Chapter 2
Basic Multi-Item, Single-Location Inventory Model

2.1 Introduction

In this chapter, we consider a basic multi-item, single-location inventory model for spare parts. This model is appropriate to demonstrate various optimization methods, to describe multiple system-oriented service measures, and to show the effects of alternative or additional model assumptions. The basic model is appropriate for planning in stationary situations. It may be used for both initial supply and inventory planning during the exploitation phase. For the initial supply problem, one has no starting stock, which leads to a cleaner problem formulation than for planning during the exploitation phase. Therefore, we will describe the initial supply problem first, and later we explain how this model can be used for the inventory planning during the exploitation phase.

The single-location model will fit in case one has only one location where spare parts are stocked. But, the model may also be used as a building block in planning concepts for spare parts networks (for both initial supply and the planning during the exploitation phase). This model may fit for:

- A central warehouse in a two-echelon network;
- A local warehouse;
- The total or aggregate stock in a two-echelon network with one central depot and multiple local warehouses when all these stockpoints are at closed distance of each other and emergency/lateral shipments are possible between each pair of stockpoints (in this case, one can operate as if the network as a whole forms one large virtual stockpoint).

This chapter is organized as follows. We will start with the description of the basic model under the assumption of backordered demands, i.e., without the assumption of emergency shipments. This means that a demand is backordered when it can not be satisfied from stock. In line with this assumption we will start with a service level constraint in terms of the aggregate mean number of backordered

G.J. van Houtum, B. Kranenburg, *Spare Parts Inventory Control under System Availability Constraints*, International Series in Operations Research & Management Science 227, DOI 10.1007/978-1-4899-7609-3_2

demands. For the optimization within this basic model, we will use a greedy heuristic. The description of the basic model, its evaluation, and its optimization are given in Sects. 2.2–2.4, respectively. After that, in Sect. 2.5, we discuss two alternative optimization techniques: Lagrangian relaxation and Dantzig-Wolfe decomposition. We will see that both approaches are kind of equivalent. Subsequently, in Sect. 2.6, a comparison is made with the so-called item approach, which is a straightforward inventory optimization approach without a direct connection with system availability. Next, in Sect. 2.7, we discuss alternative service measures, among which average availability. Then the application of the model for the inventory planning during the exploitation phase is discussed in Sect. 2.8. Subsequently, in Sect. 2.9, we discuss the model with emergency shipments, and we describe the changes in the analysis and optimization approaches under this alternative assumption. It will appear that these changes are limited. In Sect. 2.10, we discuss a number of practically relevant extensions for both the case without and with the use of emergency shipments. Finally, we make concluding remarks in Sect. 2.11.

Most of the material in this chapter stems from [20].

2.2 Basic Model

Consider a *single warehouse* where several spare parts are kept on stock to serve an installed base of machines of the same type. The machines consist of multiple components, which may be classified as *critical* and *non-critical* components. When a critical component of a machine fails, the whole machine goes down, while a machine can continue its functioning (i.e., to a sufficiently large extent) upon the failure of a non-critical component. We limit ourselves to the inventories of the spare parts for the critical components. When a critical component fails in a given machine, then the failed part is replaced by a spare part from the warehouse if it is available or as soon as a spare part becomes available; i.e., we have *repair by replacement*. The failed part is returned to the warehouse and is immediately sent into repair. We assume that all critical components are repairable.

We refer to the critical components as Stock-Keeping Units (SKU's). The set of SKU's is denoted by I, and the number of SKU's is denoted by $|I|$ ($\in \mathbb{N} := \{1,2,\ldots\}$). For notational convenience, the SKU's are assumed to be numbered $i = 1,2,\ldots,|I|$. We assume an infinite time horizon $[0,\infty)$. For each SKU $i \in I$, demand occurs according to a Poisson process with a constant rate m_i (≥ 0). The rate m_i denotes the demand rate for all machines together. The total demand rate for all SKU's together is denoted by $M = \sum_{i\in I} m_i$ and we assume that $M > 0$. A demand is fulfilled immediately if possible, and otherwise it is backordered and fulfilled as soon as possible. Each demand is accompanied by the return of a failed part, and the failed part is immediately sent into repair. The time that a failed part is in repair is called the repair leadtime, which consists of waiting time and repair time. Repair leadtimes of parts of different SKU's are assumed to be independent and repair leadtimes of parts of the same SKU are assumed to be independent and identically distributed (i.i.d.). The mean repair leadtimes for SKU i are denoted by t_i (> 0). Because each failed part is immediately sent into repair, the inventory position of SKU i, defined as the physical stock minus backordered demand plus parts in repair, is constant. This constant amount is denoted by S_i ($\in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$).

Instead of saying that each failed part is immediately sent into repair, we may also say that for each SKU the stock is controlled by a continuous-review *basestock policy*, with *basestock level S_i* for SKU i. Basestock level S_i represents the initial stock of SKU i and is a decision variable.

The price of a part of SKU i is c_i^{a} (> 0). We look at the initial supply problem at time instant $t = 0$, i.e., at the investment in spare parts at the beginning of the time horizon. The objective is to minimize the total investment subject to a constraint on the aggregate mean number of backorders. The investment in, or budget spent to, spare parts of SKU i is given by $C_i(S_i) = c_i^{\mathrm{a}} S_i$ and the total investment is given by:

$$C(\mathbf{S}) = \sum_{i \in I} C_i(S_i) = \sum_{i \in I} c_i^{\mathrm{a}} S_i,$$

where $\mathbf{S} = (S_1, \ldots, S_{|I|})$ denotes a vector consisting of all basestock levels. The *mean number of backorders* of SKU i, in steady state (i.e., at an arbitrary point in time in the long run), is denoted by $EBO_i(S_i)$. The *aggregate mean number of backorders*, in steady state, is:

$$EBO(\mathbf{S}) = \sum_{i \in I} EBO_i(S_i). \tag{2.1}$$

The target level for $EBO(\mathbf{S})$ is given by EBO^{obj} and the solution space is:

$$\mathscr{S} = \{\mathbf{S} = (S_1, \ldots, S_{|I|}) \mid S_i \in \mathbb{N}_0, \forall i \in I\}.$$

Hence, in mathematical terms, our optimization problem is as follows:

$$\begin{array}{lll} \text{(P)} & \min & C(\mathbf{S}) \\ & \text{subject to} & EBO(\mathbf{S}) \le EBO^{\mathrm{obj}}, \\ & & \mathbf{S} \in \mathscr{S}. \end{array}$$

The optimal cost of Problem (P) is denoted by C_{P}. Problem (P) has a linear objective function, a nonlinear constraint, and integral decision variables. It thus is a *nonlinear integer programming problem*.

The mean backorder position $EBO_i(S_i)$ denotes the number of parts of SKU i that is missing in all machines of the installed base together. A part is said to be missing in case a failed part has not been replaced yet by a ready-for-use part because there was no ready-for-use spare part available. Similarly, $EBO(\mathbf{S})$ denotes the total number of missing parts in all machines together, and thus is a measures for the inconvenience due to insufficient stock of ready-for-use spare parts. The constraint on the aggregate mean number of backorders is closely related to an *availability constraint*, where the availability $A(\mathbf{S})$ denotes the fraction of machines that is not down due to a missing part, or equivalently, the fraction of time that any given machine is not down due to a missing part. See Sect. 2.7.2 for a detailed description of this relation. In short, if it hardly occurs that any machine has two or more parts missing, then

$$A(\mathbf{S}) \approx 1 - \frac{1}{Z} EBO(\mathbf{S}),$$

where Z denotes the total number of machines, and thus setting a maximum level EBO^{obj} for the aggregate mean number of backorders is equivalent to setting a minimum level $A^{\text{obj}} = 1 - \frac{1}{Z} EBO^{\text{obj}}$ for the availability.

In the description above, we used the terminology that is common for repairable spare parts. Nevertheless, the model is easily generalized to situations where all SKU's, or a subset of SKU's, is consumable, or where *condemnation* for repairable SKU's occurs; see Sect. 2.10.1.

2.2.1 Overview of Assumptions

We summarize and discuss the main assumptions made above:

1. *Demands for the different SKU's occur according to independent Poisson processes.*
   The assumption of independent Poisson processes is justified when a failure of a component does not lead to additional failures of other components in the same machine. In general this is true. The assumption of Poisson processes is justified either when lifetimes of components are exponential or when lifetimes are generally distributed and the number of machines that is served by the warehouse is sufficiently large.
2. *For each SKU, the demand rate is constant.*
   The single warehouse serves multiple machines. When one or more machines fail and failed parts cannot be provided immediately, then some machines may be down for a while and the demand rates for a given SKU decreases accordingly. However, when the fraction of machines that is down is always sufficiently small, either because downtimes are short in general or because downtimes occur only rarely, then the decrease in demand rate is small, and thus it is reasonable to assume a constant demand rate.
3. *Repair leadtimes for different SKU's are independent and repair leadtimes for parts of the same SKU are independent and identically distributed.*
   For repairable SKU's, this assumption is justified if *planned repair leadtimes* have been agreed with repair companies (or departments). It then is the responsibility of the repair company to meet the planned leadtimes. In practice, planned leadtimes often occur either because repair is executed by an external company or in order to decompose the inventory control from the control of the repair facilities.

4. *A one-for-one replenishment strategy is applied for all SKU's.*
   This is justified as long as there are no fixed ordering costs or fixed ordering costs are small relative to the prices of the SKU's (or, thinking of the EOQ rule, relative to price multiplied by demand rate). If fixed ordering costs are relevant, then fixed order quantities may be appropriate to assume and one may follow an (s,Q) instead of a basestock policy for each SKU. This extension is described in Sect. 2.10.3.

Example 2.1. We now describe an illustrative example that is used throughout this chapter. A manufacturer of capital goods keeps spare parts on stock in a single warehouse to support a reasonably large number of installed machines. All spare parts demands are fulfilled from this warehouse. We consider three different SKU's ($|I| = 3$). The average number of failures per year is 15 for SKU 1 ($m_1 = 15$), 5 for SKU 2 ($m_2 = 5$) and 1 for SKU 3 ($m_3 = 1$). The average repair leadtimes are equal to 2 months (or $\frac{1}{6}$ year) for all three SKU's ($t_1 = t_2 = t_3 = \frac{1}{6}$). The price of SKU 1 is 1,000 Euros ($c_1^{\mathrm{a}} = 1{,}000$), the price of SKU 2 is 3,000 Euros ($c_2^{\mathrm{a}} = 3{,}000$) and the price of SKU 3 is 20,000 Euros ($c_3^{\mathrm{a}} = 20{,}000$). It is specified that the aggregate mean number of backorders may not exceed 0.1 ($EBO^{\mathrm{obj}} = 0.1$).

2.3 Evaluation

In this section, we evaluate the steady-state behavior and the aggregate mean number of backorders $EBO(\mathbf{S})$ for a given basestock policy $\mathbf{S}$. Because parts of different SKU's have no interaction, the steady-state behavior can be evaluated per SKU. This leads to a closed-form expression for $EBO_i(S_i)$. $EBO(\mathbf{S})$ itself then follows from (2.1).

Consider an arbitrary SKU i, and assume that the basestock level S_i is given. The repair and fulfilment process of this SKU is depicted by the Petri net in Fig. 2.1. On the left-hand side in this figure, demands for ready-for-use parts, accompanied with failed parts, arrive with rate m_i. The failed parts follow the upper stream in the figure. That is, they first go into repair which takes on average t_i time units. Then they arrive in a queue with ready-for-use parts. Actually this queue represents the physical stock, also called stock on hand. The demands for ready-for-use parts follow the lower stream. That is, these requests are sent to the warehouse, where they are fulfilled immediately if there is enough stock on hand and after some delay otherwise. Delayed requests are fulfilled according to a First-Come, First-Served (FCFS) discipline. When both a request and a ready-for-use part are available, they merge (i.e., the transition on the righthand side in the figure 'fires') and leave the system.

It always holds that at least one of the two queues on the righthand side in the figure is empty. If the stock on hand is positive then there will be no requests waiting for a ready-for-use part. If the number of requests in the queue (= number of backorders) is positive, then there cannot be any part in the queue with on hand stock. The number of backorders is identical to the length of the queue with requests.

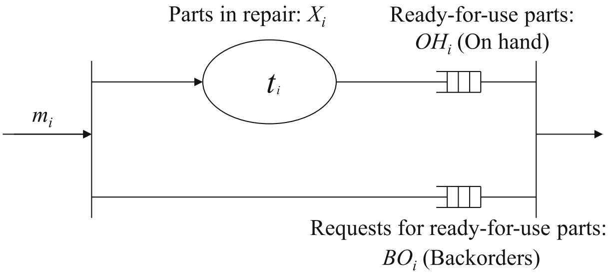

Fig. 2.1 Petri net of the repair and demand fulfilment process of SKU i

The state of the whole system at time instant t may be described by the tuples $(X_i(t), OH_i(t), BO_i(t))$, where $X_i(t)$ denotes the number of parts in repair at time t, $OH_i(t)$ denotes the stock on hand of ready-for-use parts at time t, and $BO_i(t)$ denotes the number of backordered demands at time t. The amount $X_i(t)$ represents the number of parts in the *repair pipeline* and therefore is also called the *(repair) pipeline stock*. Notice that $(X_i(t), OH_i(t), BO_i(t))$ constitutes a partial description because repair leadtimes are generally distributed, and thus for a full description one also has to denote how long parts are in repair already.

The possible values for the tuples $(X_i(t), OH_i(t), BO_i(t))$ are given by:

$$(0, S_i, 0), (1, S_i - 1, 0), \ldots, (S_i - 1, 1, 0), (S_i, 0, 0), (S_i + 1, 0, 1), (S_i + 2, 0, 2), \ldots.$$

The first S_i states in this sequence are with positive stock on hand, the state $(S_i, 0, 0)$ is the unique state where both the stock on hand and the number of backordered demands is zero, and after that the states with a positive number of backordered demands are obtained. A transition is made from one state to the next state in this sequence when a demand occurs, while a completion of a repair leads to a transition from one state to a previous state in this sequence. From the sequence with all possible states, we observe that the values of $OH_i(t)$ and $BO_i(t)$ follow directly from the value of $X_i(t)$. It holds that

$$OH_i(t) = (S_i - X_i(t))^+, \tag{2.2}$$

$$BO_i(t) = (X_i(t) - S_i)^+, \tag{2.3}$$

where $x^+ = \max\{0, x\}$ for any $x \in \mathbb{R}$. These equations imply that

$$OH_i(t) - BO_i(t) = S_i - X_i(t),$$

or, equivalently, that

$$X_i(t) + OH_i(t) - BO_i(t) = S_i.$$

This latter equation is known as the *stock balance equation* (cf. Sherbrooke [18]) and shows that the number of parts in the upper stream of the Petri net in Fig. 2.1 is always S_i more than the number of requests in the lower stream.

