Econometric Institute Reports EI 2002 - 43

A decomposition approach to a stochastic model for supply-and-return network design

Ovidiu Listes

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

Abstract

This paper presents a generic stochastic model for the design of networks comprising both supply and return channels, organized in a closed loop system. Such situations are typical for manufacturing/re-manufacturing type of systems in reverse logistics. The model accounts for a number of alternative scenarios, which may be constructed based on critical levels of design parameters such as demand or returns. We propose a decomposition approach for this model based on the branch and cut procedure known as the integer L-shaped method. Computational results show a consistent performance efficiency of the method for the addressed location problem. The stochastic solutions obtained in a numerical setting generate a significant improvement in terms of average performance over the individual scenario solutions. A solution methodology as presented here can contribute to overcoming notorious challenges of stochastic network design models, such as increased problem sizes and computational difficulty.

Keywords: re-manufacturing, location, uncertainty, decomposition, integer L-shaped

1 Introduction

Multiple case studies on reverse logistics networks propose deterministic mixed integer linear facility location models in order to support the network design decisions. Examples of such cases can be found in a review paper by Fleischmann et al.(2000), which also provides a thorough analysis of network design issues in the context of recovery networks. This analysis points out that a high level of uncertainty is characteristic for product recovery management in general, while the available case studies support this vision with respect to network design issues in particular. The availability of used products on the disposer market as well as the demand for recovered products are given as examples of major uncertain factors. Despite the well identified sources of uncertainty, the authors note that basically such uncertain factors are not included explicitly in any of the analysed cases.

In Realff et al. (2000) a robust approach extension of a previously considered case (see Ammons et al. (1999)) on recycling carpet in US is proposed. In this approach nine scenarios are identified and a solution is sought which minimizes the maximum deviation to the optimal objective values of the individual scenarios. The authors conclude that the robust solution performs well over the set of scenarios and suggest the extension of the approach to different other situations with similar system features. However, since solution times ranging from half an hour to nine hours had been recorded, the need was stated for the solution approach to be improved in order to be able to address larger models in reasonable amounts of time. The authors suggest that, given the network structures embedded in the model and the limited interaction of scenarios, a traditional decomposition approach may be applicable.

Listes and Dekker (2001) have also considered a stochastic programming based approach to an illustrative case study on recycling sand from demolition waste in The Netherlands, previously reported by Barros et al. (1998). The use of the extensive forms for the two-stage stochastic models incurred comparable running times as those reported in the carpet recycling case mentioned above. Moreover, a large increase of computational time was recorded when testing a three stage approach model formulations which accounts for a gradual revealing of information with respect to the uncertain factors. Hence, based on this own experience the authors have stated in their turn that there is need for specialized algorithms capable to more efficiently tackle such models and to generate solutions in significantly lower running times.

Laporte et al. (1994) present an application of the method known in stochastic programming as the integer L-shaped method to a facility location model with stochastic demands. The proposed model aims to locate one type of facilities and to allocate them to a number of given customers, whose demands are described in terms of normal distributions. The proposed procedure was tested and performed well on a large number of instances. In the implementation though some simplifying assumptions were made with respect to some of the model parameters. The references included in this paper may further provide the interested reader with more extensive discussions on theoretical developments in stochastic location problems.

The aim of this paper is set up a generic stochastic model for the design of integral networks (i.e. comprising return channels as well) and to proposed a decomposition based approach which can tackle this model in an efficient manner. Therefore, we propose an extension of the methodology in Laporte et al. (1994) to the design of networks comprising both supply and return channels organized in a closed loop system. Thus, we show that such a stochastic modelling arising from rather theoretical considerations related to facility location problems may be naturally extended in order to represent a situation of more practical interest within the context of reverse logistics and integral supply chains. Our main purpose is to comply with the above mentioned requirements for more efficient solution methods for recovery networks design models which take the uncertainty explicitly into account. Consequently we test the proposed methodology on a number of model instances in a numerical setting. Given the strategic nature of the considered problem as well as a rather rough division of available information, typical for more practical considerations, also in application presented here we stick to the representation of uncertainty by some discrete alternative scenarios.

This paper is organized as follows. In Section 2 we present the model elements and discuss the two-stage formulation to be considered. A detailed description of the decomposition approach is given in Section 3. It includes both identifying the relevant elements related to the decomposition of the linear relaxation of the model as well as the description of the actual integer L-shaped based approach. The idea is to retrieve all the important arguments which are then referred to in the implementation discussion in Section 4. Computational results, including the analysis of procedure efficiency and the detailed discussion of one application instance, are subsequently presented in Section 5. Potential extensions of the model are briefly mentioned in Section 6. Finally, in Section 7 we make a short summary of our findings and present our documented conclusions.

2 A generic supply-and-return network model

Let $j = 1, 2, ..., J$ denote the markets to be served by the network, $i = 1, 2, ..., I$ the potential plant sites and $k = 1, 2, ..., K$ the potential facility sites. Let $s = 1, 2, ..., S$ represent a number of scenarios (based on the amounts of demand and return at markets). The decision variables are defined as follows:

κ_i	binary variable equal to 1 if a plant is located at site i and equal to 0 otherwise;
κ	$= (\kappa_1, ..., \kappa_I)$;
ϵ_k	binary variable equal to 1 if a facility is located at site i and equal to 0 otherwise;
ϵ	$= (\epsilon_1, ..., \epsilon_K)$;
x_{ij}	binary variable equal to 1 if a transportation link is established between plant i and market j;
x	$= (x_{11}, ..., x_{IJ})$;
y_{jk}	binary variable equal to 1 if a transportation link is established between market j and facility k;
y	$= (y_{11}, ..., y_{JK})$;
z_{ki}	binary variable equal to 1 if a transportation link is established between facility k and plant i;
z	$= (z_{11}, ..., z_{KI})$;
$u_{ij}(s)$	quantity delivered from plant i to market j in scenario s;
$u(s)$	$= (u_{11}(s), ..., u_{IJ}(s))$;
$v_{jk}(s)$	quantity delivered from market j to facility k in scenario s;
$v(s)$	$= (v_{11}(s), ..., v_{JK}(s))$;
$w_{ki}(s)$	quantity delivered from facility k to plant i in scenario s.
$w(s)$	$= (w_{11}(s), ..., w_{KI}(s))$.

The model parameters are defined as follows:

p_s	probability (weight) of scenario s;
$d_j(s)$	demand at market j in scenario s;
$r_j(s)$	returns from market j in scenario s;
b	average recovery fraction;
a_i	fixed costs (per planning period) of opening a plant at site i;
a	$= (a_1, ..., a_I)$;
f_k	fixed costs (per planning period) of opening a facility at site k;
f	$= (f_1, ..., f_K)$;
c_{ij}	average costs (per planning period) of operating a transportation service between plant i and market j;
c	$= (c_{11}, ..., c_{IJ})$;
e_{jk}	average costs (per planning period) of operating a transportation service between market j and facility k;
e	$= (e_{11}, ..., e_{JK})$;

g_{ki}	average costs (per planning period) of operating a transportation service between facility k and plant i;
g	$= (g_{11}, ..., g_{KI})$;
$prod_i$	variable costs of producing one unit at plant i;
$price_j$	price of one unit at market j;
$test_k$	costs of testing one unit at facility k;
$repr_i$	variable costs of reprocessing one unit at plant i;
t^0_{ij}	transportation costs for one unit from plant i to market j;
t^1_{jk}	transportation costs for one unit from market j to facility k;
t^2_{ki}	transportation costs for one unit from facility k to plant i;
q_j	penalty costs for not collecting one unit of returns at market j;
h_k	disposal costs of one unit at facility k;
P_i	(maximum) capacity of a plant at site i;
R_k	(maximum) capacity of a facility at site k;
C_{ki}	$= \min(R_k, P_i)$

Given the notation above, the network structure can be schematically illustrated as in Figure 1.

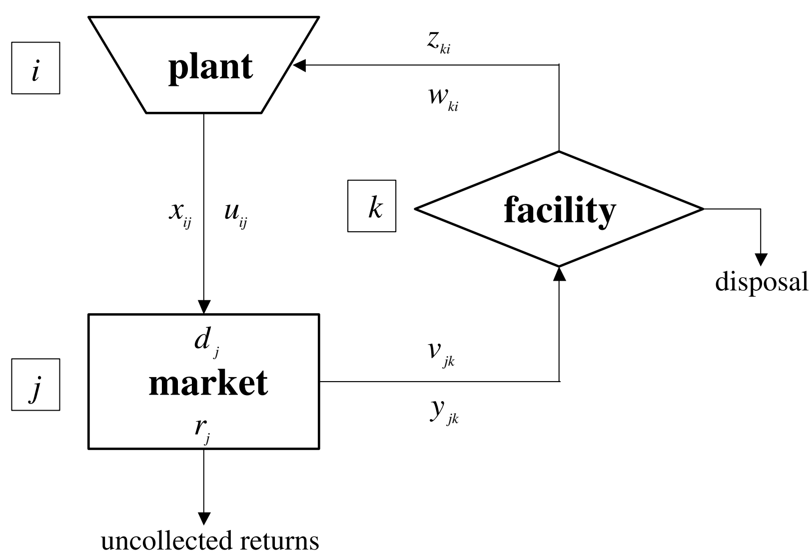

Figure 1: Structure of supply-and-return network model

We assume that all costs and revenues are defined for the same planning period. They can be as well interpreted as average values over that period of time. The entries of the vectors c, e, g represent the fixed costs only for establishing a transportation link between two sites and are not related to the actual amount to be transported between the corresponding sites. The amount related figures incorporated into the model are additionally defined. Furthermore, we assume that there are some sole servicing requirements, namely that demand of each market can be satisfied from at most one plant, returns from each market are directed to at most one facility and each facility directs return flow to at most one plant. These requirements lead to a typical situation with a large number of markets, a low number of facilities and an even lower number of plants.

