and $\psi(S)$ respectively:

$$\phi(S) := \{\phi \in I\!R \ : \ C(\phi, S) = c(\phi)\}, \tag{4.3}$$

$$\psi(S) := \{\psi \in I\!R \ : \ h(\psi, S) = 0\}. \tag{4.4}$$

Since (4.1) has a unique solution for every S (see Lemma 2.1) ϕ is a well defined function. Consider now an arbitrarily fixed S_0. Since $h(s^*, S_0) > 0$, while $h(-\infty, S_0) < 0$, the equation $h(s, S_0) = 0$ certainly has a solution in $(-\infty, s^*]$. Knowing that $h_s(s, S_0) > 0$ (cf. Lemma 3.6), we can conclude that this solution is unique. Hence, ψ is also a well defined function. Now, if equations (4.1) and (4.2) are simultaneously satisfied for a pair (s_0, S_0), then this point is a stationary point for C. By the definition of ϕ and ψ, for

this stationary point $\phi(S_0) = \psi(S_0)$, thus an intersection point of the two functions. But can we possibly find every intersection point of ϕ and ψ, and at what cost (complexity)? Which of these intersection points represent a local minimum for the total cost, and how can we filter them? This section deals with these questions.

Taking derivatives in relations (4.1) and (4.2) with respect to S, yields the first order derivatives of ϕ and ψ:

$$\phi_S(S) = \frac{C_S(\phi(S), S)}{c_\phi(\phi(S))}.$$

Relations (3.5), (4.1) and Lemma 2.1 imply together that $C_S(\phi(S), S) = h(\phi(S), S)$, yielding

$$\phi_S(S) = \frac{h(\phi(S), S)}{c_\phi(\phi(S))}. \tag{4.5}$$

Similarly,

$$\psi_S(S) = \frac{-h_S(\psi(S), S)}{h_\psi(\psi(S), S)}.$$

By relation (3.8) this becomes

$$\psi_S(S) = 1 - \frac{\Omega(\psi(S), S)}{h_\psi(\psi(S), S)}. \tag{4.6}$$

Remark 4.1. The function ϕ has a stationary point in the intersection points with ψ, that is, if $\phi(S_0) = \psi(S_0)$ then $\phi_S(S_0) = 0$, and these are the only stationary points for ϕ.

What do these first order derivatives tell us about ϕ and ψ? The answer is summarized in the following lemma.

Lemma 4.2.

$$\phi_S(S) < 1 \quad \textit{and} \quad \psi_S(S) < 1; \tag{4.7}$$

that is, neither of the functions increases steeper than the bisector of the first quarter. Moreover, $\phi(S) < s^*$, $\phi_S(s^*) > 0$, $\psi(s^*) = s^*$.

Proof. Relation (3.11) and $c_s(s) < 0$ for all $s < s^*$ imply together that $h(\phi(S), S)/c_\phi(\phi(S)) < 1$, that is $\phi_S(S) < 1$ (cf. (4.5)). The strict positivity of $\Omega(s, S)$ (cf. Lemma 3.5) and $h_s(s, S)$ (cf. Lemma 3.6) yield trivially that $\psi_S(S) < 1$. The last statement of the Lemma is trivial, and can be verified by direct computations. □

The second order derivative $\phi_{SS}(S)$ of ϕ is given by

$$\frac{h_\phi(\phi(S), S)\phi_S(S) + h_S(\phi(S), S)}{c_\phi(\phi(S))} - \frac{h(\phi(S), S)c_{\phi\phi}(\phi(S))\phi_S(S)}{c_\phi^2(\phi(S))}.$$

We already know by Remark 4.1 that ϕ has a local minimum or maximum in the intersection points with ψ. The second order derivative gives more information, namely, if $\phi(S_0) = \psi(S_0)$, then

$$\phi_{SS}(S_0) = \frac{h_S(\psi(S_0), S_0)}{c_\phi(\psi(S_0))},$$

since $\phi_S(S_0) = 0$. Multiplying by $h_\psi(\psi(S_0), S_0)/h_\psi(\psi(S_0), S_0)$ and using expression (3.7) yields

$$\phi_{SS}(S_0) = \psi_S(S_0)\frac{m(S_0 - \psi(S_0))}{U(S_0 - \psi(S_0))}. \tag{4.8}$$

In conclusion, we have two types of intersection points: the first is such that ψ is decreasing and it intersects ϕ in a local maximum, the second type is when ψ is increasing and it meets ϕ in a local minimum (see Figure 1). This gives us a lot of information about the behaviour of the two functions. Before the first type of intersection points the function ψ is decreasing, and since the intersection point itself is a maximum point for ϕ, it will increase until the intersection and it decreases afterwards. Let us summarize this in the following proposition.

