Approach to designing reverse logistics network for product recovery

Ratih Dyah Kusumastuti

2006

Ratih, D. K. (2006). Approach to designing reverse logistics network for product recovery.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/6558>

<https://doi.org/10.32657/10356/6558>

Nanyang Technological University

Downloaded on 26 Apr 2023 23:34:22 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**AN APPROACH TO DESIGNING REVERSE LOGISTICS
NETWORK FOR PRODUCT RECOVERY**

**RATIH DYAH KUSUMASTUTI
SCHOOL OF MECHANICAL & AEROSPACE ENGINEERING
2006**

An Approach to Designing Reverse Logistics Network for Product Recovery

Ratih Dyah Kusumastuti

School of Mechanical & Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2006

ABSTRACT

Shortened product life cycle has resulted in an increasing number of obsolete products, leading to growing environmental concerns. One important way of minimising waste is to implement product recovery by means of repair, refurbishing, remanufacturing, cannibalisation or recycling. To effectively implement product recovery activities, a reverse logistics network design that can facilitate reverse flow of goods in an efficient way is required.

This research aims to develop an approach to the design of reverse logistics networks for discrete product recovery that considers certain characteristics that usually exist in real systems, but are considered only partially in existing models. The characteristics are, namely, modular product design and multi-product configuration, multi-objective functions, multi-period planning horizons and uncertainties. As the main driver of product recovery is environmental concerns, besides maximising net revenue, minimising environmental impact in terms of minimising energy consumption and minimising CO₂ emission are also considered as objective functions.

The approach consists of both mathematical and simulation models; the former utilizes mixed integer programming to model the multi-objective, multi-period problem of network design, whereas the latter is employed to model uncertainty. Spanning-tree based genetic algorithms are utilized to find non-dominated solutions. Genetic algorithm is chosen as it is known to be efficient in finding the set of non-dominated solutions for multi-objective optimisation problems in a single run. Preferred non-dominated solutions (selected based on decision

maker's preferences) are re-evaluated under several scenarios of uncertainty to determine the best-preferred reverse logistics network design. Therefore, the proposed approach reduces the complexities by only re-evaluating the preferred solutions.

A case study in a company (called Compucare, not its real name), with business in providing computer repair in the Asia-Pacific region is carried out. The proposed approach is applied to explore the possibility of implementing part recovery from used computers. Results show that implementing part recovery (to replace unrecoverable out-of-warranty faulty parts instead of using newly purchased parts) can generate additional net revenue for the company.

ACKNOWLEDGEMENTS

I would like to express my gratitude to both my supervisors, A/P Rajesh Piplani and A/P Lim Geok Hian for their invaluable guidance and support throughout the research. I also would like to thank Mr. Philip Saw and Mr. Tan Cheng Huat from Compucare for their support in the case study. I further appreciate the doctorate program at the School of Mechanical and Aerospace Engineering, for the financial support and facilities provided.

I owe a great deal to the three anonymous reviewers whose useful comments and thoughtful suggestions have improved the quality of this thesis. I also wish to thank Ms. Leong Yuit Lin and fellow students in the Centre for Supply Chain Management, for their support and suggestions during the course of compiling this thesis. To all my friends out there, thank you for all the encouragement that you all gave me at times when I needed the most. Lastly to my family, especially my husband, Irzan, thanking him for his continual encouragement, support and patience that he has given me, and helped me keep things in perspective.

TABLE OF CONTENTS

ABSTRACT	i
ACKNOWLEDGMENTS	ii
TABLE OF CONTENTS	iv
LIST OF FIGURES	viii
LIST OF TABLES	x
CHAPTER 1: INTRODUCTION	1
1.1. Reverse logistics definitions and characteristics	3
1.2. Designing reverse logistics network for product recovery	7
1.3. Research objectives	9
1.4. Research scope and methodology	10
1.5. Organising the thesis	11
CHAPTER 2: LITERATURE REVIEW	13
2.1. Product recovery	13
2.1.1. Drivers	13
2.1.2. Definitions	15
2.1.3. Remanufacturing	16
2.1.4. Product recovery activities	17
2.1.5. Implementation in industries	19
2.1.5.1. Computer industry	19
2.1.5.2. Single-use camera industry	21
2.1.5.3. Photocopier industry	23
2.2. Reverse logistics: issues and aspects	25

2.2.1. Inventory management with used parts	27
2.2.2. Production planning with used parts	25
2.2.3. Reverse distribution	26
2.3. Quantitative models of reverse logistics	30
2.3.1. Models of inventory controls for product recovery	30
2.3.2. Models of production planning with used parts	36
2.3.3. Models of reverse logistics network	39
2.4. Comparison of relevant reverse logistics network models	47
2.5. Summary	50
CHAPTER 3: MODELLING REVERSE LOGISTICS FOR PRODUCT RECOVERY	51
3.1. Model development	52
3.1.1. Single objective, multi-period model	52
3.1.1.1. Model assumptions	52
3.1.1.2. Model notation	57
3.1.1.3. Model formulation	59
3.1.1.4. A numerical example	65
3.1.2. Multi-objective, multi-period model	69
3.1.2.1. Additional assumptions	69
3.1.2.2. Additional model notation	70
3.1.2.3. Model formulation	70
3.1.2.4. Revisiting the numerical example	71
3.1.2.5. Solving the multi-objective problem	73
3.2. Summary	75

CHAPTER 4: THE METHODOLOGY	76
4.1. The methodology	76
4.2. Spanning tree-based multi-objective GA for reverse logistics Design	77
4.2.1. Genetic representation and chromosome feasibility	77
4.2.2. Genetic operators	83
4.2.3. Evaluation and selection	84
4.2.3.1. Decoding procedure	85
4.2.3.2. Fitness assignment methods	88
4.2.3.3. Preserving the non-dominated solutions	90
4.2.4. Overall procedure	91
4.3. Selection of preferred solutions	94
4.4. Simulation modelling	95
4.5. Determining the best-preferred solution	97
4.6. A numerical example	97
4.6.1. Problem description	98
4.6.2. Setting the GA parameters and determining fitness assignment Method	98
4.6.2.1. Comparing two fitness assignment methods and population sizes	99
4.6.2.2. Comparing different crossover and mutation rates	101
4.6.2.3. Comparing different numbers of generations	102
4.6.3. Determination of the best-preferred network design	103
4.7. Summary	108
CHAPTER 5: CASE STUDY AT COMPUCARE ASIA-PACIFIC	109
5.1. The company's background	109
5.2. Case study	111

5.2.1. The potential network for part recovery	111
5.2.2. Designing reverse logistics network for part recovery	115
5.2.2.1. Data	115
5.2.2.2. Finding non-dominated solutions & selecting preferred solutions	117
5.2.2.3. Determination of the best-preferred solution	122
5.2.2.4. Summary	126
CHAPTER 6: CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS	129
6.1. Conclusions	129
6.2. Contributions	131
6.2. Future research directions	132
LIST OF PUBLICATIONS	134
REFERENCES	135
APPENDIX A: Data for Numerical Example of Chapter 3	142
APPENDIX B: Result for Numerical Example of Chapter 3 (Single Period, Maximising Net Revenue)	150
APPENDIX C: Result for Numerical Example of Chapter 3 (Multi-period, Maximising Net Revenue)	156
APPENDIX D: Additional Data for Numerical Example of Chapter 3 and 4	163
APPENDIX E: Result for Numerical Example of Chapter 3 (Multi-period, Minimising Environmental Impacts)	164
APPENDIX F: Multi-objective Optimisation Methods	178
APPENDIX G: The Best Three Solutions of Numerical Example of Chapter 4	184
APPENDIX H: Data for Case Study of Chapter 5	199
APPENDIX I: The Best Three Solutions of Case Study of Chapter 5	206

LIST OF FIGURES

Figure 1.1: Reverse logistics flows in the supply chain	6
Figure 2.1: Product recovery chain	18
Figure 2.2: IBM reverse channel of used materials	21
Figure 2.3: Product process for single use camera	22
Figure 2.4: Recovery process for photocopiers	24
Figure 2.5: Framework of inventory management with returns	27
Figure 2.6: Framework of production planning with recovery activities	28
Figure 2.7: Reverse distribution	29
Figure 3.1: The reverse process	53
Figure 3.2.: The potential network	55
Figure 3.3.: Quantity decision variables	58
Figure 3.4: Potential network for numerical example	66
Figure 3.5: The optimal network for single period problem	67
Figure 3.6: The optimal network for multi-period problem	68
Figure 3.7: The optimal network that minimises the energy consumption	72
Figure 3.8: The optimal network that minimises the CO ₂ emission	73
Figure 4.1: The methodology	76
Figure 4.2: Representation scheme	79
Figure 4.3: Illustration of Prüfer number decoding procedure	82
Figure 4.4: Crossover process	83
Figure 4.5: Mutation Process	84
Figure 4.6: The overall GA procedure	93

Figure 4.7: Selecting preferred solutions	94
Figure 4.8: Determining best-preferred solution	97
Figure 4.9: Solutions found in different population sizes	100
Figure 4.10: Comparisons of fitness assignment methods	101
Figure 4.11: Comparisons of different crossover and mutation rates	102
Figure 4.12: Comparisons of generations	103
Figure 4.13: The best run	104
Figure 4.14: A depiction of Solutions 1 and 9	105
Figure 4.15: A depiction of Solution 2	105
Figure 5.1: Compucare's repair network	110
Figure 5.2: The potential network for part recovery	112
Figure 5.3: Potential locations of recovery facilities	114
Figure 5.4: The best runs	118
Figure 5.5: Network structure of Solutions 22 and 55	120
Figure 5.6: Network structure of Solutions 52	121
Figure 5.7: Simulation results (total net revenue)	124
Figure 5.8: Simulation results (energy consumption)	125
Figure 5.9: Simulation results (CO ₂ emission)	125
Figure A.1: Product configuration of PC1, PC2, and PC5	142
Figure A.2: Product configuration of PC3, PC4, and PC6	142
Figure F.1: GA Procedure	182

LIST OF TABLES

Table 2.1: Comparison study of relevant reverse logistics network models	48
Table 3.1: The payoff table	73
Table 4.1: Nine-scale scoring in paired-comparisons	94
Table 4.2: Comparisons of population size and fitness assignment method	99
Table 4.3: Comparisons of crossover and mutation rates	102
Table 4.4: Comparisons of generations	103
Table 4.5: Three solutions with the highest scores	105
Table 4.6: Scenarios of Uncertainty	106
Table 4.7: Description of each parameter variability	106
Table 4.8: Overall scores for preferred solutions (numerical example)	107
Table 5.1: Comparisons of generations in the case study	118
Table 5.2: Ten solutions with the highest score	119
Table 5.3: Scenarios of uncertainties	123
Table 5.4: Descriptions of each parameter in scenarios of uncertainties	123
Table 5.5: Overall score of preferred solutions	127
Table A.1: Used and new modules of PC1-PC6	143
Table A.2: Weight of used and refurbished products	143
Table A.3: Weight of used modules	143
Table A.4: Modified supply and demand data	144
Table A.5: Distance data	144
Table A.6: Transportation cost	145
Table A.7: Fixed costs to open new facilities or operate existing/outsourced facilities	146

Table A.8: Fixed costs to operate new facilities	146
Table A.9: Holding costs	146
Table A.10: Processing costs	147
Table A.11: Prices of refurbished products, recovered, new and recyclable modules	147
Table A.12: Capacity of all facilities	148
Table D.1: Energy, and CO ₂ emission for transportation	163
Table D.2: Energy consumption during processes	163
Table F.1: Conventional multi-objective optimisation methods	179
Table H.1: Product data	199
Table H.2: Supply at each primary market	200
Table H.3: Demand at each Compucare's RDCs	200
Table H.4: Growth rate of computer sale	200
Table H.5: Capacity at each potential collection centre	201
Table H.6: Capacity at each potential disassembly centre	201
Table H.7: Distance data	202
Table H.8: Consolidation volumes	202
Table H.9: Fixed costs at new CCs	203
Table H.10: Fixed costs at new DCs	203
Table H.11: Fixed costs at other CCs	203
Table H.12: Fixed costs at other DCs	204
Table H.13: Variable holding costs at potential DCs	204
Table H.14: Purchasing cost	204
Table H.15: Energy consumption at potential disassembly centres	205
Table H.16: Recyclability & recoverability fractions	205

CHAPTER 1

INTRODUCTION

Product life cycle especially for electronics products has shortened significantly in recent years, due mainly to rapid advances in technology. For instance, the typical lifecycle of personal computers (PCs) is approximately two to three years in the workplace and three to five years in the home (Boon et al., 2000). This results in an increasing number of obsolete products. It is estimated that 100 million units of cell phones were taken out of service in the US in 2003 (Drucker, 2004) and worldwide, there would be 120 million units of used computers ready for disposal in 2005 (Yamada et al., 2004). The circumstances along with limited resources have made product recovery gain its popularity in recent years.