Before actually formulating the two-stage stochastic model we remark that for a given network configuration (that is for fixed κ and ϵ) and a given scenario s the objective function to be maximized in the second stage is to be expressed as

$$\sum_{i=1}^{I}\Big[\sum_{j=1}^{J}(price_j - t^0_{ij})u_{ij}(s) - prod_i\Big[\sum_{j=1}^{J}u_{ij}(s) - \sum_{k=1}^{K}w_{ki}(s)\Big]\Big]+$$

(*sales revenues, transportation costs from plants to markets and production costs for new products*)

$$\sum_{j=1}^{J}\sum_{k=1}^{K}(-test_k - t^1_{jk})v_{jk}(s) - \sum_{j=1}^{J}q_j\big[r_j(s) - \sum_{k=1}^{K}v_{jk}(s)\big]+$$

(*testing costs, transportation costs from markets to facilities and penalty costs for not collected returns*)

$$\sum_{k=1}^{K}\sum_{i=1}^{I}(-repr_i - t^2_{ki})w_{ki} - \sum_{k=1}^{K}h_k\Big[\sum_{j=1}^{J}v_{jk}(s) - \sum_{i=1}^{I}w_{ki}(s)\Big]$$

(*reprocessing costs, transportation costs from facilities to plants and disposal costs*)

Grouping the coefficients by decision variables yields the following expression

$$\sum_{i=1}^{I}\sum_{j=1}^{J}(price_j - prod_i - t^0_{ij})u_{ij}(s)+$$

$$\sum_{j=1}^{J}\sum_{k=1}^{K}(q_j - h_k - test_k - t^1_{jk})v_{jk}(s) - \sum_{j=1}^{J}q_j r_j(s)+$$

$$\sum_{k=1}^{K}\sum_{i=1}^{I}(prod_i - repr_i + h_k - t^2_{ki})w_{ki}(s)$$

For each s, the term $\sum_{j=1}^{J}q_j r_j(s)$ being constant can be dropped from the second stage objective function (and re-added after optimizing the second stage for each scenario). For simplicity we make the following further notations

$$\rho_{ij} = price_j - prod_i - t^0_{ij}$$

(ρ_{ij} is net revenue per unit transported from plant i to market j)

$$\mu_{jk} = q_j - h_k - test_k - t^1_{jk}$$

(μ_{jk} is the net gain per unit transported from market j to facility k)

$$\lambda_{ki} = prod_i - repr_i + h_k - t^2_{ki}$$

(λ_{ki} is the net gain per unit transported from facility k to plant i)

For every x, y, z and every scenario s we denote the second stage recourse function by $Q(x, y, z, s)$. The expected recourse function is denoted by $\mathcal{Q}(x, y, z)$, that is $\mathcal{Q}(x, y, z) = \mathbb{E}_s[Q(x, y, z, s)] = \sum_{s=1}^{S} p_s\, Q(x, y, z, s)$ where $\mathbb{E}_s$ denotes the expectation with respect to s. These functions are defined in terms of x, y, z only (and not of κ and ϵ) as it will be explained after formulating the model.

Given these preparations, the problem under consideration is then to

$$\begin{aligned} \min \quad & \sum_{i=1}^{I} a_i\, \kappa_i + \sum_{k=1}^{K} f_k\, \epsilon_k + \sum_{i=1}^{I}\sum_{j=1}^{J} c_{ij} x_{ij} + \sum_{j=1}^{J}\sum_{k=1}^{K} e_{jk} y_{jk} + \sum_{k=1}^{K}\sum_{i=1}^{I} g_{ki} z_{ki} - \\ & - \mathcal{Q}(x, y, z) \end{aligned} \tag{1}$$

subject to

$$x_{ij} \le \kappa_i \qquad \forall\, i,\ \forall\, j \tag{2}$$
$$y_{jk} \le \epsilon_k \qquad \forall\, j,\ \forall\, k \tag{3}$$
$$z_{ki} \le \kappa_i \qquad \forall\, k,\ \forall\, i \tag{4}$$
$$z_{ki} \le \epsilon_k \qquad \forall\, k,\ \forall\, i \tag{5}$$
$$\sum_{i=1}^{I} x_{ij} \le 1 \qquad \forall\, j \tag{6}$$
$$\sum_{k=1}^{K} y_{jk} \le 1 \qquad \forall\, j \tag{7}$$
$$\sum_{i=1}^{I} z_{ki} \le 1 \qquad \forall\, k \tag{8}$$
$$\kappa_i,\ \epsilon_k,\ x_{ij},\ y_{jk},\ z_{ki} \in \{0, 1\} \qquad \forall\, i,\ \forall\, j,\ \forall\, k \tag{9}$$

where the recourse function $Q(x, y, z, s)$ is defined by

$$Q(x, y, z, s) = \max \sum_{i=1}^{I}\sum_{j=1}^{J} \rho_{ij}\, u_{ij}(s) + \sum_{j=1}^{J}\sum_{k=1}^{K} \mu_{jk}\, v_{jk}(s) + \sum_{k=1}^{K}\sum_{i=1}^{I} \lambda_{ki}\, w_{ki}(s) \tag{10}$$

subject to

$$u_{ij}(s) \le d_j(s)\, x_{ij} \qquad \forall\, i,\ \forall\, j \tag{11}$$
$$\sum_{j=1}^{J} u_{ij}(s) \le P_i \qquad \forall\, i \tag{12}$$
$$v_{jk}(s) \le r_j(s)\, y_{jk} \qquad \forall\, j,\ \forall\, k \tag{13}$$
$$\sum_{j=1}^{J} v_{jk}(s) \le R_k \qquad \forall\, k \tag{14}$$
$$w_{ki}(s) \le C_{ki} z_{ki} \qquad \forall\, k,\ \forall\, i \tag{15}$$

$$\sum_{i=1}^{I} w_{ki}(s) \;\leq\; b \sum_{j=1}^{J} v_{jk}(s) \qquad\qquad \forall\ k \tag{16}$$

$$\sum_{k=1}^{K} w_{ki}(s) \;\leq\; \sum_{j=1}^{J} u_{ij}(s) \qquad\qquad \forall\ i \tag{17}$$

$$u_{ij}(s),\ v_{jk}(s),\ w_{ki}(s) \;\geq\; 0 \qquad\qquad \forall\ i,\ \forall\ j,\ \forall\ k \tag{18}$$

By the first stage problem we will refer to the problem defined by (1) – (9), while the second stage problem refers to the problem defined by (10) – (18). In the objective of the first stage problem, the first and the second terms represent the fixed costs for opening plants respectively facilities, the third, fourth and fifth terms denote the average costs (per planning period) of operating transportation links between the corresponding sites, while the last term represents minus the expected revenue resulting from operating the network configuration and the links given by the first stage variables. Constraints (2) – (5) impose that links can be established only using open plants or facilities, while constraints (6) – (8) are the sole servicing requirements mentioned before. The second stage objective computes for each scenario s the maximum net revenue under restrictions owing to the existing demands (constraints (11)), existing returns (constraints (13)) and available capacity (constraints (12),(14),(15)). Moreover balance constraints (16) and (17) at facility and plant level respectively are added. At the facility level the excess inbound amount corresponds to the part of returns which is disposed of. Similarly, the returns from a facility to a plant are required not to exceed the distribution from the plant (the excess outbound corresponding to new production). From the model formulation it becomes now clear that the second stage function $\mathcal{Q}$ may be defined as a function of x, y and z only, because in any first stage solution $\kappa_i = 1$ if and only if $x_{ij} = 1$ for one j or $z_{ki} = 1$ for one k, respectively $\epsilon_k = 1$ if and only if $y_{jk} = 1$ for one j or $z_{ki} = 1$ for one i.

3 The decomposition approach

Using the notation proposed by Laporte and Louveaux (1993), the model described in the previous section belongs to the class $B/C/D$, where B stands for binary first stage variables, C for continuous second stage variables, while D means that the model uses discrete random parameters. Moreover, the model has so-called relatively complete recourse, that is for any feasible first stage solution, the second stage problem is feasible (the recourse function is finite), since $u(s) = 0$, $v(s) = 0$, $w(s) = 0$ is always a feasible second stage solution. Such models can be tackled by a type of branch-and-cut procedure proposed by Laporte and Louveaux (1993), called the Integer L-shaped method. In the sequel we present a fairly detailed description of the arguments and the terms in which an algorithm of this type may be applied to our problem. During the exposure we will retrieve as well all the important elements which play a role in the actual implementation.

We start by noting that the first stage problem is equivalent to the following reformulation in which a new variable θ is introduced:

$$\min \sum_{i=1}^{I} a_i\,\kappa_i + \sum_{k=1}^{K} f_k\,\epsilon_k + \sum_{i=1}^{I}\sum_{j=1}^{J} c_{ij}x_{ij} + \sum_{j=1}^{J}\sum_{k=1}^{K} e_{ij}y_{jk} + \sum_{k=1}^{K}\sum_{i=1}^{I} g_{ki}z_{ki} + \theta \tag{19}$$

subject to (2), (3), (4), (5), (6), (7), (8) and

$$\theta \;\geq\; -\,\mathcal{Q}(x,y,z) \tag{20}$$

$$\kappa_i,\,\epsilon_k,\,x_{ij},\,y_{jk},\,z_{ki} \;\in\; \{0,1\} \qquad \forall\, i,\; \forall\, j,\; \forall\, k \tag{21}$$

From this formulation two main issues can be distinguished. Firstly, the inequality (20) can not be used computationally as a constraint since $\mathcal{Q}(x,y,z)$ is not defined explicitly, but only implicitly by a number of optimization problems. The classical L-shaped decomposition method (Van Slyke and Wets (1969)) provides means to relax this constraint and replace it by a number cuts, which may be gradually added during an iterative solving process. This method results from applying Benders (1962) decomposition to two-stage stochastic models and is applicable to problems with continuous variables. Consequently, the second issue arising from the formulation above is the integrality of the first-stage variables. Combining in a certain sense the L-shaped method with the well known branch-and-bound method for mixed integer problems results in the Integer L-shaped method. In the next subsection we expose first the ideas of the L-shaped method as applied to our problem. Then we return to the original problem and address its discrete nature in subsection 3.2.

