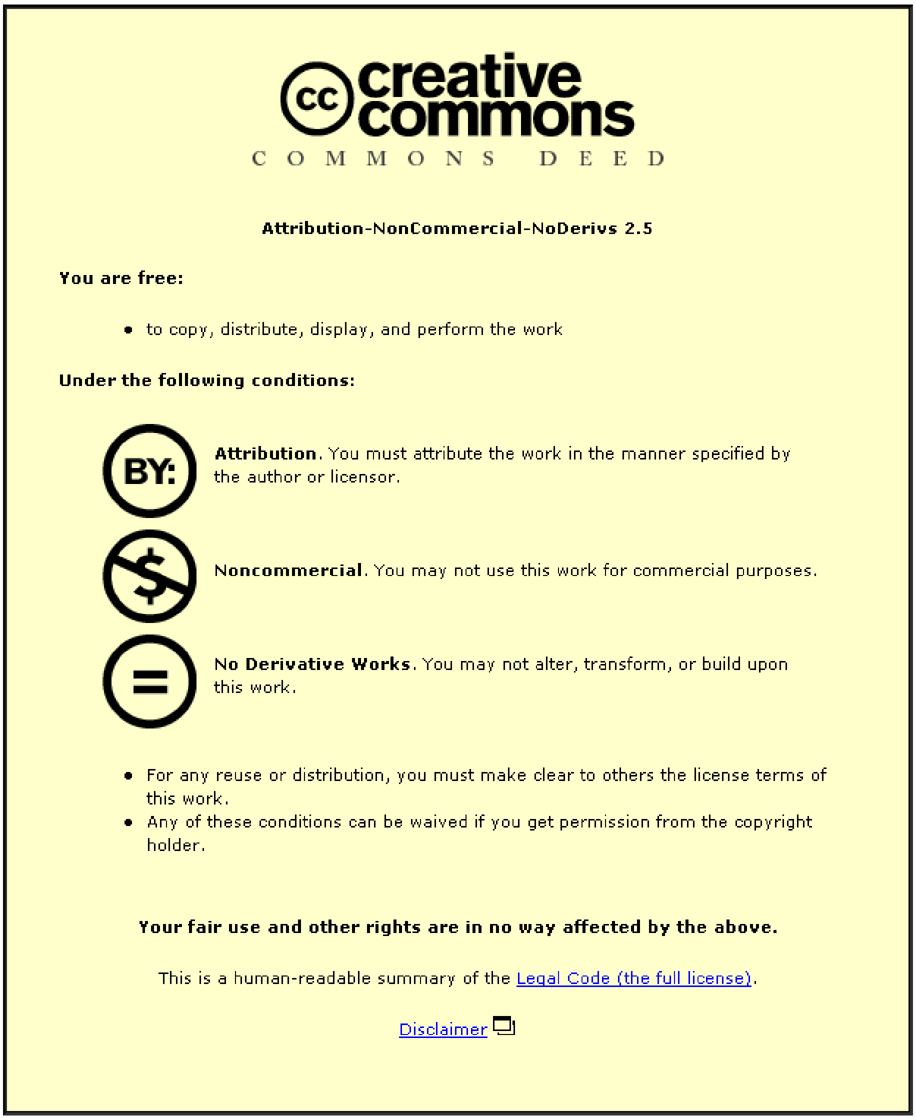


Modelling and Optimization of a Product Recovery Network

Rahul Swarnkar and Jennifer A. Harding
Wolfson School of Mechanical and Manufacturing Engineering
Loughborough University, Loughborough LE11 3TU, UK
Email: {r.swarnkar,j.a.harding}@lboro.ac.uk

Rahul Swarnkar

Dr. Rahul Swarnkar is a research associate in the Wolfson School of Mechanical and Manufacturing Engineering at Loughborough University. He obtained his doctorate degree from Loughborough University and B.Tech. degree from National Institute of Foundry and Forge Technology, India. Prior to joining Loughborough University, he worked in the Department of Industrial and Manufacturing Systems Engineering in University of Hong Kong as research assistant. His research interests are mathematical modelling, optimization techniques, simulation modelling, distributed simulation, reverse logistics and enterprise modelling.

Jennifer A Harding

Dr. Jenny Harding is a Senior Lecturer in the Wolfson School of Mechanical and Manufacturing Engineering at Loughborough University. She has substantial industrial experience having worked for over 15 years in the engineering and textile industries before joining Loughborough University in January 1992. Her research areas include knowledge management and reuse, tools to support knowledge sharing within collaborative teams, knowledge discovery and data mining applications in manufacturing, and 'Best Practice' information and knowledge. Her research has been funded by Europe and in the UK by EPSRC and Industry. Dr Harding is a member of the Innovative Digital Manufacturing Research Group and of the Product Realisation Technologies Research Group at Loughborough University.

Modelling and Optimization of a Product Recovery Network

Abstract

An appropriate logistics network is an important element of the infrastructure of any product recovery company. Small and medium enterprises (SMEs) constitute a major fraction of the product recovery industry with a different business objective and scale of operation from those of original equipment manufacturers. This paper addresses the network design issues for SMEs involved in product recovery activities. A mathematical formulation is presented in an SME context and a subsequent simulation model is developed. A genetic algorithm approach is presented for optimizing the network for single product scenario.

Keywords: product recovery, reverse logistics, simulation, optimization, genetic algorithm, Arena

1 Introduction

Recovery of used products and materials has attracted researchers' attention for many years. However in the recent years, the enforcement of environment friendly policies by different governments and customer enthusiasm for greener production has encouraged companies to start product take–back activities. The products are collected after their end of life with the aim of recovery or safe disposal.

Product recovery is the transformation of used and discarded products into useful condition through reuse, remanufacture and recycling. Implementation of product recovery requires setting up an appropriate logistics infrastructure for the arising flows of used and recovered products. Physical locations, facilities, and transportation links need to be chosen to convey used products from their former users to a producer and to future markets again. Reverse logistics encompasses the logistics activities all the way from used products no longer required by the user to recovered products that are again usable in a market. The study of reverse logistics can be broadly divided into three areas: *distribution planning*, which involves the physical transportation of used products from the end user back to the producer; *inventory management*, which is the process of managing the timing and the quantities of goods to be ordered and stocked, so that demands can be met satisfactorily and economically; and finally *production planning*, which despite not being a logistics activity, influences the other two greatly (Salema *et al.* 2007). One of the initial publications addressing distribution issues was by (Gottinger 1988). Thereafter, several models have been proposed which focus on aspects such as product recycling and planning/distribution (Caruso *et al.* 1993, Fleischmann *et al.* 2001, Giannikos 1998). A more general classification of the research areas related to reverse logistics is provided by (Dekker *et al.* 2004) and presented by (Rubio *et al.* 2008), identifying the following areas:

- Management of the recovery and distribution of end-of-life products.
- Production planning and inventory management
- Supply chain management issues in reverse logistics

The main activities in reverse logistics are the collection of the products to be recovered and the redistribution of the reprocessed goods. The reverse logistics problem looks quite similar to the normal forward distribution problem; however there are some differences too. Reverse flow of goods is convergent in nature, so the products need to be collected from many points. Therefore cooperation of the senders becomes important as product packaging is generally problematic. Products flowing in the network tend to have low value. On the other hand, time is not so important an issue as it is in forward distribution. Taking these issues into consideration, reverse logistics need new networks to be constructed. The major issues concerning design of a recovery network are the determination of the number of tiers in the network, the number and location of collection/drop–off and intermediate depots and the interaction of the reverse chain with the forward chain.

Recently, a number of case studies have been reported in the literature addressing the design of logistic networks in the product recovery context. (Kroon and Vrijens 1995) address the design of a logistics system for reusable transportation packaging. They discuss the role of the different actors in the system, economy, cost allocation, amount of containers and locations of the depots. (Castillo and Cochran 1996) discuss the distribution and collection of reusable bottles for a soft drink company while (Duhaime *et al.* 2001) address the same issues for reusable containers for Canada Post. (Alshamrani *et al.* 2007) develop a heuristic procedure for route design and pickup strategy for a network inspired by blood distribution by the American Red Cross. (Krikke *et al.* 1998) address the remanufacture of photocopiers and as remanufacturing is a labour intensive process they compare two remanufacturing options for the company; one coinciding with the existing manufacturing network and the other in another country where labour is cheap. (Barros *et al.* 1998) report a case study discussing the design of a logistics network for recycling sand coming from construction sites as waste. Recycling of carpet waste is addressed by (Louwers *et al.* 1999) and (Realff *et al.* 2000) and a mixed integer linear programming (MILP) model for the recycling of industrial by–products in German steel industry has been developed by (Spengler *et al.* 1997). An overview of key papers on reverse supply chain modelling is presented and a quantitative model is developed to support decision making concerning the design structures of both the product and the logistic network by (Krikke *et al.* 2003).

The above examples highlight the fact that most research in the area of reverse logistics network design has been case specific. The most generic model for the design of a reverse logistic network is the one proposed by (Fleischmann *et al.* 2001). This model considers the impact of inclusion of product recovery on the forward network and the model is optimized taking into account both the flows. A MILP formulation is proposed extending the traditional warehouse location problem and integrating the forward chain with the reverse chain. This work has subsequently been extended by (Salema *et al.* 2007) where capacity constraints, multi–product scenario and uncertainty were added.