Proposition 4.3. *$\phi(S) > \psi(S)$ if and only if $\phi_S(S) < 0$, while $\phi(S) < \psi(S)$ if and only if $\phi_S(S) > 0$. Moreover, if $\psi_S(S_0) < 0$ and $\phi(S_0) = \psi(S_0)$ then $\phi_{SS}(S_0) < 0$ and if $\psi_S(S_0) > 0$ and $\phi(S_0) = \psi(S_0)$ then $\phi_{SS}(S_0) > 0$.*

Proof. Suppose that $\phi(S) > \psi(S)$. Then, knowing that $h(s, S)$ is increasing in s (see Lemma 3.6) $h(\phi(S), S) > h(\psi(S), S) = 0$. Furthermore, since $c_s(s) < 0$ $(s < s^*)$, we obtain

$$\phi_S(S) = \frac{h(\phi(S), S)}{c_\phi(\phi(S))} < 0.$$

If, in turn, we suppose that $\phi_S(S) < 0$, this implies immediately by (4.5) that $h(\phi(S), S) > 0$. By the definition of ψ, $h(\psi(S), S) = 0$, and knowing that h is increasing in its first variable, these statements imply together that $\phi(S) > \psi(S)$.

When $\phi(S) < \psi(S)$, using the same reasoning as before, we obtain $\phi_S(S) > 0$. The remainder of the Proposition was proven by relation (4.8). □

It only remains to check now which of these intersection points of ϕ and ψ represent a local minimum for the total cost function C. Suppose that (s_0, S_0) is a stationary point for $C(s, S)$ (obviously $s_0 = \phi(S_0) = \psi(S_0)$), then the Hessian of C in (s_0, S_0) is

$$H(s_0, S_0) = \begin{bmatrix} -c_s(s_0)\frac{m(S_0 - s_0)}{U(S_0 - s_0)} & 0 \\ 0 & c_s(s_0)\frac{m(S_0 - s_0)}{U(S_0 - s_0)} + \Omega(s_0, S_0) \end{bmatrix}, \tag{4.9}$$

where Ω was defined by relation (3.9). In view of the definitions of $\psi_S(S)$ (see relation (4.6)) and Ω, the Hessian can be written in the form

$$H(s_0, S_0) = -c_s(s_0)\frac{m(S_0 - s_0)}{U(S_0 - s_0)} \begin{bmatrix} 1 & 0 \\ 0 & -\psi_S(S_0) \end{bmatrix}. \tag{4.10}$$

Indeed, this form yields us immediately the answer for the question: which intersection points of ϕ and ψ represent a local minimum for C?

Proposition 4.4. *If (s_0, S_0) is a stationary point for $C(s, S)$ such that $\psi_S(S_0) < 0$ and $\phi_S(S_0) = 0$, that is, S_0 is a local maximum for ϕ, then (s_0, S_0) is a local minimum for $C(s, S)$.*

4.2. **The algorithm.** What is left to do now is to define a search algorithm which finds all the intersection points of ϕ and ψ. For this purpose Lemma 4.2 and Proposition 4.3 will be of great help. Indeed, Lemma 4.2 asserts that neither ϕ nor ψ increases steeper than 45 degrees. Assume now, that having taken an arbitrary S_0, $\phi(S_0) < \psi(S_0)$. We know then from Proposition 4.3 that $\phi(S)$ is increasing. Let us take in this point a secant of 45 degrees. $\psi_S(S) < 1$ guarantees that this line will intersect ψ in strictly one point, say S_1. Furthermore $\phi_S(S) < 1$ guarantees that the 45 degree secant will not intersect ϕ in any other point than S_0. Repeat now the previous step for $\phi(S_1)$, obtaining S_2, and so on. We want to prove that $\{S_k : k \geq 1\}$ is converging monotonically to S^*, where $S^* := \inf\{S > S_0 : \phi(S) = \psi(S)\}$. When $\phi(S_0) > \psi(S_0)$ we proceed in exactly the same way and then the algorithm will converge monotonically to the left from the starting point.