3.1 Decomposing the linear relaxation of the model

Hence, we ignore for the moment the integrality constraints and allow the first stage variables to be continuous in the interval $[0;1]$. Let us denote the initial feasible set after dropping constraint (20) by

$$\mathcal{F}_0 := \{\, (\kappa,\epsilon,x,y,z,\theta) \mid (2)-(8),\; \theta\in\mathbb{R},\; \kappa_i,\,\epsilon_k,\,x_{ij},\,y_{jk},\,z_{ki}\in[0;1]\;\;\forall\, i,\forall\, j,\forall\, k\,\}$$

We want to construct a sequence of additional constraints which can define a monotonically decreasing feasible set $\mathcal{F}_1$ such that eventually the problem

$$\min\{\, a\,\kappa + f\,\epsilon + c\,x + e\,y + g\,z\; + \theta \mid (\kappa,\epsilon,x,y,z,\theta)\in\mathcal{F}_0\cap\mathcal{F}_1\,\} \tag{22}$$

yields a solution which satisfies $\theta \geq -\,\mathcal{Q}(x,y,z)$. For any particular $\mathcal{F}_1$ available at a given point during the iterative process, the problem (22) is referred to as the *current problem* (or the master problem). We will show later that there exists a lower bound θ_0 for $-\mathcal{Q}(x,y,z)$ for any feasible x,y,z. The L-shaped method for our problem can be applied along the following main steps.

Step 1. Let initially

$$\mathcal{F}_1 := \{\, (\kappa,\epsilon,x,y,z,\theta) \mid \theta\geq\theta_0,\; \kappa_i,\,\epsilon_k,\,x_{ij},\,y_{jk},\,z_{ki}\in[0;1]\;\;\forall\, i,\forall\, j,\forall\, k\,\}$$

We solve the current problem corresponding to this $\mathcal{F}_1$. Denote an optimal solution by $(\hat\kappa,\hat\epsilon,\hat x,\hat y,\hat z,\hat\theta)$.

Step 2. Consider the optimal (first-stage) solution $(\hat\kappa,\hat\epsilon,\hat x,\hat y,\hat z,s)$ resulted from the last current problem solved. Clearly in our case $Q(\hat x,\hat y,\hat z,s)$ is finite for every s, so implicitly $\mathcal{Q}(\hat x,\hat y,\hat z)$ is also finite. For general (x,y,z) we have, as before,

$$\mathcal{Q}(x,y,z) = \sum_{s=1}^{S} p_s\, Q(x,y,z,s)$$

where for every scenario s

$$Q(x,y,z,s) = \max\{\, \rho\, u(s) + \mu\, v(s) + \lambda\, w(s) \mid (11) - (17),\; u(s), v(s), w(s) \geq 0 \,\}$$

The recourse function $Q(x,y,z,s)$ can be as well expressed as the optimal value of the dual problem associated to the second stage problem in (10) - (18). More precisely, if $\alpha_{ij}(s), \phi_i(s), \beta_{jk}(s), \psi_k(s), \gamma_{ki}(s), \sigma_k(s), \tau_i(s)$ are the dual variables corresponding respectively to the constraints (11),(12),(13),(14),(15),(16),(17), then

$$\begin{aligned} Q(x,y,z,s) \;=\; \min\;\; & \sum_{i=1}^{I}\sum_{j=1}^{J} d_j(s) x_{ij}\, \alpha_{ij}(s) + \sum_{i=1}^{I} P_i\, \phi_i(s) + \\ & \sum_{j=1}^{J}\sum_{k=1}^{K} r_j(s) y_{jk}\, \beta_{jk}(s) + \sum_{k=1}^{K} R_k\, \psi_k(s) + \\ & \sum_{k=1}^{K}\sum_{i=1}^{I} C_{ki} z_{ki}\, \gamma_{ki}(s) \end{aligned} \tag{23}$$

subject to

$$\begin{aligned} \alpha_{ij}(s) + \phi_i(s) - \tau_i(s) &\geq \rho_{ij} \quad \forall\, i,\ \forall\, j && (24)\\ \beta_{jk}(s) + \psi_k(s) - b\,\sigma_k(s) &\geq \mu_{jk} \quad \forall\, j,\ \forall\, k && (25)\\ \gamma_{ki}(s) + \sigma_k(s) + \tau_i(s) &\geq \lambda_{ki} \quad \forall\, k,\ \forall\, i && (26)\\ \alpha_{ij}(s),\, \beta_{jk}(s),\, \gamma_{ki}(s),\, \phi_i(s),\, \psi_k(s),\, \sigma_k(s),\, \tau_i(s) &\geq 0 \quad \forall\, i,\ \forall\, j,\ \forall\, k && (27) \end{aligned}$$

Now, if $\hat{\theta} \geq -\,\mathcal{Q}(\hat{x},\hat{y},\hat{z})$, we are done: we stop with $(\hat{\kappa},\hat{\epsilon},\hat{x},\hat{y},\hat{z})$ being an optimal solution. If $\hat{\theta} < -\,\mathcal{Q}(\hat{x},\hat{y},\hat{z})$, then $(\hat{\kappa},\hat{\epsilon},\hat{x},\hat{y},\hat{z})$ is not optimal. In this case, for every scenario s, let $(\hat{\alpha}_{ij}(s),\, \hat{\beta}_{jk}(s),\, \hat{\gamma}_{ki}(s),\, \hat{\phi}_i(s),\, \hat{\psi}_k(s),\, \hat{\sigma}_k(s),\, \hat{\tau}_i(s))$ be an optimal dual solution of the second stage problem corresponding to $(\hat{\kappa},\hat{\epsilon},\hat{x},\hat{y},\hat{z})$. We note that the feasible set of the dual second stage problem does not depend on (x,y,z), that is, for any first stage decision the recourse function is optimized over the same feasible region. Using this argument, we can construct the following optimality cut:

$$\begin{aligned} \theta \;\geq\; -\sum_{s=1}^{S} p_s \Big[& \sum_{i=1}^{I}\sum_{j=1}^{J} d_j(s) x_{ij}\, \hat{\alpha}_{ij}(s) + \sum_{i=1}^{I} P_i\, \hat{\phi}_i(s) + \sum_{j=1}^{J}\sum_{k=1}^{K} r_j(s) y_{jk}\, \hat{\beta}_{jk}(s) + \\ & + \sum_{k=1}^{K} R_k\, \hat{\psi}_k(s) + \sum_{k=1}^{K}\sum_{i=1}^{I} C_{ki} z_{ki}\, \hat{\gamma}_{ki}(s) \Big] \end{aligned}$$

which must be satisfied by any optimal solution (x,y,z,θ), but is not satisfied by the (non-optimal) found solution $(\hat{x},\hat{y},\hat{z},\hat{\theta})$. This inequality can be re-written in the following form

$$\begin{aligned} \theta \;\geq\; & -\sum_{i=1}^{I}\sum_{j=1}^{J} \mathbb{E}_s[d_j(s)\, \hat{\alpha}_{ij}(s)]\, x_{ij} - \sum_{i=1}^{I} \mathbb{E}_s[P_i\, \hat{\phi}_i(s)] - \sum_{j=1}^{J}\sum_{k=1}^{K} \mathbb{E}_s[r_j(s)\, \hat{\beta}_{jk}(s)]\, y_{jk} \\ & - \sum_{k=1}^{K} \mathbb{E}_s[R_k\, \hat{\psi}_k(s)] - \sum_{k=1}^{K}\sum_{i=1}^{I} \mathbb{E}_s[C_{ki}\, \hat{\gamma}_{ki}(s)]\, z_{ki} \end{aligned} \tag{28}$$

where $\mathbb{E}_s$ denotes the expectation with respect to s, or concisely as

$$D\,x + E\,y + F\,z + \theta \geq \delta \tag{29}$$

where D, E, F are vectors of corresponding sizes and δ is a real value.

So if $\hat{\theta} < -\mathcal{Q}(\hat{x}, \hat{y}, \hat{z})$ then we add an optimality cut of the form (29), correspondingly redefine $\mathcal{F}_1 := \mathcal{F}_1 \cap \{\, (\kappa, \epsilon, x, y, z, \theta) \mid D\,x + E\,y + F\,z + \theta \geq \delta \,\}$ and continue with step 3.

Step 3. Solve the current problem with the updated $\mathcal{F}_1$ and return to *Step 2*.