Product recovery encompasses the management of all used and discarded products, components, and materials (Thierry et al., 1995). It aims to minimise the amount of total waste sent to landfills by recovering materials and parts from old or outdated products by means of recycling and remanufacturing, including reusing parts and products (Gungor and Gupta, 1999). The implementation of product recovery by manufacturers or other independent parties are motivated by three reasons, namely legislation by governments in several countries, the opportunity to recover hidden economic values in used products, and finally the customers' expectation of "greener" products.

In recent years, government regulations in certain countries have charged manufacturers with responsibility for the whole product life cycle that includes product safe disposal (extended producer responsibility, EPR). Legislations in the

European Union oblige its members to recover and recycle many products and components, such as containers, glass, paper, plastic, consumer goods, automotive components, electronics, etc (Bautista and Pereira, 2005). Japan also enacted EPR law for large electrical home appliances (Spicer and Johnson, 2004) and Taiwan passed recycling regulation that mandates home appliances and computer manufacturers to take back their products (Shih, 2001). While legislations in the aforementioned countries mandate manufacturers to do product recovery, legislation in the United States, on the other hand, tends to encourage indirectly, for instance by the use of tax credit (Guide et al., 2002).

To comply with government regulations and in recognition of the opportunity to recover economic values of used products, original equipment manufacturers (OEMs) have implemented product recovery activities. They include, Kodak in the single use camera industry (Guide et al., 2002), IBM, Dell and Hewlett-Packard in the computer industry (Sheu et al., 2004), Xerox and Canon in the photocopier industry (Guide et al., 2002; Fleischmann, 2000), Electrolux Europe in the home appliances industry (Istvan and Garamvolgyi, 2000), and General Motors and BMW in the automotive industry (Dowlatsahi, 2000).

Besides government regulations and economic values, customers' expectation of greener products is also a motivation behind product recovery activities. Vandermerwe and Oliff (1990) observed that there is a shift in customer demands, i.e. customers show preference for pro-green firms and are prepared to pay more for a better environment.

The implementation of product recovery results in reverse flows of goods from consumers to manufacturers or other independent re-processors that require

special treatment, as forward channels are not ready to handle the reverse flows. Management of these reverse flows is addressed as reverse logistics.

Reverse logistics is an important part of product recovery. Hansen (2000) indicated that up to 70% of the recycling costs are allocated to logistics. To successfully exploit the opportunities to recover economic value from used products, OEMs need to set up and manage the reverse logistics network that facilitates the efficient reverse flows of returned goods.

However, product recovery is not the only activity that causes reverse flows of goods; other activities such as the return of unsold merchandise and the return of failed products will also cause the product to flow the other way. In the following section, definitions and examples of reverse logistics flows are discussed.

1.1. Reverse logistics definitions and characteristics

Murphy (1986) defined reverse logistics as the movement of goods from a consumer towards producer in a channel of distribution.

Carter and Ellram (1998) defined reverse logistics as reverse distribution that includes resource reduction, whereby reverse distribution is defined as the return, upstream movement of a good or material resulting from reuse, recycling, or disposal. The upstream movement can be associated with environmental as well as quality issues, and is often performed by new, auxiliary channel members. Resource reductions on the other hand is defined as the minimisation of waste which results in more efficient forward and reverse distribution process.

Rogers and Tibben-Lembke (1998) defined reverse logistics as a process of planning, implementing, and controlling the efficient, cost-effective flow of raw materials, in-process inventory, finished goods, and related information from the

point of consumption to the point of origin for the purpose of recapturing value or proper disposal.

Dowlatshahi (2000) defined reverse logistics as a process in which the manufacturer systematically accepts previously shipped products or parts from the point of consumption for possible recycling, remanufacturing, or disposal.

From the above definitions, it is noted that Carter and Ellram (1998) implicitly point out that goods can have an open or closed loop flow by stating that the process is often performed by a new or auxiliary channel. They also mentioned that the reasons behind reverse flows can be environmental or quality issues. Furthermore, they mentioned the purpose of reverse flows as reuse, recycling or disposal. Rogers and Tibben-Lembke (1998), on the other hand, specifically referred to the management of closed loop flows of goods and information, including reverse flows within each supply chain channel. They also mentioned the general purposes of reverse flows, which are to recapture value or proper disposal. Lastly, it can be noted that Murphy's (1986) and Dowlatshahi's (2000) definitions are a sub set of the Rogers and Tibben-Lembke's (1998) definition, by specifically referring to closed loop flows of goods, and its purpose.

Fleischmann (2000) generalised Rogers and Tibben-Lembke's (1998) definition by considering that used goods do not always flow back to the point of origin, and by adding the aspect of inbound logistics. He defined reverse logistics as a process of planning, implementing, and controlling the efficient, effective inbound flow and storage of secondary goods and related information opposite to the traditional supply chain direction for the purpose of recovering value or proper disposal. He further distinguished different categories of secondary goods flows,

namely

- End-of-use returns,
- Commercial returns,
- Warranty returns,
- Production scrap and by-products, and
- Packaging.

End-of-use returns denote the flows of goods that are disposed of after their use has been completed. This type of return includes the return of goods that have reached the end of their technical or economic ‘life’ (e.g., single-use cameras and toner cartridges) and the return of end-of-lease products (e.g., computers and photocopiers).

Commercial returns refer to product returns undoing a preceding business transaction. In this type of return, the buyer returns the products to the original sender for refund. Commercial returns primarily depend on the channel power of different supply chain parties and represent a transfer of financial risk from buyer to seller. This kind of return is particularly relevant for products involving a high risk of obsolescence, for example, due to seasonality (e.g., sun-screen products) or short product life cycle (e.g., personal computers and fashion clothes).

Warranty returns represent failed products that are returned to the original sender. This may involve products that fail during use and goods that are damaged during delivery. This kind of return also includes rotatable spare parts, and product recalls due to security hazards. Rotatable spare parts refer to parts that are recovered from used products, which may circle the repair network as long as the conditions are still good.

Production scrap and by-products refer to excess material in the production process, off-specification products that may be reworked to meet quality targets, and by-products of production process.

Lastly, packaging is another category of reverse logistics flows. The best-known examples of packaging are crates, refillable bottles, and reusable boxes. Reuse of these products is economically attractive since they can often be reused directly without major reprocessing except for cleaning. Figure 1.1 describes the overview of reverse flows and their position in the supply chain.

Figure 1.1: Reverse logistics flows in the supply chain (Fleischmann, 2000)

The forward channels are usually not setup to handle the reverse logistics flows because there are several characteristics that differentiate reverse from forward logistics. Articles by Pohlen and Farris (1992), Fleischmann et al. (1997), Krikke (1998), and Tibben-Lembke and Rogers (2002) have identified these characteristics.

First, returned goods do not necessarily flow backward through the same channel. Instead, a reverse channel may emerge to accumulate, transport, and

reprocess the returned products.

Second, the reverse flows do not follow clearly defined paths, as the final destinations of the returned products (e.g., reuse, remanufacturing, recycling, and disposal) can only be known once they undergo certain processes and tests.

Third, the structure of reverse logistics networks is convergent (from many sources to few demand points) as returned products from many different locations flow back into few collection and reprocessing locations.

Fourth, the reverse flows have a high degree of uncertainty in the supply of the returned products (in terms of time, quality and quantity).

Fifth, the end market for recovered products may not be well known as the market for the recovered products can be in the primary market or in the secondary market.

Sixth, reverse logistics is a combination of push and pull systems due to the fact that there are clients on both sides of the chain (suppliers of used products and consumers of recovered products).

Lastly, transformation processes tend to be incorporated in the distribution network, covering the entire 'production' process from supply (disposal) to demand (reuse), and operations are spread over a large number of echelons.

With regard to the abovementioned complicating characteristics, reverse logistics flows must be handled carefully, so that efficient flows of goods can be realised.

1.2. Designing reverse logistics network for product recovery

As earlier mentioned, product recovery has been implemented by several manufacturers in certain countries, especially in the electronic industry. However,

there is one important aspect that makes product recovery seem unattractive to manufacturers or other independent companies, which is the logistics costs. As used goods must be physically moved from end-users back to the manufacturers or other parties, the logistics costs may conflict with the opportunity to make revenues out of it and most importantly may outweigh the environmental benefit.

Furthermore, the frequent launch of new products to the market and the decaying value of recovered products cause the time to market the recovered products to be limited. Consequently, time plays an important role in product recovery, as the longer time to collect and reprocess will only cause lower value of the recovered products. Hence, proper design of the reverse logistics network is essential.

In designing the logistics network, there are several aspects that must be considered. First, in implementing the product recovery activities, manufacturers will have two options: either to implement the activities in the existing forward facilities (assembly plants, distribution centres, and retailers) or by building the dedicated facilities for product recovery. Implementing product recovery in existing forward facilities seems to be an economic option. However, the complexity that may arise is how to ensure that the existing facilities are sufficient and economically feasible to cover the entire recovery activities without compromising the service level of new product, since there is uncertainty both in the forward and reverse flow of the supply chain, such as uncertainty in the supply of used products and in the demand of recovered and new products.

Second, as most electronic product come in multi-product configurations and modular product design, these product characteristics are need to be considered in the

network design. Modular product design refers to product design that separates the composition of end products into parts or subassemblies, some of which are common to all product options, while others are not.

Third, besides the usual maximisation of profit or minimisation of cost, it is preferred that the reverse logistics network design also considers the minimisation of environmental impact, in order to build the green image in assisting the marketing effort, and to comply with government regulations. Besides, investments in “green” can be resource saving, waste eliminating and productivity improving (Porter and van der Linde, 1995).

Lastly, Network design can be categorised as a strategic decision, hence the design should be valid for medium to long-term, despite the changes in the conditions. Therefore, designing reverse logistics for product recovery becomes a multi-objective, multi-period problem under uncertainty.

1.3. Research objectives

Research on reverse logistics has been done extensively in the past decade. It can be classified into development of theoretical framework and development of quantitative models for reverse logistics. Research in the latter area can be categorised into development of models for reverse logistics network design, inventory models in manufacturing/remanufacturing environment, and lastly, production planning in manufacturing/remanufacturing environment.

This research specifically aims to:

1. Understand the product recovery activities issues, especially the reverse logistics network design issue.
2. Develop a decision making approach to the design of reverse logistics network

for product recovery that considers the following characteristics:

- a. Multi-product configuration and modular product design
- b. Multi-objective functions. The approach considers maximising profit and minimising environmental impacts in terms of minimising energy consumption and CO₂ emission as objective functions.
- c. Multi-period planning horizon.
- d. The possibility of having resource sharing between forward and reverse flows, the possibility of performing several recovery activities in one recovery facility, and the flexibility of opening new recovery facility as needed at any period in the planning horizon.
- e. Uncertainty.

1.4. Research scope and methodology

This research is concerned with decision making in reverse logistics for product recovery, especially in developing a decision making approach to the design of reverse logistics network. In the approach, only discrete products (such as electronic products) are considered as the source for product recovery activities. Furthermore, only part recovery and product refurbishment are taken into account as the options for recovery activities.

This research is conducted by:

- Reviewing existing literature pertaining to product recovery, its implementation in several industries, reverse logistics issues and aspects, quantitative models of reverse logistics, especially reverse logistics network models and their solution approaches.
- Developing an approach for the design of reverse logistics network for discrete

product recovery that takes into account the aforementioned characteristics. The approach would consist of mathematical and simulation models. Mathematical model in the form of mixed integer programming (MIP) would be used to model the multi-objective, multi-period problem of the network design, whereas spanning-tree based genetic algorithms (st-GA) would be employed to find the set of non-dominated solutions for the problem. Simulation models, on the other hand, would be used to capture uncertainty.

- Applying the proposed approach to a case study on part recovery for a company specialising in providing repair service for computers and their peripherals.

1.5. Organisation of the thesis

The thesis is organised as follows. Chapter 1 provides an introduction to product recovery, various definitions and characteristics of reverse logistics, the complexities in designing reverse logistics network for discrete product recovery, research objectives, and research scope and methodology.

Chapter 2 reviews existing literature related to product recovery activities, reverse logistics issues and aspects, quantitative models of reverse logistics for product recovery, especially the reverse logistics network models and comparison study of the relevant reverse logistics network models.

Chapter 3 explains the modelling process of reverse logistics network design for discrete product recovery, which includes model development and a numerical example.

Chapter 4 discusses the approach to the design of reverse logistics network when multi-objective, uncertainty and the aforementioned characteristics are considered, and presents a numerical example to verify the approach.