2 The Recovery Network

In the present literature, much of the published work addresses problems involving big market players like Hewlett Packard, Canon, Dell and other original equipment manufacturers (OEMs) in the electronics industry. Similarly, the published research work dealing with other types of industries focuses on the original manufacturers' point of view. However, as previously highlighted, the recovery industry largely consists of smaller, independent recovery companies. These companies are not OEMs, so for them merging their procurement process with the distribution is not of great importance as their markets are quite different from those of OEMs and their markets for recovered products may well be different to the sources of products for remanufacture. As these companies are SMEs and recovery is their main job, the design of an efficient recovery network is extremely important as the damage caused by network inefficiency cannot be compensated from other means. This paper presents a mathematical model for the design of the network of a third party recovery firm. The formulation is based on (Fleischmann *et al.* 2001) however the context is quite different as (Fleischmann *et al.* 2001) present a generic model for companies wanting to integrate reverse logistics into their existing supply chain. In contrast the context of the initial network design formulation presented in the next section is to address network design issues for SMEs dealing with remanufacturing of returned items. The need for developing the network optimization model was realised when the authors visited one of UK's leading companies in the independent recovery industry. This visit built authors' understanding about the type of problems that they are facing. These include storage space for its facilities and the uncertainty regarding the returned items. Hence the model develop in this research includes capacity constraints and the simulation approach is employed to map the uncertainty. For the sake of simplicity, first a single product scenario has been formulated and the optimization tool is developed with it. Then it is converted to a multiple product model. The model is optimized using a genetic algorithm in conjunction with a simulation approach. The use of simulation helps in incorporating the uncertainty associated with the product returns. The computational setup is discussed in later sections of this paper and the proposed mathematical model is described below.

2.1 Mathematical Model for a Single Product Recovery Network

As mentioned earlier, the motivation for the model comes from the author's experience with industry. Three facility levels are considered, i.e. collection points which are responsible for collecting the used products and initial inspection if they are equipped with adequate facility, warehouses where returned products are stored and plants which finally reprocess them (Figure 1).

While establishing a distribution network, it should be taken into account that facilities have limitations on the number of products they can store or process. These limitations are due to various factors like availability of space, number of workers and workstations etc. The network model addresses these limitations by incorporating capacity constraints for each facility.

— Take in Figure 1 —

The proposed recovery network model involves the following index sets, variables and parameters:

Index Sets

$i \in I$; where $I = \{1, \dots, N_c\}$ fixed locations for collection points

$j \in J$; where $J = \{1, \dots, N_w\}$ potential locations for warehouses

$k \in K$;where $K = \{1, \dots, N_p\}$potential locations for plants

Costs

F_i^c Fixed cost for enabling collection point i for inspection

F_j^w Fixed cost of opening warehouse j

F_k^p Fixed cost of opening plant k for disassembly and reprocessing

T Collective cost of storage at collection points, warehouses and plants

P Unit penalty cost for not processing returned product

t^{cw} Unit transportation cost from collection point i to warehouse j

t^{wp} Unit transportation cost from collection point i to disposal site l

c^p Unit cost of reprocessing

c^d Unit cost of disposal

C_{ijk} Cost of reprocessing returned product from collection point i coming through warehouse j at plant k

D_{ijk} Cost of disposing of the returned product coming from collection point i through warehouse j and plant k

S_{ijk} Cost saving by disposing the discarded returned product at inspection enabled collection point i (and not traverse it through warehouse j and plant k)

If d_{pq} be the distance between points p and q in the distance matrix; we calculate the above costs as follows:

$$C_{ijk} = t^{cw} d_{ij} + t^{wp} d_{jk} + c^p + T \tag{1}$$

$$D_{ijk} = t^{cw} d_{ij} + t^{wp} d_{jk} + c^d + T \tag{2}$$

$$S_{ijk} = t^{cw} d_{ij} + t^{wp} d_{jk} + T \tag{3}$$

Variables

$$x_i^c = \begin{cases} 1; \text{ if collection point } i \text{ is enabled with inspection facility} \\ 0; \text{ otherwise} \end{cases}$$

$$x_j^w = \begin{cases} 1; \text{ if warehouse } j \text{ is opened} \\ 0; \text{ otherwise} \end{cases}$$

$$x_k^p = \begin{cases} 1; \text{ if plant } k \text{ is opened} \\ 0; \text{ otherwise} \end{cases}$$

y_{ijk} fraction of returned products served by collection point i, warehouse j and plant k

z_i fraction of the returned product at collection point i which can not be reused (chosen with a random distribution)

Parameters

R_i return from collection point $i; i \in I$

M_i^c maximum capacity of collection points $i; i \in I$

M_j^w maximum capacity of warehouse $j; j \in J$

M_k^p maximum capacity of plant $k; k \in K$

Using the above notation, the mathematical formulation to minimise the sum of the fixed, variable and penalty costs is as follows:

$$\min FC + VC + PC \tag{4}$$

Where,

$$FC = \sum_{i \in I} F_i^c x_i^c + \sum_{j \in J} F_j^w x_j^w + \sum_{k \in K} F_k^p x_k^p \tag{5}$$

$$VC = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} R_i C_{ijk} y_{ijk} + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} R_i z_i \left(D_{ijk} - S_{ijk} x_i^c \right) \tag{6}$$

$$PC = \sum_{i \in I} R_i \left(1 - \sum_{j \in J} \sum_{k \in K} y_{ijk} \right) P \tag{7}$$

Subject to:

$$\sum_{j\in J}\sum_{l\in K} y_{ijk} + z_i = 1, \quad \forall i \in I \tag{8}$$

$$\sum_{j\in J}\sum_{k\in K} R_i y_{ijk} \le M_i^c x_i^c \quad \forall i \in I \tag{9}$$

$$\sum_{k\in K}\sum_{i\in I} R_i y_{ijk} \le M_j^w x_j^w \quad \forall j \in J \tag{10}$$

$$\sum_{i\in I}\sum_{j\in J} R_i y_{ijk} \le M_k^p x_k^p \quad \forall k \in K \tag{11}$$

$$0 \le y_{ijk}, z_i \le 1 \tag{12}$$

$$x_i^c, x_j^w, x_k^p \in \{0,1\} \tag{13}$$

The above formulation minimises the fixed cost for the setup of facilities and costs involved in the recovery/disposal processes. The three terms in equation (5) represent the cost of installing inspection facilities at collection/drop–off points and setup costs for warehouses and reprocessing plants. The first term in equation (6) maps transportation costs and reprocessing/disposing costs for the reprocessing/disposing of product, while the second term in this equation involves cost savings for the product if the collection point it is coming from has inspection facilities installed. The returned products which are not processed due to the capacity constraints pose a loss and are mapped by equation (7). Constraint (8) ensures that all the returns are taken into consideration. Equations (9–11) make sure that the capacities of the facilities are not exceeded.

The formulation is generic in nature and can reflect recovery scenarios for various kinds of products. The disposal of unusable products from collection points as well as from plants may involve sending them to a third party recycler/disposer or to the remanufacturer’s own facility and the associated transportation cost. This model just requires the flow of such items to leave the network after sorting.

3 Optimization of the Model

One of the major characteristics of problems concerning reverse logistics activities is the uncertainty associated with the return of products, including quantity, quality and timing. The stochastic nature of these problems means that most of the analytical models become either too simplistic or exceptionally complex. Discrete event simulation is regarded as the most suitable analysis tool for such situations and is largely used to evaluate “what-if” scenarios (Fishman 2001, Schroer and Tseng 1988, Smith *et al.* 1994). In this research, a simulation based approach is used for the optimization of the model.

3.1 Solution Methodology

A general simulation based optimization method consists of two essential components: an optimization module that guides the search direction and a simulation module for evaluating the performance of candidate solutions. The decision variables create the environment in which the simulation is run while the output of the simulation runs is used by the optimiser to progress the search for optimal solution (Figure 2).

— Take in Figure 2 —

In the existing literature, a number of simulation–based optimization methods have been reported, which include gradient based search, stochastic approximation, sample path optimization, response surface, heuristic search methods and evolutionary algorithms (Andradóttir 1998, Azadivar 1999). There are several metaheuristic optimization algorithms present in the literature like tabu search (TS), simulated annealing (SA), ant colony optimization (ACO) and genetic algorithm (GA). The performance of TS and SA deteriorate significantly as the problem size and solution space increases (Woodruff 1994). ACO approach is best suited for travelling salesman problem and needs to be manipulated to addresses other types of optimization problems (Dorigo *et al.* 1996, Dorigo and Gambardella 1997). According to an empirical comparison of search algorithms by (Lacksonen 2001), GA appears to be the most robust to solve large problems though it requires a large number of replications.

In the past, GA has been successfully applied to classical combinatorial problems such as capacitated plant location (Gen *et al.* 1999), fixed charge location (Jaramillo *et al.* 2002), minimum spanning tree (Zhou and Gen 1999), network design (Palmer and Kershenbaum 1995), and warehouse allocation (Zhou *et al.* 2003). GA has been applied to the network design problem for reverse logistics as well (Min *et al.* 2006, Ko and Evans 2007, Lieckens and Vandaele 2007). Because of this proven effectiveness of GA for various combinatorial problems, it has been adopted in this research to perform stochastic search for solutions. The details of the algorithm are discussed in later subsections.