Let X_i, OH_i, and BO_i be the steady-state variables corresponding to $X_i(t)$, $OH_i(t)$, and $BO_i(t)$, respectively; i.e., they are random variables denoting the number of parts in repair, the number of ready-for-use parts, and the number of backordered demands in steady state. By (2.2) and (2.3),

$$OH_i = (S_i - X_i)^+, \tag{2.4}$$

$$BO_i = (X_i - S_i)^+. \tag{2.5}$$

In our model failed parts enter the repair pipeline according to a Poisson process and each failed part stays on average a time t_i in the repair pipeline. The repair pipeline may be seen as a queueing system with infinitely many servers and service times t_i. In other words, the repair pipeline is an $M|G|\infty$ queueing system and thus we may apply Palm's theorem (cf. Palm [13]):

Palm's theorem: *If jobs arrive according to a Poisson process with rate λ at a service system and if the times that the jobs remain in the service system are independent and identically distributed according to a given general distribution with mean EW, then the steady-state distribution for the total number of jobs in the service system is Poisson with mean λEW.*

Application of this theorem to the repair pipeline leads to part (i) of the following lemma; the parts (ii) and (iii) of this lemma follow from part (i) and Eqs. (2.4) and (2.5).

Lemma 2.1. *Let $i \in I$.*

(i) The pipeline stock X_i is Poisson distributed with mean $m_i t_i$, i.e.,

$$P\{X_i = x\} = \frac{(m_i t_i)^x}{x!} e^{-m_i t_i}, \qquad x \in \mathbb{N}_0;$$

(ii) The distribution of the stock on hand OH_i is given by

$$P\{OH_i = x\} = \begin{cases} \sum_{y=S_i}^{\infty} P\{X_i = y\} & \text{if } x = 0; \\ P\{X_i = S_i - x\} & \text{if } x \in \mathbb{N},\ x \le S_i; \end{cases}$$

(iii) The distribution of the number of backordered demands BO_i is given by

$$P\{BO_i = x\} = \begin{cases} \sum_{y=0}^{S_i} P\{X_i = y\} & \text{if } x = 0; \\ P\{X_i = x + S_i\} & \text{if } x \in \mathbb{N}. \end{cases}$$

Lemma 2.1 contains the main results for the evaluation of a given policy. From this lemma, we easily obtain relevant service measures, among which the mean backorder positions $EBO_i(S_i)$:

$$\begin{aligned} EBO_i(S_i) = E\{BO_i(S_i)\} &= \sum_{x=S_i+1}^{\infty} (x - S_i) P\{X_i = x\} \\ &= m_i t_i - S_i + \sum_{x=0}^{S_i} (S_i - x) P\{X_i = x\}, \qquad S_i \in \mathbb{N}_0. \end{aligned} \tag{2.6}$$

Notice that the latter expression for $EBO_i(S_i)$ is most appropriate for computational purposes as it avoids complications because of sums with infinitely many terms.

2.4 Optimization

Instead of solving Problem (P) directly, we consider a closely related Problem (Q) with two objectives, minimization of the investment $C(\mathbf{S})$ and minimization of the aggregate mean number of backorders $EBO(\mathbf{S})$:

$$\begin{array}{lll} \text{(Q)} & \min & C(\mathbf{S}) \\ & \min & EBO(\mathbf{S}) \\ & \multicolumn{2}{l}{\text{subject to } \mathbf{S} \in \mathscr{S}.} \end{array}$$

This problem is a *multi-objective programming problem*. For this problem, we will derive so-called *efficient solutions*. A solution $\mathbf{S} \in \mathscr{S}$ is efficient for Problem (Q) if and only if there is no other solution $\mathbf{S}^{'} \in \mathscr{S}$ with $C(\mathbf{S}') \leq C(\mathbf{S})$ and $EBO(\mathbf{S}^{'}) \leq EBO(\mathbf{S})$, and strict inequality for at least one of these inequalities. Alternatively stated, a solution $\mathbf{S} \in \mathscr{S}$ is efficient for Problem (Q) if and only if $C(\mathbf{S}^{'}) > C(\mathbf{S})$, or $EBO(\mathbf{S}^{'}) > EBO(\mathbf{S})$, or $(C(\mathbf{S}^{'}), EBO(\mathbf{S}^{'})) = (C(\mathbf{S}), EBO(\mathbf{S}))$ for all $\mathbf{S}^{'} \in \mathscr{S}$. Let $\mathscr{E}^*$ denote the set of all efficient solutions for Problem (Q). Then the points $(C(\mathbf{S}), EBO(\mathbf{S}))$, $\mathbf{S} \in \mathscr{E}^*$, constitute an *efficient frontier* for the total inventory investment vs. aggregate mean number of backorders. From this efficient frontier, an appropriate solution for Problem (P) may be picked.

Example 2.1 (continued). For our illustrative example, we compute the mean number of backorders and inventory investment for all plausible solutions with an investment of at most 85,000 Euros. These solutions are plotted in an $C(S)$ vs. $EBO(S)$ figure; see Fig. 2.2. We are interested in the efficient solutions for problem (Q), which are denoted by squares. From this figure, we easily obtain an optimal solution of Problem (P) with $EBO^{\text{obj}} = 0.1$. That solution is the first efficient solution in Fig. 2.2 with $EBO(\mathbf{S}) \leq 0.1$. This leads to the solution $\mathbf{S} = (6,2,1)$, for which $EBO(\mathbf{S}) = 0.098$ and $C(\mathbf{S}) =$ 32,000 Euros.

Problem (Q) has the following structure:

$$\begin{aligned} C(\mathbf{S}) &= \sum_{i \in I} C_i(S_i), \\ EBO(\mathbf{S}) &= \sum_{i \in I} EBO_i(S_i), \\ \mathscr{S} &= \mathscr{S}_1 \times \mathscr{S}_2 \times \ldots \times \mathscr{S}_{|I|}, \end{aligned}$$

where $\mathscr{S}_i = \mathbb{N}_0$ represents the solution space for S_i for all $i \in I$, i.e., the objective functions are separable and the solutions space is a Cartesian Product, and thus Problem (Q) as a whole is *separable* (cf. Fox [8]). In addition, the functions $C_i(S_i) = c_i^{\text{a}} S_i$, $i \in I$, are linear, and, as we shall derive in Sect. 2.4.1, the functions $EBO_i(S_i)$, $i \in I$, are decreasing and convex. Therefore, a greedy procedure may be applied to generate efficient solutions; see Sect. 2.4.2.

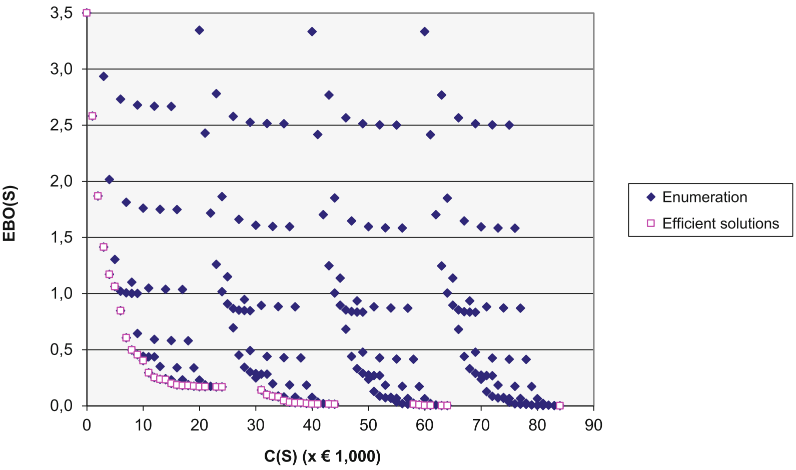

Fig. 2.2 Efficient solutions for Example 2.1

2.4.1 Convexity of the Mean Backorder Positions

Definition 2.1. Let $f(x)$ be a function on $\mathbb{Z}$, and $x_0 \in \mathbb{Z}$.

(i) $f(x)$ is decreasing for $x \geq x_0$ if

$$\Delta f(x) = f(x+1) - f(x) \leq 0, \qquad x \geq x_0;$$

(ii) $f(x)$ is convex for $x \geq x_0$ if

$$\Delta^2 f(x) = \Delta f(x+1) - \Delta f(x) \geq 0, \qquad x \geq x_0.$$

Notice that $\Delta f(x+1) - \Delta f(x) = f(x+2) - 2f(x+1) + f(x)$, $x \in \mathbb{Z}$. The definitions for strictly decreasing and strictly convex are obtained by replacing the inequality signs by strict inequality signs. The definitions for (strictly) increasing and (strictly) concave are obtained by turning the (strict) inequality signs around.

The mean number of backorders $EBO_i(S_i)$ for SKU $i \in I$ is a function on $\mathbb{N}_0$. Lemma 2.2 says that $EBO_i(S_i)$ is decreasing and convex on its whole domain.

Lemma 2.2. *For each SKU $i \in I$, $EBO_i(S_i)$ is decreasing and convex for $S_i \in \mathbb{N}_0$.*

Proof. Let $i \in I$. By (2.6),

$$\begin{aligned} \Delta EBO_i(S_i) &= EBO_i(S_i+1) - EBO_i(S_i) \\ &= -\sum_{x=S_i+1}^{\infty} P\{X_i = x\} \leq 0, \qquad S_i \in \mathbb{N}_0, \end{aligned} \tag{2.7}$$

which shows that $EBO_i(S_i)$ is decreasing on its whole domain. Further,

$$\begin{aligned}\Delta^2 EBO_i(S_i) &= \Delta EBO_i(S_i+1) - \Delta EBO_i(S_i)\\ &= P\{X_i = S_i+1\} \geq 0, \qquad S_i \in \mathbb{N}_0,\end{aligned}$$

which shows that $EBO_i(S_i)$ is convex on its whole domain. □

2.4.2 *Greedy Algorithm*

Problem (Q) is separable and the functions $EBO_i(S_i)$ are decreasing and convex on their whole domains. Hence we can prove that a set of efficient solutions can be generated by a greedy algorithm. A first efficient solution $\mathbf{S} = (S_1, \ldots, S_{|I|})$ is obtained by setting $S_i = 0$ for each SKU $i \in I$. This solution is efficient because it has the lowest possible investment $C(\mathbf{S}) = 0$. Next, for each SKU i, we compute the decrease in $EBO(\mathbf{S})$ relative to the increase in $C(\mathbf{S})$ when S_i would be increased by one unit. The increase in $C(\mathbf{S})$ equals c_i^{a}, while the change in $EBO(\mathbf{S})$ equals (use (2.7))

$$\Delta_i EBO(\mathbf{S}) = \Delta EBO_i(S_i) = -\sum_{x=S_i+1}^{\infty} P\{X_i = x\} = -\left(1 - \sum_{x=0}^{S_i} P\{X_i = x\}\right).$$

The decrease in $EBO(\mathbf{S})$, which is equal to $-\Delta_i EBO(\mathbf{S})$, divided by the increase in $C(\mathbf{S})$ is denoted by Γ_i. The SKU with the highest value for Γ_i is selected (also referred to as “biggest bang for the buck”), and the corresponding basestock level is increased by one unit (ties may be broken with equal probabilities). The new solution $\mathbf{S}$ is also efficient and is added to a set of efficient solutions. The generation of efficient solutions is continued until a given aggregate mean number of backorders or inventory investment has been reached, or until some other stop criterium is met. The formal procedure is described in Algorithm 2.1, where $\mathbf{e}_k$ is an $|I|$-dimensional unit row-vector.

Algorithm 2.1 *(Greedy algorithm)*

Step 1 Set $S_i := 0$ for all $i \in I$, and $\mathbf{S} = (0, 0, \ldots, 0)$;
$\mathscr{E} := \{\mathbf{S}\}$;
$C(\mathbf{S}) := 0$ and $EBO(\mathbf{S}) := \sum_{i \in I} m_i t_i$.
Step 2 $\Gamma_i := (1 - \sum_{x=0}^{S_i} P\{X_i = x\})/c_i^{\mathrm{a}}$ for all $i \in I$;
$k := \arg\max\{\Gamma_i : i \in I\}$;
$\mathbf{S} := \mathbf{S} + \mathbf{e}_k$;
$\mathscr{E} := \mathscr{E} \cup \{\mathbf{S}\}$.
Step 3 Compute $C(\mathbf{S})$ and $EBO(\mathbf{S})$;
If ‘stop criterium’, then stop, else goto Step 2.

In the following lemma, it is formally stated that Algorithm 2.1 generates a set of efficient solutions for Problem (Q). The proof of this lemma follows directly from Theorem 2 in Fox [8].

Lemma 2.3. *At termination of Algorithm 2.1, the set $\mathscr{E}$ consists of efficient solutions for Problem (Q), i.e., $\mathscr{E} \subset \mathscr{E}^*$.*

For the computation of the pipeline stock probabilities $P\{X_i = x\}$ in Algorithm 2.1, we advice to use the following recursion for the sake of efficiency and to avoid numerical problems:

$$\begin{aligned} P\{X_i = 0\} &= e^{-m_i t_i}, \\ P\{X_i = x+1\} &= \frac{m_i t_i}{x+1} P\{X_i = x\} \text{ for } x \in \mathbb{N}_0. \end{aligned} \tag{2.8}$$

Example 2.1 (continued). Application of the greedy algorithm to our example with stop criterium '$EBO(\mathbf{S}) \leq 0.1$' leads to the efficient solutions displayed in Table 2.1. After 11 iterations we obtain the first efficient solution that satisfies $EBO(\mathbf{S}) \leq 0.1$. This solution is $\mathbf{S} = (7,3,1)$, for which $EBO(\mathbf{S}) = 0.031$ and $C(\mathbf{S}) = 36{,}000$ Euros. This solution is optimal for Problem (P) with $EBO^{\text{obj}} = 0.031$. Further, this solution is feasible for Problem (P) with $EBO^{\text{obj}} = 0.1$, but apparently not optimal. Earlier, we found that $\mathbf{S} = (6,2,1)$ is optimal for Problem (P) with $EBO^{\text{obj}} = 0.1$. Notice that the gap in costs between $\mathbf{S} = (7,3,1)$ and the optimal solution is equal to $\frac{4{,}000}{32{,}000} = 12.5\,\%$. The efficient solution $\mathbf{S} = (6,2,1)$ is not generated by the greedy algorithm. In general, the greedy algorithm generates only a subset of all efficient solutions. This follows clearly from Fig. 2.3, where both the efficient solutions from the enumeration and the efficient solutions from the greedy algorithm are displayed.