Chapter 5 explains a case study in a company that offers repair services for computers. The possibility of implementing part recovery in the respective company is explored by applying the developed approach.

Finally, Chapter 6 provides the conclusions of this research, the contributions and the recommendations for future research directions.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews the existing literature related to product recovery and reverse logistics issues and models. The focus is placed on literature related to quantitative models of reverse logistics, especially reverse logistics network modelling. Thereafter, comparison study of relevant reverse logistics network models is presented.

2.1. Product recovery

In this section, the drivers and definitions of product recovery are presented. Remanufacturing is also addressed followed by product recovery activities and implementations in several industries.

2.1.1. Drivers

Waste reduction has received growing attention in the industrialised countries due to the depletion of landfill and incineration capacities. To overcome the waste problem, several countries including the European Union, Japan and Taiwan have enforced environmental legislation, charging producers with extended producer responsibility (EPR). EPR can be implemented through the following approaches (Spicer and Johnson, 2004):

- OEM take-back, in which the producer takes direct responsibility,
- Pooled take-back, in which the responsibility is shared through a consortium of producers,
- Third party take-back, in which contracted product responsibility providers assume responsibility of end-of-life (EOL) on behalf of the producers.

Another approach to the end-of-life (EOL) management issues is charging the responsibility for disposing the used products to the final owners (Boon et al, 2000). For example, in the United States, several states like Massachusetts, Minnesota and Wisconsin have either banned or are considering banning the dumping of computer-related equipment in their landfills (Stough and Benson, 2000). Therefore, the final owners must find a way to dispose their equipment properly, usually by asking a company specialising in handling EOL products to dispose it properly for a certain disposal fee.

Even if the environmental legislation is less strict, customers' expectations of green product impose strong pressure on companies to take environmental protection into account (Vandermerwe and Oliff, 1990). Kopicki et al. (1993) mentioned that there were three approaches in environmental management: reactive, proactive and value seeking approach. In the value-seeking approach, the perspective of "greening" changes from a burden to a potential source of competitiveness (van Hoek, 1999). The basic reasoning of greening as a competitive initiative is that investments in greening can be resource saving, waste eliminating and productivity improving (Porter and van der Linde, 1995).

Besides the above considerations, economic reason is also a strong motivation behind product recovery. The aim is to capture the value still encapsulated in used products, where overhauled products can be used as spares or sold to the secondary markets while requiring only a small fraction of the original production costs for repair (Fleischmann et al., 1997). Overall, product recovery offers manufacturers potential advantages, including increased profitability through reduced requirements for materials and improved market share based on

environmental image (Ayres et al, 1997; Ferrer, 1997).

2.1.2. Definitions

Thierry et al. (1995) categorised the recovery processes into repair, refurbishment, remanufacturing, cannibalisation, and recycling. Description of each activity is given next.

- Repair: the purpose is to return the used products to the market. Product repair involves fixing and/or replacement of non-functioning parts, so that they are in working order.
- Refurbishment: the purpose is to bring used products up to a specified quality. Quality standards are less rigorous than those for new products. Refurbishing involves modular disassembly of used products, inspection of all critical modules, and replacing or fixing them.
- Remanufacturing: the purpose is to bring used products up to quality standards that are as rigorous as those for new products. Remanufacturing involves complete disassembly of used products, extensive inspection of modules and parts, and replacement of old and outdated modules and parts.
- Cannibalisation: the purpose is to recover reusable parts from used products or components. These parts are reused in the repair, refurbishing, or remanufacturing of other products and components.
- Recycling: the purpose is to reuse materials from used products and components. These materials can be reused in the production of original parts or other parts.

While Thierry et al. (1995) divided the recovery processes into five processes as mentioned above, Fleischmann et al. (1997) categorised the recovery processes simply as material recovery (recycling) and added value recovery (repair and

remanufacturing). Gungor and Gupta (1999) also classified the recovery processes simply as material recovery (recycling) and product recovery (remanufacturing). Several articles like Ferrer (1997) and White et al. (2003) described remanufacturing similar to refurbishment as explained by Thierry et al. (1995). Lastly, White et al. (2003) also mentioned demanufacturing, which referred to both recycling and cannibalisation.

Repair, refurbishment, and remanufacturing often lead to a closed-loop system, where the products are returned to the original producer, whereas recycling can be described as an open loop system, where the products do not return to the original producer but are used in other industries. Cannibalisation, on the other hand, may lead to both closed and open loop systems as good parts can be used within the producing company or sold to other parties.

2.1.3. Remanufacturing

Ferrer and Ayres (2000) investigated the impact of remanufacturing on the economy. They evaluated the impact of remanufacturing in a hypothetical situation where remanufacturing held a significant share of the economy, presently dominated by OEMs. Analysis with the French economy led to the following observations:

1. Remanufacturing activity promotes demand for labour.
2. Remanufacturing activity reduces the level of inter-industry transaction.
3. Suppliers of sectors subject to competition from remanufacturing sectors have their inter-industry transaction significantly reduced. The situation worsens if the sector is also subjected to remanufacturing activity.

Lund (1998) developed seven criteria for remanufacturability, which are:

1. The product is a durable goods item.

2. The product fails functionally.
3. The product is standard, and the parts are interchangeable.
4. The remaining value added is high.
5. The cost to obtain failed products is low compared to the remaining value-added.
6. The product technology is stable.
7. The consumers are aware that remanufactured products are available.

The average price of a remanufactured product is usually between 40% and 80% of the new product with an average price of 60% of the new product (Steinhilper, 2001). A survey done by Nasr et al. (1998) found that remanufacturing firms had high profit margins of around 20%. However, also based on a survey, Guide (2000) and Guide et al. (2000) identified seven complicating characteristics of remanufacturing, which are as follows:

1. Uncertain timing and quantity of returns.
2. The need to balance returns with demands.
3. The disassembly of returned products.
4. The uncertainty in materials recovered from returned items.
5. The requirement for a reverse logistics network.
6. The complication of material matching restrictions.
7. The problem of stochastic routings for materials for remanufacturing operations and highly variable processing times.

2.1.4. Product recovery activities

Fleischmann et al. (2000) identified several activities that are recurrent in the product recovery network, namely collection, inspection/separation, reprocessing, disposal, and redistribution. Collection refers to all activities making used products available

and physically moving them to some point where further treatment is done. Inspection/separation denotes all operations which determine whether a given product is in fact reusable and in which way. Inspection/separation results in splitting the flow of used products according to distinct product recovery options. Reprocessing means the actual transformation of used products into usable products again. Disposal is required for products that cannot be used for technical or economical reasons. Redistribution refers to directing reusable products to a potential market and to physically moving them to future users. The product recovery chain is shown in Figure 2.1.

Based on process orientation, Fleischmann et al. (2000) distinguished three types of product recovery network, namely bulk recycling network, assembly product remanufacturing network, and reusable item network. Bulk recycling network is concerned with material recovery from low value products. Assembly product remanufacturing network is concerned with the reuse on a product or part level of a relatively high value product. Reusable item network on the other hand is concerned with reusable items such as reusable packages.

Figure 2.1: Product recovery chain (Fleischmann et al., 2000)

Product recovery design system consists of the following decisions (Realf et al., 2000):

- Number and size of collection and processing sites.
- Allocation of functions in the reverse chain to geographic locations.
- Routes for products and materials to take through the potential task network.
- Modes of transport to be used to connect the physical sites.
- Amount of material to allocate to each potential end-use.

2.1.5. Implementation in industries

Product recovery has been implemented in several industries. This section discusses examples of product recovery implementation in computer, single use camera and photocopier industries.

2.1.5.1. Computer industry

Similar to Boon (2000), some studies have reported that the useful life of a PC is now in the three-year range (Grenchus et al., 2001). Driven by the opportunities in the secondary market and the economic value of used products, recovery activities have become an important part of the computer industry.

According to Ferrer (1997), owners dispose of their computers not because of breakdowns or excessive wear, but because of technological changes that push the market to another performance level. The consequences are:

1. The majority of computers arrive at the end of their first life fully operational. Some defects might occur, but most problems can be solved with the replacement of the failed module.
2. The mainstream market for computers is not interested in machines that are more than 2 or 3 years old.
3. Computers cannot be remanufactured into products with the same performance as the new ones on the market. Although the design allows some upgrading and

refurbishing, the product cannot evolve into a model comparable with ones launched even a few years after the introduction of the original unit.

Based on the above, there are two main options for computer recovery. These are:

1. Used computers can be remanufactured (refurbishing-oriented) to produce renovated computers (Ferrer, 1997).
2. Used computers can be demanufactured (recycling-oriented) for component or material recovery. The process includes partial disassembly and atomising assets to divert waste and recapture value wherever practicable (White et al., 2003).

The output of computer remanufacturing (refurbished computers) is in great demand from those who cannot afford or do not need to invest in the newest technology, such as schools, small businesses, and consumers in developing countries, whereas the output of computer demanufacturing (used components in good condition and materials) can be sold as spare parts or for maintenance purposes, or can be sold to other industries.

As an example of the real world practice, IBM has already implemented the product recovery activities through an organisation called Global Asset Recovery Services (GARS). The recovery activity at IBM's GARS was discussed in detail in Grenchus et al. (2001), and the demanufacturing process of IT equipment at IBM was presented in Grenchus et al. (1997). Fleischmann et al. (2003) presented the product flow for different return dispositions at IBM, which can be seen in Figure 2.2.

According to a survey by Wong (2001) in the US, other computer manufacturers namely Dell, Gateway, Hewlett Packard, Micron, and Sun

Microsystems have already implemented the “take-back” program. However, according to Knemeyer et al. (2002), success in the product recovery of EOL computers is highly dependent on ensuring sufficient input stream and a consistent and high quality stream of refurbished products.

Figure 2.2: IBM reverse channel of used materials (Fleischmann et al, 2003)

Furthermore, the value of computers drops rapidly as the technology ages fast even though most components remain operational far beyond their typical life cycle. A challenge facing computer renovation is improving the reverse logistics such that the refurbished computer becomes available to an even broader customer base in its limited marketing time (Ferrer, 1997).

2.1.5.2. Single-use camera industry

In the single-use camera recovery, the reusable parts, such as circuit board, plastic body and lens aperture of the returned camera are put back into production after inspection. The printed circuit board is the primary driver of the product. Used

boards are valuable for the manufacturer as long as the product design allows them to be reused, and have minimal salvage otherwise.

Several articles discuss the product recovery in the single-use camera industry. They include Toktay et al. (2000), which focused on the inventory management and Guide et al. (2002) that emphasised the production planning and control. Both articles used Kodak as the case study. Next, product recovery activity at Kodak is presented.

In response to heavy environmental criticism, Kodak started in 1990 to take back, reuse, and recycle its single-use cameras, which had originally been designed as disposable units (Fleischmann, 2000). Since 1990, Kodak reused 310 million cameras in over 20 countries (Guide et al., 2002). The product recovery process is depicted in Figure 2.3.

Figure 2.3: Recovery process for single use camera (Guide et al., 2002)

The product recovery process starts with consumers returning the cameras to the photofinisher to develop the films. The photofinisher then places the cameras into specially designed shipping containers and sends each batch to collection centres.

Kodak has entered into agreements with other manufacturers (Fuji, Konica, and others) so that single use cameras are sent to common collection centres. Thereafter, the accumulated cameras are sorted according to the manufacturer and model.

After sorting, the cameras are shipped to a subcontractor facility, whereby all packaging and batteries are removed and the cameras are cleaned and transferred to an assembly line for disassembly and inspection. Some parts are routinely reused, others, such as batteries, are replaced. After inspection, the subassemblies are sent to manufacturing facilities for final assembly. The final products, which contain remanufactured parts and recycled materials, are indistinguishable to consumers from products containing no reused parts. They are then distributed to retailers for resale.

The complexity of remanufacturing operations of single use cameras is very limited. However, the uncertainty and unobservability associated with return flows of used cameras (in terms of time, volume, and quality) gives additional complexities to production planning.

2.1.5.3. Photocopier industry

Major manufacturers of photocopiers are putting substantial efforts into remanufacturing of used equipment. Manufacturers report on annual savings due to remanufacturing, at the same time minimising waste. Several papers have discussed product recovery in photocopier industry: Thierry et al. (1995), focused on the strategic issues of product recovery management, Krikke et al. (1999) emphasised reverse logistics network design, and Guide et al. (2002) focused on production planning and control. The aforementioned articles used CopyMagic, Océ, and Xerox respectively, as case studies. Hereinafter, product recovery at Xerox is explained.

Xerox Europe accounts for 25% of Xerox worldwide business reporting US\$ 53 billion in revenue from sales, leasing and service (Guide et al., 2002). In 1991, Xerox Corporation set a goal to become waste-free. The photocopier recovery process is depicted in Figure 2.4.