Simulation Model

A simulation–based optimization method has been developed for the optimization of a network design problem, keeping the constraints within the model logic. The simulation model has been created in Arena 10.0 (Kelton *et al.* 2007), which was selected over other available simulation software because of its seamless integration with other software supporting Microsoft technologies. Arena exploits two Windows technologies that are designed to enhance the integration of desktop applications. The first, ActiveX Automation allows applications to control each other and themselves via a programming interface. The second technology exploited by Arena for application integration addresses the programming interface issue. In this research, the code for the optimization algorithm has been written in Visual Basic (Deitel *et al.* 1999) and uses the Arena model for the evaluation of candidate solutions.

In the simulation model, the entities representing returned products in the model are generated on a daily basis and the number of products is decided by normal distribution with a mean proportional to the population of the customer zone. Each entity carries attributes of its origin customer zone, warehouse and plant locations and reusability. The decision whether a facility is open or not is coded in the candidate solution sent over by the GA code while Arena VBA blocks decide what alternative facilities are available for the entity to use.

Genetic Algorithm Representation and Operations

One of the most important aspects of genetic optimization is the chromosome encoding for representation of a typical solution. The encoding depends largely on the nature of the problem. In this case, the chromosome is an array of binary variables as shown in Figure 3. The individual binary arrays for facilities at all the tiers are concatenated to form chromosomes for binary representation of the solution. As the chromosome consists of variables of uniform nature, the genetic operations are performed on the whole of the chromosome at once. Each binary variable in the chromosome represents the installation of the associated facility.

— Take in Figure 3 —

Crossover and *mutation* are two basic genetic operations for the optimization search. The crossover method applied in this work is Single Point Crossover illustrated in Figure 4. In this method, a position is selected randomly. The binary string from the beginning of the chromosome to the crossover point is taken from one parent and the rest is copied from the other parent. As shown in Figure 4, this operation can produce two offspring chromosomes using one crossover point. In this work, the parents are selected by roulette wheel selection and both the produced offspring chromosomes are included in the new population. To maintain the diversity of the population, and save the search from getting trapped in local optima, GA uses another operation called mutation. Based on the mutation probability, the produced offspring is subjected to mutation operation, using Bit Inversion Method (although several other methods could be used instead). In this method, the binary bits of the chromosome are inverted (a NOT binary operation is applied) as shown in Figure 4.

— Take in Figure 4 —

Determination of Probability Values

The best values of these probabilities for a particular problem is decided with a small set of experiments. First a couple of arbitrary sets of values are chosen. Then a sample problem is chosen and it's time horizon is greatly reduced and accordingly the costs are adjusted. For example if a 1 year problem is reduced to 1 day, the associated annual costs are also reduced accordingly. Now the average iteration time comes down to a few seconds from the original 2–4 minutes. With the help of these reduced examples, the algorithm was run for different sets of the probabilities and the appropriate values of the probabilities were determined. These sets of probabilities vary for different types of problems and hence need to be determined for each individual problem.

3.2 Test Problems

Problem Description

A hypothetical example of a single product recovery enterprise has been used to analyse the model. The structure and functionality of the hypothetical company is based on experience gained from the product recovery industry and the design of the reverse logistic network for an SME dealing with printer cartridge remanufacture is considered. It is assumed that the SME procures used cartridges from certain customer zones through its collection points spread across the UK (Figure 5). The collection points procure used cartridges from independent retailers and high volume users irrespective of their condition (reusable/unusable). The returns coming from the customer zones are assumed to be proportional to their population. The collection points may or may not have facilities to sieve out unusable products. If the products are found to be unusable in an inspection enabled collection point they are sent directly to the disposal site. This saves costs of storage and transportation of the unusable product at different tiers of the network. The transportation cost involved in the transit between different tiers of the network varies and generally it is higher in case of transit from collection points to the warehouse than in the case of transit from warehouse to the plant. From the collection points, the cartridges are sent to warehouses for storage. Plants have facilities to inspect and reprocess the products.

— Take in Figure 5 —

The design problem poses several questions for the decision maker in the SME. For example, depending on the location of collection points, the nature of the returned product will vary. Some collection points with large volumes of returns might actually have benefits if they are enabled with inspection facilities. The location of the warehouses and plants is another strategic issue to save transportation and handling cost. The complexity of the problem multiplies as the numbers of tiers and products increase.

Generation of Example Problems

Based on the above description and understanding built from a survey of the available literature, the ranges of costs and parameter values were decided as listed in Table 1. Ten data files were created with values uniformly distributed in the ranges as shown in Table 1 for the optimization tool to create random example problems. The example problems are created in accordance with the problems presented in (Fleischmann *et al.* 2001).

— Take in Table 1 —

3.3 Simulation Based Optimization Tool

A tool has also been developed in Visual Basic to handle the GA based optimization task for the network configuration, and a screenshot of this optimization tool is shown in Figure 6. This tool works in conjunction with a simulation model template created in Arena. This template contains the modules and VB codes common to all types of problems under consideration. The optimization tool gets the basic data from the user through its GUI and loads the detailed information specific to the current network problem from this user specified data. The tool then invokes Arena to load the model template to modify the existing modules and create new ones. A screenshot of an Arena model created by this tool is shown in Figure 7. The model shown has 10 collection points generating a number of entities (representing returned products) on every working day based on a uniform distribution. Attributes representing origin, destination warehouse and reprocessing plant and inspection tags are created for each entity. The entity travels through the various VBA logic blocks and decision modules, which determine the destination warehouses and plants of the entity and assign it to the respective attributes. These decisions are based on the model constraints and input in the form of candidate chromosomes. Associated costs are calculated as the entity travels through process blocks before being disposed (representing products being sent to market or recycled/disposed due to infeasibility of remanufacture). Once the model is created, the data related to the model run and parameters of the optimization are entered by user or retrieved from file to start the optimization. The optimization results are stored on a spreadsheet.

— Take in Figure 6 and Figure 7 —

Performance of the Optimization Approach

To test the genetic algorithm based optimization approach, a small example of 10 candidate collection points (CP), 5 warehouses (WH) and 2 plants (PL) was considered as shown in Figure 5. The generated model was optimised with the developed tool and gave reasonably good solutions at around the 300th generation. Figure 8 shows the plot of best solutions obtained in generations for the first test set. The continuous plot in the figure shows the convergence of solution. Figure 9, Figure 10 and Figure 11 show different configurations of the network as obtained by the iterative optimization process and associated costs. Note that the cost associated with the network in Figure 10 is not much higher than that in Figure 11 despite the longer traverse paths. This is due to the fact that the latter involved a fixed cost for setup of additional facilities.

—Take in Figure 8, Figure 9, Figure 10 and Figure 11 —

4 Performance of the Model

After generating various simulation models specific to the problem the optimization is started. The fixed costs associated with the setup of facilities are annual costs hence the simulation horizon is set to 1 year. The warm–up period, required for the simulation model to reach the steady state is set to 1 month. The duration of warm–up period is decided based on the observations of pilot runs of the simulation. For each candidate solution, ten replications of simulation were run to smooth out residual randomness. A large GA population will result in higher computational time and a lower one may lead to premature convergence of the solution. Hence there is always a trade–off between the computational time and solution quality while deciding the GA parameters. The GA population for this problem size is set to 25 after observing few test iterations while generating the initial population, 1 solution is predefined with all the facilities setup and the rest of the solutions in the population are randomly generated. At each generation, solutions are selected for crossover or mutation operations based on their respective probabilities. The optimization tool is run on a Pentium 4HT Dual Processor PC running Windows XP at a clock speed of 3.06 GHz on 2 GB of RAM and took around 1-2 minutes per generation of GA.

Simulation model

The simulation model built for the optimization is generic in nature and is modified according to the data provided by the optimization tool and hence the model run for each problem is unique. The simulation model built for the optimization is quite flexible in nature and the decisions with multiple influencing parameters/variables such as determination of destination facilities are taken by the VBA blocks built within it. The logic of these blocks can be slightly modified to give a competitive edge to certain facilities according to the problem scenario. Such tweaks in the logic are useful in the cases where facilities at geographically dispersed locations have different overall cost and time for processing products. Once the optimization is finished, the models can be simulated without the help of the optimization tool with the optimum or other set of configuration for further investigations.

In order to verify the correctness of the model, the values calculated by its logic were compared with manual calculations. It can be understood that performing the manual calculations of an entire simulation run will be impossible in terms of time taken. Therefore, a random sampling approach was taken. While running the model with a specific network configuration, the simulation was stopped at certain date and all the parameters were calculated. Then based calculations were done for one day of operations and compared with the values obtained from the model simulation stopped at next day. These random verifications are carried out throughout the development of the simulation model so that its correctness was checked and assured.