We provide now details on the computation of the lower bound θ_0. Define $\overline{\rho}_i = \max_j \{\rho_{ij}\}$, $\tilde{\rho}_j = \max_i \{\rho_{ij}\}$, $\overline{\mu}_j = \max_k \{\mu_{jk}\}$, $\tilde{\mu}_k = \max_j \{\mu_{jk}\}$, $\tilde{\lambda}_k = \max_i \{\lambda_{ki}\}$, $\tilde{C}_k = \max_i \{C_{ki}\}$. Given the definition of $Q(x, y, z, s)$ and the constraints (12), (14), (17) we have that

$$\begin{aligned} Q(x, y, z, s) &\leq \sum_{i=1}^{I} \sum_{j=1}^{J} \overline{\rho}_i\, u_{ij}(s) + \sum_{j=1}^{J} \sum_{k=1}^{K} \tilde{\mu}_k\, v_{jk}(s) + \sum_{k=1}^{K} \sum_{i=1}^{I} \tilde{\lambda}_i\, w_{ki}(s) \\ &= \sum_{i=1}^{I} \overline{\rho}_i \big(\sum_{j=1}^{J} u_{ij}(s) \big) + \sum_{k=1}^{K} \tilde{\mu}_k \big(\sum_{j=1}^{J} v_{jk}(s) \big) + \sum_{i=1}^{I} \tilde{\lambda}_i \big(\sum_{k=1}^{K} w_{ki}(s) \big) \\ &\leq \sum_{i=1}^{I} \overline{\rho}_i P_i + \sum_{j=1}^{J} \tilde{\mu}_k R_k + \sum_{i=1}^{I} \tilde{\lambda}_i P_i \end{aligned}$$

that is, by taking expectations on both sides we can write

$$-\mathcal{Q}(x, y, z) \geq -\sum_{i=1}^{I} \overline{\rho}_i P_i - \sum_{j=1}^{J} \tilde{\mu}_k R_k - \sum_{i=1}^{I} \tilde{\lambda}_i P_i \tag{30}$$

Now using constraints (11), (13), (15) as well as (6), (7), (8) we get

$$\begin{aligned} Q(x, y, z, s) &\leq \sum_{i=1}^{I} \sum_{j=1}^{J} \rho_{ij}\, d_j(s)\, x_{ij} + \sum_{j=1}^{J} \sum_{k=1}^{K} \mu_{jk}\, r_j(s)\, y_{jk} + \sum_{k=1}^{K} \sum_{i=1}^{I} \lambda_{ki}\, C_{ki}\, z_{ki} \\ &\leq \sum_{j=1}^{J} \tilde{\rho}_j\, d_j(s) \big(\sum_{i=1}^{I} x_{ij} \big) + \sum_{j=1}^{J} \overline{\mu}_j\, r_j(s) \big(\sum_{k=1}^{K} y_{jk} \big) + \sum_{k=1}^{K} \tilde{\lambda}_k\, \tilde{C}_k \big(\sum_{i=1}^{I} z_{ki} \big) \\ &\leq \sum_{j=1}^{J} \tilde{\rho}_j\, d_j(s) + \sum_{j=1}^{J} \overline{\mu}_j\, r_j(s) + \sum_{k=1}^{K} \tilde{\lambda}_k\, \tilde{C}_k \end{aligned}$$

and taking again expectations at both sides yields

$$-\mathcal{Q}(x, y, z) \geq -\sum_{j=1}^{J} \tilde{\rho}_j\, \mathbb{E}_s[d_j(s)] - \sum_{j=1}^{J} \overline{\mu}_j\, \mathbb{E}_s[r_j(s)] - \sum_{k=1}^{K} \tilde{\lambda}_k\, \tilde{C}_k \tag{31}$$

Combining (30) and (31), we define the lower bound θ_0 to be the maximum of the two right hand sides.

Having prepared the ingredients corresponding to the L-shaped method, we return in the next subsection to our original problem where the first-stage variables are required to be integer (binary).

3.2 The integer L-shaped based algorithm

As already mentioned the idea is to combine an usual branch-and-bound scheme for the first-stage problem with the iterative cutting planes procedure of the L-shaped method described above. So we operate with a list of waiting nodes, each node in the list corresponding to a form of the current problem.

Our *current problem* states as

$$\min \sum_{i=1}^{I} a_i\,\kappa_i + \sum_{k=1}^{K} f_k\,\epsilon_k + \sum_{i=1}^{I}\sum_{j=1}^{J} c_{ij}x_{ij} + \sum_{j=1}^{J}\sum_{k=1}^{K} e_{ij}y_{jk} + \sum_{k=1}^{K}\sum_{i=1}^{I} g_{ki}z_{ki} + \theta \qquad (32)$$

subject to (2), (3), (4), (5), (6), (7), (8) and

$$\begin{aligned}
D_n\,x + E_n\,y + F_n\,z + \theta &\geq \delta_n \qquad n = 1, ..., m && (33)\\
\sum_{i=1}^{I} \overline{\rho}_i P_i\,\kappa_i + \sum_{j=1}^{J} \tilde{\mu}_k R_k\,\epsilon_k + \theta &\geq -\sum_{i=1}^{I} \tilde{\lambda}_i P_i && (34)\\
\theta &\geq \theta_0 && (35)\\
\kappa_i,\ \epsilon_k,\ x_{ij},\ y_{jk},\ z_{ki} &\in [0;1] \qquad \forall\ i,\ \forall\ j,\ \forall\ k && (36)
\end{aligned}$$

Denote by $\overline{\mathcal{Z}}$ the objective value of the best found integer first stage solution, by m the total number of optimality cuts generated up to the current point and by ν the iteration index. The procedure consists of the following steps:

Step 0. Set $\nu := 0$, $m := 0$, $\overline{\mathcal{Z}} := \infty$. The list consists of one node corresponding to the initial current problem.

Step 1. Choose a node from the list. If the list is empty, stop.

Step 2. Let $\nu := \nu + 1$. Solve the current problem and denote an optimal solution by $(\kappa^\nu, \epsilon^\nu, x^\nu, y^\nu, z^\nu, \theta^\nu)$.

Step 3. If $a\,\kappa^\nu + f\,\epsilon^\nu + c\,x^\nu + e\,y^\nu + g\,z^\nu + \theta^\nu > \overline{\mathcal{Z}}$, fathom the current node and return to Step 1.

Step 4. If there are unsatisfied integrality constraints, pick a variable with fractional value and create two new nodes corresponding to setting its value at 0 or 1. Replace the current node by the two new nodes in the list and return to Step 1.

Step 5. Using the dual formulation in (23) – (27) determine $Q(x^\nu, y^\nu, z^\nu, s)$ for every s. Along the solving, use the dual solutions to compute the vectors D_{m+1}, E_{m+1}, F_{m+1} and the value δ_{m+1} as given by (28). Compute $\mathcal{Q}(x^\nu, y^\nu, z^\nu)$. Let

$$\mathcal{Z}^\nu := a\,\kappa^\nu + f\,\epsilon^\nu + c\,x^\nu + e\,y^\nu + g\,z^\nu - \mathcal{Q}(x^\nu, y^\nu, z^\nu)$$

If $\mathcal{Z}^\nu < \overline{\mathcal{Z}}$, set $\overline{\mathcal{Z}} := \mathcal{Z}^\nu$ and retain the values $\kappa^\nu, \epsilon^\nu, x^\nu, y^\nu, z^\nu$.

Step 6. If $\theta^\nu \geq -\mathcal{Q}(x^\nu, y^\nu, z^\nu)$, fathom the current node and return to Step 1. Otherwise, impose the optimality cut $D_{m+1}\,x + E_{m+1}\,y + F_{m+1}\,z + \theta \geq \delta_{m+1}$, set $m := m + 1$ and return to Step 2.

As it can be noticed from the procedure description above, the fathoming rules here result from the combination of two methods. In particular, they differ from what is usually done in branch-and-bound, in the sense that nodes are not necessarily fathomed when integrality conditions are satisfied. Instead, an optimality cut may be further added in order to assure the extra optimality condition involving the θ variable and the expected recourse function. The procedure eventually generates an exact solution to the problem by exhausting the list of nodes to be investigated.

4 Implementation issues

The method was implemented on a Windows NT-based 933MHz Pentium III PC with 256MB RAM through a C application which makes use of the CPLEX Callable Library version 7.1 (see ILOG (2000)). Specifically we work with two LP problem objects, one for the current problem and one for the dual recourse problem. In the current problem object, the objective function (32) and the constraints (2), (3), (4), (5), (6), (7), (8), (34), (35) and (36) are maintained during computation, while the (branching) integrality constraints and the optimality cuts (33) corresponding to a node are added to the problem object upon entering that node. For the dual recourse problem, the constraints (24), (25), (26), (27) are maintained, while the coefficients of the objective function (23) are modified every time the recourse problem is solved for one scenario. The CPLEX Primal Simplex Optimizer was used for solving the current problem, since it showed enhanced ability to produce solutions with a larger number of integer values. For the dual second-stage problem we used the CPLEX Barrier LP solver, which seemed appropriate for faster evaluations of the recourse function.

The pendant nodes in the list are in fact the leaves of a branch-and-bound tree. If we concatenate all first stage decision into one vector $(\kappa, \epsilon, x, y, z)$ of dimension N, then there are $2N$ possible integrality constraints (corresponding to fixing each first stage variable at 0 or 1). So, they can be indexed from 1 to $2N$ and referred to by their indices. On the other hand, during the computation a dynamic list of optimality cuts is maintained as follows. Each optimality cut is indexed upon generation by $m+1$, where m is the total number of optimality cuts generated up to that point. Such a cut is stored as a list containing (only) its non-zero coefficients of the first stage variables and its value of the right hand side. The optimality cuts generated in one node are imposed for all descendant nodes of the search tree, but removed when backtracking. So, when a node is fathomed, all the optimality cuts generated inside that node are removed both from the current problem formulation and physically from memory. Upon removal of some optimality cuts from the physical memory, all remaining optimality cuts keep the same index which was assigned to them upon generation.