The recovery process starts with the transfer of used products from customers to regional distribution centres. The accumulated products are then sent to a disassembly centre, where returned products are inspected and graded (based on economical use of returned products) into four categories, which represent the most economically attractive use of the returned products.

Figure 2.4: Recovery process for photocopiers (Guide et al., 2002)

Category 1 products are practically unused machines requiring only minor servicing. Category 2 machines are in good conditions; these machines require parts and components to be replaced during remanufacturing process. Category 3 machines are in good condition but not economically fit for remanufacturing, therefore, parts and components are stripped off the frame of the machines and

entered into reused parts inventories. Machines are categorised as category 2 or 3 based, in part, on the demand for remanufactured machines and the levels of reused parts inventories. Category 4 machines are economically fit only for materials recycling. The categorisation decision is based on a variety of factors, including the overall condition of the machine, the age of the machine, the demand for reused parts, and current inventory level of reused parts.

Repaired and remanufactured copiers are distributed to customers through Xerox's traditional forward logistics channel. Parts and components are distributed in machines containing reused parts, via the traditional forward channel and as spare parts via the service supply network. The service network is responsible for repair and maintenance activities and relies on a flow of quality remanufactured parts and components.

The major barrier to the success of Xerox's equipment remanufacture and parts reuse program has been the misperception among some customers that products made with recycled parts are inferior to those built from all-new parts (Guide et al., 2002).

2.2. Reverse logistics: issues and aspects

As can be seen from the above examples, product recovery implementation requires efficient reverse flow of used goods from customers to manufacturers or third party specialising in product recovery. Reverse logistics is described as the process of moving goods from their typical final destination for the purpose of capturing value or proper disposal (Rogers and Tibben-Lembke, 1998).

Several researchers have addressed issues related to reverse logistics. Wu and Dunn (1995) examined logistics issues relative to the natural environment and

discussed measures that can be undertaken to achieve a proactive environmental management focus. A thorough literature review on reverse logistics was compiled by Carter and Ellram (1998), based on which they proposed a model of the factors affecting a firm's reverse logistics, reverse logistics hierarchy, and a model of the drivers (motivations) and constraints of reverse logistics. Furthermore, Van Hoek (1999) highlighted the changes that are needed in the evolution from reverse logistics to green supply chains, while Ferguson and Browne (2001) examined the emerging issues in reverse logistics, in particular the information requirements for reverse logistics within the extended enterprise. Lastly, Tibben-Lembke (2002) studied the way reverse logistics is impacted by changes in sales over the product life cycles.

As mentioned earlier, product recovery can also be performed by third party companies. Krumwiede and Sheu (2002) proposed a decision making model to examine the feasibility of implementing reverse logistics at third party providers. Maede and Sarkis (2002) proposed a conceptual model for selecting third party reverse logistics provider, and Spicer and Johnson (2004) explored the option to outsource demanufacturing activities to a third party.

For closed-loop supply chains, Guide et al. (2000) discussed the distinguishing features of closed-loop supply chains, and Krikke et al. (2001c) extended the design principles in traditional logistics to closed-loop supply chains. Krikke et al. (2001b) discussed the success factors and information technology (IT) areas in the closed-loop supply chains, while Kokkinaki et al. (2001) presented the web-based system with business processes in the closed-loop supply chain and Guide et al. (2003) discussed the challenges of closed-loop supply chain. Lastly, quantitative modelling of all aspects in closed-loop supply chain is discussed in

Dekker et al. (2004).

2.2.1. Inventory management with used parts

The objective of inventory management that considers the reuse of components/materials is to control external component orders and the internal component recovery process to guarantee a required service level and minimise both fixed and variable costs (Fleischmann et al., 1997).

To integrate the return flow of used products into material planning, appropriate control mechanisms are required. The framework of inventory management with return flow is shown in Figure 2.5 and the effect of remanufacturing on inventory control in the hybrid manufacturing/remanufacturing (HMR) environment was investigated by Van der Laan (1997).

Figure 2.5: Framework of inventory management with returns (Fleischmann et al., 1997)

2.2.2. Production planning with used parts

The kind of problem that arises in production planning with used parts/materials and the adequacy of traditional production planning methods to deal with used parts and materials depend to a large extent on the specific form of reuse considered (Fleischmann et al., 1997). In the case of direct reuse whereby used products can be directly reused (possibly after cleaning or minor repair), no additional production

process has to be taken into account. However, in remanufacturing environment, individual repairs to used products and the coordination of several interdependent activities make production planning a highly sophisticated task (Lund, 1984). A framework for the activities involved in production planning with used parts is shown in Figure 2.6.

Guide et al. (2002) classified three kinds of remanufacturing environment based on the nature of process and product volume, just as in the manufacturing case. They are:

- Remanufacture-to-Stock (RMS),
- Reassemble-to-order (RATO), and
- Remanufacture-to-order (RMTO).

Figure 2.6: Framework of production planning with recovery activities (Fleischmann et al., 1997)

2.2.3. Reverse distribution

Reverse distribution is the collection and transportation of used products and packages (Fleischmann et al., 1997). It can take place through the original forward channel, through a separate reverse channel or through combinations of the forward and reverse channel, as shown in Figure 2.7.

Figure 2.7: Reverse distribution (Fleischmann et al., 1997)

According to Fleischmann et al. (1997), in an efficient reverse distribution channel, decisions have to be made with respect to actors in reverse distribution channel, functions have to be carried out in the reverse distribution channel along with the locations, and relation between the forward and the reverse distribution channel.

Several software tools have been developed for the purpose of building an efficient reverse logistics network structure. Hirsch et al. (1998) discussed a software tool that helped create a logistics infrastructure including the assessment of environmental impact. Nagel and Meyer (1998) discussed a software tool called EDS-Rlog to support planning of integrated and cost-optimised take-back and recovery systems. Hansen (2000) presented the importance of logistics network in product recovery activities and once again discussed the EDS-Rlog planning tool. Lastly, Istvan and Garamvolgyi (2000) presented a discussion on a special software tool developed by BAYLOGI (Bay Zoltan Foundation for Applied Research Institute of Logistics and Production Engineering) that aimed to help the management system at the collection sites and in the disassembly and recycling plants.

2.3. Quantitative models of reverse logistics

In this section, the literature on quantitative models in reverse logistics is reviewed. Discussion is based on three aspects of reverse logistics discussed earlier, which are inventory management and production planning with used parts, and reverse logistics network. The focus is placed on reverse logistics network models.

2.3.1. Models of inventory control for product recovery

Quite a few inventory models have been developed for a system with recovery option. The models were developed solely for remanufacturing environment or for systems with hybrid manufacturing/remanufacturing. Most models were related to inventory management and some models also dealt with the pricing of used and/or recovered products. Some inventory control models also considered the disposal of excess used items. Next, some of the recent inventory models are briefly discussed.

Van der Laan and Salomon (1997) considered a stochastic inventory system with production, remanufacturing and disposal operations, in which customer demands must be fulfilled from either manufacturing or remanufacturing. They extended the PUSH and PULL strategies to control a system in which all returned products were remanufactured and no planned disposals occurred, gave indications as to when and why planned disposals were economically beneficial, and compared the PUSH disposal strategy with PULL disposal strategy over different stages of product life cycle.

Van der Laan et al. (1999) proposed an inventory control policy for hybrid manufacturing/remanufacturing systems. In this system, both manufacturing and remanufacturing output can be used to satisfy demand. They compared the traditional systems without remanufacturing to PUSH and PULL controlled systems with

remanufacturing and presented some managerial insights, such as:

- It may be unwise to remanufacture all remanufacturables, even when the return intensity is lower than the demand intensity.
- Remanufacturing companies should attempt to keep uncertainty in the timing and quality of returned products as low as possible.
- Remanufacturing companies should keep track of correlations between product returns and product demands.

Richter and Sombrutzki (2000) extended the Wagner/Whitin algorithm to remanufacturing planning. The algorithm was applied to a reverse model with given demand for remanufactured products and to the model of alternate manufacturing /remanufacturing.

Teunter et al. (2000) proposed five different methods of setting the holding cost rates for non-serviceable, remanufactured and manufactured items. Each holding cost rate consisted of out-of-pocket holding rate and some opportunity cost rate depending on the method that was used and the interest rate. The proposed methods, incorporated in an average cost (AC) inventory model, were compared to discounted cash flow (DCF) model. For the PULL model, it turned out that one of the methods led to near-optimal policies from a DCF point of view.

Toktay et al. (2000) addressed the procurement of new components for recyclable products such as single-use camera. The objective was to find an ordering policy that minimised the total expected procurement, inventory holding, and lost sales costs. Distinguishing characteristic of the system were uncertainty and unobservability associated with return flows of used camera. They modelled the system as a closed queuing network, developed a heuristic procedure for adaptive

estimation and control, and used Kodak as a case study.

Inderfurth et al. (2001) proposed a periodic review model for product recovery with multiple remanufacturing options, which also included a disposal option and incorporated uncertainties in returns and demands for different serviceable options. The objective was to determine the quantity of reusable items that should be allocated to different remanufacturing options to minimise the total remanufacturing, disposal, inventory holding and backordering costs. The optimal policy parameters were determined by using a computational procedure.

Kiesmuller and Van der Laan (2001) developed an inventory model for a single, reusable product with dependent product demand and return. The model considered backordering and purchasing lead-time, and the probabilities of lost product and disposed product were assumed to be constant. The objective of the modelling process was to find an optimal order-up-to level that minimised the total cost. The optimisation model was then solved numerically.

Minner (2001) proposed a safety-stock planning approach for general supply chain networks with integrated return and reuse material flows for single product and multi-level bill of material (BOM). The integrated reverse material flows consisted of external product return and internal by-products whereby demands were assumed to be normally distributed and the product assumed to be randomly returned by the external customer. Correlation of return quantities in time and correlation between demands and returns were neglected. The objective was to place strategic safety stocks that minimised the inventory holding cost, and the model was solved heuristically.

Teunter (2001) proposed a deterministic economic order quantity (EOQ)

model for an inventory system with recoverable items. He considered general policies with alternating manufacturing and recovery batches, included disposal and allowed disposal rate to vary. He also used different holding cost rates for manufactured and recovered items as recovered items had a lower quality than new items and repair/refurbishing cost was much lower than manufacturing cost.

Fleischmann et al. (2002) developed a single item inventory model with continuous review policy with Poisson distributed demand and fixed procurement lead-time, and random returns that could be used to serve demand. Returns occurred unit-by-unit according to a Poisson process and independent of demand. The objective was to find an order policy that minimised the long-run expected average costs per time. Cost structure comprised fixed costs, linear holding costs, and backlogging costs. The optimal control parameters were computed by using a sequential procedure.

Dobos (2003) proposed optimal production-inventory strategies for a specific type of reverse logistics system. This system comprised two stores. Demand was satisfied from the first store, where manufactured and remanufactured items were stored. The returned products, on the other hand, were collected in the second store and then remanufactured or disposed. It was assumed that demand was a known continuous function over a given planning horizon, return rate of used items was a known function and there was a constant delay between use and return processes. The objective was to minimise the sum of squared deviation from inventory levels in stores, and the costs of manufacturing, remanufacturing and disposal.

Kiesmuller (2003a) proposed a new approach for inventory control of a stochastic recovery system with two stocking points and different lead times for

production and remanufacturing. The objective was to decide on the number of items to produce and the number of items to remanufacture which minimised the average cost. In this new approach, the decisions were based on two aggregate variables called inventory positions, which are defined differently for different lead times.

Kiesmuller (2003b) considered a recovery system consisting of manufacturing and remanufacturing facilities, inventory for returned and recoverable items, and inventory for serviceable items, with the possibility of disposing returned items. Kiesmuller determined the optimal costs of manufacturing, remanufacturing and disposal rates under the assumptions of linear cost structure, a finite planning horizon and deterministic and dynamic demand and return rates. Two classes of policies whereby backorders were forbidden and allowed were studied.

Fleischmann and Kuik (2003) proposed an optimal inventory control for a single inventory point facing independent stochastic demand and item returns. The model was a variant of a traditional stochastic single item inventory model with both positive and negative demand. Using results from Markov decision process, they showed average cost optimality of an (s, S) control policy in this model.

Mahadevan et al. (2003) proposed a periodic review, push inventory control policy for single remanufacturing facility with stockpile of returned products and stockpile of serviceable products (manufactured or remanufactured). Returns and demands were assumed to be Poisson distributed, with fixed manufacturing and remanufacturing lead-times. The authors assumed that all returns were fit for remanufacturing and also considered backordering. The objective was to choose an appropriate order-up-to-level that minimises the costs. Heuristics were developed to search for an appropriate order-up-to-level and simulations were used to test the

heuristics. Analytical method was used to narrow the simulation work, to develop bounds and heuristics.