In general, the greedy algorithm generates an ordered set $\mathscr{E} = \{\mathbf{S}^0, \mathbf{S}^1, \mathbf{S}^2, \ldots\}$ of efficient solutions for Problem (Q), where $EBO(\mathbf{S}^0) > EBO(\mathbf{S}^1) > EBO(\mathbf{S}^2) > \ldots$ and $0 = C(\mathbf{S}^0) < C(\mathbf{S}^1) < C(\mathbf{S}^2) < \ldots$. The set $\mathscr{E}$ is a subset of the set $\mathscr{E}^*$ with all efficient solutions. For Problem (P) with a given target EBO^{obj}, one can easily obtain a feasible solution from the subset $\mathscr{E}$ generated by the greedy algorithm. One just takes the first solution $\mathbf{S}^l \in \mathscr{E}$ with $EBO(\mathbf{S}^l) \leq EBO^{\text{obj}}$. This solution is optimal if and only if there is no solution $\mathbf{S} \in \mathscr{E}^*$ with $EBO(\mathbf{S}^l) < EBO(\mathbf{S}) \leq EBO^{\text{obj}}$. In general, the solution $\mathbf{S}^l$ will be close to optimal if $EBO(\mathbf{S}^l)$ is close to EBO^{obj}. In the above example, we were a bit unlucky, because in the 11-th iteration the basestock level of SKU 3 was increased, which led to a large jump for the aggregate mean number of backorders. For real-life problems, one often has many SKU's and then such large jumps become less likely. See also the following example.

Example 2.2. The data in this example are taken from a real life situation of a repair shop which has 99 SKU's in stock. The prices of the SKU's range from 135 Euros to 61,828 Euros and the average price is 2,236 Euros. The number of failures ranges from 1 to 18 per year and the average number of failures is 2.55 per year. Figure 2.4 gives an overview of the prices and failures rates (per year) of the SKU's. The repair

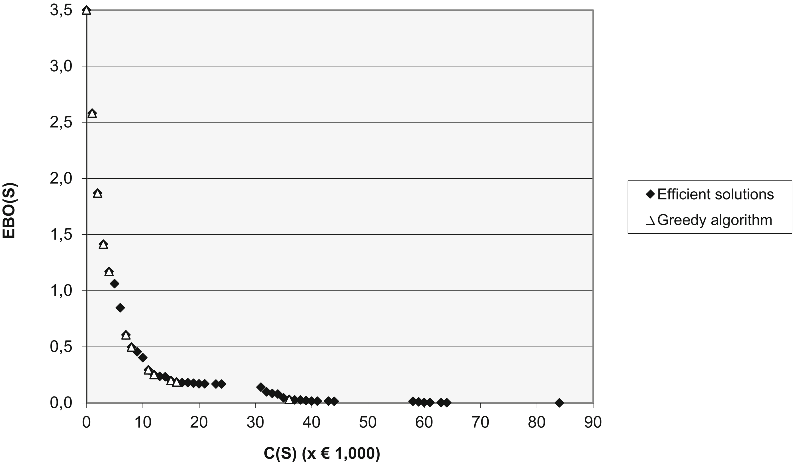

Fig. 2.3 Comparison of efficient solutions generated by the greedy algorithm and the whole set of efficient solutions

Table 2.1 Steps of the greedy algorithm for Example 2.1

Iteration	Γ_1	Γ_2	Γ_3	k	S_1	S_2	S_3	$EBO(\mathbf{S})$	$C(\mathbf{S})$ (Euros)
0	–	–	–	–	0	0	0	3.500	0
1	$9.18 \cdot 10^{-4}$	$1.88 \cdot 10^{-4}$	$7.68 \cdot 10^{-6}$	1	1	0	0	2.582	1,000
2	$7.13 \cdot 10^{-4}$	$1.88 \cdot 10^{-4}$	$7.68 \cdot 10^{-6}$	1	2	0	0	1.869	2,000
3	$4.56 \cdot 10^{-4}$	$1.88 \cdot 10^{-4}$	$7.68 \cdot 10^{-6}$	1	3	0	0	1.413	3,000
4	$2.42 \cdot 10^{-4}$	$1.88 \cdot 10^{-4}$	$7.68 \cdot 10^{-6}$	1	4	0	0	1.171	4,000
5	$1.09 \cdot 10^{-4}$	$1.88 \cdot 10^{-4}$	$7.68 \cdot 10^{-6}$	2	4	1	0	0.605	7,000
6	$1.09 \cdot 10^{-4}$	$6.77 \cdot 10^{-5}$	$7.68 \cdot 10^{-6}$	1	5	1	0	0.497	8,000
7	$4.20 \cdot 10^{-5}$	$6.77 \cdot 10^{-5}$	$7.68 \cdot 10^{-6}$	2	5	2	0	0.293	11,000
8	$4.20 \cdot 10^{-5}$	$1.74 \cdot 10^{-5}$	$7.68 \cdot 10^{-6}$	1	6	2	0	0.251	12,000
9	$1.42 \cdot 10^{-5}$	$1.74 \cdot 10^{-5}$	$7.68 \cdot 10^{-6}$	2	6	3	0	0.199	15,000
10	$1.42 \cdot 10^{-5}$	$3.47 \cdot 10^{-6}$	$7.68 \cdot 10^{-6}$	1	7	3	0	0.185	16,000
11	$4.25 \cdot 10^{-6}$	$3.47 \cdot 10^{-6}$	$7.68 \cdot 10^{-6}$	3	7	3	1	0.031	36,000

lead time is 4 months ($\frac{1}{3}$ year) for all SKU's. We are interested in the solution of Problem (P) with a target $EBO^{\text{obj}} = 3.3$ (which is comparable to the target in Example 2.1).

We applied the greedy algorithm to this data set, which led to a feasible solution $\mathbf{S}$ after 239 iterations, with $EBO(\mathbf{S}) = 3.02$ and $C(\mathbf{S}) = 277{,}749$ Euros. In Fig. 2.5, the solutions from the greedy algorithm are displayed. In this figure we see that the iterations of the greedy algorithm produce a very smooth curve. Towards the end we see somewhat bigger gaps in the line. The cost of the last but one solution was 266,593 Euros, and constitutes a lower bound for the optimal cost. Hence, we know that the generated heuristic solution is within $\frac{277{,}749-266{,}593}{266{,}593} = \frac{11{,}156}{266{,}593} = 4.2\,\%$ of

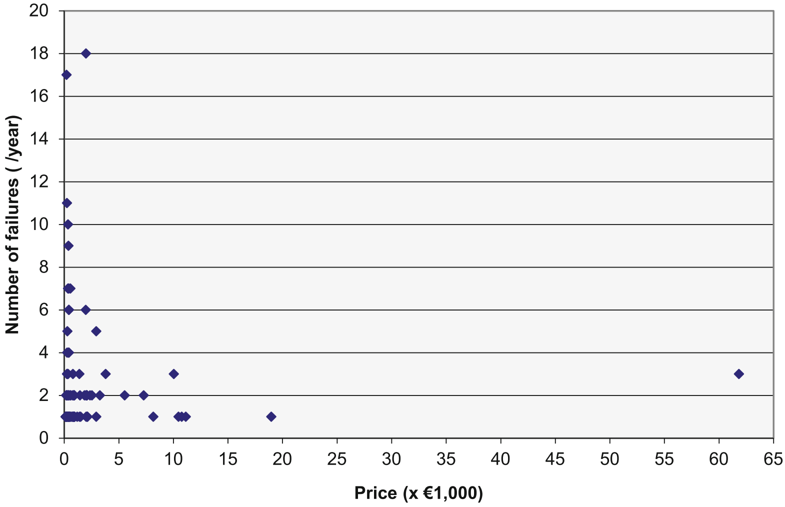

Fig. 2.4 Representation of all SKU's

the optimal solution (this is a bound, the actual gap is smaller). For applications in practice, this is sufficiently good.

We can conclude that for instances with sufficiently many SKU's, the greedy algorithms will generate good heuristic solutions for Problem (P). Besides, a greedy algorithm is efficient, it is easy to implement in practice and it is an algorithm that is easy to understand by practitioners. If one would be interested in optimal solutions, then Problem (P) may be solved by a similar exact approach as for knapsack problems. A disadvantage of such an approach is that a small change in input parameters (cost prices of the SKU's, demand rates, or the target EBO^{obj}) may lead to large changes in the optimal solution. The solution generated by the greedy algorithm, however, will be rather robust. That is another big advantage of the greedy algorithm.

2.5 Alternative Optimization Techniques

We can also approach the Problems (P) and (Q) with other techniques than the greedy algorithm of Sect. 2.4.2. In this section we will describe two alternative optimization techniques, Lagrangian relaxation is described in Sect. 2.5.1 and the Dantzig-Wolfe decomposition is described in Sect. 2.5.2.

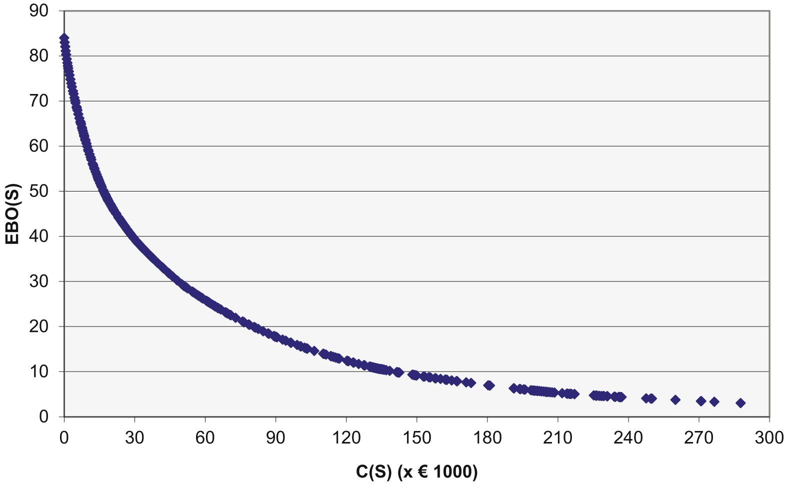

Fig. 2.5 Outcome of the greedy algorithm

2.5.1 Lagrangian Relaxation

We apply the Lagrangian relaxation technique to Problem (P); for a general description of this technique, we refer to Appendix B of Porteus [14] (other well-known references on Lagrangian relaxation are Everett [4] and Fisher [6, 7]). The *Lagrangian function* for (P) is defined as

$$L(\mathbf{S},\lambda) := \sum_{i\in I} c_i^{\mathrm{a}} S_i + \lambda\left(\sum_{i\in I} EBO_i(S_i) - EBO^{\mathrm{obj}}\right)$$

where $\lambda \geq 0$ is a *Lagrange multiplier*.

It has been noticed before that Problem (Q) is separable. This also holds for Problem (P) (see also the definition of separable problems in [14], Appendix B). We can separate Problem (P) because it is a linear combination of SKU objectives and constraints. It is known that in separable problems, the Lagrangian function is also separable. The Lagrangian function is now defined as

$$L(\mathbf{S},\lambda) = \sum_{i\in I} L_i(S_i,\lambda) - \lambda EBO^{\mathrm{obj}}, \tag{2.9}$$

where

$$L_i(S_i,\lambda) := c_i^{\mathrm{a}} S_i + \lambda EBO_i(S_i)$$

is the *decentralized Lagrangian function* for SKU i. Notice that, in Eq. (2.9), we have $|I|$ different Lagrangian functions, one for every SKU. Notice that we have only one λ because we have only one constraint in our problem.

For any given value of λ, we can easily find a base stock level that minimizes the decentralized Lagrangian function $L_i(S_i, \lambda)$, $i \in I$. Since each decentralized Lagrangian function is a convex functions, we know that it has either one unique minimum or multiple minima in subsequent points. One way to find a minimum is to start with $S_i = 0$ and increase this S_i by 1 at a time, until the value for the decentralized Lagrangian function start increasing. We can do this for all $i \in I$, and the resulting base stock vector $\mathbf{S}$ is a feasible solution for problem (P) (under any choice for EBO^{obj}). We can now vary the value of λ to find different solutions for Problem (P). Then, we calculate the corresponding values of $EBO(\mathbf{S})$ and $C(\mathbf{S})$.

Example 2.1 (continued). Applying Lagrangian relaxation to our problem, gives the following decentralized Lagrangian functions:

$$L_i(S_i, \lambda) = \begin{cases} 1{,}000 \cdot S_1 + \lambda EBO_1(S_1) & \text{if } i = 1 \\ 3{,}000 \cdot S_2 + \lambda EBO_1(S_2) & \text{if } i = 2 \\ 20{,}000 \cdot S_3 + \lambda EBO_1(S_3) & \text{if } i = 3. \end{cases}$$

If we vary λ from 0 to 300,000, we find the solutions that are displayed in Table 2.2. We can see that Lagrangian relaxation yields exactly the same solutions as our greedy algorithm (compare Tables 2.1 and 2.2). In each row of Table 2.2, a range of values for λ is given for which that specific base stock vector minimizes the Lagrangian function.