Reverse Logistics Network

The optimization was run on 10 examples generated using the values in Table 1. Figure 12 shows costs associated with the initial and optimal configurations for the various example problems. It is observed that most of the configurations came out with only a few collection points enabled with inspection facilities. This is due to the fact that printer cartridges are one of the most ‘remanufacturable’ products, so the probability of being reusable is high (hence the parameter settings in the model). Cartridges have very short life cycle and are generally handled with care. The short life span helps in two ways: the cartridges do not get much time to be mishandled and they do not become obsolete by the time they are returned. Hence they have a high probability of being in a reusable condition when they reach the remanufacturer. However, for a more complex product, the case is different. For example, a mobile device (phone or laptop) becomes obsolete within half of its lifespan! In such cases, having intermediate inspection sites would be helpful for channelling the product to disposal or other type of reuse/recycle site. The above observation is also backed by Table 2, which shows the variation in number of inspection enabled collection points obtained from simulation runs with different values of reusability of the returned product.

— Take in Figure 12 and Table 2 —

5 Multiple Product Scenario

The formulation presented above addresses the network issue for an SME dealing with a single product. However, an SME in the independent product recovery business essentially deals with multiple products. For inclusion of multiple products, an index set of products is introduced

$l \in L$; where $L = \{1, \ldots, N_r\}$ set of products

The notations for costs, variables and parameters will change to

Costs

F_i^c Fixed cost for enabling collection point i for inspection

F_j^w Fixed cost of opening warehouse j

P_l Unit penalty cost for not processing returned product l

T_l Collective cost of storage for product l at collection points, warehouses and plants

t_l^{cw} Unit cost of transporting product l from collection point to warehouse

t_l^{wp} Unit cost of transporting product l from warehouse to plant

c_l^p Unit cost of reprocessing product l

c_l^d Unit cost of disposing product l

C_{ijkl} Cost of reprocessing returned product l from collection point i coming through warehouse j at plant k

$$C_{ijkl} = t_l^{cw} d_{ij} + t_l^{wp} d_{jk} + c_l^p + T_l \tag{14}$$

D_{ijkl} Cost of disposing of the returned product l coming from collection point i through warehouse j and plant k

$$D_{ijkl} = t_l^{cw} d_{ij} + t_l^{wp} d_{jk} + c_l^d + T_l \tag{15}$$

S_{ijkl} Cost saving by disposing the discarded returned product l at inspection enabled collection point i (and not traverse it through warehouse j and plant k)

$$S_{ijkl} = t_l^{cw} d_{ij} + t_l^{wp} d_{jk} + T_l \tag{16}$$

Variables

y_{ijkl} fraction of returned product l served by collection point i, warehouse j and plant k

z_{il} fraction of returned product l at collection point i which can not be reused (chosen with a random distribution)

Parameters

R_{il} total return of product l from collection point $i; i \in I$

Using the above notations, the mathematical formulation for the fixed, variable and penalty costs is as follows:

$$FC = \sum_{i \in I} F_i^c x_i^c + \sum_{j \in J} F_j^w x_j^w + \sum_{k \in K} F_k^p x_k^p \tag{17}$$

$$VC = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} R_{il} C_{ijkl} y_{ijkl} + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} R_{il} z_{il} \left(D_{ijkl} - S_{ijkl} x_i^c \right) \tag{18}$$

$$PC = \sum_{l \in L} \sum_{i \in I} R_{il} \left(1 - \sum_{j \in J} \sum_{k \in K} y_{ijkl} \right) P_l \tag{19}$$

The constraints need to include the consideration of multiple products as well. So the modified constraints are:

$$\sum_{j \in J} \sum_{l \in K} y_{ijkl} + z_{il} = 1, \quad \forall i \in I, l \in L \tag{20}$$

$$\sum_{l \in L} \sum_{j \in J} \sum_{k \in K} R_{il} y_{ijkl} \le M_i^c x_i^c \quad \forall i \in I \tag{21}$$

$$\sum_{l \in L} \sum_{k \in K} \sum_{i \in I} R_{il} y_{ijkl} \le M_j^w x_j^w \quad \forall j \in J \tag{22}$$

$$\sum_{i \in L} \sum_{i \in I} \sum_{j \in J} R_{il} y_{ijkl} \le M_k^p x_k^p \quad \forall k \in K \tag{23}$$

$$0 \le y_{ijkl}, z_{il} \le 1 \tag{24}$$

$$x_i^c, x_j^w, x_k^p \in \{0,1\} \tag{25}$$

The above formulation brings the model closer to the real world scenario by including multiple products. The same simulation based optimization approach is used for the evaluation of the model, i.e. the optimization in multiple product scenario is the same as that in the single product scenario. However, the simulation model needed alteration as the modules responsible for entering entities representing products needed to be modified to produce multiple products and assign attributes to the model. Also the logic of the model needs to be modified to handle multiple products.

For initial experimentation on the multiple product scenarios, The problem described earlier is extended to two products. Going from one product to two may seem to be a small improvement, however it serves the purpose of allowing the possibility of inclusion of more than one products in the logistic network. The parameters and costs associated with the products are listed in Table 3. The fixed costs remain the same as in Table 1. The computational time increases to 2-3 minutes for a generation. With crossover and mutation probabilities of 0.85 and 0.1 respectively, the solution converges at approximately the 457th generation. Figure 13 shows costs of “best solution so far” at every 10th generation of the optimization iterations.

—Take in Table 3 and Figure 13—

5.1 Inclusion of Multiple Product in the Network

A company involved in the recovery business essentially deals with multiple products. The presence of multiple products makes the network optimization problem more complex and has great impact on the output. Figure 14 shows the optimal configurations for a network optimization problem with 13 collection points, 6 warehouse and 3 plants. Figure 14a is the optimal configuration with two products, in which all the collection points are inspection enabled and only one warehouse is not installed. The facility which is not installed can be identified as the one which is not connected by any arrow. However, when the same problem was run with either of the products, the optimal configuration was different due to the variability of availability and quality of products at different locations (Figure 14b and Figure 14c).

— Take in Figure 14 —

6 Concluding Remarks

To successfully implement the product recovery activities, an appropriate logistics network needs to be established for the flow of returned products. Logistic network planning involves decisions about the location of facilities as well as capacity planning for the concerned facility. For an independent recovery SME, decisions regarding storage are also vital as the uncertainty of quality and quantity of recovered products is high because of the wide range of products coming in the facility. This paper presents a mathematical formulation addressing the network design issues for SMEs involved in independent product recovery activities. The formulation examines the inspection, separation and remanufacturing stages. Based on the formulation, a simulation model is created. The network configuration is optimised using a GA with fitness functions calculation done by the simulation model. The simulation model approach enables company managers to examine and compare the possibilities offered by various possible configurations through what–if types of experiments. The optimum configurations obtained from the optimization are then utilised to perform further investigations.

The initial formulation presented in this paper addresses the network issues for a SME dealing with a single product. However, the SMEs in the independent recovery business essentially deal with multiple products. Therefore a multiple product formulation is also presented later in the paper. The example shown with the multiple product formulation takes two products at a time, which at first glance does not look a big step ahead of single product scenario. However, the major difference in formulation and modelling for both the scenarios lies in the fact that even for two products, all the calculations are done with the help of iterations, which makes it mathematically possible to introduce any number of products.

This kind of approach could be applied by the industry in an iterative way. The simulation approach utilised in calculating the fitness function for optimisation, makes use of predefined sets of values and parameters. These values and parameters are decided and/or calculated from previously available data either by simple averaging or by complex functions depending on the requirements. A decision on the configuration of the network is reached and the operations start to take place. Subsequently, management could come back to their model and check the optimality of the network. As the approach enables (and indeed benefits from) the use of “what–if” type of experiments, management will be able to evaluate the various scenario in the altered environment.

7 Reference

ALSHAMRANI, A., MATHUR, K. and BALLOU, R.H., 2007. Reverse logistics: simultaneous design of delivery routes and returns strategies. *Computers & Operations Research,* **34**(2), pp. 595-619.

ANDRADÓTTIR, S., 1998. A Review of Simulation Optimization Techniques, D.J. MEDEIROS, E.F. WATSON, J.S. CARSON and M.S. MANIVANNAN, eds. In: *Proceedings of the 1998 Winter Simulation Conference,* 1998, pp151-158.

AZADIVAR, F., 1999. Simulation Optimization Methodologies, P.A. FARRINGTON, H.B. NEMBHARD, D.T. STURROCK and G.W. EVANS, eds. In: *Proceedings of the 1999 Winter Simulation Conference,* 1999, pp93-100.

BARROS, A.I., DEKKER, R. and SCHOLTEN, V., 1998. A two-level network for recycling sand: A case study. *European Journal of Operational Research,* **110**(2), pp. 199-214.

CARUSO, C., COLORNI, A. and PARUCCINI, M., 1993. Regional urban solid waste management system: A modelling approach. *European Journal of Operational Research,* **70**(1), pp. 16-30.

CASTILLO, E.D. and COCHRAN, J.K., 1996. Optimal Short Horizon Distribution Operations in Reusable Container Systems. *Journal of the Operational Resarch Society,* **47**, pp. 48-60.

DEITEL, H.M., DEITEL, P.J. and NIETO, T.R., 1999. Visual Basic 6 : how to program. New Jersey: Prentice Hall.

DEKKER, R., FLEISCHMANN, M., INDERFURTH, K. and VAN WASSENHOVE, L. N., 2004. Reverse Logistics: Quantitative Models for Closed-Loop Supply Chains. 1 edn. Springer.

DORIGO, M. and GAMBARDELLA, L.M., 1997. Ant colonies for the travelling salesman problem. *BioSystems,* **43**, pp. 73-81.