The main algorithm is implemented through a recursive procedure, which can be concisely described as follows:

procedure SolveNode(I_p, O_p, O_n, ic)
begin
  if (ic > 0) **then** $I = I_p \cup \{\text{ic}\}$;
  $O = O_p \cup O_n$;
  BuildCurrent(I, O);
  SolveCurrent;
  if $(a\,\kappa + f\,\epsilon + c\,x + e\,y + g\,z + \theta > \overline{\mathcal{Z}})$ **then** Remove(O_n);   {*fathom node*}
  else begin

CheckIntegrality;
if (exists *var* fractional)
  then begin
    nic1 = IndexIntConstr($var = 1$);
    SolveNode(I, O, $\emptyset$, nic1);
    nic0 = IndexIntConstr($var = 0$);
    SolveNode(I, O, $\emptyset$, nic0);
    Remove(O_n); {*fathom node*}
  end;
  else begin
    Perform **Step 5** described above;
    Compute D_{m+1}, E_{m+1}, F_{m+1} and δ_{m+1};
    if ($\theta \geq -\mathcal{Q}(x, y, z)$) **then** Remove($O_n$); {*fathom node*}
      else begin
        $O_n = O_n \cup$ {opt cut $m + 1$};
        $m = m + 1$;
        SolveNode(I, O_p, O_n, 0);
      end;
  end;
end;
end;

where the symbols not yet defined represent the following:

I_p	set of integrality constraints inherited from the parent node;
ic	index of the integrality constraint to be imposed inside the node;
O_p	set of optimality cuts inherited from the parent node;
O_n	set of optimality cuts previously generated inside the same node;
I	set of integrality constraints for the current problem;
O	set of optimality constraints for the current problem.

The algorithm was executed by giving branching priorities to the first stage binary variables in the following manner. First we branched on the κ_i variables, which correspond to the opening decisions for plants. A second priority level was assigned to the ϵ_k variables, corresponding to the opening decisions for testing centers. Finally, a third priority level was associated to all the transportation links variables x_{ij}, y_{jk}, z_{ki}. As already suggested in the procedure described above, the next node to process when backtracking was chosen as the most recently created node. At each selected node, within each priority group, the next variable to branch on was chosen as the variable belonging to the group, which had the largest fractional value in the current problem solution from that node. The upwards variable branching direction was chosen first in order to attempt a more aggressive strategy for determining feasible network configurations.

5 Computational results

We used the model described above to illustrate the analysis of a supply-and-return network in a numerical setting. Specifically, this example follows in broad terms the direction of a case study on (re)manufacturing of electronic equipment as presented in Fleischmann

(2000). So we assume that, owing to take-back obligations of the original manufacturer, used equipment should be collected from customers for remanufacturing or proper disposal. A substantial penalty is imposed for the returns which are not collected. Due to a strict legislation prescribing re-processing to the largest extent possible, no collected equipment may be disposed at the collection sites. Instead, all collected equipment has to be shipped to some center for inspection, disassembly and testing (in this context, the term 'facility' will denote a disassembly/testing center). In order to allow for remanufacturing, the used equipment must meet certain quality standards. The equipment failing such quality tests is disposed of at the local level, while the acceptable equipment is shipped to the original manufacturing plants, which are assumed to accommodate also the remanufacturing operation. After recovering the intrinsic left value, the remanufactured equipment is re-sold in the same markets as the new products. Apparent drawbacks of this setting are the implicit assumptions that the plant capacity may be split for production and remanufacturing at various amount levels, while the remanufactured products are considered equal substitutes for the new products in the market. Clearly, the remedy would be to consider a multicommodity network flow model, where different quality streams and different corresponding capacities are distinguished. Moreover, an intermediate warehouse level may be considered for the distribution in the forward supply chain instead of supplying markets directly from the plants. However, we feel that these elements would not change the essence of our investigations. Therefore we stick here to the above assumptions for the sake of clarity of our exposition.

However, in contrast with Fleishmann (2000), we consider also some different elements in our setting. Firstly, we assume that plants may have two possible capacities P^1 and P^2 and respectively that disassembly centers may have two possible capacities R^1 and R^2. More precisely, we incorporate a notion of 'economy of scale' for the processing capabilities and assume that P^1 can be doubled to P^2 at about 65% more fixed costs, while R^1 can be doubled to R^2 at 50% more fixed costs. Secondly, in our setting the volume dependent transportation costs are - relatively to the fixed costs of facilities - slightly higher. This may stimulate somewhat the choice of more decentralized network configurations. Thirdly, we consider average costs c_{ij}, e_{jk}, g_{ki} related only to the existence of transportation links. These costs are assumed proportional to the distance between the corresponding sites. Finally, our model does consider penalty costs for not collecting (part of) returns from the markets, but does not consider penalties for not satisfying (part of) demand of the markets. Thus, we assume that demand at markets is seen only as a profit opportunity and not as an obligation by the network investor. The parameters settings for this example are summarized in Table 1.

We consider three problem settings based on a total number of potential markets of $J = 60$, $J = 80$ and respectively $J = 100$. The market locations are generated in the square $S = [0; 4000] \times [0; 4000]$ as described in the sequel. In each problem setting, three demand scenarios are considered, where demands grow both in amount and in locational diversity at a low (L), medium (M) and high (H) level. We accomplish this by increasing the number of markets as well as their demand limits at successive levels in the following manner. First, we randomly generate a low number of markets l according to a uniform distribution in the square $S_0 = [1000; 3000] \times [1000; 3000]$, where demand of each market (in thousand pieces) is randomly generated according to a uniform distribution in $[15; 25]$. These values define the low demand scenario L. Then, we keep the l sites from scenario

Description	Value
Capacity plant P^1	600,000
Capacity plant P^2	1,200,000
Capacity facility R^1	150,000
Capacity facility R^2	300,000
Fixed costs plant (per annum) P^1	9,000,000
Fixed costs plant (per annum) P^2	15,000,000
Fixed costs facility (per annum) R^1	1,200,000
Fixed costs facility (per annum) R^2	1,800,000
Average costs of a link plant – market (per km per annum)	800
Average costs of a link market – facility (per km per annum)	500
Average costs of a link facility – plant (per km per annum)	300
Transportation costs (per prod per km)	
plant $\rightarrow$ market	0.040
market $\rightarrow$ facility	0.025
facility $\rightarrow$ plant	0.015
Penalty costs for not collected returns (per prod)	60
Recovery fraction	0.6
Cost savings = production – reprocessing (per prod)	150
Disassembly/Testing costs (per prod)	35
Disposal costs (per prod)	10

Table 1: Parameter settings for the numerical results

L and besides them we randomly generate a further number of $m-l$ markets, uniformly in $S_1 = [500; 3500] \times [500; 3500] \setminus S_0$ (up to a medium number of markets m). Now, for each of the m sites a demand value (in thousand pieces) is randomly generated according to a uniform distribution in $[20; 40]$. These figures result in the medium demand scenario M. Finally, we keep the m sites from scenario M, generate some $h - m = J - m$ more markets uniformly in $S_2 = S \setminus (S_0 \cup S_1)$ (up to a high number of markets $h = J$) and randomly generate a demand figure (in thousand pieces) uniformly in $[25; 55]$ for each of the h markets. The last demand values define now the high demand scenario H. For the problem setting with $J = 60$ we consider $l = 20$, $m = 35$ and $h = 60$; when the total number of potential markets is $J = 80$ we set $l = 30$, $m = 50$ and $h = 80$; finally, for the setting with $J = 100$ we take $l = 40$, $m = 70$ and $h = 100$. In any problem setting, the three demand scenarios are assumed to have equal weight (1/3).

Besides demand scenarios, we also consider demand dependent returns scenarios in the following sense. We assume that $r_j = \eta\, d_j$ for all j, that is there is a uniform return rate η across all the markets. We consider four equally probable values for η, namely 0.2, 0.4, 0.6 and 0.8. Three alternative settings for demand and four for the return rate result in $3 \times 4 = 12$ overall scenarios to be investigated. Clearly, as in the previous chapter, the underlying assumption is that scenarios as such may be build based on field knowledge of some experts. Moreover, the associated probabilities are of subjective nature and tend to reflect the relative importance given to each of the identified alternatives.

In all problem settings we consider 15 potential plant locations and 25 potential disassembly center locations. However, since at any such location 2 possible capacities may be chosen, this leads to a total number of $I = 30$ potential plant sites and a total number of $K = 50$ potential center sites. In all settings, 7 plant locations and 10 center locations are first randomly generated according to a uniform distribution in S_0 and then 8 more plant locations and 15 more center locations are uniformly generated in S_1. The distance between any two locations is considered to be the euclidean distance (with kilometer as distance unit).

The settings given in Table 1 together with the assumptions made above define specific mechanisms which underly the system under consideration. The generation of scenarios aims to reflect situations where the markets (possibly for newly designed and developed products) may expand or alternatively, shrink in the long-run planning time, in terms of both demand levels and locational diversity. On the other hand, returns may vary at different levels, depending on the level of demand. Given this close dependency as well as the co-location of manufacturing and remanufacturing operations, the issue addressed here is that of an appropriate integral network design, that is the simultaneous optimization of both the forward and the reverse network channels. Clearly, for any *given* level of demand, in general it is profitable to invest in sufficient production capacity in order to meet all demanded volumes. Yet, another important observation is that for any *given* level of returns, the substantial penalty for not collecting (part of) returns combined with significant savings resulting from re-manufacturing lead to strong incentives to invest in sufficient testing capacity for all returned volumes. Since the two channels are closely interrelated at the plant level, the integral network configuration for *given* demand/returns values results from the interaction of these two tendencies. More specifically, the new products will be in general substituted by remanufactured products to the largest extent possible, such that both the new and the remanufactured products together cover as much as possible from markets demand. It is expectable that these functioning mechanisms of the system will be accurately reflected in the optimal solutions of the individual scenarios. However the aim of the stochastic approach is to generate a balanced solution, able to make a proper trade-off between the investments which would be chosen just on an individual scenario basis. Within the context of integrated forward and reverse logistics chains as considered here, the nature of this trade-off should be carefully analyzed, since besides fixed investment costs and transportation costs, the (potential) penalties may be an important driver as well. Lower investments may be appropriate for low or medium levels of demand/returns, but are likely to result in significant loss of market opportunities and/or high penalties for the returns which can not be collected in other scenarios due to the lack of testing capacity. By contrary, higher investments will in general avoid these consequences, but will probably generate a substantial loss in low/medium demand scenarios due to a substantial part of unnecessary (i.e. unused) opened capacity. Consequently, a trade-off solution should adequately balance the investments against the gains and the negative effects (including substantial potential penalties!) which may occur in all the considered scenarios. We discuss in more detail the impact of this uncertainty on the network design decisions in section 5.2, after addressing the computational efficiency of the proposed approach in the next section.