Inderfurth (2004) proposed optimal policies in a hybrid manufacturing /remanufacturing system with product substitution. In this system, returned items which were not remanufactured were disposed of, serviceable products were sold at a type-specific price, products leftover after the demand period earned specific salvage values and excess stock of manufactured serviceable could be used to satisfy excess demand for remanufactured product to avoid stock-out costs. Manufacturing, remanufacturing and disposal were charged at a specific price. The objective was to determine the manufacturing and remanufacturing order quantities given an arbitrary starting inventory of serviceable products that maximised expected profit from all decisions.

Sheu et al. (2004) developed a multi-objective optimisation model for a logistics system where both forward and reverse chains existed. They considered a single product with the objective to maximise the systematic net profit as aggregated from chain based net profit associated with manufacturing supply chain and recovery supply chain. The decision variables were inventory amount associated with each member of the chain.

Teunter (2004) presented a lot sizing model for inventory systems with product recovery. It was assumed that recovered items were as good as new and satisfied the same demand as new items, and demand rate and return fraction were deterministic. The objective was to find the optimal lot sizes for production/procurement of new items and for the recovery of returned items.

Tang and Grubbstrom (2005) developed inventory models for systems with

stochastic lead times in manufacturing/remanufacturing environment. Firstly, they considered two sources for replenishing serviceable inventory (from manufacturing and remanufacturing), and provided a solution procedure when a cycle ordering policy was used. Secondly, they investigated the possibility of using dual sourcing ordering policy whereby each order was split between a manufacturing and remanufacturing process.

2.3.2. Models of production planning with used parts

The following are quantitative models of production planning with used parts, which can be classified into production planning and forecasting, recovery strategy decision-making models, and disassembly planning and scheduling.

Clegg et al. (1995) developed a linear programming model for production planning for remanufacturing with multi-products and two-level BOM, modular product design, and multi-period planning horizon. It was assumed that the company recovered the products from the field and had options of disposing of them, reusing some components after disassembly, and remanufacturing the product or assembling the product from scratch. New product assembly and remanufacturing were assumed to be carried out in the same manual assembly facility while the costs considered were production, disposal, and inventory holding costs. The objective was to determine a maximum-profit of production and procurement plan with the amount of products or components used in different options and inventory levels of various intermediate items as decision variables.

Kasmara et al. (2001) introduced a cyclic optimisation approach for optimal production planning in a remanufacturing/manufacturing production system. The basic idea was to simultaneously consider the whole cyclical process in the system

that caused the return of the end-of-use products, and to allow product release and return processes to be treated as variables to be determined in generating optimal production plan. Along with the remanufacturing/manufacturing activities, disposing of, or sending excessive returned parts to material recycler and the respective inventories were considered. They developed a linear programming model (LP) with maximisation of profit throughout production planning period as the objective, and sales, assembly, disassembly, sending parts to recycler, manufacturing, and remanufactured parts to be used in assembly as decision variables. The model was developed and solved using mathematical programming tool XPRESS-MP release 10.

Marx-Gomez et al. (2002) proposed a three-stage approach to forecast returns based on life cycle data and influencing factor using simulation approach and fuzzy and neuro-fuzzy techniques. Firstly, they developed a simulation model to generate reliable data to forecast time and amount of scrapped products returning to the recovery process. Secondly, they developed a rule-based fuzzy reasoning model for one-period-prognosis based on life cycle analysis and the previous simulation results, and lastly, they applied neuro-fuzzy approach for multi-period prognosis. The approach was applied to the photocopier industry.

Krikke et al. (1998a) proposed a model that determines an optimal product recovery and disposal strategy for one product type. The objective function included technical, commercial, and ecological criteria as well as uncertainty of those criteria regarding the quality level of the returned product. Optimisation was conducted on overall net profit using a two-phased dynamic programming (DP) algorithm.

Krikke et al. (1998b) extended the previous model so that it could be used to

determine a recovery strategy for multiple-type consumer products. The aim was to find an optimal balance between maximising net profit and meeting constraints such as recovery target, limited market volumes, and processing capacities. Later, Krikke et al. (1999a) applied the optimisation models to a business case (monitors recovery) with the assumption that various types of returned products belong to one product group.

Johnson and Wang (1995) proposed a methodology for the evaluation and optimisation of the disassembly process for material recovery. The methodology consisted of a four-level disassembly analysis:

- Level 1: Feasibility study of material recovery opportunities (MRO)
- Level 2: Optimal disassembly sequence generation
- Disassembly optimisation
- Design for disassembly (DFD) guidelines

Four criteria were used to optimise the generation of disassembly sequence, namely material compatibility, clustering for disposal, concurrent disassembly operations, and maximisation of yield.

According to Penev and De Ron (1996), the disassembly problem comprised different significant items that should be resolved:

- Specifying the valuable and hazardous materials and components per product;
- Maximising the profit and minimising the cost of execution of the disassembly operations;
- Finding a less expensive way to remove the hazardous components and to recover the valuable ones;
- Sending the product to a shredder as soon as no profit can be realised by

executing disassembly operations.

Hence the disassembly process could be formulated as finding the best solution when all of the aforementioned items were considered simultaneously. They used graph theory and dynamic programming (DP) to generate and evaluate the feasible disassembly plans and an economic model was introduced to determine the optimal level of disassembly.

Guide et al. (1997) studied a remanufacturing facility and examined two key activities in planning and control, namely disassembly release mechanisms (DRMs) and priority dispatching rules (PDRs) using a simulation model.

Gungor and Gupta (1998) proposed a methodology to develop a framework for dealing with uncertainty in disassembly sequence planning (DSP) implementation, which were defective parts in the returned products, upgrading/downgrading the products during consumer use and disassembly damage.

Gupta and Veerakamolmal (2000) developed a procedure to determine the disassembly schedule for products. The procedure included an integer programming (IP) model to determine the lot sizes of product for disassembly to be obtained from the distributor, and net profit/loss from resale and disposal. They considered multi-products with multi-level BOM, and multi-period planning horizon. The procedure followed the material requirement planning (MRP) logic.

2.3.3. Models of reverse logistics network

Most reverse logistics network models were specifically developed for product recovery while other models were developed for waste management or for managing product returns. Some models were also developed for closed-loop supply chain problems and others were developed especially for reverse supply chains only.

Furthermore, most mathematical models were usually an extension of facility location-allocation model in the form of mixed integer programs (MIP) and either commercial software or heuristics was used to solve them. Others employed genetic algorithms to solve the models. Some of the recent models are briefly explained next.

Gottinger (1988) proposed a single product, single period, deterministic MIP model for solid waste management. The network model consisted of three stages namely sources, intermediate processing facilities, and landfills. The objective was to minimise the total costs, whereby locations and flows of waste between locations were the decision variables. Pseudo-facilities were used to represent the existing facilities or new facilities that might be built. Procedure for solving the model was developed using the branch and bound technique.

Caruso et al. (1993) proposed a location-allocation model for planning the urban solid waste management system (USWMS). The proposed model was for a single product, with single-period and deterministic supply of waste, and was formulated as MIP with multiple objective functions namely cost, wastes of resources, and environmental impact. The model was solved using a heuristic technique and applied to the case of the Italian region, Lombardy.

Kroon and Vrijens (1995) presented a quantitative model to plan a reverse logistics system for returnable containers. Five parties were involved in the system, namely the central agency, a logistics service organisation, the senders of the containers, the recipients, and the carriers that actually transport the containers from senders to recipients. The agency was the owner of the containers, and most of the logistics operations were subcontracted to the logistics service organisation. They developed a single-product, single-period, deterministic MIP model to decide on the

number of containers that circulate in the network and which distribution centre should be used as a depot to minimise the total logistics cost.

Spengler et al. (1997) developed a strategic recycling management system (SRMS) for the iron and steel industry. SRMS was a multi-product MIP-model based on the multistage capacitated warehouse location problem (WLP). In the WLP, demand was fixed and production had to be optimised while in the SRMS, production was fixed and recycling options that corresponded to demand were capacitated. The objective was to determine the locations to be opened and how flows were routed from the sources through the intermediary facilities to the sinks, to minimise the total costs (transportation costs and processing costs). The model was solved using the commercial solver GAMS/OSL.

Barros et al. (1998) proposed a two-level location model for sand recycling and optimised it using heuristic procedures. The sand recycling problem could be described as follows: given the potential locations of regional depots and treatment facilities, the capacity of these facilities, the cost structures (transporting, processing and fixed cost) and the location of the sieved sand as well as the sites of the projects and their respective availability and demand for sand, determine which facilities should be built and how the sand should be classified, stored, cleaned and delivered to the projects so that the total cost of the system was minimised. The model was formulated as a MIP and solved using a heuristic procedure.

Jayaraman et al. (1999) proposed a multi-product, single-level BOM, and single-period deterministic MIP model called REVLOG, to design a reverse distribution network with a small electronics company as a case study. The reverse logistics network consisted of collection zones, distribution/remanufacturing

facilities, and customer outlets. The objective was to decide which set of distribution/remanufacturing locations to open and operate, what products to stock in those locations, and in what quantities, to minimise the total reverse logistics cost. Remanufacturing facility and storage/distribution centre for remanufactured products were assumed to share the same location, and GAMS was used to obtain the optimal solutions.

Krikke et al. (1999b) proposed a single-product, single-level BOM, single-period, deterministic MIP model for reverse logistics network redesign for copiers. The reverse logistics network consisted of dismantling, preparation, and reassembly facilities. The objective was to decide the locations and the capacities of recovery process to be installed (when there were several options of locations for each recovery activities), which minimised the total costs (processing, transportation, and inventory cost). There were two major inputs into the reverse logistics network design: (1) the recovery strategies applied to return flows (represented by a processing graph for each product), and (2) the available logistics system that could potentially be used in the reverse logistics network design (represented by a transportation graph). The transportation graph and the processing graph were combined into a network graph. The final design was to be found by the MIP model and the model was solved using the MIP solver LINDO.

Louwers et al. (1999) presented a facility location-allocation model for the collection, pre-processing, and redistribution of carpet waste. Their objective was to determine the physical locations where different activities should take place, the capacity of the facilities at those locations for storing and pre-processing the disposed carpets, and the allocations of disposed carpet waste flow to those locations

and facilities as well as the transportation mode to be used. The locations were not preset but calculated, and expenses for buildings and buying facilities were explicitly taken into account using linear depreciations. The model was solved using standard software.

Realff et al. (1999) proposed a deterministic multi-period MIP model for reverse logistics network design for carpet recycling. The carpet recovery activities included collection, storage, sorting and separating, material recycling, and transportation. The objective of the model was to choose recovery sites to perform the above activities and the allocation of goods between sites that maximised net revenue, with capacity at all sites, the transportation mode, and maximum amount of goods that can be sold at all sites as constraints. Later, the model was used by Realff et al. (2000), who adopted a robust modelling approach, introduced by Kouvelis and Gang (1997), to incorporate uncertainties. In the robust modelling approach, all scenarios deemed potentially realisable were enumerated, and a solution was sought which was suitably close to the optimal solution for each scenario. FORTRAN 90 was used to model and solve the problem.

Based on the models that were developed for several case studies, Fleischmann et al. (2001) proposed a generic facility location model for recovery network design. They considered three intermediate levels of facilities, namely disassembly centres where inspection and separation functions were carried out, factories for reprocessing and/or new production, and distribution warehouses. Moreover, they considered two disposition alternatives, which were recovery and disposal. The generic model was a single-period, single-product, single-level BOM, and deterministic MIP model. The objective was to determine whether to open a new

facility (plant, warehouse, disassembly centre) and the flow of goods that minimised the total cost. The standard MIP-solver of CPLEX 6.0 based on LP relaxation was used to solve the model. Several extensions to the basic model such as the integration of forward and reverse locations, the combination of forward and reverse transportations, and the selection of recovery processing technologies were presented. Sim et al. (2004) extended the basic model by taking into account multi-product and multi-period aspects, and used LP-based genetic algorithm (GA) to solve the problem, by firstly relaxing the binary variables in the model and then solving the resulting LP-relaxed problem.

Krikke et al. (2001a) developed a quantitative model to design closed loop supply chains for refrigerator recycling which was a deterministic, multi-product, multi-level BOM, and single-period MIP model. Their objective was to determine the network structure (number, capacity and locations for supply chain process facilities combined with allocation of product flows in the system) that optimised deviational variables of the total supply chain costs, energy use and residual waste. In the scenario analysis, the first scenario covered a comparison of a centralised and decentralised network, while the second scenario compared the results for alternative product designs. For each scenario, they considered ten cases with different weights for supply chain costs, energy use and residual waste in the objective function.