DORIGO, M., MANIEZZO, V. and COLORNI, A., 1996. The ant systems: optimization by a colony of cooperative agents. *IEEE Transactions on Man, Machine and Cybernetics: Part B,* **26**, pp. 1-13.

DUHAIME, R., RIOPEL, D. and LANGEVIN, A., 2001. Value Analysis and Optimization of Reusable Containers at Canada Post. *Interfaces,* **31**(3), pp. 3-15.

FISHMAN, G.S., 2001. Discrete-Event Simulation: Modeling, Programming, and Analysis. 1st edn. New York: Springer.

FLEISCHMANN, M., BEULLENS, P., BLOEMHOF-RUWAARD, J.M. and VAN WASSENHOVE, L.N., 2001. The impact of product recovery on logistics network design. *Production and Operations Management,* **10**(2), pp. 156-173.

GEN, M., CHOI, J. and TSUJIMURA, Y., 1999. Genetic algorithm for the capacitated plant location problem with single source constraints, *Proceedings of seventh European congress on intelligent techniques and soft computing, Session CD-7,* 1999, .

GIANNIKOS, I., 1998. A multiobjective programming model for locating treatment sites and routing hazardous wastes. *European Journal of Operational Research,* **104**(2), pp. 333-342.

GOTTINGER, H.W., 1988. A computational model for solid waste management with application. *European Journal of Operational Research,* **35**(3),.

JARAMILLO, J.H., BHADURY, J. and BATTA, R., 2002. On the use of genetic algorithms to solve location problems. *Computers & Operations Research,* **29**(6), pp. 761-779.

KELTON, W.D., SADOWSKI, R.P. and SADOWSKI, D.A., 2007. Simulation with Arena. 4 edn. New York: McGraw-Hill.

KO, H.J. and EVANS, G.W., 2007. A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs. *Computers & Operations Research,* **34**(2), pp. 346-366.

KRIKKE, H., BLOEMHOF-RUWAARD, J.M. and VAN WASSENHOVE, L. N., 2003. Concurrent product and closed-loop supply chain design with an application to refrigerators. *International Journal of Production Research,* **41**, pp. 3689-3719.

KRIKKE, H.R., VAN HARTEN, A. and SCHUUR, P.C., 1998. On a medium term product recovery and disposal strategy for durable assembly products. *International Journal of Production Research,* **36**(1), pp. 111-139.

KROON, L. and VRIJENS, G., 1995. Returnable containers: an example of reverse logistics. *International Journal of Physical Distribution & Logistics Management,* **25**(2), pp. 56-68.

LACKSONEN, T., 2001. Empirical comparison of search algorithms for discrete event simulation. *Computers & Industrial Engineering,* **40**(1-2), pp. 133-148.

LIECKENS, K. and VANDAELE, N., 2007. Reverse logistics network design with stochastic lead times. *Computers & Operations Research,* **34**(2), pp. 395-416.

LOUWERS, D., KIP, B.J., PETERS, E., SOUREN, F. and FLAPPER, S.D.P., 1999. A facility location allocation model for reusing carpet materials. *Computers & Industrial Engineering,* **36**(4), pp. 855-869.

MIN, H., JEUNG KO, H. and SEONG KO, C., 2006. A genetic algorithm approach to developing the multi-echelon reverse logistics network for product returns. *Omega,* **34**(1), pp. 56-69.

PALMER, C.C. and KERSHENBAUM, A., 1995. An approach to a problem in network design using genetic algorithms. *Networks,* **26**(3), pp. 151-163.

REALFF, M.J., AMMONS, J.C. and NEWTON, D., 2000. Strategic design of reverse production systems. *Computers & Chemical Engineering,* **24**(2-7), pp. 991-996.

RUBIO, S., CHAMORRO, A. and MIRANDA, F.J., 2008. Characteristics of the research on reverse logistics (1995–2005). *International Journal of Production Research,* **46**(4), pp. 1099.

SALEMA, M.I.G., BARBOSA-POVOA, A.P. and NOVAIS, A.Q., 2007. An optimization model for the design of a capacitated multi-product reverse logistics network with uncertainty. *European Journal of Operational Research,* **179**(3), pp. 1063-1077.

SCHROER, B.J. and TSENG, F.T., 1988. Modelling complex manufacturing systems using discrete event simulation. *Computers & Industrial Engineering,* **14**(4), pp. 455-464.

SMITH, J.S., WYSK, R.A., STURROCK, D.T., RAMASWAMY, S.E., SMITH, G.D. and JOSHI, S.B., 1994. Discrete event simulation for shop floor control, *Simulation Conference Proceedings, 1994. Winter,* 1994, pp962-969.

SPENGLER, T., PUCHERT, H., PENKUHN, T. and RENTZ, O., 1997. Environmental integrated production and recycling management. *European Journal of Operational Research,* **97**(2), pp. 308-326.

WOODRUFF, D.L., 1994. Simulated annealing and tabu search: Lessons from a line search. *Computers & Operations Research*, **21**(8), pp. 823-839.

ZHOU, G. and GEN, M., 1999. Genetic algorithm approach on multi-criteria minimum spanning tree problem. *European Journal of Operational Research*, **114**(1), pp. 141-152.

ZHOU, G., MIN, H. and GEN, M., 2003. A genetic algorithm approach to the bi-criteria allocation of customers to warehouses. *International Journal of Production Economics*, **86**(1), pp. 35-45.

Table 1: Parameters and Costs (in GBP) for the SME example

Description	***Parameter***	***Value***
Fixed installation cost per collection point	F_i^c	[4000, 8000]
Fixed setup cost per warehouse	F_j^w	[8000, 11000]
Fixed setup cost per plant	F_k^p	[32000, 40000]
Transportation costs per mile (to warehouse)	t^{cw}	0.008
Transportation costs per mile (to plant)	t^{wp}	0.005
Reprocessing cost per product	c^p	10.0
Disposal cost per product	c^d	2.5
Penalty cost per product	P	4.5
Collective cost of storage per product	T	3.0
Return per 1000 residents	R_i	0.1

Table 2: Reusability vs. Inspection Enabled Facilities

Probability of being reusable	0.90	0.80	0.75	0.70
Percentage of inspection enabled collection points in optimal configuration	0	9	61	83

Table 3: Parameters and Associated Costs in the Multiple Product Scenario

Description	***Parameter***	***Product 1***	***Product 2***
Transportation costs per mile (to warehouse)	t_l^{cw}	0.007	0.012
Transportation costs per mile (to plant)	t_l^{wp}	0.004	0.007
Reprocessing cost per product	c_l^p	14.0	20.0
Disposal cost per product	c_l^d	2.0	1.5
Penalty cost per product	P_l	4.5	4.0
Collective cost of storage per product	T_l	2.3	3.4
Return per 1000 resident	R_{il}	0.25	0.15