Lemma 4.5. *Consider s and S fixed. The function $\Gamma(t) := C(s+t, S+t)$ is strictly convex in t. Moreover, t_0 is optimal for Γ if and only if $h(s+t_0, S+t_0) = 0$.*

Proof. The optimality condition is that the derivative of Γ w.r.t. t has to be 0; that is, $C_s(s+t_0, S+t_0) + C_S(s+t_0, S+t_0) = h(s+t_0, S+t_0) = 0$. It only remains to check if the second derivative is positive:

$$\Gamma_{tt}(t) = h_s(s+t, S+t) + h_S(s+t, S+t) = \Omega(s+t, S+t) > 0,$$

where Ω is given by (3.9). □

Let us summarize the results we found so far in a subroutine (which finds a local minimum for C) and prove convergence formally.

Subroutine: LM(S_0)

1. input S_0, $k := 0$;
2. calculate s_0 such that $C(s_0, S_0) = c(s_0)$;

repeat

3. $t_k := \arg\min_t C(s_k + t, S_k + t)$;
4. $S_{k+1} := S_k + t_k$;

5. calculate s_{k+1} such that $C(s_{k+1}, S_{k+1}) = c(s_{k+1})$;
6. $k := k + 1$

until $t_k = 0$.

7. $LM := S_k$;

Theorem 4.6. *If S_0 is the starting point and the previously described subroutine converges monotonically to S^*, then the cost reaches a (local) minimum in S^*. Moreover, the subroutine always finds the closest[2] minimum point to S_0, such that, if $S^* < S_0$ then there is no other stationary point in $[S^*, S_0]$ and if $S^* > S_0$ then there is no other stationary point in $[S_0, S^*]$.*

Proof. Suppose that $\psi(S_k) > \phi(S_k)$. Since $\psi_S(S) < 1$ the 45 degree secant $(\Phi(S_k) + t, S_k + t),\ t \geq 0$ intersects $\psi(S)$ strictly in one point, S_{k+1}. That is,

$$\Gamma_t(t)|_{t=0} = C_t(\Phi(S_k) + t, S_k + t)|_{t=0} = h(\Phi(S_k), S_k) < 0,$$

since $h(\psi(S_k), S_k) = 0$, $\psi(S_k) > \phi(S_k)$ and $h(s, S)$ is increasing in s. Hence Γ is decreasing from $\phi(S_k)$ and since it is strictly convex it has strictly one minimum, say S_{k+1} (the intersection point, cf. Lemma 4.5), such that

$$S_{k+1} > S_k. \tag{4.11}$$

The fact that $\phi_S(S) < 1$ and $\psi_S(S) < 1$ implies for all $S_k < S < S_{k+1}$ that

$$\phi(S) < \phi(S_k) + (S - S_k), \tag{4.12}$$

$$\psi(S) > \psi(S_{k+1}) + (S - S_{k+1}). \tag{4.13}$$

Hence, for every $S_k \leq S < S_{k+1}$, relations (4.12) and (4.13) imply together that

$$\phi(S) < \phi(S_k) + (S_{k+1} - S_k) + (S - S_{k+1}) = \psi(S_{k+1}) + (S - S_{k+1}) < \psi(S).$$

[2]By "closest minimum" we understand the following: if the starting point is between two stationary points of ϕ, then the subroutine will find the stationary point which is a maximum point for ϕ, that is, a minimum point for C; it is not necessarily the closest in norm (distance).

In conclusion, $\phi(S) < \psi(S)$ for all $S_k \leq S < S_{k+1}$, that is, there are no stationary points for C in (S_k, S_{k+1}). Since S_k is an increasing sequence (cf. relation (4.11)) and it is bounded by the intersection point of the two functions $\phi(S) = \psi(S)$[3], S_k is convergent, hence the subroutine converges monotonically, too, and we have $C(s_{k+1}, S_{k+1}) < C(s_k, S_k)$ for all $k \geq 0$.