5.1 Computational efficiency

In order to assess the computational efficiency of the method, in each of the three problem settings (that is, J=60, J=80 and respectively J=100) we applied the described algorithm on five problem instances. The instances for different values of J were generated independently (i.e. there is no correspondence between instances with the same label from two different problem settings). Computational results are presented in Table 2, Table 3 and Table 4. These tables report the CPU running time (in seconds), the number of nodes and cuts generated during the searching process as well as the number of plants and facilities of each type opened in every instance. The CPU times include both the branch and cut part of the procedure and the building of the problem objects for the current problem and for the dual second stage problem. These results indicate a remarkable performance consistency of the method for the considered problem.

In the problem setting with $J = 60$ markets the model contains 6,380 binary first stage variables (80 opening decisions and 6,300 links) and 6,300 continuous second stage variables for each scenario, i.e. 75,600 continuous variables for all scenarios. The running times varied roughly between 3 and 7 minutes, with an average running time of about 5 minutes.

When $J = 80$ markets are considered, the model makes use of 7,980 binary first stage variables (80 opening decisions and 7,900 links) and 7,900 continuous second stage variables for each scenario (93,600 continuous variables in total). The running times in this setting varied between about 4 1/2 minutes and about 10 1/2 minutes. The average running time on the five instances was slightly less than 8 minutes.

Finally, in the setting with $J = 100$ markets the model contains 9,580 binary first stage variables (80 opening decisions and 9,500 links) and 9,500 continuous second stage variables for one scenario (resulting in a total of 114,000 continuous variables for all scenarios). The running times among the five instances varied roughly between 7 1/2 min and 20 1/2 minutes, with an average running time of about 14 minutes.

There is a larger increase in the average running times when going from 80 to 100 (potential) markets, than from 60 to 80 markets. However, all the computation times are rather low, even though a relatively large number of nodes are investigated and a significant number of cuts are generated. The context of the problem addressed here may not be directly comparable to that of the specific case study we have previously reported (see Listes and Dekker (2001)). Nevertheless, the two problems include comparable numbers of location decision variables, while obviously, the running times resulted here are with one order of magnitude lower. The difference in speed of the used computers contributes only to a very small extent to this remarkable reduction in computation times. Clearly, the essential difference relates to different modelling and solution methodology. In particular, the first stage problem as formulated here has a special structure, which can be exploited during solving. Moreover, owing to the decomposition strategy, it is no longer necessary to work with an extensive form of the model and a large number of second stage variables (i.e. one set for every scenario). Instead, the number of continuous variables which suffice for one scenario can be iteratively used for all the scenarios in order to compute the expected recourse function and where appropriate, the coefficients of the optimality cuts. Once determined, these elements are then transferred and used within the (first stage) current problem. It is this modularity which contributes essentially to the efficiency of the approach.

Instance	**Time**(sec)	**Nodes**	**Cuts**	**Plants** P^1	**Plants** P^2	**Facilities** R^1	**Facilities** R^2
1	231	523	69	2	0	4	4
2	426	1021	185	0	2	3	5
3	284	240	91	1	1	3	4
4	178	414	59	2	0	3	4
5	375	732	143	1	1	3	5
Average	298.8	586	109.4	1.2	1	3.2	4.4

Table 2: Computational results on 5 problem instances for J=60

Instance	**Time**(sec)	**Nodes**	**Cuts**	**Plants** P^1	**Plants** P^2	**Facilities** R^1	**Facilities** R^2
1	573	1242	183	2	1	5	5
2	278	406	67	0	2	5	4
3	486	351	145	1	2	4	6
4	621	1013	212	2	1	5	6
5	369	435	98	1	2	4	5
Average	465.4	689.4	141	1.2	1.6	4.6	5.2

Table 3: Computational results on 5 problem instances for J=80

Instance	**Time**(sec)	**Nodes**	**Cuts**	**Plants** P^1	**Plants** P^2	**Facilities** R^1	**Facilities** R^2
1	876	1073	193	1	3	6	7
2	619	959	129	2	2	6	6
3	1013	1347	156	1	3	5	8
4	442	731	74	2	2	4	7
5	1228	1785	248	2	3	6	9
Average	835.6	1179	160	1.6	2.6	5.4	7.4

Table 4: Computational results on 5 problem instances for J=100

Such specialized modelling strategies and solution methods may substantially improve the computational tractability of stochastic models for logistics network design. In particular, they may significantly reduce the proficiency gap between stochastic methods and computationally less demanding, yet theoretically less powerful methods such as sensitivity or scenario analysis. This is important since scenario analysis generally requires a much finer gradation of the scenarios to be considered, followed by a cumbersome performance evaluation of each individual scenario solution over all scenarios. If the running time of a stochastic method can be reduced far below the prohibitive level (as it is apparently the case here), multiple runs of such models may be performed in order to investigate in more comprehensive steps the robustness relative to the uncertainty in parameter values.

5.2 Impact of uncertainty

As mentioned previously, we dedicate this section to a more detailed discussion of the impact of uncertainty on network design decisions. In order to make things concrete, we present in more detail the results obtained in instance 5 of the problem setting with $J = 80$ potential markets. The optimal solutions for each of the twelve scenarios and the stochastic solution can be found in Table 5. We note that in this table the plant sites and test center sites are numbered separately, either numbering being done in increasing order of distance to the central point $(2000, 2000)$. The network configuration in the stochastic solution is further illustrated in Figure 2 (for the sake of clarity, the links to and from the markets are omitted in the picture).

The stochastic solution differs from any of the individual scenario solutions on both the opened plants and the opened testing centers, but seems closer to the configuration from M0.8 scenario. However, at first sight it can be noticed that the plant capacity opened in the stochastic solution is higher that in the M scenarios, while the opened test center capacity shifts even more to the upper side of the investment levels involved in scenarios. In order to explain this effect we take a closer look to the actual profit figures. To this end, Table 6 gives the optimal profit values for each individual scenario, the profit generated by the stochastic solution in each scenario as well as the difference between the optimal profit and the profit generated by the stochastic solution configuration (regret). One remark is in order. When evaluating the stochastic solution over the scenarios, only the location of the plants and the test centers is fixed, the transportation links are allowed to be re-chosen according to the situation in each scenario. This will hold as well when evaluating an arbitrary individual scenario solution configuration over all the scenarios.

Table 6 shows that the stochastic solution configuration achieves an expected profit equal to 69.92% of the weighted average of the individual optima. Moreover, it gives regret figures which are spread over a rather short range. Higher regret values are given in the L scenarios due to too much investment costs and in scenario H0.8 due to the payment of a penalty for not collecting a small fraction of returns. The stochastic solution configuration achieves the lowest regret in scenario M0.8.

Scenario	**Open plants**		**Open test centers**	
	P^1 sites (**total**)	P^2 sites (**total**)	R^1 sites (**total**)	R^2 sites (**total**)
L0.2	1, 2 (**2**)	(**0**)	2 (**1**)	3 (**1**)
L0.4	1, 2 (**2**)	(**0**)	2, 4 (**2**)	5 (**1**)
L0.6	1, 2 (**2**)	(**0**)	2 (**1**)	4, 5 (**2**)
L0.8	1, 2 (**2**)	(**0**)	5, 8 (**2**)	2, 3 (**2**)
M0.2	3, 7 (**2**)	6 (**1**)	6 (**1**)	4, 5 (**2**)
M0.4	3, 7 (**2**)	6 (**1**)	6, 8, 13 (**3**)	4, 10 (**2**)
M0.6	3, 7 (**2**)	6 (**1**)	10, 13, 16 (**3**)	1, 4, 6 (**3**)
M0.8	3, 7 (**2**)	6 (**1**)	14,18, 19, 20 (**4**)	3, 5, 6, 16 (**4**)
H0.2	15 (**1**)	3, 6, 9 (**3**)	9, 16 (**2**)	5, 6, 14 (**3**)
H0.4	15 (**1**)	3, 6, 9 (**3**)	3, 6, 23 (**3**)	5, 15, 19, 21, 25 (**5**)
H0.6	15 (**1**)	3, 6, 9 (**3**)	2, 6, 14, 19 (**4**)	3, 5, 15, 21, 23, 24, 25 (**7**)
H0.8	15 (**1**)	3, 6, 9 (**3**)	7, 13, 17, 19, 20 (**5**)	2, 3, 5, 14, 15, 21, 23, 24, 25 (**9**)
Stoch	8 (**1**)	3, 6 (**2**)	2, 14, 23, 24 (**4**)	3, 5, 15, 21, 25 (**5**)

Table 5: Optimal scenario solutions and stochastic solution, J=80, instance 5 (plant sites and test center sites are numbered separately; the numbering is done in increasing order of distance to the common center of the square regions)