Shih (2001) developed a multi-product, single-period, deterministic MIP model for reverse logistics system planning for electrical appliances and computer recycling in Taiwan. The reverse logistics network consisted of collection points, storage sites, disassembly/recycling plants, and material market/final treatment. The objective was to determine the optimal network flow and the number and locations of

the new facilities that minimised the total costs. Scenario analysis with different take-back rates and storage sites sharing conditions were presented.

Jayaraman et al. (2003) proposed a reverse distribution model for hazardous products where the reverse distribution network consisted of origination sites, collection sites, and refurbishing facilities. The model was a multi-product, single-level BOM, single-period, and deterministic MIP model with cost minimisation as the objective function, and location and allocation of flows between facilities as the decision variables. A procedure to obtain the best solution was developed and AMPS as front-end of CPLEX was used to solve the MIP model. The solution procedure had three components, which were the random selection of potential collection sites, heuristic concentration section, and the heuristic expansion component. In the second stage, the best solution found in the first stage is re-evaluated by including the frequently used collection and destination sites, whereas in the third stage, the best solution found in the second stage is re-evaluated iteratively by adding one collection or destination site not chosen in the best solution.

Schultmann et al. (2003) proposed a hybrid approach to the design of closed-loop supply chains for used batteries. The approach consisted of an optimisation model for planning the network and flow-sheeting process that enabled a simulation tailored to potential recycling options. Facilities considered in the model were collection points, battery sorting facilities and recycling/disposal sites. The optimisation model was a two-stage facility location allocation model, and the objective was to determine locations of battery sorting facilities and allocations between facilities that minimised the total fixed and transportation costs.

Beamon and Fernandes (2004) developed a model to design closed-loop

supply chain networks for manufacturers that produce new products and remanufacture used products. Single type of product was considered. The decisions to be made were: which warehouses and collection centres should be open, which warehouses should have sorting capabilities and how much material should be transported between each pair of sites. A multi-period MIP was developed with minimising investment and operational costs as the objective function. The present worth method was employed to jointly analyse the investment and operational costs.

Min et al. (2004) proposed a GA approach to the design of reverse logistics networks for product returns. Firstly they proposed a nonlinear MIP model to represent the network design problem of product returns. Two facilities were considered in the model, namely initial collection points and centralised return centres. The objective of the model was to determine the location of all facilities and the allocations between facilities that minimised the reverse logistics costs.

Bautista and Pereira (2005) proposed reverse logistics network models for urban waste management. Two formulations of unicost set covering problem were developed. The objective was to locate a minimum set of waste collection points so that each citizen was able to encounter a collection point within a certain distance of his/her home. They used GA and GRASP heuristics to solve the models, whereby the latter is a constructive multi-start procedure allowing the generations of multiple solutions for a single problem.

Listes and Dekker (2005) developed a stochastic programming model to handle uncertainties in product recovery network design. They developed a deterministic model and later extended it to a two-stage stochastic model to approach locational uncertainty of demand. The first stage corresponded to the investment that

must be made for opening facilities prior to the actual realisation of the random parameters. The second stage corresponded to the allocation of flows through the established network after the values of the random parameters were known. To cope with the possible combinations of supply-demand realisation, they further developed a three-stage stochastic programming model in which location decisions were made in the first and second stages, and the corresponding flow decisions were reserved for the third stage. The methodology was applied to a case study concerning the recycling of sand from demolition waste. They used GAMS as the modelling language and CPLEX 6.5 to solve the mixed integer models.

2.4. Comparison of relevant reverse logistics network models

This section provides a comparison study of existing reverse logistics network models, especially the network models for discrete product recovery. This is carried out based on the following characteristics, namely, the type of the supply chain (whether it modelled only the reverse chain or forward chain, or both), assumptions of the nature of product, the modelling approach, the type of optimisation, whether uncertainties is taken into account, the planning horizon, and lastly whether they considered implementing the recovery activities at existing forward chain (facility sharing). This comparison study is summarised in Table 2.1.

Three models concentrated on the reverse part of the supply chain (the flow of products that going back to the manufacturer) and the other four models considered the entire closed loop supply chain (forward and reverse chain) or modelled the closed and open loop supply chain. In addition to modelling the closed loop supply chain; they also modelled the flow of used products from consumers to other independent recovery company such as recycler.

Most of the models assumed the single level BOM and single product type or single level BOM and multi-product type. Only Krikke et al. (2001a) and Shih (2001) considered multi-level BOM and multi-product type. Some models were developed as a generic model without specifying the product type, and some models were developed specifically for certain products like refrigerators, electrical appliances, computers, cell phones, and photocopiers.

Table 2.1: Comparison study of relevant reverse logistics network models

No.	Models	Type of supply chain	Nature of product	Modelling approach	Type of optimisation	Uncertainty considered?	Planning horizon	Facility sharing at forward chain
1.	Jayaraman et al. (1999)	Reverse chain	Cell phones, single product, single level BOM	MIP	Minimising costs	No	Single period	No
2.	Krikke et al. (1999)	Reverse chain	Photocopiers, single product, single level BOM	MIP	Minimising costs	No	Single period	No
3.	Fleischman et al. (2001)	Forward and reverse chains	Single product, single level BOM	MIP	Minimising costs	No	Single period	Yes
4.	Krikke et al. (2001a)	Forward and reverse chains	Refrigerator, multi product, multi level BOM	Goal programming	Minimising costs, energy consumption	No	Single period	No
5.	Shih (2001)	Reverse chain	Electrical appliances and computers, Multi-product, multi level BOM	MIP	Minimising costs	No	Single period	No
6.	Beamon and Fernandes (2004)	Forward and reverse chains	Single product, single level BOM	MIP	Minimising costs	No	Multi period	Yes
7.	Sim et al. (2004)	Forward and reverse chains	Multi product, single level BOM	MIP	Minimising costs	No	Multi period	Yes

Most of the models used MIP as modelling approach, and usually the models were extension of the classical model of warehouse location problem (WLP), with minimising cost as the objective function and balance of flow between facilities and capacity limit as constraints. Only the model by Krikke et al. (2001a) considered the minimisation of energy consumption as another objective function besides

minimising cost.

All of the models assumed deterministic supply of used product and most models consider the single period planning horizon, except for Sim et al. (2004) and Beamon and Fernandes (2004) that considered multi-period planning horizon.

Most of the models did not consider the possibility of implementing product recovery activities at existing forward chain facilities, except for Fleischmann et al. (2001), Sim et al. (2004), and Beamon and Fernandes (2004) which considered the possibility of using the existing manufacturing plant for reprocessing, and the distribution warehouse for storing the new products and the reprocessed products.

As discussed earlier, the emphasis of this research is to develop an approach to the design of reverse logistics networks for discrete product recovery that can be used in the real systems. Therefore, several characteristics that exist in the real system, such as multi-product configuration and modular design, and uncertainty must also be considered in the approach. Furthermore, as network planning is a strategic decision, the design should be valid for several periods of time. Moreover, as environmental concern is one of the main drivers of product recovery, it must also be considered in the network design. Lastly, as cost is the main factor that makes OEMs reluctant in implementing product recovery activities, economical ways of implementing it, such as the possibility of having facility sharing in the forward chain facilities, the possibilities of performing several recovery activities in one facility, and the flexibility of opening new recovery facilities as needed at any period of the planning horizon must also be considered in the network design.

Concerning the existing reverse logistics network models for discrete product recovery, it can be inferred that, even though several models have considered some

of the preferred characteristics, none of them considered all the characteristics altogether. The closest applicable model is model by Sim et al. (2004). They extended previous model by Fleischmann et al. (2001) to relax the single product and single period assumptions. However, their objective function was still based solely on minimising costs and they did not consider uncertainty. Therefore, a decision approach that takes into account those characteristics is still needed.

2.5. Summary

Literature related to product recovery and reverse logistics models have been discussed in this chapter. Concerning the existing reverse logistics network models for discrete product recovery, it can be concluded that none of them considered all the characteristics that usually exist in a real system. Therefore, a decision approach that takes into account those characteristics is still needed.

CHAPTER 3

MODELLING REVERSE LOGISTICS NETWORK FOR PRODUCT RECOVERY

The objective of the research is to develop a decision approach for designing reverse logistics networks for discrete product refurbishment and/or module (part) recovery that considers all the following characteristics:

1. Multi-product configuration with modular product design,
2. Facility sharing between forward and reverse flows,
3. Possibility of having several recovery activities carried out in one facility (integrated recovery facility),
4. Multi-period planning horizon,
5. Flexibility to build new recovery facilities as needed, and
6. Multiple objectives: maximising net revenue and minimising environmental impacts,
7. Uncertainty.

The research is conducted as follows: A mathematical model that considers the first six characteristics is developed in two stages. In the first stage, a model that considers the first five characteristics is developed, and in the second stage, the model is modified to take into account the multiple objectives. Thereafter, a methodology to solve the multi-objective optimisation problem and to determine the solution that has the best performance under uncertainty is developed.

The remainder of this chapter explains the modelling process which includes the mathematical model development and a numerical example, whereas the solution

methodology and a numerical example are presented in Chapter 4.

3.1. Model development

In this research, MIP is chosen as the approach for modelling reverse logistics networks because of its effectiveness in modelling the location problem as indicated in Shapiro (2001). A classic application of MIP is the facility location-allocation problem. The simplest application of the facility location-allocation problem is to determine the allocation of goods and to choose the location of facilities from among several options when other parameters are fixed. Salkin et al. (1989) presented the MIP model of the aforementioned problem.

3.1.1. Single-objective, multi-period model

The proposed mathematical model is developed based on the facility location-allocation model. The objective is to determine locations of recovery facilities and the allocation of goods between facilities that maximise the net revenue.

3.1.1.1. Model assumptions

The model is developed based on the following assumptions:

1. Product recovery activities considered in the research comprise collection, modular disassembly, testing, grading, minor repairs of used modules, and reassembly. These activities are assumed to be performed at three separate facilities, namely, collection centres to collect used product from consumers in the primary market, disassembly centres where used products are subject to modular disassembly and used modules undergo testing, grading, and minor repairs, and recovery centre where reassembly is carried out.
2. The reverse process (Figure 3.1) is assumed to be as follows: Customers from the primary markets dispose of their used products at the nearby collection centres

(1). The accumulated used products are sent to disassembly centres (2). Once there is a demand from recovery centres, used products are disassembled. Used modules are then tested, graded, and if necessary, undergo minor repairs. Recovered used modules are sent to the recovery centre as required (3). The surplus is sent to the spare-part stock centre within the company (4), if required, or sold to third parties (5). Used modules that are unrecoverable but recyclable (6) are sold to recycling companies for material recovery, while the remaining products that cannot be recovered or recycled are sent to disposal sites (7). Once the recoverable used modules arrive at the recovery centres, they are immediately processed (re-assembled) and newly manufactured modules may be required in the recovery process (8). The output of this process is refurbished products (9).

Figure 3.1: The reverse process

3. Product considered in the model is assumed to have multi-product configurations and modular product design.
4. Revenues considered in the model are from selling refurbished products, surplus of good-condition used modules (as spare-parts), recyclable used modules, and

savings from using used modules (instead of newly manufactured modules) as spare-parts within the company.

5. Costs considered in the model are transportation costs between recovery facilities, taxes, inventory holding costs at collection and disassembly centres, processing costs at disassembly and recovery centres, purchasing costs of newly manufactured modules to be used in the recovery process, disposal cost, and fixed (overhead) costs at all facilities. Testing, grading, and minor repair costs of recoverable used modules are included in the processing cost at disassembly centre.
6. The movements of goods among facilities are assumed to be carried on by using different modes of transportation.
7. Supply of used products and demand for refurbished products are independent of each other.
8. Recovery activities can be done in the forward chain facilities (retailers, distribution centres, and assembly plants), new recovery facilities or third party facilities. Collection, disassembly, testing and grading, and minor repairs can be done in one location (integrated collection and disassembly centre), and disassembly, testing and grading, minor repairs and reassembly processes can also be done in one location (integrated disassembly and recovery centre). However, it is assumed that only the existing or new facilities in the next stage are the feasible locations for integrated facilities as third party facilities may not be equipped to handle more than one function. The potential network is depicted in Figure 3.2.
9. Used products are assumed to be held for some period of time in the collection

centres (before being sent to the disassembly centres) and in the disassembly centres (before being disassembled into used modules). If those facilities are assigned to the same location (integrated facilities) then it is assumed that used products are stored until disassembly time.