In this example, we observe a few important properties. The first property is that using the Lagrangian relaxation method gives us optimal solutions of Problem (P) for specific values of EBO^{obj}. This follows from the so-called Everett result (Everett [4]), which for our problem reads as follows:

The Everett result: *If, for a given $\lambda \geq 0$, $\mathbf{S}(\lambda)$ minimizes $L(\mathbf{S}, \lambda)$ over $\mathbf{S} \in \mathscr{S}$, then $\mathbf{S}(\lambda)$ is optimal for Problem (P) for every $EBO^{\text{obj}} \in (0, \infty)$ that satisfies*

$$EBO^{\text{obj}} \geq EBO(\mathbf{S}(\lambda)) \text{ and } \lambda\,(EBO(\mathbf{S}(\lambda)) - EBO^{\text{obj}}) = 0.$$

If we take $\lambda = 0$, then each Lagrangian function $L_i(S_i, \lambda)$ is strictly increasing, and we find $\mathbf{S}(0) = (0, \ldots, 0)$ and $EBO(\mathbf{S}(0)) = \sum_{i \in I} m_i t_i$. The solution $\mathbf{S}(0) = (0, \ldots, 0)$ is optimal for Problem (P) for every $EBO^{\text{obj}} \geq \sum_{i \in I} m_i t_i$. For each $\lambda > 0$, the solution $\mathbf{S}(\lambda)$ is optimal for Problem (P) for $EBO^{\text{obj}} = EBO(\mathbf{S}(\lambda))$; i.e., then the optimality of $\mathbf{S}(\lambda)$ is guaranteed for one specific value of EBO^{obj} (but the solution might also be optimal for slightly lower values of EBO^{obj}).

The second property that we see back in the above example is that the Lagrangian relaxation method gives efficient solutions for Problem (Q). This follows directly from Theorem 1 in Fox [8].

A third property that we observe is that Lagrangian relaxation yields exactly the same solutions as the greedy algorithm. This is not a coincidence. When we study the details of the execution of the greedy algorithm and the execution of the Lagrangian relaxation method, we see the similarities. The key is that a one-to-one

Table 2.2 Solutions generated with Lagrangian relaxation

λ	S_1	S_2	S_3	$EBO_1(S_1)$	$EBO_2(S_2)$	$EBO_3(S_3)$	$EBO(\mathbf{S})$	$C(\mathbf{S})$ (Euros)
$\in [0,1089)$	0	0	0	2.500	0.833	0.167	3.500	0
$\in [1{,}089, 1{,}403)$	1	0	0	1.582	0.833	0.167	2.582	1,000
$\in [1{,}403, 2{,}192)$	2	0	0	0.869	0.833	0.167	1.869	2,000
$\in [2{,}192, 4{,}125)$	3	0	0	0.413	0.833	0.167	1.413	3,000
$\in [4{,}125, 5{,}306)$	4	0	0	0.171	0.833	0.167	1.171	4,000
$\in [5{,}306, 9{,}189)$	4	1	0	0.171	0.268	0.167	0.605	7,000
$\in [9{,}189, 14{,}761)$	5	1	0	0.062	0.268	0.167	0.497	8,000
$\in [14{,}761, 23{,}798)$	5	2	0	0.062	0.065	0.167	0.293	11,000
$\in [23{,}798, 57{,}324)$	6	2	0	0.020	0.065	0.167	0.251	12,000
$\in [57{,}324, 70{,}486)$	6	3	0	0.020	0.012	0.167	0.199	15,000
$\in [70{,}486, 130{,}278)$	7	3	0	0.006	0.012	0.167	0.185	16,000
$\in [130{,}278, 287{,}985)$	7	3	1	0.006	0.012	0.013	0.031	36,000

relationship exists between the Γ_i values computed in the greedy algorithm and the values of λ for which $\mathbf{S}(\lambda)$ changes to the next solution in the Lagrangian relaxation method.

2.5.2 Dantzig-Wolfe Decomposition

A method that one may also apply to solve Problem (P) is Dantzig-Wolfe decomposition, as introduced by Dantzig and Wolfe [3]. In order to be able to apply Dantzig-Wolfe decomposition to Problem (P), we need to redefine it. A *Master Problem* is introduced in which all possible basestock levels per SKU are listed as columns. A constraint is added to ensure that only one basestock level is chosen per SKU. Set $K := \mathbb{N}_0$ contains all possible basestock levels for all SKU's i, $i \in I$. Let S_i^k, $i \in I$, $k \in K$, be a variable referring to a fixed policy k for SKU i. We introduce a new variable $x_i^k \in \{0,1\}$, $i \in I$, $k \in K$, that indicates whether a specific policy is selected for item i or not. For example, if $x_i^3 = 1$ then policy 3 is selected for SKU i, this is equivalent with $S_i = 2$. By relaxing the integrality constraint on x_i^k, we allow for randomized policies. The Master Problem (MP) related to Problem (P) is now defined as follows:

$$
\begin{aligned}
\text{(MP)} \qquad & \min \sum_{i \in I} \sum_{k \in K} C_i(S_i^k) x_i^k \\
\text{subject to } & \sum_{i \in I} \sum_{k \in K} EBO_i(S_i^k) x_i^k \le EBO^{\text{obj}} \\
& \sum_{k \in K} x_i^k = 1, && i \in I \\
& x_i^k \ge 0, && i \in I,\ k \in K.
\end{aligned}
$$

Note that the costs $C_i(S_i^k)$ and average backorder $(EBO_i(S_i^k))$ are pre-determined for all basestock levels S_i^k and that we use these numbers as input for Problem (MP). This Master Problem can have many input variables, or possible policies. The size of the problem may restrict fast calculation of the solution. Therefore, a *Restricted Master Problem* (RMP) is introduced that considers only a subset of all possible policies ($K_i \subseteq K$) per SKU i and therefore requires less computational effort:

$$\text{(RMP)} \qquad \min \sum_{i\in I}\sum_{k\in K_i} C_i(S_i^k)x_i^k$$

$$\text{subject to} \sum_{i\in I}\sum_{k\in K_i} EBO_i(S_i^k)x_i^k \le EBO^{\text{obj}}$$

$$\sum_{k\in K_i} x_i^k = 1, \qquad i \in I$$

$$x_i^k \ge 0, \qquad i \in I,\ k \in K_i.$$

Initially, the Problem (RMP) will be solved with one policy per SKU ($|K_i| = 1$). Thereafter, we will employ a procedure called column generation to add policies, or columns, to the policy set K_i. We define an initial solution that is feasible but not optimal. Let K_i initially consist of one policy k that is defined by $S_i^k := \min\{S_i | EBO_i(S_i) \le \frac{m_i}{M} EBO^{\text{obj}}, S_i \in \mathbb{N}_0\}$.

From the optimal solution of the (RMP) we also obtain the dual prices of all constraints. Let u be the dual price that corresponds to the EBO constraint and let v_i be the dual price that corresponds to the constraint that ensures that only one policy is chosen for SKU i. Dual price u needs to be nonpositive and v_i is unrestricted in sign.

In the column generation *subproblem* of SKU i, we search for policies that have not yet been considered and improve our solution for (RMP), and these policies are then added to K_i. To obtain these policies we need to solve a subproblem for each SKU i. Subproblem i is defined as follows:

$$(\text{SUB}(i)) \qquad \min C_i(S_i) - EBO_i(S_i)u - v_i$$

$$\text{subjext to } S_i \in \mathbb{N}_0.$$

We are only interested in a basestock level that minimizes the subproblem and that has a so-called negative reduced cost coefficient (i.e., under this basestock level, the objective function of (SUB(i)) has a negative value). We know that the solution of (SUB(i)) is straightforward because its objective function is the sum of a convex term $(-EBO_i(S_i)u)$, a linear term $(C_i(S_i))$ and a constant term $(-v_i)$. After solving the subproblem (SUB(i)), we add the obtained optimal policy to the set K_i if it has a negative reduced cost. We do this for all subproblems (SUB(i)), $i \in I$. We then solve the (RMP) again for the extended set of policies and use the new dual prices as input for the subproblems (SUB(i)). We continue this until all subproblems (SUB(i)) have a nonnegative reduced cost coefficient.

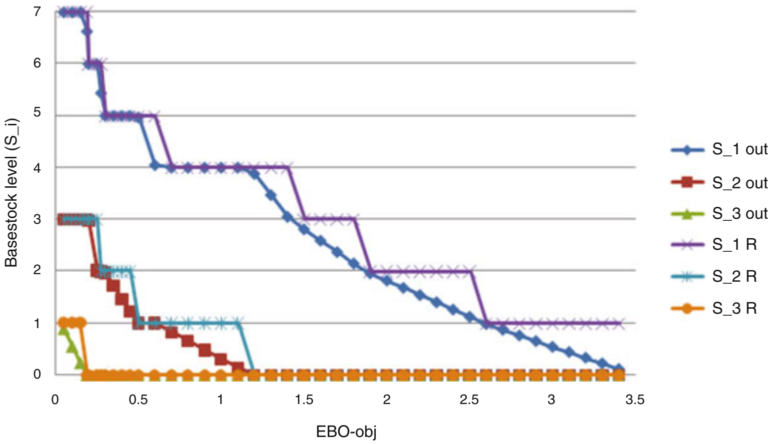

Fig. 2.6 Basestock vectors generated with Dantzig-Wolfe decomposition

One final step in the Dantzig-Wolfe method is to determine the basestock vector that corresponds to the optimal solution for Problem (P). If the solution that we obtain from the (RMP) consists of some fractional values, we need to round up these basestock levels to the nearest integer. It appears that at most one SKU has a fractional base stock value.

Example 2.1 (continued). We applied Dantzig-Wolfe decomposition to the example. For different value of EBO^{obj} we determined the optimal basestock level. As said before, rounding up was used to obtain integer values for basestock levels. The results of this analysis are graphically displayed in Fig. 2.6. For varying levels of EBO^{obj}, we have depicted the value of the basestock levels for all SKU's (indicated by "S_i out"), and the rounded values (indicated by "S_i R"). (A basestock value of 2.5 of SKU i corresponds to $x_i^2 = x_i^3 = 0.5$.) From this figure, we see that the fractional basestock levels decrease almost linearly in the target EBO between two integer values.

The results are aggregated in Table 2.3. If we compare Tables 2.3 and 2.2, we see that the solutions we obtain are equal. Apparently for this problem, the two optimization methods yield exactly the same solutions. This fact also implies that with Dantzig-Wolfe decomposition we obtain the same results as with the Greedy Algorithm. That we obtain the same results for Dantzig-Wolfe decomposition and Lagrangian relaxation is due to the strong similarities between these two methods; see also [10] (Sect. 1.4.2).

Table 2.3 Solutions generated with Dantzig-Wolfe decomposition

EBO^{obj}	S_1	S_2	S_3	$EBO_1(S_1)$	$EBO_2(S_2)$	$EBO_3(S_3)$	$EBO(\mathbf{S})$	$C(\mathbf{S})$ (Euros)
≥ 3.500	0	0	0	2.500	0.833	0.167	3.500	0
$\in [3.500, 2.582)$	1	0	0	1.582	0.833	0.167	2.582	1,000
$\in [2.582, 1.869)$	2	0	0	0.869	0.833	0.167	1.869	2,000
$\in [1.869, 1.413)$	3	0	0	0.413	0.833	0.167	1.413	3,000
$\in [1.413, 1.171)$	4	0	0	0.171	0.833	0.167	1.171	4,000
$\in [1.171, 0.605)$	4	1	0	0.171	0.268	0.167	0.605	7,000
$\in [0.605, 0.497)$	5	1	0	0.062	0.268	0.167	0.497	8,000
$\in [0.497, 0.293)$	5	2	0	0.062	0.065	0.167	0.293	11,000
$\in [0.293, 0.251)$	6	2	0	0.020	0.065	0.167	0.251	12,000
$\in [0.251, 0.199)$	6	3	0	0.020	0.012	0.167	0.199	15,000
$\in [0.199, 0.185)$	7	3	0	0.006	0.012	0.167	0.185	16,000
$\in [0.185, 0.031)$	7	3	1	0.006	0.012	0.013	0.031	36,000

2.6 Item Approach

So far, we treated all SKU's in one model and thus we followed a so-called system approach (cf. Sect. 1.3). A straightforward way to get a feasible solution for Problem (P) is to decompose the constraint for the aggregate mean number of backorders into constraints per SKU. One then gets a simple decision problem per SKU. This simplified approach is a so-called *item approach* and is obviously suboptimal. In this section, we describe the item approach and we compare its solution for Problem (P) to the solutions found by the methods of the previous sections.

The item approach consists of the following steps:

1. Set a target EBO^{obj} for the aggregate mean number of backorders;
2. Divide this target over the SKU's based on the demand rates; i.e., set a target EBO_i^{obj} for the mean number of backorders of SKU i via the following formula: $EBO_i^{obj} = (m_i/M)EBO^{obj}$;
3. For each SKU i: determine S_i such that the target EBO_i^{obj} is reached against minimal costs; i.e., determine S_i as the smallest S_i for which $EBO_i(S_i) \leq EBO_i^{obj}$;
4. Determine $EBO(\mathbf{S})$ and $C(\mathbf{S})$.
   Because all S_i are integer valued, the actual $EBO_i(S_i)$ can be significantly smaller than the corresponding target EBO_i^{obj} for a SKU i, and thus the actual $EBO_i(\mathbf{S})$ can be quite a bit smaller than the aggregate target EBO^{obj}. By varying EBO^{obj}, we obtain an $EBO(\mathbf{S})$ versus $C(\mathbf{S})$ curve, as under the approaches of Sects. 2.4 and 2.5. Via this curve, a heuristic solution is obtained for Problem (P) with a given target level EBO^{obj}.