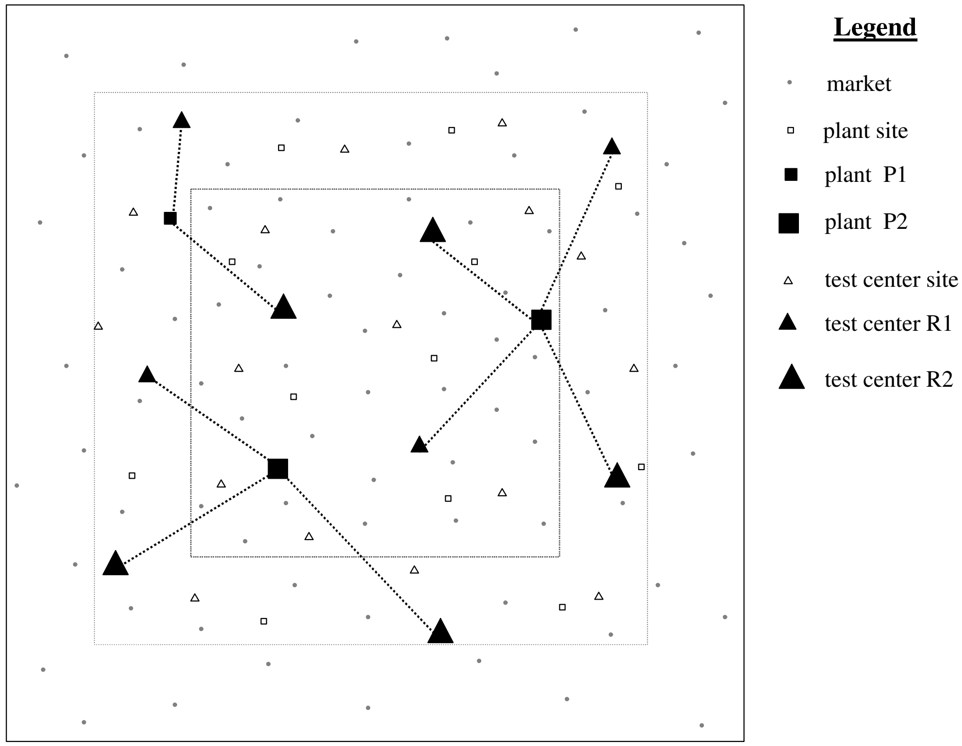

Figure 2: Network configuration in the stochastic solution (J=80, instance 5)

In Table 7 a worst case analysis for this instance is presented. The worst case configuration for each scenario is considered to be the configuration taken from the twelve scenarios which gives the lowest profit (or alternatively, the highest loss) in that scenario. It turns out that the solution from scenario H0.8 gives the worst performance on the first seven scenarios due to too high investment costs, while the solution from scenario L0.2 performs worst on the last five scenarios due to too less manufacturing/testing capacity which results in important loss of market opportunities and in huge penalties for not being able to test large volumes of returns. We note that in the worst case an actual loss is being generated in the first four and the last two scenarios, such that the expected worst case outcome is in the end a small negative value (loss). This means that for the considered system and its dynamics the expected worst regret basically equals the expected profit which would be only ideally achieved in case of perfect information (that is the weighted average of the individual optima). The loss generated by too high investments has about two times lower extent than the loss generated when paying the largest penalties. On the other hand, the loss due to high penalties in scenarios H0.8 and H0.6 is reduced in larger steps in scenarios H0.4 and H0.2 (which involve less returns), while the loss generated by too high investments are to a lower rate attenuated when processing increasing volumes of returns in scenarios L0.2 - L0.8. These remarks show that both the investments costs and the potential penalties are strong drivers of system performance. However, the impact of penalties shows a potentially more striking effect on the final outcome. For example, in the

Scenario	Optimal profit	Stochastic profit	Stoch Regret = Opt − Stoch
L0.2	29,235,100	1,168,400	28,066,700
L0.4	33,647,300	6,433,100	27,214,200
L0.6	37,861,700	11,306,600	26,555,100
L0.8	41,849,200	15,893,500	25,955,700
M0.2	57,863,400	35,652,100	22,211,300
M0.4	65,241,700	45,046,900	20,194,800
M0.6	72,126,400	54,488,200	17,638,200
M0.8	78,419,900	65,293,800	13,126,100
H0.2	103,835,800	84,489,900	19,345,900
H0.4	115,432,600	96,779,800	18,652,800
H0.6	126,057,300	108,832,700	17,224,600
H0.8	137,041,100	102,992,700	34,048,400
Expected	**74,884,291**	**52,364,808**	**22,519,483**

Table 6: Optimal values for scenarios and stochastic solution, J=80, instance 5

Scenario	Optimal profit	Worst Case profit (loss)	Worst Regret = Optimal−Worst
L0.2	29,235,100	− 23,326,500	52,561,600
L0.4	33,647,300	− 18,498,400	52,145,700
L0.6	37,861,700	− 13,594,800	51,456,500
L0.8	41,849,200	− 9,326,300	51,175,500
M0.2	57,863,400	11,567,500	46,295,900
M0.4	65,241,700	21,385,200	43,856,500
M0.6	72,126,400	32,599,100	39,527,300
M0.8	78,419,900	19,219,800	59,200,100
H0.2	103,835,800	44,842,400	58,993,400
H0.4	115,432,600	14,877,800	100,554,800
H0.6	126,057,300	− 23,246,800	149,304,100
H0.8	137,041,100	− 59,483,700	196,524,800
Expected	**74,884,291**	**− 248,725**	**75,133,016**

Table 7: Worst case analysis, J=80, instance 5

worst case performance, the penalties (associated as well with a loss of market shares) in the last three scenarios lead to up to four times higher absolute regret (relative to the individual optima) than the regret due to high investments in the first seven scenarios. This explains why the the stochastic solution shifts more to the upper side of the investment range, especially with regard to the test center capability.

We extend further our analysis and present in Table 8 a full listing of the regret associated with the individual solutions of three scenarios, L0.8, M0.8 and H0.8. It can be noticed that the three scenarios feature different patterns of regret when evaluated over all the 12 scenarios. While L0.8 scenario solution incurs huge regret in the H scenarios, the H0.8 scenario solution incurs large regret especially on the L scenarios. The M0.8 scenario solution performs well on the first 8 scenarios, but still incurs substantial regret in the last 4 scenarios, especially in the very last one due to a higher penalty. Nevertheless, the solution in scenario M0.8 achieves the lowest expected regret among the individual scenario solutions (and implicitly the highest expected profit among scenarios). The trade-off made by the stochastic solution becomes now more clear, in that it incurs as less differences between the various regret values as possible. Overall, the network configuration from the stochastic solution achieves 2.67% higher expected profit and 6.05% less expected regret than the one from scenario M0.8 solution (that is, the best one in expectation among scenarios). This analysis points out that the variations in demand and returns have a critical impact on the profitability of system operation. Moreover, the actual uncertainty in such factors at the moment of strategic planning has a considerable impact on the network design decisions as well. While the impact of uncertainty seems less dependent on the specific design in the "middle" scenarios, the design based on extreme scenarios shows highly increased risk of instability. From here, the need arises for a coherent method for balancing the investment decisions, as actually achieved through the stochastic approach.

The optimal network configurations for L0.8, M0.8 and L0.8 scenarios are illustrated in Figure 3, Figure 4 and respectively Figure 5. A first remark is that the consideration of multiple processing capacities significantly enhances the ability of adjustment of network configuration to the specific requirements in each situation. Besides contributing to tailoring the scenario solutions as close as possible to the necessities in each case, the multiple capacities assumption also generates more flexibility when it comes to the design of a network which should be balanced between the individual scenarios. When compared with the configuration in the stochastic solution, the last three pictures reflect once again through a visual representation the trade-off explained by the numerical results analysis. It is the reverse channel dimension with its associated possibility for large penalty application which drives investments in higher testing capability in the stochastic solution than in an imaginary "middle" scenario. Moreover, given the usage of this testing capacity and the potentially larger volumes it could direct for re-manufacturing, a slightly "higher than average" manufacturing/re-manufacturing capacity as well is chosen in the stochastic solution. In this choice, the economies of scale involved by high capacity plants and high capacity testing centers are also exploited. As a result of uncertainty, the stochastic solution points out the in general the logistics system tends to be somewhat redundant. At the same time, however, the stochastic approach meets the implicit requirement to determine the most effective form of redundancy required, as well as an operating strategy which will be able to exploit it. The findings in this section support the main conclusion that volume is a strong driver in the design of logistics systems with remanufacturing options.