Figure 3.2: The potential network

10. Used products are collected and disassembled into modules. Only certain types of used modules are required in the refurbishment process. For these types of module, the recovered modules along with newly manufactured modules are used to build new types of product configuration (refurbished products). It is assumed that newly manufactured modules are used to replace outdated modules, and this assumption is justified since upgrading is a usual practice in product recovery. It is also assumed that the volume of recovered used modules is always equal to or larger than, the number required to satisfy the demand and this assumption is justified as usually there will be more supplies than demands in case of product

recovery. The surplus (if any) can be used as spare-parts (if there is a demand within the company) or can be sold as good parts to third parties. For modules that are not required in the refurbishment process, they are sold to recycler companies if recyclable, otherwise they are considered as waste and disposed of at disposal sites.

11. Recyclability (α) and recoverability (β) fractions may be different for each module. It is assumed that for each type of used module, recoverability is a subset of recyclability. It means that modules that can be recovered can also be recycled, however modules that can be recycled, can or can not be recovered.
12. Multi-period planning horizon is incorporated in the model. Supply, demand, prices, costs and capacities of the forward chain facilities, recyclability and recoverability fractions may vary across periods. Used products collected in each period are assumed to be disassembled during the respective period, and used modules obtained in the disassembly process are assumed to be used in recovery process, used as spare-parts within the company, sold as good parts to third party, sold to recycler companies, or disposed of during the period itself and not carried over to the next period. This assumption is justified because it is more profitable to recycle rather than hold them until next period's demand is received.
13. Due to its high starting fixed cost, once a new facility is opened, it has to remain open for the subsequent periods.
14. Fixed costs of new facilities comprise the fixed-opening and fixed-operating costs. Fixed-opening cost applies in the opening period only, while the fixed-operating cost applies in the subsequent periods. The two costs are treated separately as fixed costs for opening a facility and operating it in the subsequent

periods may differ significantly.

15. The reverse logistics network may span across countries, however complexities regarding currencies and environmental regulations are not considered in the model.

3.1.1.2. Model notation

Set Indices

$I = \{1, \dots, N_i\}$	Fixed locations of primary market
$J = \{1, \dots, N_j\}$	Feasible locations for collection centre, $J = J_E \cup J_N \cup J_O \cup K_E \cup K_N$
$K = \{1, \dots, N_k\}$	Feasible locations for disassembly centre, $K = K_E \cup K_N \cup K_O \cup L_E \cup L_N$
$L = \{1, \dots, N_l\}$	Feasible locations for recovery centre, $L = L_E \cup L_N \cup L_O$
$M = \{1, \dots, N_m\}$	Fixed locations of secondary market
$N = \{1, \dots, N_n\}$	Fixed location of spare-parts stock centres within the company
$P = \{1, \dots, N_p\}$	Countries
$O = \{1, \dots, N_o\}$	Transportation modes
O_C	Modes of transportation from collection centres to disassembly centres, $O_C \subset O$
O_D	Modes of transportation from disassembly centres to recovery centres, $O_D \subset O$
O_S	Modes of transportation from disassembly centres to spare-part stock centres, $O_S \subset O$
O_R	Modes of transportation from recovery centres to secondary markets, $O_R \subset O$
I_p	Primary market locations in country p , $I_p \subset I$, $p \in P$
J_p	Feasible locations of collection centres in country p , $J_p \subset J$, $p \in P$
J_E	Existing retailers
K_E	Existing distribution centres
L_E	Existing assembly plants
J_N	Potential locations for new collection centres
K_N	Potential locations for new disassembly centres
L_N	Potential locations for new recovery centres
J_O	Third party collection centres
K_O	Third party disassembly centres
L_O	Third party recovery centres
$Z = \{1, \dots, N_z\}$	Used and newly manufactured modules, $Z = Z_U \cup Z_N$
Z_U	Used modules
Z_R	Used modules that will be used in producing refurbished product, $Z_R \subset Z_U$
Z_N	Newly manufactured modules
$R = \{1, \dots, N_r\}$	Used and refurbished products, $R = R_U \cup R_R$
R_U	Used products
R_R	Refurbished products
$T = \{1, \dots, N_t\}$	Time periods
$T_O = \{2, \dots, N_t\}$	Time periods for operating new facilities after their opening, $T_O \subset T$

Decision Variables

Q_{rijt}^A	Flow of used product r from primary market i to collection centre j in time period t
Q_{rjko}^B	Flow of used product r from collection centre j to disassembly centre k using transportation mode o in time period t
Q_{zkt}^M	Total number of used module z obtained at disassembly centre k in time period t
Q_{zklot}^C	Flow of recovered used module z from disassembly centre k to recovery centre l using transportation mode o in time period t
Q_{zknott}^G	Flow of recovered used module z from disassembly centre k to spare-part stock centre n using transportation mode o in time period t
Q_{zkt}^H	Surplus of recovered used module z at disassembly centre k in period t
Q_{zlt}^R	Total number of newly manufactured module z required to build refurbished products at recovery

Chapter 3: Modelling reverse logistics network for product recovery

Q_{rlmot}^E	centre l in time period t
	Flow of refurbished product q from recovery centre l to secondary market m using transportation mode o in time period t
YC_{jt}	Indicator of assigning a collection centre to a feasible location j in time period t
YD_{kt}	Indicator of assigning a disassembly centre to a feasible location k in time period t
YR_{lt}	Indicator of assigning a recovery centre to a feasible location l in time period t
YOC_{jt}	Indicator of operating a new collection centre j in time period t
YOD_{kt}	Indicator of operating a new disassembly centre k in time period t
YOR_{lt}	Indicator of operating a new recovery centre l in time period t

Figure 3.3: Quantity decision variables

Parameters

S_{rit}	Supply of used product r from primary market i in time period t
D_{rmt}	Demand for refurbished product r from secondary market m in time period t
DS_{znt}	Demand for spare-part z from spare-part stock centre n in time period t
WP_r	Weight of 1 unit of product r
WM_z	Weight of 1 unit of used module z
XD_{zr}	Number of used module z yielded from 1 unit of used product r
XU_{zr}	Number of used module z required to build 1 unit of refurbished product r
XN_{zr}	Number of new module z required to build 1 unit of refurbished product r
PP_{rt}	Price of 1 unit of refurbished product r in time period t
PM_{zt}	Price of 1 kg of recyclable used module z in time period t
PU_{zt}	Price of 1 unit of recovered used module z in time period t
PN_{zt}	Price of 1 unit of newly manufactured module z in time period t
PD_{rkt}	Processing cost of 1 unit of used product r at disassembly centre k in time period t
PR_{rlt}	Processing cost of 1 unit of refurbished product r at recovery centre l in time period t
HC_{rjt}	Holding cost of 1 unit of used product r for a period time at collection centre j in time period t
HD_{rkt}	Holding cost of 1 unit of used product r for a period of time at disassembly centre k in time period t
TC_{jkot}	Transportation cost of 1 kg of used product from collection centre j to disassembly centre k using transportation mode o in time period t
TD_{klot}	Transportation cost of 1 kg of recovered modules from disassembly centre k to recovery centre l using transportation mode o in time period t
TS_{knot}	Transportation cost of 1 kg of recovered modules from disassembly centre k to spare-part stock centre n using transportation mode o in time period t
TR_{lmot}	Transportation cost of 1 kg of refurbished product from recovery centre l to secondary market m , using transportation mode o in time period t
UC_{rjkt}	Tax on 1 unit of used product r from collection centre j to disassembly centre k in time period t
UD_{zklt}	Tax on 1 unit of recovered modules z from disassembly centre k to recovery centre l in time period t
US_{zknt}	Tax on 1 unit of recovered modules z from disassembly centre k to spare-part demand site n in time period t
UR_{rlmt}	Tax on 1 unit of refurbished product r from recovery centre l to secondary market m , in time period t
DC_{zt}	Disposal cost of 1 unit of unrecoverable and unrecyclable used module z in time period t
FC_{jt}	Fixed cost at collection centre j in time period t

Chapter 3: Modelling reverse logistics network for product recovery

FD_{kt}	Fixed cost at disassembly centre k in time period t
FR_{lt}	Fixed cost at recovery centre l in time period t
FOC_{jt}	Fixed cost to operate a new collection centre j after its opening in time period t
FOD_{kt}	Fixed cost to operate a new disassembly centre k after its opening in time period t
FOR_{lt}	Fixed cost to operate a new recovery centre l after its opening in time period t
VC_{rjt}	Capacity to store used product r at collection centre j in time period t
VD_{rkt}	Capacity to store/process used product r at disassembly centre k in time period t
VR_{rlt}	Capacity to process refurbished product r at recovery centre l in time period t
α_{zt}	Fraction of recyclable used modules z in period t , $0 \leq \alpha_{zt} \leq 1$
β_{zt}	Fraction of recoverable used modules z in period t , $\beta_{zt} \leq \alpha_{zt}$

3.1.1.3. Model formulation**Objective function: Maximise net revenue**

Net revenue of all period = total revenue of all period – total costs of all period.

Total revenue of all period:

Revenue from selling refurbished products:

$$\sum_{r \in R_R} \sum_{l \in L} \sum_{m \in M} \sum_{o \in O_R} \sum_{t \in T} PP_{rt} Q_{rlmot}^E$$

Revenue from selling recovered modules as good parts:

$$+ \sum_{z \in Z_R} \sum_{k \in K} \sum_{t \in T} PU_{zt} Q_{zkt}^H$$

Revenue from selling recyclable modules:

$$+ \sum_{k \in K} \sum_{t \in T} \left(\sum_{z \in Z_R} (\alpha_{zt} - \beta_{zt}) PM_{zt} WM_z Q_{zkt}^M + \sum_{z \in Z_U, z \notin Z_R} \alpha_{zt} PM_{zt} WM_z Q_{zkt}^M \right)$$

Savings from using recovered modules (instead of new ones) as spare-parts within the company:

$$+ \sum_{z \in Z_R} \sum_{k \in K} \sum_{n \in N} \sum_{o \in O_S} \sum_{t \in T} PN_{zt} Q_{zknott}^G$$

Total costs of all period:

Transportation costs and taxes:

$$\begin{aligned} & - \sum_{r \in R_U} \sum_{j \in J} \sum_{k \in K} \sum_{o \in O_C} \sum_{t \in T} Q_{rjkot}^B (WP_r TC_{jkot} + UC_{rjkt}) - \sum_{z \in Z_R} \sum_{k \in K} \sum_{l \in L} \sum_{o \in O_D} \sum_{t \in T} Q_{zklot}^C (WM_z TD_{klot} + UD_{zklot}) \\ & - \sum_{z \in Z_R} \sum_{k \in K} \sum_{n \in N} \sum_{o \in O_S} \sum_{t \in T} Q_{zknott}^G (WM_z TS_{knott} + US_{zknott}) - \sum_{r \in R_R} \sum_{l \in L} \sum_{m \in M} \sum_{o \in O_R} \sum_{t \in T} Q_{rlmot}^E (WP_r TR_{lmot} + UR_{rlmot}) \end{aligned}$$

Chapter 3: Modelling reverse logistics network for product recovery

Holding costs at collection and distribution centres:

$$- \sum_{r \in R_U} \sum_{j \in J} \sum_{i \in I} \sum_{t \in T} HC_{rjt} Q_{rijt}^A - \sum_{r \in R_U} \sum_{k \in K} \sum_{j \in J} \sum_{o \in O_C} \sum_{t \in T} HD_{rkt} Q_{rjkot}^B$$

Processing costs at distribution and recovery centres:

$$- \sum_{r \in R_U} \sum_{k \in K} \sum_{j \in J} \sum_{o \in O_C} \sum_{t \in T} PD_{rkt} Q_{rjkot}^B - \sum_{r \in R_R} \sum_{l \in L} \sum_{m \in M} \sum_{o \in O_R} \sum_{t \in T} PR_{rlt} Q_{rlmot}^E$$

Purchasing costs of newly manufactured modules used in reassembly of refurbished products:

$$- \sum_{z \in Z_N} \sum_{l \in L} \sum_{t \in T} PN_{zlt} Q_{zlt}^R$$

Disposal costs of unrecoverable and unrecyclable modules:

$$- \sum_{z \in Z_U} \sum_{k \in K} \sum_{t \in T} (1 - \alpha_{zt}) DC_{zkt} Q_{zkt}^M$$

Fixed costs at all facilities:

$$\begin{aligned} & - \left(\sum_{j \in J} \sum_{i \in I} FC_{ji} YC_{ji} + \sum_{k \in K} \sum_{i \in I} FD_{ki} YD_{ki} + \sum_{l \in L} \sum_{i \in I} FR_{li} YR_{li} - \sum_{j \in J_N} \sum_{i \in I} FC_{ji} YOC_{ji} - \sum_{k \in K_N} \sum_{i \in I} FD_{ki} YOD_{ki} - \sum_{l \in L_N} \sum_{i \in I} FR_{li} YOR_{li} \right) \\ & - \left(\sum_{j \in J_N} \sum_{t \in T_O} FOC_{jt} YOC_{jt} + \sum_{k \in K_N} \sum_{t \in T_O} FOD_{kt} YOD_{kt} + \sum_{l \in L_N} \sum_{t \in T_O} FOR_{lt} YOR_{lt} \right) \end{aligned} \quad (3.1)$$