Example 2.1 (continued). We apply the item approach to our problem for varying levels of the target EBO^{obj}, which leads to the solutions displayed in Table 2.4 and Fig. 2.7. The generated solutions are different from the solutions that we obtained before. For $EBO^{obj} = 0.1$, the item approach leads to the solution $(5, 3, 2)$ and a

Table 2.4 Inventory investment for different values of target item mean backorder

EBO^{obj}	EBO_1^{obj}	EBO_2^{obj}	EBO_3^{obj}	S_1	S_2	S_3	$EBO(\mathbf{S})$	$C(\mathbf{S})$ (Euros)
0.03	0.02	0.01	0.00	6	4	2	0.02	58,000
0.08	0.05	0.02	0.00	6	3	2	0.03	55,000
0.15	0.11	0.04	0.01	5	3	2	0.09	54,000
0.20	0.14	0.05	0.01	5	3	2	0.09	54,000
0.30	0.21	0.07	0.01	4	2	1	0.25	30,000
0.60	0.43	0.14	0.03	3	2	1	0.49	29,000
0.90	0.64	0.21	0.04	3	2	1	0.49	29,000
1.20	0.86	0.29	0.06	3	1	1	0.69	26,000
1.50	1.07	0.36	0.07	2	1	1	1.15	25,000
1.80	1.29	0.43	0.09	2	1	1	1.15	25,000
2.10	1.50	0.50	0.10	2	1	1	1.15	25,000
2.40	1.71	0.57	0.11	1	1	1	1.86	24,000
2.70	1.93	0.64	0.13	1	1	1	1.86	24,000

corresponding investment of 54,000 Euros. We can see from the graph that the item approach does not work well. In general, the item approach works well if the differences between the prices of the SKU's are small. The solutions will deviate considerably from the efficient solutions under large price differences.

2.7 Alternative Service Measures

Now consider the situation that we are not interested in the aggregate mean number of backorders but in some other service measure. Fortunately, it is possible to adjust the greedy algorithm to deal with a different service measure. In Sects. 2.7.1 and 2.7.2, we discuss the implications of using the aggregate mean waiting time and availability, respectively, as the relevant service measure. Furthermore, we deal with the sum of the backorder probabilities in Sect. 2.7.3. In Sect. 2.7.4, we show how to deal with the aggregate fill rate. Lastly, in Sect. 2.7.5, we describe the implications of using the aggregate mean number of stockouts as a service measure.

2.7.1 Aggregate Mean Waiting Time

A service measure that is quite often considered is the expected waiting time until an arbitrary spare part demand is fulfilled. This measure, which we denote as the *aggregate mean waiting time*, is obtained as follows. Let $W_i(S_i)$ be the mean waiting time for a spare part demand of SKU $i \in I$ under base stock level S_i. Then the aggregate mean waiting time $W(\mathbf{S})$ is equal to

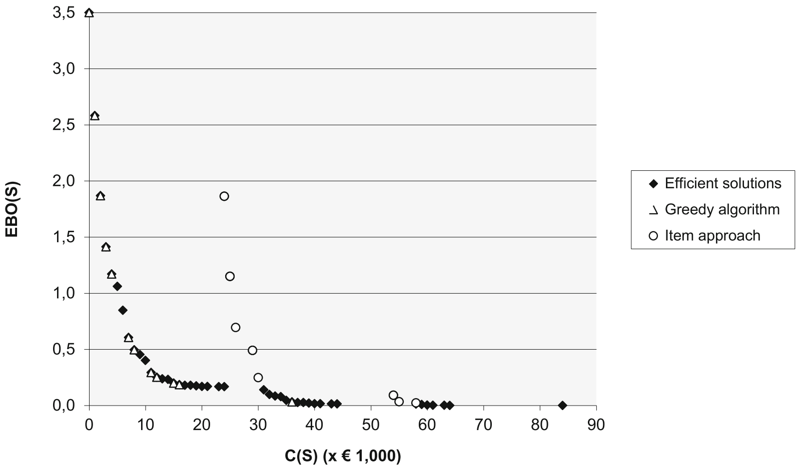

Fig. 2.7 Outcome of the item approach

$$
\begin{aligned}
W(\mathbf{S}) &= \sum_{i \in I} P\{\text{an arbitrary demand is for SKU } i\} \\
&\qquad\qquad \times (\text{expected waiting time for SKU } i) \\
&= \sum_{i \in I} \frac{m_i}{M} W_i(S_i).
\end{aligned}
$$

In Problem (P), the constraint $EBO(\mathbf{S}) \leq EBO^{\text{obj}}$ is replaced by the constraint $W(\mathbf{S}) \leq W^{\text{obj}}$, where W^{obj} is the target level for the aggregate mean waiting time.

By Little's law [12], it is easily seen that $W_i(S_i) = EBO_i(S_i)/m_i$, $i \in I$. As a result,

$$
W(\mathbf{S}) = \sum_{i \in I} \frac{m_i}{M} \frac{EBO_i(S_i)}{m_i} = \frac{1}{M} \sum_{i \in I} EBO_i(S_i) = \frac{1}{M} EBO(\mathbf{S}),
$$

and thus the constraint $W(\mathbf{S}) \leq W^{\text{obj}}$ is equivalent to the constraint $EBO(\mathbf{S}) \leq EBO^{\text{obj}}$ with $EBO^{\text{obj}} = MW^{\text{obj}}$ (notice that the relation $W(\mathbf{S}) = EBO(\mathbf{S})/M$ also is obtained by applying Little's formula to the total number of backordered demands). This means that the problem with an aggregate mean waiting time constraint can be solved in the same way as Problem (P) of the basic model.

2.7.2 Average Availability

As stated in Sect. 2.2, the constraint on the aggregate mean number of backorders is closely related to an *availability constraint*. The average availability is equal to the fraction of time that any given machine is available. Let Z be the total number of machines, and let Z_i be the number of parts of SKU i installed per machine.

We can approximate the average availability as follows. The average number of backorders of SKU i is $EBO_i(S_i)$. Hence, the probability that a given part of SKU i in a given machine is working is equal to $1 - \frac{EBO_i(S_i)}{ZZ_i}$. Next, ignoring dependencies between these probabilities for the various parts in a given machine, we obtain the following approximation for $A(\mathbf{S})$ (which is an accurate estimate in case the number of machines is sufficiently large and the total number of backordered demands is low):

$$A(\mathbf{S}) \approx \prod_{i \in I} \left(1 - \frac{EBO_i(S_i)}{ZZ_i}\right)^{Z_i}.$$

For sufficiently high values of $A(S)$ (i.e., sufficiently low values of the $EBO_i(S_i)$), the product on the right hand side may be approximated by its first order approximation:

$$\begin{aligned}
A(\mathbf{S}) &\approx \prod_{i \in I} \left(1 - \frac{EBO_i(S_i)}{ZZ_i}\right)^{Z_i} \\
&= 1 - \sum_{i \in I} Z_i \frac{EBO_i(S_i)}{ZZ_i} + \sum_{i \in I} \frac{Z_i(Z_i - 1)}{2} \left(\frac{EBO_i(S_i)}{ZZ_i}\right)^2 \\
&\quad + \sum_{i,j \in I,\ i \neq j} Z_i Z_j \frac{EBO_i(S_i)}{ZZ_i} \frac{EBO_j(S_j)}{ZZ_j} - \ \ldots \\
&\approx 1 - \sum_{i \in I} Z_i \frac{EBO_i(S_i)}{ZZ_i} = 1 - \frac{1}{Z} \sum_{i \in I} EBO_i(S_i) = 1 - \frac{1}{Z} EBO(\mathbf{S}).
\end{aligned}$$

Hence, for a sufficiently high target A^{obj} for $A(\mathbf{S})$, a heuristic solution for the problem with a target average availability may be obtained via the heuristic solution for Problem (P) with target $EBO^{\text{obj}} = Z(1 - A^{\text{obj}})$ for the aggregate mean number of backorders.

2.7.3 Sum of Backorder Probabilities

In our basic model, we assumed that a relatively large group of machines is supported by the single stockpoint. In some applications, only one machine is supported. This situation occurs on board of a frigate, or when customers have one machine only and spare parts are kept on stock at the customer itself. The corresponding optimization problem has been studied by Rustenburg [15]. In this problem, the availability of the machine is equal to $A(\mathbf{S}) = \prod_{i \in I}(1 - PBO_i(S_i))$, where $PBO_i(S_i)$ denotes the backorder probability of SKU i. This is the steady-state probability for a positive number of backorders for SKU i. This probability is equal to:

$$PBO_i(S_i) = \sum_{x = S_i + 1}^{\infty} P\{X_i = x\}.$$

For sufficiently high values of the availability (i.e., sufficiently low values of the $PBO_i(S_i)$), it holds that the availability is well approximated by its first order approximation

$$\begin{aligned} A(\mathbf{S}) &= 1 - \sum_{i \in I} PBO_i(S_i) + \sum_{i,j \in I,\, i \neq j} PBO_i(S_i) PBO_j(S_j) - \ldots \\ &\approx 1 - \sum_{i \in I} PBO_i(S_i), \end{aligned}$$

and then aiming for a high availability is almost equivalent to aiming for a low sum of backorder probabilities:

$$PBO(\mathbf{S}) := \sum_{i \in I} PBO_i(S_i).$$

Then we get Problem (P) with a constraint for $PBO(\mathbf{S})$. The greedy algorithm can be slightly modified to solve this problem.

For the first order difference function of $PBO_i(S_i)$, $i \in I$, we find that

$$\Delta PBO_i(S_i) = PBO_i(S_i+1) - PBO_i(S_i) = -P\{X_i = S_i + 1\} \leq 0, \qquad S_i \in \mathbb{N}_0;$$

i.e., $PBO_i(S_i)$ is decreasing on its whole domain. Via the second order difference function of $PBO_i(S_i)$, one can show that $PBO_i(S_i)$ is convex for $S_i \geq \max\{\lceil m_i t_i - 2 \rceil, 0\}$ (see Problem 2.4). The amount $m_i t_i$ represents the average number of parts in the repair pipeline. If this average pipeline stock is smaller than or equal to 2, then $\max\{\lceil m_i t_i - 2 \rceil, 0\} = 0$ and thus $PBO_i(S_i)$ is convex on its whole domain. If the average pipeline stock is larger than 2 (for slow moving SKU's this will not happen in general), then $\max\{\lceil m_i t_i - 2 \rceil, 0\} > 0$ and we exclude solutions $\mathbf{S}$ with $S_i \leq \max\{\lceil m_i t_i - 2 \rceil, 0\}$ from the solution space. The excluded solutions have a high value for $PBO_i(S_i)$ and thus also for $PBO(\mathbf{S})$. Hence, they are not feasible for the relevant problem instances with a low target for the sum of backorder probabilities.

In the greedy algorithm, the exclusion of solutions $\mathbf{S}$ with $S_i \leq \max\{\lceil m_i t_i - 2 \rceil, 0\}$ for some $i \in I$ simply implies that we use the solution $\mathbf{S}$ with $S_i = \max\{\lceil m_i t_i - 2 \rceil, 0\}$ for all $i \in I$ as the starting solution. Next, in each iteration of the greedy algorithm, the ratio of the decrease in $PBO(\mathbf{S})$ and the increase in $C(\mathbf{S})$, due to an increase of the basestock level of SKU i by 1, is measured by Γ_i, which equals:

$$\Gamma_i = \frac{P\{X_i = S_i + 1\}}{c_i^{\mathrm{a}}}.$$

In each iteration, we increase the basestock level of the SKU with the highest Γ_i.

2.7.4 Aggregate Fill Rate

The *aggregate fill rate* is defined as the probability that an arbitrary demand for the total group of SKU's is fulfilled immediately, or, equivalently, as the fraction of the total demand stream that is fulfilled from stock. Let the fill rate for SKU i, also called *item fill rate*, be denoted by $\beta_i(S_i)$, then

$$\beta(\mathbf{S}) = \sum_{i \in I} \frac{m_i}{M} \beta_i(S_i). \tag{2.10}$$

The target aggregate fill rate is given by β^{obj}.

Demand for SKU i arrives according to a Poisson process, and thus, by the PASTA (Poisson Arrivals See Time Averages) property, an arbitrary arriving demand observes the system in steady state. Hence, with probability $P\{OH_i > 0\} = P\{X_i < S_i\}$, a positive stock on hand is observed and the demand can be fulfilled immediately, and otherwise not. So,

$$\beta_i(S_i) = \sum_{x=0}^{S_i-1} P\{X_i = x\}. \tag{2.11}$$

The item fill rate $\beta_i(S_i)$ for an SKU $i \in I$ is a function with domain $\mathbb{N}_0$. Lemma 2.4 says that $\beta_i(S_i)$, and thus also $f_i(S_i) = \frac{m_i}{M}\beta_i(S_i)$, is increasing on its whole domain and concave for $S_i \geq \max\{\lceil m_i t_i - 1 \rceil, 0\}$, where $\lceil x \rceil$ denotes the rounded value to above for any $x \in \mathbb{R}$.

Lemma 2.4. *For each SKU $i \in I$, the item fill rate $\beta_i(S_i)$ is increasing on its whole domain and concave for $S_i \geq \max\{\lceil m_i t_i - 1 \rceil, 0\}$.*

Proof. Let $i \in I$. By (2.11),

$$\Delta\beta_i(S_i) = \beta_i(S_i+1) - \beta_i(S_i) = P\{X_i = S_i\} \geq 0, \qquad S_i \in \mathbb{N}_0, \tag{2.12}$$

which shows that $\beta_i(S_i)$ is increasing on its whole domain. Further,

$$\Delta^2\beta_i(S_i) = P\{X_i = S_i+1\} - P\{X_i = S_i\}, \qquad S_i \in \mathbb{N}_0. \tag{2.13}$$

By Eq. (2.8),

$$P\{X_i = S_i+1\} = \frac{m_i t_i}{S_i+1} P\{X_i = S_i\}, \qquad S_i \in \mathbb{N}_0,$$

and by substitution of this recursive relation into (2.13), we find

$$\Delta^2\beta_i(S_i) = \left(\frac{m_i t_i}{S_i+1} - 1\right) P\{X_i = S_i\}, \qquad S_i \in \mathbb{N}_0.$$

From this formula, it follows that $\Delta^2\beta_i(S_i) \leq 0$ if and only if $\frac{m_i t_i}{S_i+1} - 1 \leq 0$, i.e., if and only if $S_i \geq m_i t_i - 1$. In other words, $\beta_i(S_i)$ is concave for $S_i \geq m_i t_i - 1$. Because of the integrality and nonnegativity of S_i, the condition $S_i \geq m_i t_i - 1$ is equivalent to $S_i \geq \max\{\lceil m_i t_i - 1 \rceil, 0\}$. □

The amount $m_i t_i$ represents the average number of parts in the repair pipeline. If this average pipeline stock is smaller than or equal to 1, then $\max\{\lceil m_i t_i - 1 \rceil, 0\} = 0$ and thus $\beta_i(S_i)$ is concave on its whole domain. If the average pipeline stock is larger than 1 (for slow moving SKU's this will not happen in general), then $\max\{\lceil m_i t_i - 1 \rceil, 0\} > 0$ and we exclude solutions $\mathbf{S}$ with $S_i \leq \max\{\lceil m_i t_i - 1 \rceil, 0\}$ from the solution space. The excluded solutions have a very low value for $\beta_i(S_i)$ and they are generally not relevant for problem instances with a high target for the aggregate fill rate.