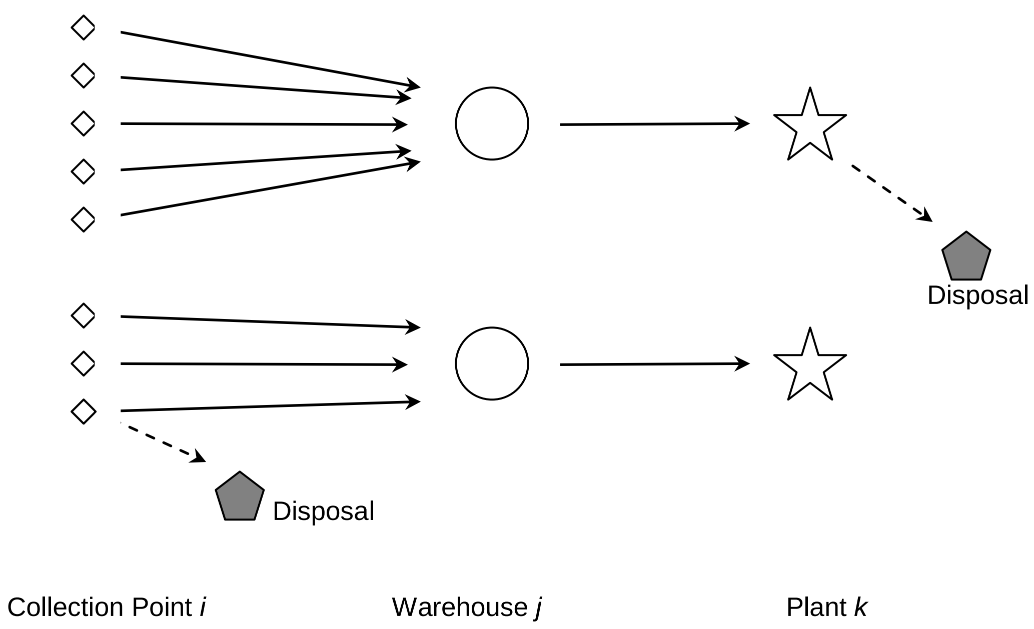


Figure 1: Recovery Network

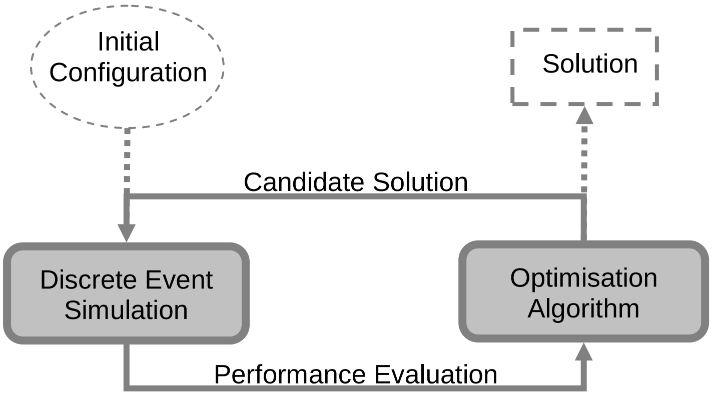


Figure 2: Simulation based optimization approach

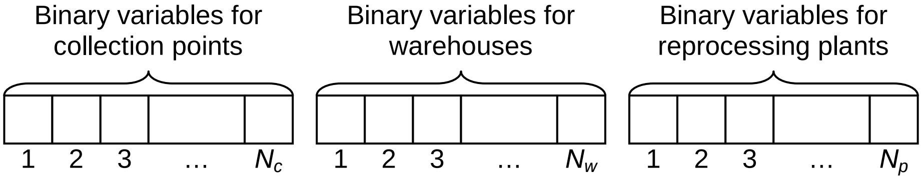

Binary variables for collection points
Binary variables for warehouses
Binary variables for reprocessing plants
1 2 3 … N_c
1 2 3 … N_w
1 2 3 … N_p

Figure 3: Genetic representation of the chromosomes

1	1	0	1	0	0	1	1	0	1	1	1

Parent 1

1	0	0	1	1	1	0	1	0	0	0	1

Parent 2

Crossover point

1	1	0	1	0	0	1	1	0	0	0	1

Offspring 1

1	0	0	1	1	1	0	1	0	1	1	1

Offspring 2

Single–point crossover operation

1	1	0	1	0	0	1	1	0	0	0	1

Offspring after crossover

0	0	1	0	1	1	0	0	1	1	1	0

Offspring after mutation

Inversion mutation operation

Figure 4: Genetic operations

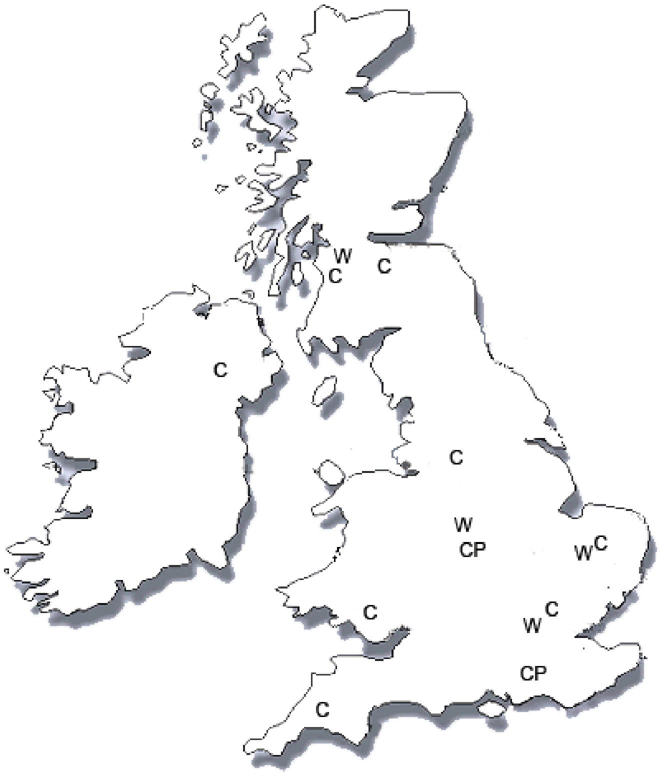
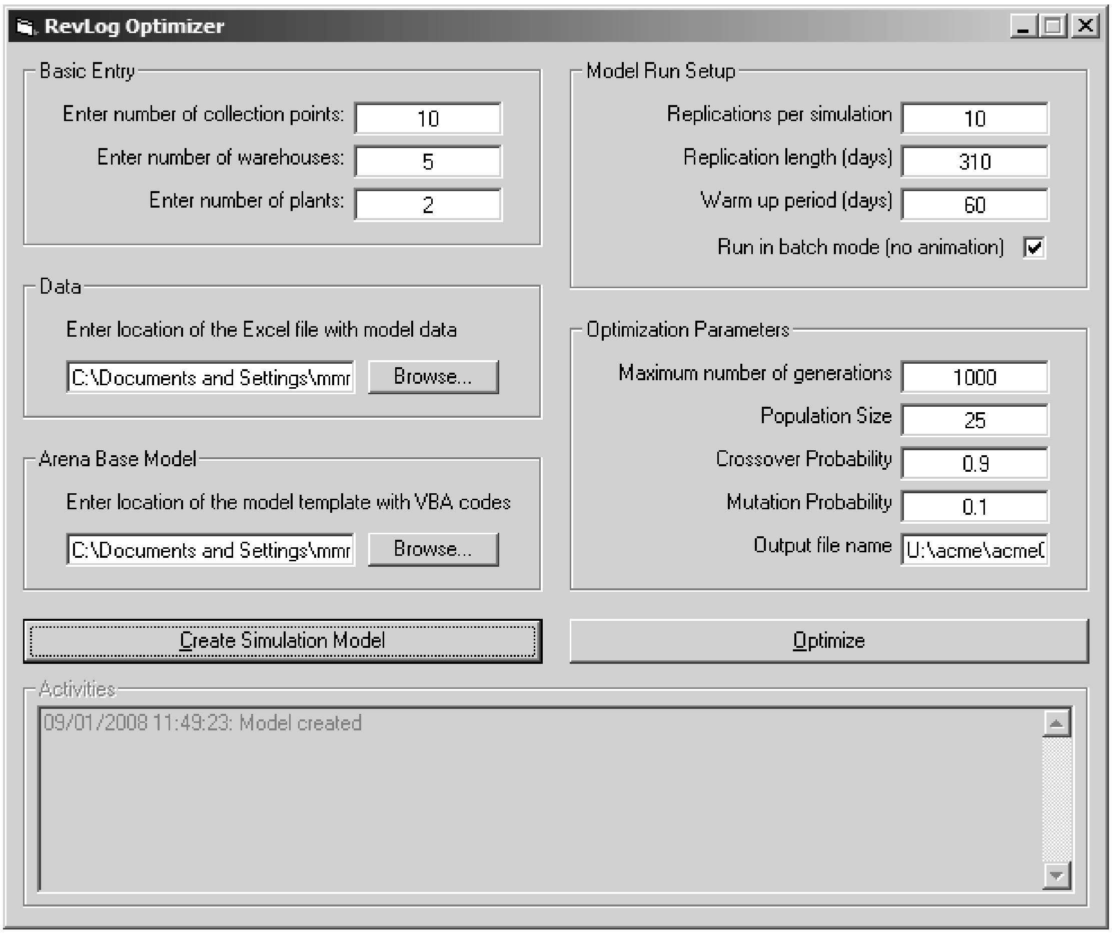


Figure 5: Location of candidate facilities (C: Collection Points, W: Warehouse, P: Plant)