In the case when $\psi(S_k) < \phi(S_k)$ we proceed exactly in the same way, obtaining a decreasing sequence $S_{k+1} < S_k$. In conclusion, the subroutine always converges to a stationary point S_n such that $\psi_S(S_n) < 0$. In view of Proposition 4.4 this implies that the subroutine always converges to a (local) minimum. □

Remark 4.7. $C(s, S) = c(s)$ and $h(s, S) = 0$ are nonlinear equations, in fact, the first means to compute $s = \phi(S)$, while the second is to compute $s = \psi(S)$. In Section 4.2.1 a detailed explanation is given about the computation of these two functions. Yet, it is important to remark now how we achieve that the local optimum $\bar{S}$ is not overshot, thus insuring that the iterations are monotone: When the iterations begin at $S_0 < \bar{S}$ (thus the sequence increases monotonically to the right), we replace Step 5 with $0 \leq C(s, S) - c(s) \leq \varepsilon$ and the stopping condition with $0 \leq t_k \leq \varepsilon$, for any $\varepsilon > 0$ (obviously, $h(s, S) < 0$). Analogously, when $S_0 > \bar{S}$ (thus the sequence decreases monotonically to the left), we replace Step 5 with $-\varepsilon \leq C(s, S) - c(s) \leq 0$ and the stopping condition with $-\varepsilon \leq t_k \leq 0$, for any $\varepsilon > 0$.

The core of the algorithm is the subroutine LM which finds the local minimum, but it is equally important to define a search which finds the global optimum in a fast and efficient way. For this purpose the next lemma will be of great help, it will assure that the algorithm is fast and efficient.

Lemma 4.8. *The global minimum of $C(s, S)$, $C^* = C(\bar{s}, \bar{S})$ is reached exactly in the global maximum of the function ϕ, $\phi(\bar{S}) = \bar{s}$.*

[3]It is also easy to check that this upper bound is at the same time the lowest upper bound of the sequence

Proof. The proof is very simple and it is based on Proposition 4.4 (any local minimum of C is reached in a local maximum of ϕ) and the assumption that c is non increasing on $(-\infty, s^*]$. $C^* = C(\bar{s}, \bar{S})$ is the global minimum, which means that $\bar{s}$ is a minimum, that is, $C(\bar{s}, \bar{S}) = c(\bar{s})$ (cf. Lemma 2.1). Moreover,

$$c(\phi(\bar{S})) = c(\bar{s}) = C(\bar{s}, \bar{S}) \ < \ C(\hat{s}, \hat{S}) = c(\hat{s}) = c(\phi(\hat{S}))$$

for any other local minimum $(\hat{s}, \hat{S})$. Since c is non increasing, it follows that $\phi(\bar{S}) > \phi(\hat{S})$, for all $\hat{S}$ local maximum for ϕ (cf. Proposition 4.4). □

Now we are ready to proceed with the description of the algorithm. Before providing a detailed description we give the general idea behind the algorithm in three major steps. **Step 0**: It is trivial to start the search at the lower bound s^* by running the subroutine $LM(s^*)$, obtaining $S_0 := LM(s^*)$, with $(\phi(S_0), S_0)$ representing a local minimum for C. (Since $\psi(s^*) = s^* > \phi(s^*)$, trivially $S_0 > s^*$.) The difficulty arises at this point. Since the subroutine always finds the closest stationary point, we have to step away "far enough" otherwise the subroutine would find back S_0 again and again. The upper bound S^u (cf. Lemma 3.2) yields the solution: **Step 1**: restart the search at the upper bound, finding an optimum S_1 such that: a) $\phi(S_1) > \phi(S_0)$ (that is $C(\phi(S_1), S_1) < C(\phi(S_0), S_0)$, cf. Lemma 4.8) and b) $\phi(S_1) > \phi(S)$ for all $S \in (S_1, S^u]$. **Step 2**: With the help of the new maximum, $\phi(S_1)$ we can restart the search in S_0, obtaining S_2, such that $\phi(S_2) > \phi(S_1)$. We construct thus iterations from both of the ends of the feasibility interval of the form $[S_{k-1}, S_k]$ (S_{k-1}, S_k local optima's) until the two ends meet, reducing the interval to 0. The last found optima is the optimal policy.

There's one more question we still have to answer before giving the algorithm, and that is: how to find the closest maximum for ϕ, which is bigger than the previously found maximum. That is, find the point where ϕ increases to the level of the last maximum (say, S_k^u and S_k^l) and in that

point restart the subroutine LM. In Step 1 we proceed in the following way: Let the feasibility interval be (S_{k-1}, S_{k-2}) at this moment, with $\hat{s}_{k-1} := \phi(S_{k-1}) > \phi(S_{k-2})$, the last found maxima for ϕ. The aim is to find the level crossing point:

$$S_k^u := \sup\{S < S_{k-2} : \phi(S) = \hat{s}_{k-1}\}. \tag{4.14}$$

Due to $\phi_S(S) \leq 1$ (cf. Lemma 4.2), the 45 degree secant in the point $(S_{k-2}, \hat{s}_{k-1})$ intersects ϕ in strictly one point, say S^1, with $S^1 < S_{k-2}$ and $\phi(S^1) \leq \hat{s}_{k-1}$. Taking this secant repeatedly in the points $(S^n, \hat{s}_{k-1}), \ldots$ we obtain a decreasing sequence $\ldots < S^n < \ldots < S^1$, which converges to S_k^u. (Later we start the subroutine $LM(S_k^u)$ in this point, obtaining the new maxima $\hat{s}_k > \hat{s}_{k-1}$ with $\hat{s}_k := \phi(S_k)$, and the new feasibility interval $[S_{k-1}, S_k]$.) Let us summarize this in the subroutine:

Subroutine: ISU($S_{k-2}, \hat{s}_{k-1}$)

1. $n := 0$; $S_0 := S_{k-2}$; $t_0 = 1$;

while $t_n > \varepsilon$ do begin

2. $n := n + 1$;
3. t_n is the solution of $0 \leq C(\hat{s}_{k-1} - t, S_{n-1} - t) - c(\hat{s}_{k-1} - t) \leq \varepsilon$;
4. $S_n := S_n - t_n$;

end.

5. $ISU := S_n$;

Step 2 works similarly, with the difference that the level crossing point is given by

$$S_k^l := \inf\{S > S_{k-1} : \phi(S) = \hat{s}_k\},$$

thus we obtain an increasing sequence $\{S^n\}$, (starting from S_{k-1}), such that we take secants first in the point $(S_{k-1}, \phi(S_{k-1}))$, which intersects the line $\hat{s}_k$ in S^1. $\phi_S(S) < 1$ insures that $\phi(S^1) \leq \hat{s}_k$. Taking this secant repeatedly in the points $(S^n, \phi(S^n)), \ldots$ we obtain an increasing sequence $S^1 < \ldots < S^n < \ldots$, which converges to S_k^l. (Start then the subroutine $LM(S_k^l)$ in this

point, obtaining the new maxima $\hat{s}_{k+1} > \hat{s}_k$ with $\hat{s}_{k+1} := \phi(S_{k+1})$, and the new feasibility interval $[S_{k+1}, S_k]$). The subroutine is given as follows:

Subroutine: ISL($S_{k-1}, \hat{s}_k$)

1. $n := 0$; $S_0 := S_{k-1}$; $\delta_0 := 1$;

while $\delta_n > \varepsilon$ do begin

2. $n := n + 1$;
3. $\delta_n := \hat{s}_k - s$, where s is the solution of $0 \leq C(s, S_{n-1}) - c(s) \leq \varepsilon$;
4. $S_n := S_{n-1} + \delta_n$;

end.

5. $ISL := S_n$;

Having established these subroutines, the algorithm itself is simple. The search stops when the feasibility interval reduces to zero.

The algorithm:

1. (input ε); $S_0 := LM(s^*)$; $\hat{s}_0 := \phi(S_0)$; $c_0 := c(\hat{s}_0)$;
2. **while** $c_0 \geq c(S_0 + \Delta)$ **do** $\Delta := 2 \cdot \Delta$;
   $S_1 := S_0 + \Delta$;
   if $\phi(S_1) < \hat{s}_0$ then $RP := ISU(S_1)$; $S_1 := LM(RP)$;
   $\hat{s}_1 := \phi(S_1)$;
   $k := 1$;
3. **while** $S_k - S_{k-1} > \varepsilon$ **do begin**
4. $RP := ISL(S_{k-1}, \hat{s}_k)$;
5. $S_{k+1} := LM(RP)$; $\hat{s}_{k+1} := \phi(S_{k+1})$;
6. $RP := ISU(S_k, \hat{s}_{k+1})$;
7. $S_{k+2} := LM(RP)$; $\hat{s}_{k+2} := \phi(S_{k+2})$;
8. $k := k + 2$;

   end;
9. $S^* := S_k$; $s^* := \hat{s}_k$;

The algorithm always converges to the global minimum of C, the optimal policy is (s^*, S^*). An example for the iterations made, using the functions ϕ and ψ, is given in figure 1. The parameters are, as follows: $L = 1$,

$\lambda = 1$, $h = 1$, $p = 10$, $K = 1$. The individual demands are distributed with a Gamma distribution, with parameters $\alpha = \beta = 200$. In this case, the