We now reformulate Problem (Q) as defined in Sect. 2.4. First, we replace the minimization of $EBO(\mathbf{S})$ by the maximization of $\beta(\mathbf{S})$. Second, we limit the solution space to $\mathscr{S}' = \mathscr{S}'_1 \times \mathscr{S}'_2 \times \ldots \times \mathscr{S}'_{|I|}$, with $\mathscr{S}'_i = \{S_i \in \mathbb{N}_0 : S_i \geq m_i t_i - 1\}$, $i \in I$. Hence, we obtain the Problem (Q$'$):

$$
\begin{aligned}
(\mathrm{Q}') \qquad & \min \qquad C(\mathbf{S}) \\
& \max \qquad \beta(\mathbf{S}) \\
& \text{subject to } \mathbf{S} \in \mathscr{S}'.
\end{aligned}
$$

Problem (Q$'$) is still separable and the functions $\beta_i(S_i)$ are now increasing and concave on their whole domains $\mathscr{S}'_i$. Hence a set of efficient solutions can be generated by a greedy algorithm. A first efficient solution $\mathbf{S} = (S_1, \ldots, S_{|I|})$ is obtained by setting $S_i = \max\{\lceil m_i t_i - 1 \rceil, 0\}$ for each SKU $i \in I$. This solution is efficient because it has the lowest possible investment. Next, for each SKU i, we compute the increase in $\beta(\mathbf{S})$ relative to the increase in $C(\mathbf{S})$ when S_i would be increased by one unit. The increase in $C(\mathbf{S})$ equals c_i^{a}, while the increase in $\beta(\mathbf{S})$ equals (use (2.12))

$$\Delta_i\beta(\mathbf{S}) = \frac{m_i}{M}\Delta\beta_i(S_i) = \frac{m_i}{M}(\beta_i(S_i+1) - \beta_i(S_i)) = \frac{m_i}{M}P\{X_i = S_i\}.$$

The increase in $\beta(\mathbf{S})$ divided by the increase in $C(\mathbf{S})$ is denoted by

$$\Gamma_i = \frac{m_i P\{X_i = S_i\}}{M c_i^{\mathrm{a}}}.$$

The SKU with the highest value for Γ_i is selected, and the corresponding basestock level is increased by one unit (ties are broken with equal probabilities). The new solution S is also efficient and is added to a set of efficient solutions. The generation of efficient solutions is continued until a given aggregate fill rate or inventory investment has been reached, or until some other stop criterium is met. The formal procedure is described in Algorithm 2.2.

Algorithm 2.2 *(Greedy algorithm)*

Step 1 Set $S_i := \max\{\lceil m_i t_i - 1 \rceil, 0\}$ for all $i \in I$;
$S = (S_1, S_2, \ldots, S_{|I|})$;
$\mathscr{E} := \{\mathbf{S}\}$;
Compute $C(\mathbf{S})$ and $\beta(\mathbf{S})$.
Step 2 $\Gamma_i := (m_i P\{X_i = S_i\})/(Mc_i^{\mathrm{a}})$ for all $i \in I$;
$k := \arg\max\{\Gamma_i : i \in I\}$;
$\mathbf{S} := \mathbf{S} + \mathbf{e}_k$
$\mathscr{E} := \mathscr{E} \cup \{\mathbf{S}\}$.
Step 3 Compute $C(\mathbf{S})$ and $\beta(\mathbf{S})$;
If ‘stop criterium’, then stop, else goto Step 2.

For the computation of the pipeline stock probabilities $P\{X_i = S_i\}$ in this algorithm, we advice to use the recursive expression (2.8) for the sake of efficiency. In the following lemma, it is formally stated that Algorithm 2.2 generates efficient solutions for Problem (Q′) (again, the proof follows directly from Theorem 2 in Fox [8]).

Lemma 2.5. *At termination of Algorithm 2.2, the set $\mathscr{E}$ consists of efficient solutions for Problem (Q′).*

Example 2.1 (continued). If we apply the greedy algorithm to our example with $\beta^{\mathrm{obj}} = 0.98$, we find the efficient solutions that are displayed in Table 2.5. After twelve iterations we obtain basestock levels that fulfill the target aggregate fill rate at an inventory investment of 41,000 Euros. In Fig. 2.8, the efficient solutions obtained by both enumeration and the greedy algorithm are displayed. As you can see the greedy algorithm again generates a subset of all efficient solutions.

2.7.5 Aggregate Mean Number of Stockouts

Another service measure is the aggregate mean number of stockouts. This measure counts the number of stockouts for all spare parts together. The mean number of stockouts of SKU i is calculated by the following formula:

$$\alpha_i(S_i) = m_i(1 - \beta_i(S_i)).$$

The aggregate mean number of stockouts is then the sum of the individual mean numbers of stockouts:

$$\alpha(\mathbf{S}) = \sum_{i \in I} \alpha_i(S_i).$$

The increase in $\alpha_i(S_i)$ by increasing the basestock level of SKU i by 1 is:

$$\Delta\alpha_i(S_i) = \alpha_i(S_i + 1) - \alpha_i(S_i) = -m_i P\{X_i = S_i\} \le 0, \qquad S_i \in \mathbb{N}_0.$$

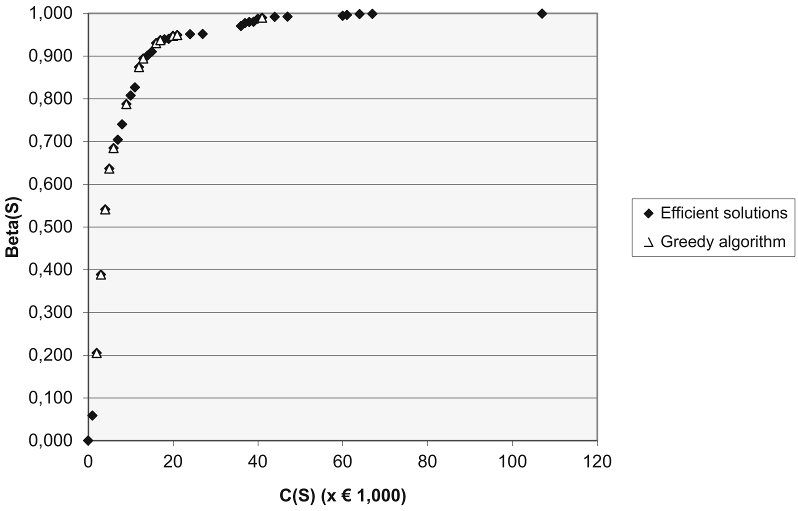

Fig. 2.8 Outcome of the greedy algorithm

Table 2.5 Steps of the greedy algorithm

Iteration	Γ_1	Γ_2	Γ_3	k	S_1	S_2	S_3	$\beta(\mathbf{S})$	$C(\mathbf{S})$ (Euros)
0	–	–	–	–	2	0	0	0.205	2,000
1	$1.83 \cdot 10^{-4}$	$3.45 \cdot 10^{-5}$	$2.02 \cdot 10^{-6}$	1	3	0	0	0.388	3,000
2	$1.53 \cdot 10^{-4}$	$3.45 \cdot 10^{-5}$	$2.02 \cdot 10^{-6}$	1	4	0	0	0.541	4,000
3	$9.54 \cdot 10^{-5}$	$3.45 \cdot 10^{-5}$	$2.02 \cdot 10^{-6}$	1	5	0	0	0.637	5,000
4	$4.77 \cdot 10^{-5}$	$3.45 \cdot 10^{-5}$	$2.02 \cdot 10^{-6}$	1	6	0	0	0.684	6,000
5	$1.99 \cdot 10^{-5}$	$3.45 \cdot 10^{-5}$	$2.02 \cdot 10^{-6}$	2	6	1	0	0.788	9,000
6	$1.99 \cdot 10^{-5}$	$2.87 \cdot 10^{-5}$	$2.02 \cdot 10^{-6}$	2	6	2	0	0.874	12,000
7	$1.99 \cdot 10^{-5}$	$1.20 \cdot 10^{-5}$	$2.02 \cdot 10^{-6}$	1	7	2	0	0.894	13,000
8	$7.10 \cdot 10^{-6}$	$1.20 \cdot 10^{-5}$	$2.02 \cdot 10^{-6}$	2	7	3	0	0.930	16,000
9	$7.10 \cdot 10^{-6}$	$3.33 \cdot 10^{-6}$	$2.02 \cdot 10^{-6}$	1	8	3	0	0.937	17,000
10	$2.22 \cdot 10^{-6}$	$3.33 \cdot 10^{-6}$	$2.02 \cdot 10^{-6}$	2	8	4	0	0.947	20,000
11	$2.22 \cdot 10^{-6}$	$6.93 \cdot 10^{-7}$	$2.02 \cdot 10^{-6}$	1	9	4	0	0.949	21,000
12	$6.16 \cdot 10^{-7}$	$6.93 \cdot 10^{-7}$	$2.02 \cdot 10^{-6}$	3	9	4	1	0.989	41,000

Lemma 2.4 states that $\beta_i(S_i)$ is increasing on its whole domain and concave for $S_i \geq \max\{\lceil m_i t_i - 1 \rceil, 0\}$. Applying Lemma 2.4 to the situation of $\alpha_i(S_i)$ results in the conclusion that $\alpha_i(S_i)$ is decreasing on its whole domain and convex for $S_i \geq \max\{\lceil m_i t_i - 1 \rceil, 0\}$. This can be taken into account for the starting solution in the greedy algorithm. In each iteration of the greedy algorithm, the ratio of the decrease in $\alpha(\mathbf{S})$ and the increase in $C(\mathbf{S})$, due to an increase of the basestock level of SKU i with 1, is measured by Γ_i, which equals:

$$\Gamma_i = \frac{m_i P\{X_i = S_i\}}{c_i^{\mathrm{a}}}.$$

In each iteration, we increase the basestock level of the SKU with the highest Γ_i.

2.8 Inventory Planning During the Exploitation Phase

The basic model of Sect. 2.2 has been formulated for the initial supply problem. In this section, we consider the use of this model, or a slightly modified version, for the tactical inventory planning during the exploitation phase of the machines that are supported. Here, we limit ourselves to the part of the exploitation phase that repair of repairables and procurement of consumables are possible and the installed base has a reasonably stable size. In that part of the exploitation phase, it is common that every month or quarter, the base stock levels are updated. We refer to these time instants as the tactical planning time instants.

At a tactical planning instant, one first generates new estimates for the demand rates m_i, $i \in I$. This is done via some forecasting method, in which one may also incorporate a changed size of the installed base. Next, based on the new demand rates, new base stock levels S_i, $i \in I$, are determined. For the latter step, one can use the basic model in Sect. 2.2. This model is based on a steady-state analysis, and looking at such an analysis is fine only if changes in the basestock levels are effectuated quickly enough for all SKU's. If one orders additional parts for some repairable SKU's and the delivery leadtimes would be 1 year (notice that these lead-times are other times than the repair leadtimes), then one should also look at the transient behavior of these SKU's in the first year. This possible complication is ignored below.

When applying the model in Sect. 2.2 during the exploitation phase, one has already parts on stock or in the repair pipeline, and these amounts have to be taken into account. Let the current level for the inventory position of SKU i be denoted by S_i^{cur}, $i \in I$. The effect of these current base stock levels S_i^{cur} depends on what one can do with the parts of SKU's for which we would like to decrease the base stock level. We distinguish two main cases.

The first case applies for the stock at a local warehouse. Parts that are not needed anymore at the local warehouse go back to the central depot, and the central depot may be able to use those parts for other local warehouses. One can then assume that decreasing the base stock level of an SKU i from S_i^{cur} to a lower level S_i reduces the inventory investment by c_i^{a} per unit decrease. And, increasing the base stock level of an SKU i from S_i^{cur} to a higher level S_i increases the inventory investment by c_i^{a} per unit increase. In other words, the marginal costs for increasing or decreasing base stock levels are the same as for the original problem of Sect. 2.2. Hence, one can optimize the new base stock levels S_i in the same way as before, i.e., as if there is no current stock.

The second case generally applies for the stock at a central depot or the stock in a network as a whole. A decrease of the basestock level for a consumable SKU i means that one part less is procured in the near future. This saves an amount c_i^{a}. Sometimes it is possible to deliver parts back to the original supplier, but one may then receive a lower price back than the original price c_i^{a} for which the part was bought. A decrease of the basestock level for a repairable SKU i may lead to less repairs in the near future or the excess stock is sold, e.g., on a second-hand parts market. In both cases one will save less than c_i^{a} per unit decrease. To capture all cases, we introduce a parameter $\hat{c}_i^{\mathrm{a}}$ ($0 \le \hat{c}_i^{\mathrm{a}} \le c_i^{\mathrm{a}}$) that denotes how much is saved per unit decrease of the basestock level of SKU i, $i \in I$. For an SKU i, the current stock represents a value of $\hat{c}_i^{\mathrm{a}} S_i^{\mathrm{cur}}$, and the cost of changing the basestock level from S_i^{cur} to S_i is

$$\begin{cases} c_i^{\mathrm{a}}(S_i - S_i^{\mathrm{cur}}) & \text{if } S_i \ge S_i^{\mathrm{cur}} \\ \hat{c}_i^{\mathrm{a}}(S_i - S_i^{\mathrm{cur}}) & \text{if } 0 \le S_i < S_i^{\mathrm{cur}}. \end{cases}$$

Adding a constant cost factor $\hat{c}_i^{\mathrm{a}} S_i^{\mathrm{cur}}$ leads to the following modified cost function $C_i(S_i)$:

$$C_i(S_i) = \begin{cases} \hat{c}_i^{\mathrm{a}} S_i & \text{if } 0 \le S_i < S_i^{\mathrm{cur}} \\ \hat{c}_i^{\mathrm{a}} S_i^{\mathrm{cur}} + c_i^{\mathrm{a}}(S_i - S_i^{\mathrm{cur}}) & \text{if } S_i \ge S_i^{\mathrm{cur}}. \end{cases}$$

For the rest, Problem (P) and Problem (Q) remain the same. The modified cost functions $C_i(S_i)$ are increasing and convex. Hence, generating efficient solutions for $C(\mathbf{S})$ and $EBO(\mathbf{S})$ can still be done by the same greedy algorithm (cf. Fox [8], Sect. 8); the factors Γ_i in Algorithm 2.1 are now computed as

$$\Gamma_i = \frac{-\Delta EBO_i(S_i)}{\Delta C_i(S_i)} = \begin{cases} (1 - \sum_{x=0}^{S_i} P\{X_i = x\})/\hat{c}_i^{\mathrm{a}} & \text{if } \hat{c}_i^{\mathrm{a}} > 0 \text{ and } 0 \le S_i < S_i^{\mathrm{cur}} \\ \infty & \text{if } \hat{c}_i^{\mathrm{a}} = 0 \text{ and } 0 \le S_i < S_i^{\mathrm{cur}} \\ (1 - \sum_{x=0}^{S_i} P\{X_i = x\})/c_i^{\mathrm{a}} & \text{if } S_i \ge S_i^{\mathrm{cur}}. \end{cases}$$

If $\hat{c}_i^{\mathrm{a}} = 0$ for some SKU i, then one could say that the first S_i^{cur} are free for this SKU and they should we selected first. We enforce this by saying that $\Gamma_i = \infty$ as long as $S_i < S_i^{\mathrm{cur}}$. The alternative would be to start with $S_i := S_i^{\mathrm{cur}}$ instead of $S_i := 0$ for such an SKU.