RevLog Optimizer

Basic Entry

Enter number of collection points: 10

Enter number of warehouses: 5

Enter number of plants: 2

Data

Enter location of the Excel file with model data

C:\Documents and Settings\mmr

Arena Base Model

Enter location of the model template with VBA codes

C:\Documents and Settings\mmr

Model Run Setup

Replications per simulation 10

Replication length (days) 310

Warm up period (days) 60

Run in batch mode (no animation) ☑

Optimization Parameters

Maximum number of generations 1000

Population Size 25

Crossover Probability 0.9

Mutation Probability 0.1

Output file name U:\acme\acme(

Activities

09/01/2008 11:49:23: Model created

Figure 6: Screenshot of the optimization tool developed

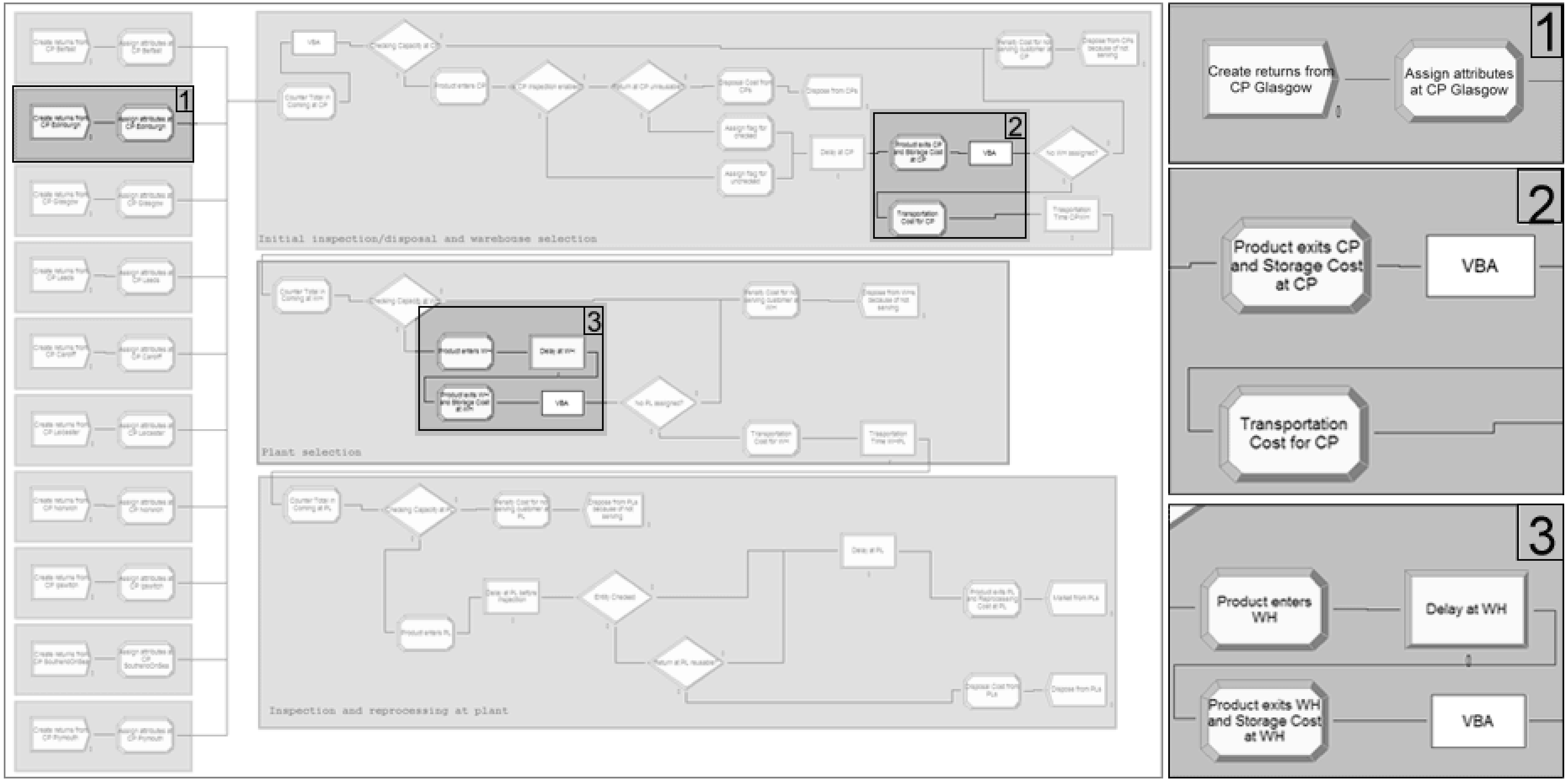

Create returns from CP Belfast
Assign attributes at CP Belfast
Create returns from CP Edinburgh
Assign attributes at CP Edinburgh
Create returns from CP Glasgow
Assign attributes at CP Glasgow
Create returns from CP Leeds
Assign attributes at CP Leeds
Create returns from CP Cardiff
Assign attributes at CP Cardiff
Create returns from CP Leicester
Assign attributes at CP Leicester
Create returns from CP Norwich
Assign attributes at CP Norwich
Create returns from CP Ipswich
Assign attributes at CP Ipswich
Create returns from CP SouthendOnSea
Assign attributes at CP SouthendOnSea
Create returns from CP Plymouth
Assign attributes at CP Plymouth
VBA
Checking Capacity at CP
Counter Total in Coming at CP
Product enters CP
Is CP inspection enabled
Return at CP unreusable
Dispose Cost from CPs
Dispose from CPs
Assign flag for checked
Assign flag for unchecked
Delay at CP
Product exits CP and Storage Cost at CP
VBA
No WH assigned?
Transportation Cost for CP
Transportation Time CPtoWH
Penalty Cost for not serving customer at CP
Dispose from CPs because of not serving
Initial inspection/disposal and warehouse selection
Counter Total in Coming at WH
Checking Capacity at WH
Product enters WH
Delay at WH
Product exits WH and Storage Cost at WH
VBA
No PL assigned?
Penalty Cost for not serving customer at WH
Dispose from WHs because of not serving
Transportation Cost for WH
Transportation Time WHPL
Plant selection
Counter Total in Coming at PL
Checking Capacity at PL
Penalty Cost for not serving customer at PL
Dispose from PLs because of not serving
Product enters PL
Delay at PL before Inspection
Entity Checked
Return at PL reusable
Delay at PL
Product exits PL and Reprocessing Cost at PL
Market from PLs
Dispose Cost from PLs
Dispose from PLs
Inspection and reprocessing at plant
1
Create returns from CP Glasgow
Assign attributes at CP Glasgow
2
Product exits CP and Storage Cost at CP
VBA
Transportation Cost for CP
3
Product enters WH
Delay at WH
Product exits WH and Storage Cost at WH
VBA

Figure 7: Screenshot of a model created by the optimization tool (1. Creation of entities and attribute assignment; 2. Assignments of various costs and VBA block for decisions and 3. Entities entering warehouse and process delays)

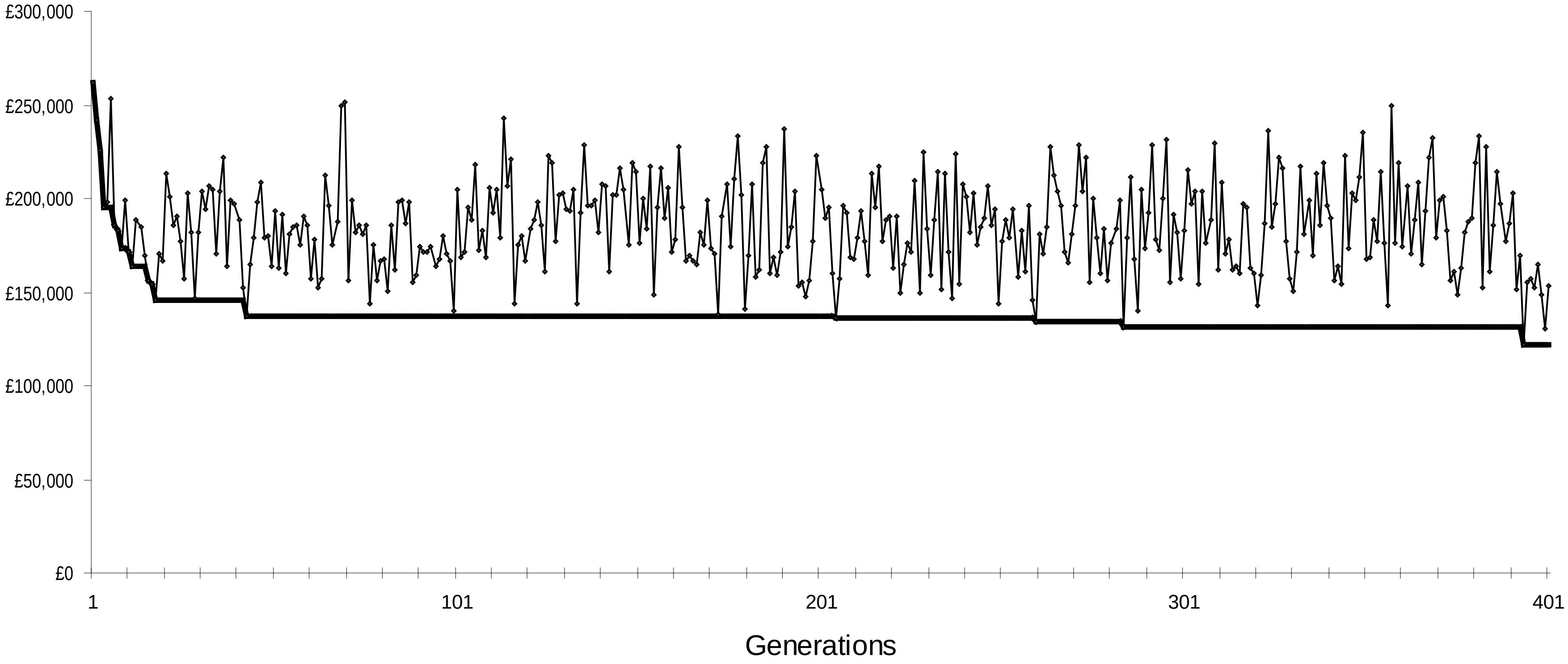


Figure 8: Costs of best solution in the populations and the convergance of solution over increasing generations

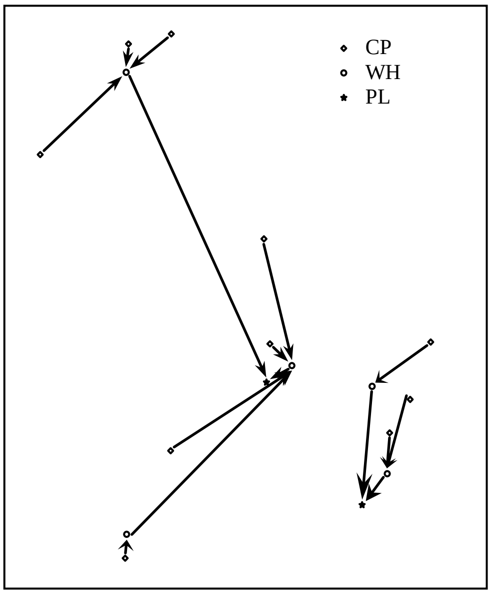


Figure 9: Initial configuration with all CPs inspection–enabled, all WHs working, all PLs working; cost: £233187