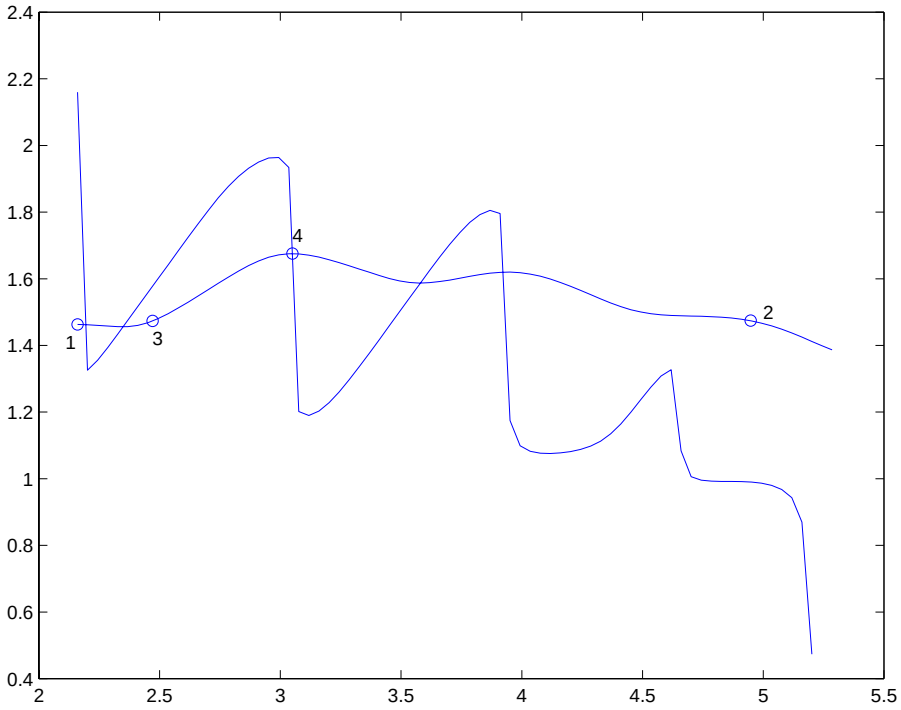

FIGURE 1. Some iterations of the algorithm

global optimum is found in four iterations, numbered on the graph as they follow. After having found the first local maximum of ϕ, S_0, of value (1), the algorithm finds the upper bound for the optimal order-up-to level, S_1, of function value (2). Since (2) is larger than (1), (2) automatically defines the new relevance level. The next step is starting $ISL(S_0, (2))$, obtaining S_3, the level crossing point, (3). $LM(S_3)$ finds (4), which turns out to be the global maximum for ϕ, since $ISU(S_2, (4))$ finds back the same point, reducing the feasibility interval to zero. The optimal policy is $(1.6754, 3.0503)$.

Remark 4.9. Steps 5 and 7 require the computation of $\hat{s}_k := \phi(S_k)$, where $\phi(S_k)$ is calculated for instance with bisection (see Section 4.2.1). To insure that that we do not overshoot the global minimum, just as we did for the sake

of monotonicity in Remark 4.7, we approximate $\hat{s}_k := \phi(S_k)$ such that $-\varepsilon \leq C(s, S_k) - c(s) \leq 0$, for any $\varepsilon > 0$. This will ensure that $\hat{s}_k \leq \hat{s}^*$, for every $k \geq 1$, where $\hat{s}^*$ represents the global maximum of ϕ. This means in fact, that the algorithm will never overshoot the global minimum of C (see Lemma 4.8), that is, for the found optimal policy $C(s^*, S^*) \geq min_{0 \leq s < S} C(s, S)$.

4.2.1. *Calculating $\phi(S)$ and $\psi(S)$.* We owe the reader one more explanation, that is, how we solve the nonlinear equations $C(s, S_0) = c(s)$ and $h(s, S_0) = 0$ (which is, in fact, calculating $s = \phi(S_0)$ and $s = \psi(S_0)$) for a fixed S_0. For most of the software packages it is standard to solve equations which have only got a single root (cf. Lemma 2.1 this is the case for $\phi(S)$, and cf. Lemma 3.6 for $\psi(S)$). However, we propose an approximation, which is easy to evaluate, so much the more for instance in Step 7 the precision of the value of ϕ does not influence the precision of the found local maximum in Step 8, thus it does not influence the precision of the optimal policy. The precision of the optimal policy is only important in the last evaluation of the subroutine LM. Since we can choose any $\varepsilon > 0$, any precision can be attained. We describe now a bisection method because it is more accessible, although there are faster methods, such as the Newton method.