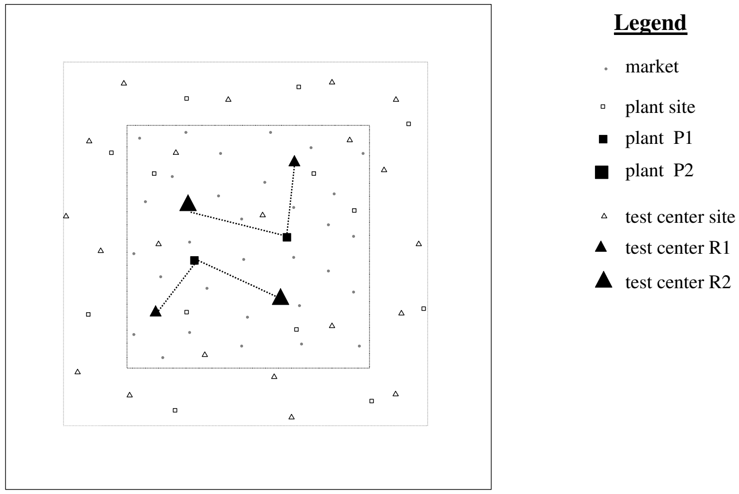

Figure 3: Optimal configuration in scenario L0.8 (J=80, instance 5)

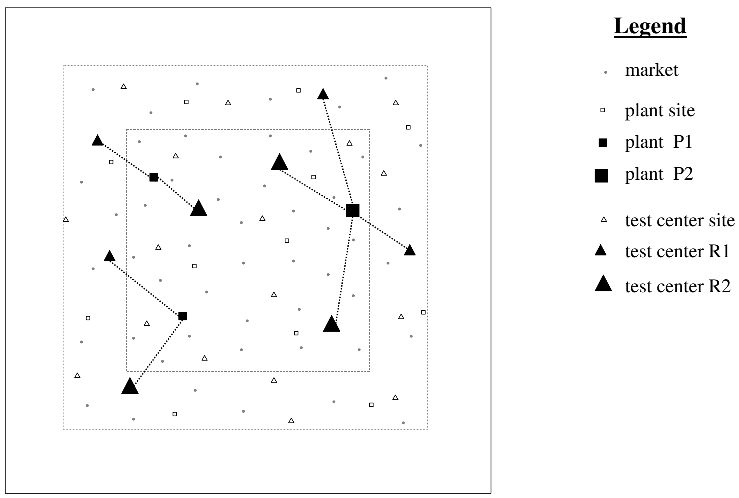

Figure 4: Optimal configuration in scenario M0.8 (J=80, instance 5)

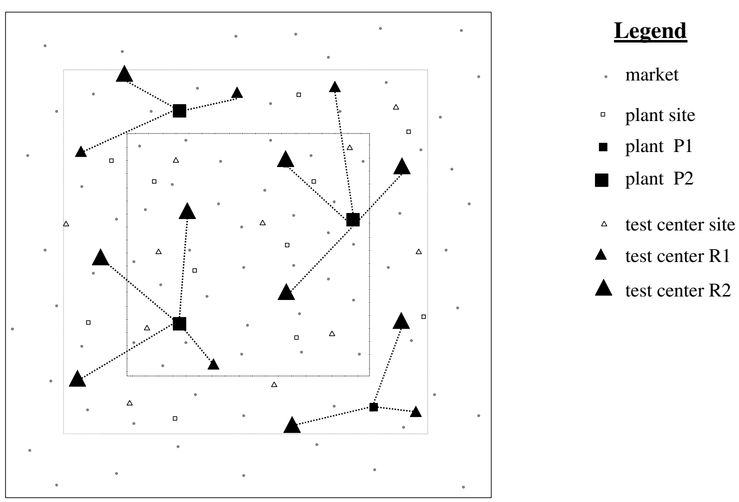

Figure 5: Optimal configuration in scenario H0.8 (J=80, instance 5)

Scenario	Regret in case of design Scenario L0.8	Scenario M0.8	Scenario H0.8
L0.2	2,329,500	20,268,900	52,561,600
L0.4	1,405,200	19,587,200	52,145,700
L0.6	583,800	18,829,300	51,456,500
L0.8	0	17,865,600	51,175,500
M0.2	20,589,300	13,718,300	46,295,900
M0.4	22,324,700	9,334,700	43,856,500
M0.6	23,812,500	5,748,500	39,527,300
M0.8	47,209,400	0	36,351,400
H0.2	57,148,900	39,193,400	18,112,800
H0.4	83,451,200	32,613,800	11,324,100
H0.6	121,560,700	36,322,500	5,502,900
H0.8	159,497,100	73,101,200	0
Expected	**44,992,691**	**23,881,950**	**34,025,850**

Table 8: Scenario regret analysis, J=80, instance 5

We resume this section with the remark that in this example the model gives insight into the appropriate network configuration based on a fairly rough division of information. Such situation is a common feature of strategic planning. The model takes such information into account based on a two-stage decision making assumption. The possible refinement of the available information, including the timing of its revealing, may require the addition of possible corrective decisions (including locational ones) to the overall analysis. However, the proposed two-stage model certainly generates valuable first step insight into the design problem by explicitly accounting for a number of alternatives. At the same time, it may serve as a potential starting point for even further advanced extensions, some of which are briefly mentioned in the next section.

6 Extensions

The presented model is a basic stochastic model for capacitated integral network design. This model may be extended in various ways in order to more accurately describe specific practical situations or to appropriately capture more complex decisional issues. As already mentioned, a possible extension may lead for instance to a multi-commodity model, where different quality streams may be distinguished concerning both the kind of products to be sold (new/re-manufactured) as well as various categories of returns. However, as resulting also from the numerical example above, probably one of the most desirable extensions is the explicit modelling of long-term, dynamic effects such as the step by step expanding (or shrinking) of the network based on the gradual revealing of extra information. Of specific need for building such multi-period or multi-stage models is the clarification of certain timing issues. For example, the actual amount of time for setting up the different facilities to be located should be more precisely specified in order to determine if corrective locational decisions are actually feasible in specific situations. More generally, the process of "accumulation" of incoming information should be more explicitly modelled in order to determine when the solution of each stage should be executed to determine the appropriate resulting infrastructure. Extensions as such certainly deserve further attention. A general drawback, however, is expected to be the computational tractability of such formulations. Presently there is a lack of a general methodology (either exact or approximative) capable to address multi-stage stochastic integer models. In particular, network design models falling into this category in are likely to incur extreme algorithmic challenges. Main means for tackling such potential extensions are expected to remain a careful and structured modelling together with the exploitation of the resulting model structure in the solution methodology.

7 Summary and conclusions

We have considered in this paper a generic stochastic model for design of networks comprising both supply and return channels organized in a closed loop system. Such situations are typical for manufacturing/re-manufacturing type of systems in reverse logistics. The model accounts for a number of alternative scenarios which may be constructed based on critical levels of design parameters such as demand or returns. We propose a decomposition approach for this model based on the branch and cut procedure known as the integer L-shaped method. Computational results show a consistent performance efficiency of the method for the addressed location problem. Such solution methodology may overcome generally recognized features of stochastic network design models, such as increased problem sizes and computational difficulty. Therefore, the approach may be further employed in multiple runs in order to make a comprehensive investigation of robustness relative to the uncertainty in parameter values.

The stochastic solutions generated in a numerical setting do not coincide in general with any of the optimal solutions for the individual scenarios considered. It generates a significant improvement in terms of average performance over most of the scenario solutions. Moreover, the stochastic approach has the ability to generate qualitatively different solutions, which give increased insight into the functioning of the system. It generally exploits features such as the flexibility offered by multiple possible capacities or by economies of scale. In the particular setting considered, the attention is drawn over the finding that if one can only roughly anticipate the levels of determinant parameters such as demand and returns, the integral network to be design tends to be somewhat redundant. A stochastic solution as generated here seems particularly suitable since it is capable to determine the most effective form of redundancy required and to properly balance its usage among various alternative situations.

The findings of the overall analysis lead to the main conclusion that volume is a powerful driver in integral networks with remanufacturing options. Furthermore, the processes which can adjust as accurate as possible to the overall requirements generally enjoy a natural advantage, provided that their investment costs are not prohibitive.

We believe that models as the one considered in this paper may considerably enhance the body of quantitative approaches able to represent and highlight important issues arising in the design of reverse logistics networks. Extensions of this model may be considered in order to contribute to a broadening of the modelling approaches presently available. Eventually such models may step by step find utilization in dedicated decision support systems able to shed light on the efficient design of logistics systems of the future.

References

Ammons, J.C., Realff, M.J., Newton, D., 1999. Carpet recycling: Determining the reverse production system design, The Journal of Polymer-Plastics Technology and Engineering 38 (3) 547–567.

Barros, A.I., Dekker, R., Scholten, V., 1998. A two-level network for recycling sand: a case study, European Journal of Operations Research 110 (2) 199–214.

Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems, Numerische Mathematik 4 238–252.

Birge, J.R., Louveaux, F., 1997. Introduction to stochastic programming, Springer Series in Operations Research, Springer Verlag, New York.

Drezner, Z., 1995. Facility Location: A Survey of Applications and Methods, Springer Series in Operations Research, Springer Verlag, New York.

Fleischmann, M., 2000. Quantitative Models for Reverse Logistics. PhD thesis, Rotterdam School of Management, Erasmus University Rotterdam, The Netherlands.

Fleischmann, M., Krikke, H.R., Dekker, R., Flapper, S.D.P., 2000. A classification of logistics networks for product recovery, Omega 28 (6) 653–666.

ILOG, 2000. ILOG CPLEX 7.0 User's manual.

Kall, P., Wallace, S.W., 1994. Stochastic Programming, John Wiley and Sons, Chichester, UK.

Laporte, G., Louveaux, F., 1993. The integer L-shaped method for stochastic integer programs with complete recourse, Operations Research Letters 13 (3) 133–142.

Laporte, G., Louveaux, F., van Hamme, L., 1994. Exact solution to a location problem with stochastic demands, Transportation Science 28 (2) 95–103.

Listes, O.L., Dekker, R., 2001. Stochastic approaces for product recovery network design: A case study, Technical Report EI 2001 - 08, Econometric Institute, Erasmus University Rotterdam, The Netherlands. (Revised February 2002 for the *European Journal of Operational Research*).

Louveaux, F., 1986. Discrete stochastic location models, Annals of Operations Research 6 (1) 23–34.

Louveaux, F., 1993. Stochastic location analysis, Location Science 1 (2) 127–154.

Morgan, M.G., Henrion, M., 1990. Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis, Cambridge University Press, New York.

Realff, M.J., Ammons, J.C., Newton, D.J., 2000. Strategic design of reverse production systems, Computers and Chemical Engineering 24 991–996.

Slyke, R.V., Wets, R., 1969. L-shaped linear programs with applications to optimal control and stochastic programming, SIAM Journal of Applied Mathematics 17 638–663.