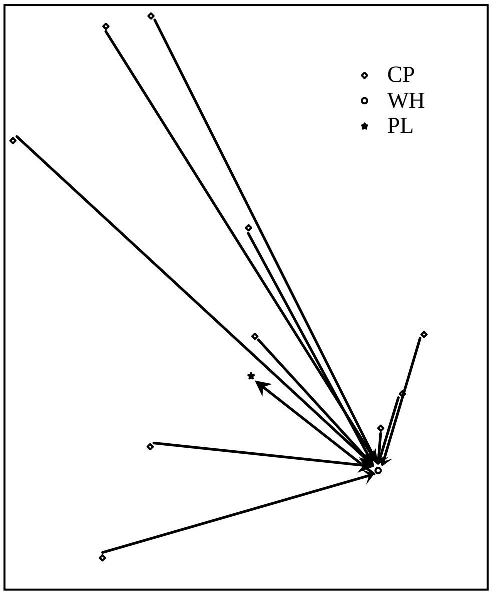


Figure 10: 284th generation; 2 inspection–enabled CPs, one WH working, and one WH working; cost: £131394

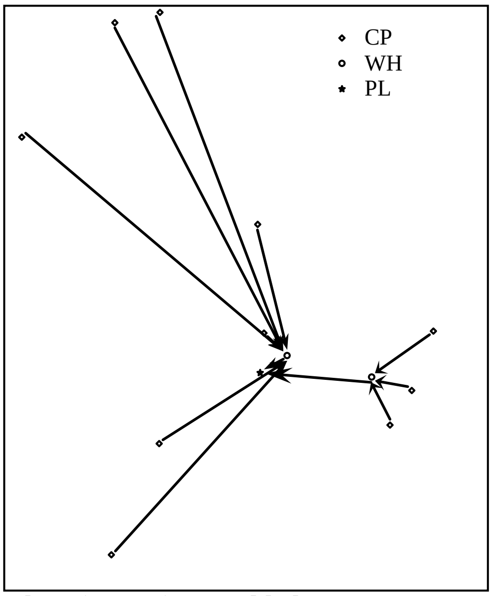


Figure 11: 394th generation; three inspection–enabled CPs, 2 WHs working, and one PL working; cost: £121564

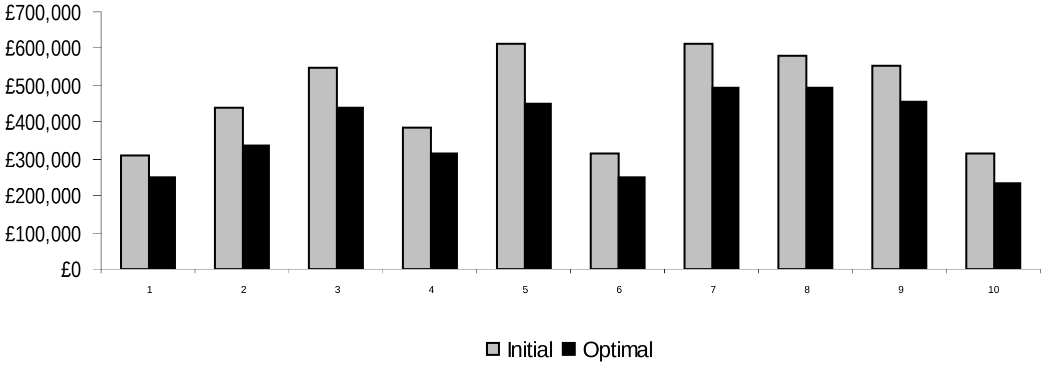


Figure 12: Costs associated with initial and optimal solutions

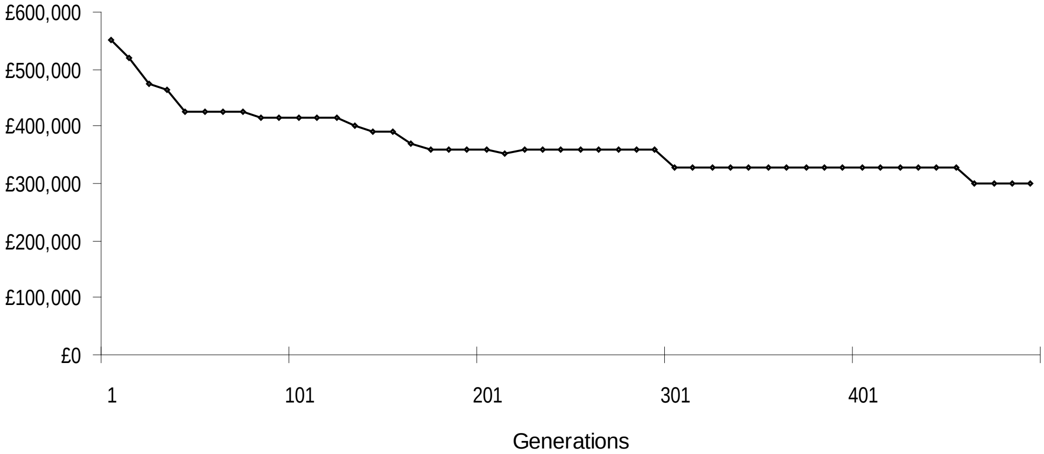


Figure 13: Costs of best solution so far at every 10th generations

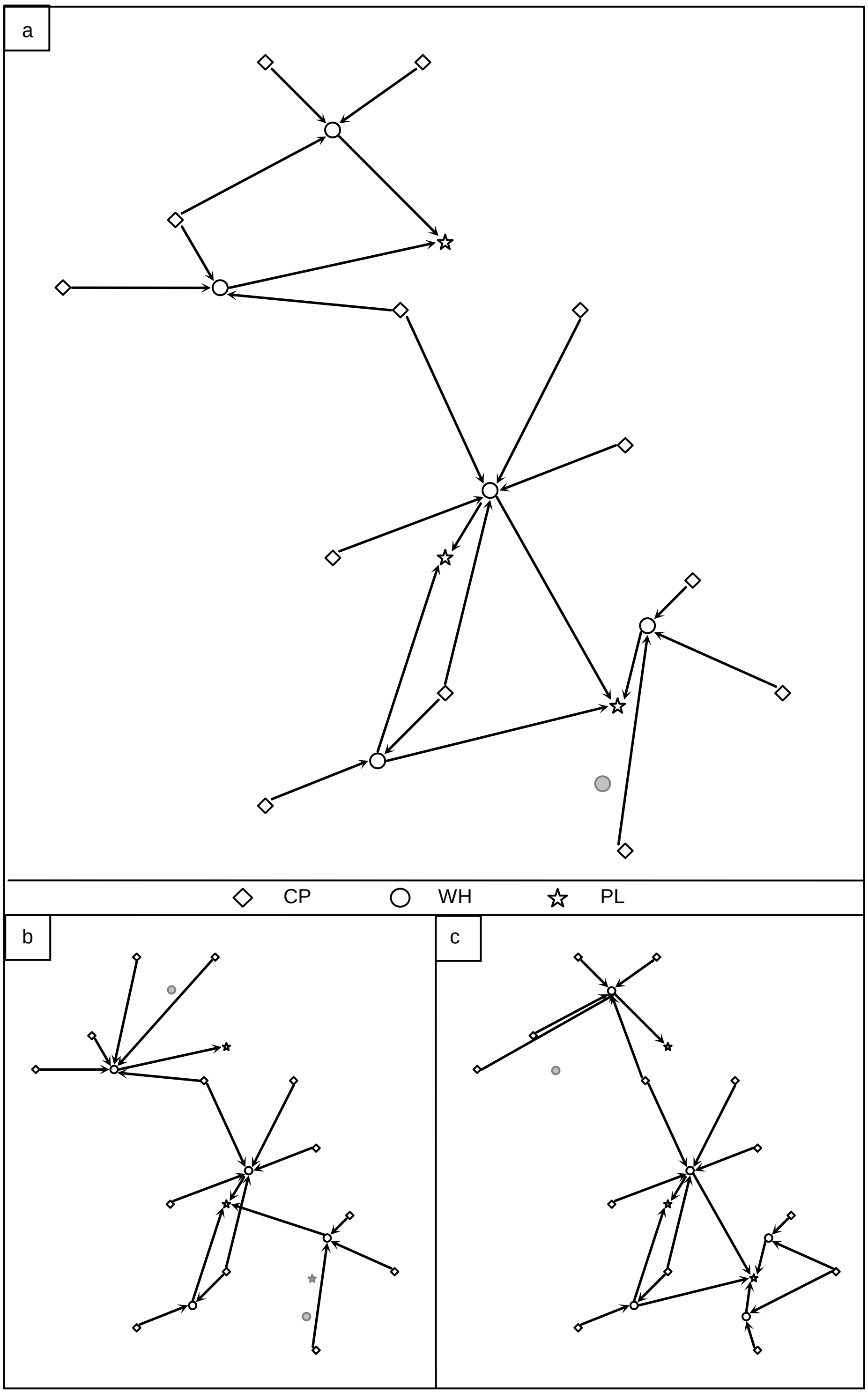


Figure 14: Optimised network configuration for a. two products, b. product 1 and c. product 2