Subroutine PHI(S)**:**

$l := \hat{s}_k$; $s_1 := \hat{s}_k - l/2$; $i := 0$;
while $l > \varepsilon$ **do begin**
  $l := l/2$; $i := i + 1$;
  If $C(s_i, S) < c(s_i)$
    then $s_{i+1} := s_i + l/2$;
    else $s_{i+1} := s_i - l/2$;
end {while}

We know from Lemma 2.1, relation (2.6) that $C(s, S) > c(s)$ if and only if $s > \bar{s}$ (where $C(\bar{s}, S) = c(\bar{s})$). The rest of the routine speaks for itself. The subroutine PSI(S) is similar to PHI(S), except that in the if case the condition is $h(s_i, S) < 0$.

4.2.2. *Comparisons with the Federgruen-Zipkin algorithm.* Although our approach is different from that of Federgruen and Zipkin (1985), there are some common ideas. Let us go through the differences and common idea's step by step.

0. The bounds for the optimal policy are tighter in the present paper.

1. We also need to compute the cost function $C(s,S)$, that is, g_R: We first calculate it's Laplace transform, which is a closed form expression, due to its convolution structure (cf. Bázsa and Iseger (2000)). Then we invert the Laplace transform, obtaining a very accurate approximation for $C(s,S)$ (the approximation is that of a machine precision, cf. Iseger (2000)).

2(a). The approach of the search for an optimal order up to level S is completely different. Federgruen and Zipkin are minimizing the relative cost $v_R(y)$ for a fixed reorder point s. Since the cost function is not convex w.r.t. S, this can be a difficult problem. We exploit the very convenient properties of the functions $\phi(S)$ and $\psi(S)$, while always obtaining a policy improvement.

2(b) Finding an optimal s for a fixed S. In (ii) of Federgruen and Zipkin, the equation $G(x) = g_R$ is in fact $C(s,S) = c(s)$, that is, calculating $\phi(S)$. We proved that a unique solution exists and we also gave a simple procedure for finding this root. The cases (i) and (ii) can be interpreted as the two conditions for being under or below the graph of the function $\phi(S)$, exactly what we also exploit in our procedure.

4.2.3. *Speed of convergence.* The algorithm basically consists of repeated evaluations of the subroutines LM, ISU, ISL, PHI, PSI, and the functions $C(s,S)$ and $c(s)$. The evaluations of the functions $C(s,S)$ and $c(s)$ are done with a Laplace transform inversion method (cf. Iseger(2000)): due to their convolution structure, their Laplace transform is easy to calculate, which is then inverted. The results are accurate (up to machine precision) and they are obtained in fractions of time. The subroutines PHI and PSI use a simple bisection or Newton method for finding the unique solution of a

nonlinear equation. The results are accurate up to an ε precision, for any $\varepsilon > 0$. However, it pays off not to choose ε very small, since it does not effect the convergence of the algorithm to the global minimum, while a larger ε can make the algorithm even faster. The three subroutines, LM, ISU, and ISL have the same speed of convergence, since in a neighborhood of the limit point they are very similar. Denoting with l_n the distance from the limit point at the nth iteration, we obtain for the different subroutines the following expressions for l_{n+1}:

- subroutine LM: $l_{n+1} = (1/(1 - \psi_S(\hat{S}_k)))l_n$, with $\psi_S(\hat{S}_k) < 0$.
- subroutine ISU: $l_{n+1} = (1/(1 - \phi_S(RP)))l_n$, with $\phi_S(RP) < 0$.
- subroutine ISL: $l_{n+1} = (1 - \phi_S(RP))l_n$, with $\phi_S(RP) > 0$,

where $\hat{S}_k$ is the intersection point of ϕ and ψ, and RP is the point where ϕ crosses the actual relevance level. This means that the subroutines converge linearly, such that $l_{n+1} = \alpha l_n$, with $0 < \alpha < 1$. It is also important to remark that if N is the number of the local optimums (N is always finite), then the algorithm will execute a local search at most N times. Furthermore, from the speed of convergence expressions we can deduce the following relations:

LM: If the subroutine goes from to the left to the right we have: $l_{n+1} - l_n = \psi_S(\hat{S}_k)l_{n+1}$ and $S_{n+1} - S_n = l_n - l_{n+1}$ imply, that taking a precision $\varepsilon > 0$ yields $\varepsilon = \psi_S(\hat{S}_k)l_{n+1}$. The Taylor expansion of ϕ in the point $\hat{S}_k$ yields $\phi(S_{n+1}) - \phi(\hat{S}_k) = 1/2\phi_{SS}(\hat{S}_k)l_{n+1}^2$, since $\phi_S(\hat{S}_k) = 0$ and $l_{n+1} = \hat{S}_k - S_{n+1}$. Substituting this into the speed of convergence expression, we obtain for the convergence of the ϕ values to the local maximum, that

$$\phi(S_{n+1}) - \phi(\hat{S}_k) = \frac{\varepsilon^2 \phi_{SS}(\hat{S}_k)}{2\psi_S^2(\hat{S}_k)}, \tag{4.15}$$

where $\phi_{SS}(\hat{S}_k) < 0$. When the subroutine converges from the right to the left we obtain the same result.

ISU: $l_{n+1} - l_n = \phi_S(RP) l_{n+1} = \phi(S_{n+1}) - \phi(RP)$ $(\phi_S(RP) < 0)$, that is, $\phi(RP) - \phi(S_{n+1}) = \varepsilon$, for an $S_n - S_{n+1} = l_n - l_{n+1} = \varepsilon$ step size $(\forall \varepsilon > 0)$.

ISL: $l_n - l_{n+1} = \phi_S(RP) l_n$ $(0 < \phi_S(RP) < 1)$, that is, $l_n - l_{n+1} = (\phi_S(RP)/(1-\phi_S(RP))) l_{n+1}$. Having $\phi_S(RP) l_{n+1} = \phi(RP) - \phi(S_{n+1})$ and $S_{n+1} - S_n = l_n - l_{n+1}$, we obtain for any $\varepsilon > 0$ that

$$\phi(RP) - \phi(S_{n+1}) = \varepsilon(1 - \phi_S(RP)),$$

with $0 < \phi_S(RP) < 1$.

Remark 4.10. With regard to the speed of convergence of the subroutines *ISU* and *ISL*, the reader might wonder what happens at the last iteration of the algorithm, when the feasibility interval reduces to zero, yielding $\phi_S(S) = 0$ at the last iterations. In particular, consider the case when $(S_k, \hat{s}_k)$, the global maximum is found, such that $S_k < S_{k-1}$ (obviously, $\hat{s}_{k-1} < \hat{s}_k$). At this instance the feasibility interval is (S_k, S_{k-1}), $|S_k - S_{k-1}| > \varepsilon$. At the next step, $ISU(S_{k-1}, \hat{s}_k)$ should find back $S_{k+1} = S_k$, yielding $|S_{k+1} - S_k| < \varepsilon$. Observe, that in a neighborhood of S_k $\phi_S(S_n) = 0$, yielding $l_{n+1} = l_n$ (check the expression for speed of convergence). However, this also means that $S_{n+1} = S_n$, that is, the subroutine *ISU* stops. Now, the subroutine *LM* is run in this point, but its speed of convergence depends on ψ_S, having $\psi_S(S) << 0$ in a neighborhood of S_k.

References

Bázsa, E. and Iseger, den P. 2000. *Single item inventory models: a time and event averages approach.* Submitted to OR.

Clark, A. and Scarf, H. 1960. *Optimal policies for a multi-echelon inventory problem.* Management Science, 6(475- 490).

Eppen, G. and Schrage, E. 1981. *Centralized order policies in a multi-warehouse system with lead times and random demand.* In Multilevel production/inventory control systems: Theory and practice, ed. L. Schwarz.

North-Holland, Amsterdam.

FEDERGRUEN, A. AND ZIPKIN, P. 1984. *Computational issues in an infinite horizon, multi-echelon inventory model.* Operation Research, 32(818-836).

FEDERGRUEN, A. AND ZIPKIN, P. 1985. *Computing optimal (s, S) policies in inventory models with continuous demands.* Advances in Applied Probabilities, 17(424-442).

ISEGER, DEN P.W. 2001 . *A new method for inverting Laplace transforms.* In preparation.

ROSLING, K. 1989. *Optimal inventory policies for assembly systems under random demands.* Operations research, 37(565-579).

VEINOTT, F. AND WAGNER, H. 1965. *Computing optimal (s, S) inventory policies.* Management Science, 11(525-552).

ZHENG, YS. AND FEDERGRUEN, A. 1991. *Finding optimal (s, S) policies is about as simple as evaluating a single policy.* Operations Research, 39(654-665).