2.9 Emergency Shipments

In several practical situations, demand will not be backordered in case of a stockout, but an emergency option will be applied. For example, when our single-location model is applied to a local warehouse, it may be so that a demand is satisfied by an emergency shipment from a central depot or from another local warehouse (in the latter case, the shipment is generally denoted as a lateral transshipment). When our model is applied for a central depot, such an emergency shipment may be possible from an external supplier or another source. Or, for repairables, it may be possible to execute an emergency repair for a part that is already at the repair shop, followed by a fast form of transport. For our spare parts stockpoint, it means that the demand is lost (or a 'lost sale') instead of backordered when it cannot be satisfied from stock. Emergency shipments (or repairs) will be expensive in general, but, in case downtime costs of machines are high, it is natural to apply them. The application of emergency shipments has several consequences for our model description and its whole analysis.

Assume that the average time for an emergency shipment is equal to t_i^{em} for SKU i. Define $W_i(S_i)$ as the mean waiting time for an arbitrary demand for SKU i. It holds that

$$W_i(S_i) = (1 - \beta_i(S_i))t_i^{\mathrm{em}},$$

where $\beta_i(S_i)$ is the fill rate for SKU i. Next, we define $W(\mathbf{S})$ as the mean waiting time for an arbitrary demand for all SKU's together, i.e., the aggregate mean waiting time:

$$W(\mathbf{S}) = \sum_{i \in I} \frac{m_i}{M} W_i(S_i).$$

Then the total downtime of all machines together is equal to $MW(\mathbf{S})$, and the average availability of the machines may be approximated by

$$A(\mathbf{S}) \approx 1 - \frac{MW(S)}{Z}.$$

Hence, setting a minimum level A^{obj} for $A(\mathbf{S})$ is equivalent to setting a maximum level $W^{\mathrm{obj}} = Z(1 - A^{\mathrm{obj}})/M$ for $W(\mathbf{S})$.

With respect to costs, one has two types of costs now. As before, one has costs for buying spare parts at the beginning, with price c_i^{a} for SKU i. In addition, one has costs for the emergency shipments. Each time that one applies an emergency shipment for SKU i, one has costs c_i^{em}. We assume that c_i^{em} contains the costs for a fast delivery from another location. When an emergency shipment is applied, one has one normal replenishment less at our stockpoint, and therefore those costs should be subtracted. The average costs per time unit for SKU i for emergency shipments are equal to $m_i(1 - \beta_i(S_i))c_i^{\mathrm{em}}$. The total costs consist of these costs per time unit and the one-time costs at the beginning. The latter costs may be transformed into inventory holding costs per time unit. We use c_i^{h} to denote the inventory holding cost per time unit per part of SKU i. Then the average costs per time unit for SKU i are equal to

$$\hat{C}_i(S_i) = c_i^{\mathrm{h}} S_i + m_i(1 - \beta_i(S_i))c_i^{\mathrm{em}}, \tag{2.14}$$

and the total average costs are equal to $\hat{C}(\mathbf{S}) = \sum_{i \in I} \hat{C}_i(S_i)$. The optimization problem that we want to solve is as follows:

$$\begin{aligned} (\mathrm{P}'') \qquad & \min \qquad \hat{C}(\mathbf{S}) \\ & \text{subject to } W(\mathbf{S}) \le W^{\mathrm{obj}}, \\ & \qquad\qquad\quad \mathbf{S} \in \mathscr{S}. \end{aligned}$$

The related multi-objective problem is:

$$\begin{aligned} (\mathrm{Q}'') \qquad & \min \qquad \hat{C}(\mathbf{S}) \\ & \min \qquad W(\mathbf{S}) \\ & \text{subject to } \mathbf{S} \in \mathscr{S}. \end{aligned}$$

The evaluation of a given policy basestock policy $\mathbf{S}$ can still be made per SKU. Under the application of emergency shipments, the number of parts in the repair pipeline of SKU i is bounded from above by S_i. I.e., the behavior of the number of parts in repair of SKU i is no longer as in an $M|G|\infty$ queue but as in an $M|G|c|c$ queue with $c = S_i$ parallel servers, arrival rate m_i, and mean service time t_i. The $M|G|c|c$ queue is also called an *Erlang loss system*. The fill rate $\beta_i(S_i)$ for SKU i is obtained via the so-called Erlang loss probability. The fill rate is equal to the fraction of time that there is at least one part on stock. This is equal to the fraction of time that at least one server is free in the corresponding Erlang loss system. The latter probability is equal to 1 minus the fraction of time that all servers are occupied, i.e., to 1 minus the Erlang loss probability. Hence,

$$\beta_i(S_i) = 1 - \frac{\frac{1}{S_i!}\rho_i^{S_i}}{\sum_{j=0}^{S_i}\frac{1}{j!}\rho_i^j}, \tag{2.15}$$

where $\rho_i := m_i t_i$.

Karush [9] has shown that the Erlang loss probability is strictly convex and decreasing as a function of the number of servers (see also Remark 2 in [11]). This implies that $\beta_i(S_i)$ is strictly concave and increasing on its whole domain. As a result:

- For each $i \in I$, $W_i(S_i)$ is decreasing and concave on its whole domain.
- For each $i \in I$, $\hat{C}_i(S_i)$ is convex on its whole domain. The function $\hat{C}_i(S_i)$ may now be decreasing for smaller values of S_i, because of the presence of the emergency costs.

Let $S_{i,\min} := \arg\min \hat{C}_i(S_i)$. Then, obviously, for Problem (P″) and its corresponding multi-objective programming problem (Q″), we may exclude solutions with $S_i < S_{i,\min}$ for some $i \in I$. Then, one can generate efficient solutions for $C(\mathbf{S})$ and $W(\mathbf{S})$ in a similar way as for Problem (Q) in Sect. 2.4 (see [8], Sect. 8); the factors Γ_i are now computed as $\Gamma_i := -\Delta_i W(\mathbf{S})/\Delta_i \hat{C}(\mathbf{S})$ with $\Delta_i W(\mathbf{S}) = (m_i/M)\Delta W_i(S_i) = (m_i/M)(W_i(S_i+1) - W_i(S_i))$ and $\Delta_i \hat{C}(\mathbf{S}) = \Delta \hat{C}_i(S_i) = \hat{C}_i(S_i+1) - \hat{C}_i(S_i)$, $S_i \geq S_{i,\min}$. This results in the following algorithm.

Algorithm 2.3 *(Greedy algorithm)*

Step 1 $S_{i,\min} := \arg\min \hat{C}_i(S_i)$ for all $i \in I$;
Set $S_i := S_{i,\min}$ for all $i \in I$, and $\mathbf{S} = (S_{1,\min}, \ldots, S_{|I|,\min})$;
$\mathscr{E} := \{\mathbf{S}\}$;
Compute $\hat{C}(\mathbf{S})$ and $W(\mathbf{S})$.
Step 2 $\Gamma_i := -(m_i \Delta W_i(S_i))/(M \Delta \hat{C}_i(S_i))$ for all $i \in I$;
$k := \arg\max\{\Gamma_i : i \in I\}$;
$\mathbf{S} := \mathbf{S} + \mathbf{e}_k$;
$\mathscr{E} := \mathscr{E} \cup \{\mathbf{S}\}$.
Step 3 Compute $\hat{C}(\mathbf{S})$ and $W(\mathbf{S})$;
If ‘stop criterium’, then stop, else goto Step 2.

In the following lemma, it is formally stated that Algorithm 2.3 generates efficient solutions for Problem (Q″). The proof of this lemma follows from Fox [8] (Sect. 8).

Lemma 2.6. *At termination of Algorithm 2.3, the set $\mathscr{E}$ consists of efficient solutions for Problem (Q″).*

2.10 Extensions

In this section we describe a few extensions to our model.

2.10.1 Consumables and Condemnation

In Sect. 2.2, we use the terminology that is common for repairable spare parts. Nevertheless, the model also applies when all SKU's, or some of the SKU's, are consumable. For a consumable, we assume that a new part is procured each time that a demand occurs, and we assume i.i.d. procurement leadtimes, which again are denoted by t_i.

One can also have a kind of mixture of the situations for a repairable and a consumable, respectively. So far, for a repairable, we assume that a repair is always executed and it is always successful. In practice, this is often more subtle. In many cases, components can fail due to various reasons. Some of the resulting defects may be repairable and others not. It may also be that a part is only repaired for a limited number of times, because its performance slowly decreases after each repair. One generally refers to these phenomena as *condemnation*. From a modeling point of view, condemnation can be easily incorporated, which was already noticed by [5]. The idea is to introduce a parameter r_i that represents the probability that a failed part of SKU i can be repaired. Next, we distinguish a mean repair leadtime t_i^{rep} and a mean procurement leadtime t_i^{proc} (the procurement leadtime for SKU i may be stochastic as long as realizations of the leadtime for different order are independent). Then an arbitrary failed part leads to the arrival of a ready-for-use/new part at the most upstream location after an average leadtime $t_i = r_i t_i^{\text{rep}} + (1 - r_i)t_i^{\text{proc}}$. These t_i's are the leadtimes that can be used in the basic model without emergency shipments, and similarly in the model with emergency shipments.

2.10.2 Excluding Pipeline Stock

In Sect. 2.10.1, we denoted that the model of Sect. 2.2 also applies for consumables. In that basic model, we look at inventory investment. In Sect. 2.9, we describe the model with emergency shipments, and we have switched to inventory holding costs so that these costs can be added to the emergency shipments costs to obtain the average costs per time unit. When determining inventory holding costs for consumables that are procured at an external supplier, it is more appropriate to exclude the pipeline stock (i.e., the parts that are on order but not on hand yet). For a consumable SKU i in the model of Sect. 2.9, replenishment orders for single units are placed with rate $m_i\beta_i(S_i)$, and, by Little's law, the average pipeline stock is $m_i\beta_i(S_i)t_i$. This leads to the following modified expression for the average costs per time unit for SKU i, as given in (2.14):

$$\begin{aligned}\hat{C}_i(S_i) &= c_i^{\mathrm{h}}(S_i - m_i\beta_i(S_i)t_i) + m_i(1-\beta_i(S_i))c_i^{\mathrm{em}} \\ &= c_i^{\mathrm{h}}(S_i - m_i t_i) + m_i(1-\beta_i(S_i))(c_i^{\mathrm{em}} + c_i^{\mathrm{h}} t_i).\end{aligned}$$

This function is still convex (like the function $\hat{C}_i(S_i)$ in (2.14)), and thus one can still follow the same solution procedure as in Sect. 2.9.

2.10.3 Batching

In our basic model, we assume one-for-one replenishments for all SKU's. For a local warehouse, this is generally justified because it receives consolidated replenishments for all SKU's together from a central depot. However, at a central depot, one may send failed parts into repair at external repair shops or one orders consumables at outside suppliers, and then some form of batching may be desired. Reasons for using batching may be fixed setup costs for certain repair activities, fixed ordering and delivery costs that are charged by external suppliers, or pack sizes that are prescribed by suppliers. Applying the logic of the EOQ rule shows that generally one-for-one replenishments will make sense for the more expensive SKU's, which have high inventory holding costs and/or low demand rates. For less expensive components, however, it may be appropriate to use a fixed batch size Q, and thus to follow an (s,Q)-policy instead of a basestock policy. Consider the basic model (without emergency shipments), but assume now that for each SKU i, a batch of Q_i failed parts is sent into repair as soon as the inventory position of SKU i drops to its reorder level s_i (≥ -1). (Notice that a basestock policy with basestock level $S_i = s_i + 1$ is obtained in case $Q_i = 1$.) The repair leadtime for such a batch is always equal to t_i (we now assume that the repair leadtime is deterministic). The performance for SKU i is then obtained by making use of the following two key properties:

- The inventory level $OH_i(t) - BO_i(t)$ of SKU i at an arbitrary time point t is given by the inventory position at time point $t - t_i$ minus the demand in the time interval $[t - t_i, t)$ (see Sect. 5.3.2 of Axsäter [1]).
- The inventory position at time point $t - t_i$ is as the inventory position at an arbitrary time point and thus is uniformly distributed on the integers $s_i + 1, s_i + 2, \ldots, s_i + Q_i$ (see Proposition 5.1 of [1]).

The demand during time interval $[t-t_i,t)$ is Poisson distributed with mean $m_i t_i$, and thus has the same distribution as the pipeline stock X_i in the basic model. Hence, $OH_i(t) - BO_i(t)$ is equal to $s_i + U_i - X_i$, where U_i is a uniformly distributed random variable on $\{1,\ldots,Q_i\}$. For the on-hand stock and the number of backorders in steady state, we then obtain:

$$\begin{aligned} OH_i &= (s_i + U_i - X_i)^+, \\ BO_i &= (X_i - (s_i + U_i))^+. \end{aligned}$$

These expressions generalize the expressions (2.4)–(2.5) of the basic model. The rest of the analysis goes along similar lines as for the basic model. In particular, the mean number of backorders is now denoted by $EBO_i(s_i)$ and may be shown to be decreasing and convex for $s_i \geq -1$.

For the model with emergency shipments, it is less easy to incorporate a fixed batch size. Emergency shipments lead to lost sales for the inventory of spare parts, and, generally spoken, lost sales models are much harder to analyze than backordering models (see also Bijvank and Vis [2]). The two key properties as described above for the backordering case, do not hold for the lost sales case. Hence, for the lost sales case, one has to rely on approximate evaluation methods; see [2] for further references.

2.10.4 Criticality

So far, we have assumed that all components are equally critical, i.e., a delay of x hours/days in fulfilling a spare parts demand is equally bad for all SKU's. When considering system availability, we assume that each component is critical. That is, an entire system goes down when a component has failed and can not be immediately replaced by a spare part. The reality in practice is often more sophisticated. The criticality of a component is related to the consequences for the system and system output if that component is not replaced immediately. These consequences may depend on the failure mode of a failed part, the position of a component in the system (an SKU may occur at multiple places in a system, and the criticality may differ per position), the level of redundancy per position, and so on. A good way to address these factors is to go back to the reliability data of a system and to incorporate the above factors in the modeling. This has been done by Van Jaarsveld and Dekker [21].

Generally, it appears to be difficult to quantify the criticality of all components. But, suppose that this has been done. Then it is possible to classify components in multiple criticality classes, and next a service target may be specified per class or one adds weights and optimizes under weighted service level constraints.

2.11 Concluding Remarks

A single-item version of the basic model in this chapter was formulated for the first time in 1966 by Feeney and Sherbrooke [5], who also discussed extensions to compound Poisson demand processes and the emergency shipments case (also denoted as the lost sales case). Shortly later, Sherbrooke [17] extended this model to a multi-item distribution system with one central warehouse and multiple local warehouse. This was the so-called METRIC model, which we will discuss extensively in Chap. 6. For a heuristic optimization of basestock levels, a so-called marginal analysis was introduced, which is like the greedy algorithm as formulated in this chapter. The paper by Sherbrooke led to a big stream of papers on spare parts models.

A better understanding for the marginal analysis of Sherbrooke [17] and the quality of its solutions was developed later; see e.g. Sherbrooke [18] and Wong et al. [22]. In Sect. 2.12 of Sherbrooke [18], a justification for the marginal analysis is given for the single-location model. However, a link with efficient solutions for a corresponding multi-objective programming problem and a real proof that it leads to optimal solutions for specific target values for the aggregate mean number of backorders (cf. Sect. 2.4.2) are absent in those works. To the best of our knowledge, that link and proof were given for the first time in Van Houtum and Hoen [20].

Differences between the system and item approach, as presented in Sect. 2.6, were studied by multiple authors; see e.g. Rustenburg et al. [16] and Thonemann et al. [19]. In the latter paper, it has been shown that cost differences between the SKU's are the main factor to lead to large cost differences between the system and item approach.

Problems

2.1. Consider a single warehouse for which the basic model of Sect. 2.2 applies. We have $|I| = 3$ SKU's. The data for the SKU's are as follows:

i	m_i (per month)	t_i (months)	c_i^{a} (Euros)
1	1.0	1.0	500
2	0.4	1.5	1,400
3	0.2	2.0	4,000

The maximum level for the aggregate mean number of backorders is 0.2.

(a) Compute the probabilities $P\{X_i = x\}$ and $P\{X_i \le x\}$ for $i = 1,2,3$ and $0 \le x \le 6$. Hint: Use the recursion in (2.8) for the computation of the probabilities $P\{X_i = x\}$.

(b) Apply Algorithm 2.1 to generate efficient solutions for Problem (Q). (Use an appropriate stop criterium.)
(c) What is the first solution $\mathbf{S}$ generated under (b) for which the aggregate mean number of backorders is at most 0.2? What are the average costs under this solution?

2.2. Consider Problem 2.1, and suppose that the item approach with $EBO^{\text{obj}} = 0.2$ is applied to generate a feasible solution. Denote this solution as $\mathbf{S}^{\text{item}}$.

(a) Determine $\mathbf{S}^{\text{item}}$, $EBO(\mathbf{S}^{\text{item}})$, and $C(\mathbf{S}^{\text{item}})$.
(b) What is the costs difference with the solution obtained under part (c) of Problem 2.1?
(c) Consider the efficient solutions obtained via the system approach, i.e., via Algorithm 2.1; see also part (b) of Problem 2.1. What is the cheapest solution $\mathbf{S}$ among them with $EBO(\mathbf{S}) \leq EBO(\mathbf{S}^{\text{item}})$, and how large is the costs difference? What is the solution $\mathbf{S}$ among them with the lowest aggregate mean number of backorders and costs $C(\mathbf{S}) \leq C(\mathbf{S}^{\text{item}})$, and how large is the difference in the aggregate mean number of backorders when comparing this solution $\mathbf{S}$ to $\mathbf{S}^{\text{item}}$?

2.3. Consider Problem 2.1. We are interested in additional performance measures for the solution $\mathbf{S}$ obtained under part (c) of that problem.

(a) Determine the aggregate mean waiting time $W(\mathbf{S})$ under solution $\mathbf{S}$.
(b) What is the aggregate fill rate $\beta(\mathbf{S})$ under solution $\mathbf{S}$?
(c) A fraction $1-\beta(\mathbf{S})$ of all demands is backordered, and they are satisfied after a certain delay. What is the average delay for an arbitrary backordered demand?

2.4. Consider the problem as described in Sect. 2.7.3. First, prove that

$$\Delta^2 PBO_i(S_i) = \left(1 - \frac{m_i t_i}{S_i + 2}\right) P\{X_i = S_i + 1\}, \qquad S_i \in \mathbb{N}_0.$$

Next, use this equation to prove that $PBO_i(S_i)$ is convex for $S_i \geq \max\{\lceil m_i t_i - 2\rceil, 0\}$.

2.5. Consider Problem 2.1, but suppose that one has already certain parts on stock: 1 part of SKU 1, 2 parts of SKU 2, and 3 parts of SKU 3. The parts are bought from an external supplier and it is not possible to sell parts back to the supplier. The requirement for the aggregate mean number of backorders remains the same.

(a) Problem (Q) as described for the standard problem of Sect. 2.2 becomes slightly different. Give the adapted problem formulation.
(b) Formulate a greedy algorithm that gives efficient solutions for the adapted Problem (Q).
(c) Apply the algorithm formulated under (b).

(d) What is the first solution $\mathbf{S}$ generated under (c) for which the aggregate mean number of backorders is at most 0.2? What are the average costs under this solution?

2.6. Consider Problem 2.5, but now assume that parts can be sold back to the supplier for 50 % of the selling price. Answer the same questions as for Problem 2.5.

2.7. Consider the extension of the basic model to the case with emergency shipments as described in Sect. 2.9. In the analysis of this extended model, one uses the property that the Erlang loss probability is strictly convex and decreasing as a function of the number of servers. For an Erlang loss system with $c \in \mathbb{N}_0$ servers and offered load $\rho > 0$, the Erlang loss probability is given by:

$$L(c,\rho) = \frac{\rho^c/c!}{\sum_{x=0}^{c} \rho^x/x!}.$$

Prove that $L(c,\rho)$ is strictly decreasing as a function of $c \in \mathbb{N}_0$.

2.8. Consider the problem of Sect. 2.9. Assume that the replenishment leadtimes are exponentially distributed (i.e., we consider a special case). Determine the steady-state distribution for the number of parts on order of an SKU i via a Markov analysis. Verify the correctness of (2.15) for this special case.

2.9. Consider the generalization of the basic model to the use of given batch sizes for the inventory control of all SKU's; see Sect. 2.10.3. For each SKU $i \in I$, one then follows an (s_i, Q_i)-policy, where Q_i represents the given batch size for SKU i and s_i (≥ -1) is the reorder level. The mean number of backorders is denoted by $EBO_i(s_i)$. Prove that $EBO_i(s_i)$ is decreasing and convex for $s_i \geq -1$.

2.10. Consider a company with its own maintenance department and a own spare parts stock. The spare parts are used for corrective maintenance at a group of machines. For the inventory control, the company wants to use a system approach. The service measure that they work with is the aggregate fill rate, cf. Sect. 2.7.4.

To learn more about how the system approach works, the company wants to consider a set of three representative spare parts. The data for these parts are as follows:

$$m_1 = 15,\ m_2 = 5,\ m_3 = 1 \ \text{ (in demands per year)},$$
$$t_1 = t_2 = t_3 = \frac{1}{6} \ \text{ (in years)},$$
$$c_1^{\mathrm{a}} = 300,\ c_2^{\mathrm{a}} = 800,\ c_3^{\mathrm{a}} = 5{,}000 \ \text{ (in Euros)}.$$

(a) Apply the greedy algorithm of Sect. 2.7.4 to generate efficient solutions for $C(\mathbf{S})$ and $\beta(\mathbf{S})$. Plot these solutions in a figure. List the solutions also in a table. What is now the cheapest solution with an aggregate fill rate of at least $\beta^{\mathrm{obj}} = 0.98$?

The company is not willing to apply a pure form of the system approach. They want to avoid too high basestock levels for cheap parts, because there is always inaccuracy in estimating demand rates and then a too high basestock level may lead to dead stock in the future. For each SKU, they want a fill rate of at most 0.998. In addition, they do not want too low basestock levels for expensive SKU's, because that may hinder acceptance of the use of the system approach. For each SKU, they want a fill rate of at least 0.90.

(b) With these extra constraints, one obtains a variant of Problem (Q′) of Sect. 2.7.4. Formulate this variant.
(c) For this variant of Problem (Q′), one can again formulate a greedy algorithm to generate efficient solutions. Formulate such a greedy algorithm and explain why it will generate efficient solutions.
(d) Apply the greedy algorithm of (c), and plot the generated solutions in the same figure as the solutions obtained under (a). List the solutions also in a table. What is now the cheapest solution with an aggregate fill rate of at least $\beta^{\text{obj}} = 0.98$? How large is the increase in costs in comparison to the solution obtained under (a)?

References

1. Axsäter, S.: Inventory Control, 2nd edn. Springer, New York (2006)
2. Bijvank, M., Vis, I.F.A.: Lost-sales inventory theory: a review. Eur. J. Oper. Res. **215**, 1–13 (2011)
3. Dantzig, G.B., Wolfe, P.: Decomposition principle for linear programs. Oper. Res. **8**, 101–111 (1960)
4. Everett, H. III: Generalized lagrange multiplier method for solving problems of optimum allocation of resources. Oper. Res. **11**, 399–417 (1963)
5. Feeney, G.J., Sherbrooke, C.C.: The $(s-1,s)$ inventory policy under compound Poisson demand. Manag. Sci. **12**, 391–411 (1966)
6. Fisher, M.L.: The Lagrangian relaxation method for solving integer programming problems. Manag. Sci. **27**, 1–18 (1981)
7. Fisher, M.L.: An application oriented guide to Lagrangian relaxation. Interfaces **15**, 10–21 (1985)
8. Fox, B.: Discrete optimization via marginal analysis. Manag. Sci. **13**, 210–216 (1966)
9. Karush, W.: A queueing model for an inventory problem. Oper. Res. **5**, 693–703 (1957)
10. Kranenburg, A.A.: Spare parts inventory control under system availability constraints. Ph.D. thesis, Eindhoven University of Technology (2006). http://w3.tue.nl/en/services/library/digilib/publications_from_tue/dissertations/
11. Kranenburg, A.A., Van Houtum, G.J.: Cost optimization in the $(S-1,S)$ lost sales inventory model with multiple demand classes. OR Lett. **35**, 493–502 (2007)
12. Little, J.D.C.: A proof of the queuing formula $L = \lambda W$. Oper. Res. **9**, 383–387 (1961)
13. Palm, C.: Analysis of the Erlang traffic formula for busy-signal arrangements. Ericsson Technics **4**, 204–212 (1938)

14. Porteus, E.L.: Foundations of Stochastic Inventory Theory. Stanford University Press, Stanford (2002)
15. Rustenburg, W.D.: A system approach to budget-constrained spare parts management. Ph.D. thesis, Eindhoven University of Technology (2000). http://w3.tue.nl/en/services/library/digilib/publications_from_tue/dissertations/
16. Rustenburg, W.D., Van Houtum, G.J., Zijm, W.H.M.: Spare parts management at complex technology-based organizations: an agenda for research. Int. J. Prod. Econ. **71**, 177–193 (2001)
17. Sherbrooke, C.C.: METRIC: a multi-echelon technique for recoverable item control. Oper. Res. **16**, 122–141 (1968)
18. Sherbrooke, C.C.: Optimal Inventory Modeling of Systems: Multi-echelon Techniques. Kluwer Academic, Boston/Dordrecht/London (2004)
19. Thonemann, U.W., Brown, A.O., Hausman, W.H.: Easy quantification of improved spare parts inventory policies. Manag. Sci. **48**, 1213–1225 (2002)
20. Van Houtum, G.J., Hoen, K.M.R.: Single-location, multi-item inventory models for spare parts. Lecture notes, Eindhoven University of Technology (2008)
21. Van Jaarsveld, W., Dekker, R.,: Spare parts stock control for redundant systems using reliability centered maintenance data. Reliab. Eng. Syst. Saf. **96**, 1576–1586 (2011)
22. Wong, H., Kranenburg, A.A., Van Houtum, G.J., Cattrysse, D.: Efficient heuristics for two-echelon spare parts inventory systems with an aggregate mean waiting time constraint per local warehouse. OR Spectr. **29**, 699–722 (2007)

Springer

http://www.springer.com/978-1-4899-7608-6

Spare Parts Inventory Control under System Availability Constraints
van Houtum, G.-J.; Kranenburg, B.
2015, XV, 215 p. 23 illus., 7 illus. in color., Hardcover
ISBN: 978-1-4899-7608-6