Constraints:

Flow balances between locations:

$$\sum_{j \in J_p} Q_{rijt}^A = \begin{cases} S_{rit} & (\forall r \in R_U)(\forall i \in I_p)(\forall p \in P)(\forall t \in T) \\ 0 & (\forall r \in R_U)(\forall i \notin I_p)(\forall p \in P)(\forall t \in T) \end{cases} \quad (3.2)$$

$$\sum_{k \in K} \sum_{o \in O_C} Q_{rjkot}^B = \sum_{i \in I} Q_{rijt}^A \quad (\forall r \in R_U)(\forall j \notin (K_E \cup K_N))(\forall t \in T) \quad (3.3)$$

$$Q_{zkt}^M = \sum_{r \in R_U} \sum_{j \in J} \sum_{o \in O_C} XD_{zr} Q_{rjkot}^B \quad (\forall z \in Z_U)(\forall k \in K)(\forall t \in T) \quad (3.4)$$

$$\sum_{k \in K} \sum_{o \in O_D} Q_{zklot}^C = \sum_{r \in R_R} \sum_{m \in M} \sum_{o \in O_R} XU_{zr} Q_{rlmot}^E \quad (\forall z \in Z_R)(\forall l \in L)(\forall t \in T) \quad (3.5)$$

$$\sum_{k \in K} \sum_{o \in O_S} Q_{zknott}^G = DS_{znt} \quad (\forall z \in Z_R)(\forall n \in N)(\forall t \in T) \quad (3.6)$$

$$\sum_{l \in L_O} \sum_{o \in O_D} Q_{zklot}^C = \beta_{zt} Q_{zkt}^M - \sum_{n \in N} \sum_{o \in O_S} Q_{zknott}^G - Q_{zkt}^H \quad (\forall z \in Z_R)(\forall k \notin (L_E \cup L_N))(\forall t \in T) \quad (3.7)$$

$$Q_{zlt}^R = \sum_{r \in R_R} \sum_{m \in M} \sum_{o \in O_R} XN_{zr} Q_{rlmot}^E \quad (\forall z \in Z_N)(\forall l \in L)(\forall t \in T) \quad (3.8)$$

$$\sum_{l \in L} \sum_{o \in O_R} Q_{rlmot}^E = D_{rmt} \quad (\forall r \in R_R)(\forall m \in M)(\forall t \in T) \quad (3.9)$$

Chapter 3: Modelling reverse logistics network for product recovery

Integrated facility constraints:

$$\sum_{o \in O_C} Q_{rjkot}^B = \begin{cases} \sum_{i \in I} Q_{rijt}^A & (\forall r \in R_U)(\forall j, \forall k \in (K_E \cup K_N))(\forall t \in T) \\ 0 & (\forall r \in R_U)(\forall j \in (K_E \cup K_N))(\forall k \notin (K_E \cup K_N))(\forall t \in T) \end{cases} \quad (3.10)$$

$$\sum_{o \in O_D} Q_{zklot}^C = \begin{cases} \beta_{zt} Q_{zkt}^M - \sum_{n \in N} \sum_{o \in O_S} Q_{zknott}^G - Q_{zkt}^H & (\forall z \in Z_R)(\forall k, \forall l \in (L_E \cup L_N))(\forall t \in T) \\ 0 & (\forall z \in Z_R)(\forall k \in (L_E \cup L_N))(\forall l \notin (L_E \cup L_N))(\forall t \in T) \end{cases} \quad (3.11)$$

Capacity limitation at each recovery facility:

$$\sum_{i \in I} Q_{rijt}^A \leq VC_{rjt} YC_{jt} \quad (\forall r \in R_U)(\forall j \in J)(\forall t \in T) \quad (3.12)$$

$$\sum_{j \in J} \sum_{o \in O_C} Q_{rjkot}^B \leq VD_{rkt} YD_{kt} \quad (\forall r \in R_U)(\forall k \in K)(\forall t \in T) \quad (3.13)$$

$$\sum_{m \in M} \sum_{o \in O_R} Q_{rlmot}^E \leq VR_{rlt} YR_{lt} \quad (\forall r \in R_R)(\forall l \in L)(\forall t \in T) \quad (3.14)$$

New facilities constraints:

$$YC_{jt+1} \geq YC_{jt} \quad (\forall j \in J_N)(\forall t \in T, t \neq N_t) \quad (3.15)$$

$$YD_{kt+1} \geq YD_{kt} \quad (\forall k \in K_N)(\forall t \in T, t \neq N_t) \quad (3.16)$$

$$YR_{lt+1} \geq YR_{lt} \quad (\forall l \in L_N)(\forall t \in T, t \neq N_t) \quad (3.17)$$

$$YOC_{jt} = YC_{jt-1} \quad (\forall j \in J_N)(\forall t \in T_O) \quad (3.18)$$

$$YOD_{kt} = YD_{kt-1} \quad (\forall k \in K_N)(\forall t \in T_O) \quad (3.19)$$

$$YOR_{lt} = YR_{lt-1} \quad (\forall l \in L_N)(\forall t \in T_O) \quad (3.20)$$

Binary variables:

$$YC_{jt}, YD_{kt}, YR_{lt}, YOC_{jt}, YOD_{kt}, YOR_{lt} \in \{0, 1\} \quad (3.21)$$

Nonnegative variables:

$$Q_{rijt}^A, Q_{rjkot}^B, Q_{zkt}^M, Q_{zklot}^C, Q_{zknott}^G, Q_{zkt}^H, Q_{zlt}^R, Q_{rlmot}^E \geq 0 \quad (3.22)$$

Formulation (3.1) represents the objective function, which is maximisation of net revenue. Maximising net revenue is chosen instead of minimising total costs as it reflects the profitability of product recovery, as for decision makers the profitability of the activities is more important than just minimising the costs. The model assumes

revenue from selling refurbished product, recovered modules as good parts, recyclable modules and lastly savings from using recovered modules (instead of new ones) as spare-parts within the company.

Regarding revenue from selling recyclable modules, used modules obtained from disassembly process can be classified into two classes, which are used modules that are required and not required in the refurbishment process. For the first class, since it is assumed that recoverability fraction is a subset of recyclability fraction, number of used modules that are actually sold for recycling is $\sum_{k \in K} \sum_{t \in T} \left(\sum_{z \in Z_R} (\alpha_{zt} - \beta_{zt}) Q_{zkt}^M \right)$. For the second class, on the other hand, since the modules are not required in refurbishment process, the number of used modules that are sold for recycling is $\sum_{k \in K} \sum_{t \in T} \left(\sum_{z \in Z_U, z \notin Z_R} \alpha_{zt} Q_{zkt}^M \right)$.

As earlier mentioned in the assumption, if there is demand for recovered modules to be used as spare-parts within the company itself, recovered modules are sent to spare-part stock centres. Therefore, savings from using recovered parts as spare-parts instead of purchasing new modules is included as the last part of the revenue.

Regarding costs, the model assumes that costs consist of transportation and taxes between facilities, holding costs at collection and disassembly centres, processing costs at disassembly and recovery centres, purchasing costs of newly manufactured modules required in refurbishment process, disposal costs of unrecoverable and unrecyclable modules and lastly fixed costs at all facilities.

Constraints (3.2) to (3.9) represent flow balances between facilities. Constraint (3.2) in particular corresponds to the assumption that all used products

from customers in a primary market located in a certain country must be returned at collection centres in the same country. This assumption is justified since it is less likely for customers to trade-in their used products outside their countries.

Constraint (3.3) represents balance of flow balances at collection centres. As there is no testing facility in the collection centres, used products accumulated at collection centres are sent entirely to disassembly centres. Constraint (3.4), on the other hand, represents modular disassembly processes that take place at disassembly centres. Q_{zkt}^M represents the quantity of module z obtained from disassembly process of all used product configurations at disassembly centre k at time period t .

Constraint (3.5) ensures that demand for recovered used modules for building refurbished products is satisfied. Constraint (3.6), on the other hand, ensures that the demand for recovered modules from spare-parts stock centres within the company is satisfied. Constraint (3.7) represents balances of flows regarding the recovered modules. Surplus in each disassembly centre will exist if the quantity of recovered modules is larger than the sum of demands from recovery centre and spare-parts stock centres.

Constraint (3.8) denotes the quantity of newly manufactured modules that are required in the product refurbishment process, and Constraint (3.9) ensures that the demand for refurbished products is satisfied.

Constraint (3.10) represents the concept of integrated recovery facility: collection and disassembly centres can be assigned to one location (existing distribution centre or new disassembly centre), and disassembly and recovery centres can also be assigned to one location (existing assembly plant or new recovery centre). Therefore, if goods from primary markets are collected at distribution/new

disassembly centres, the disassembly process must also be performed at these locations. The same condition applies to the integration of disassembly and recovery centres as represented by Constraint (3.11).

Constraints (3.12) to (3.14) represent capacity limitations of the facilities, ensuring that goods are only flowing to the chosen facilities and that total inflows to those facilities are less than their maximum capacity.

Constraints (3.15) to (3.17), on the other hand, ensure that once a new facility is opened, it will remain open in the subsequent periods due to its high fixed-opening cost. Constraints (3.18) to (3.20) indicate that for new facilities, the fixed-operating costs apply in the subsequent periods. Lastly, constraints (3.21) and (3.22) denote variable restrictions.

The above model can be used to model a product recovery network, especially product refurbishment and/or module (part) recovery (to be used as spare-parts or to be sold to other parties). As mentioned earlier, a company may be interested only in product refurbishment, or only in part recovery, or in both. The model can be used to design the network for all these purposes. For instance, if the company is only interested in recovering parts for spare parts for internal use, demand for refurbished (D_{qmt}) is set to zero, and demand for the recovered parts from the company's spare-part stock centre is represented in DS_{znt} . If the company is only interested in dismantling used product and selling the recovered parts to other companies, D_{qmt} and DS_{znt} are set to zero, and the demand for used parts is represented in Q^H_{zkt} .

Multi-product configurations and modular product design are also modelled, so that different parameters associated with different products and modules can be

considered resulting in a more accurate representation. The model also represents existing forward chain, new and outsourced facilities as different set of locations as each set of locations may be associated with different parameters (costs and capacity).

For economical reason, facility sharing between forward and reverse flows is also taken into account by the inclusion of forward chain facilities as potential locations for recovery activities. It also models the possibility of having integrated facilities so that the recovery activities can be implemented in fewer facilities

Multi-period planning horizon is also considered, therefore changes in parameters can be represented resulting in a network that is valid for several periods of time. It also models the flexibility to open new facilities in any period in the multi-period planning horizon, so that the recovery activities can be implemented gradually and in the most economical way.

The model also considers the possibility of having the network span across countries, which is possible in certain regions, such as in Asia-Pacific. Taxes for transporting goods between countries are included in the objective function. However, currency and certain environmental policy, such as the prohibition on import of used goods, are not considered in the model. To consider complexity regarding currency, the model can be extended by introducing currency exchange rate in the objective function, whereas environmental policy can be modelled by introducing high penalty in the objective function or by adding logical constraints.

3.1.1.4. A numerical example

To verify the model, it is applied to a case of designing reverse logistics network for computer refurbishment with no demand for recovered modules within the company.

The data for the network considered in the application are taken and modified from various sources.

The potential network consists of 1 primary market location, 5 feasible locations of collection centres, 3 feasible locations of disassembly centres, 3 feasible locations of recovery centres, and 3 secondary market locations. As mentioned earlier, integrated recovery facilities can only be assigned to forward facilities (existing) or new locations. Therefore, there will be 7 feasible locations for collection centres and 5 feasible locations for disassembly centres. The potential network is presented in Figure 3.4, while other data is presented in Appendix A.

Figure 3.4: Potential network for the numerical example

Four types of used PCs (with 17 types of modules) are refurbished into two types of refurbished PCs (5 newly manufactured modules are required in the

refurbishment process). There is no demand for spare-parts within the company, therefore the surplus of recovered modules (if any) are sold to a third party. Furthermore, three modes of transportation are considered (by car, truck or airplane). The problem is solved for single period (one year) and multi-period (five years) using Lingo 9.0 (Lindo Systems, 2004).

Result of the single period indicates that a new collection centre is assigned for used computers collection, whereas the assembly plant is assigned as an integrated disassembly and recovery centre. Transportation from collection to disassembly centres, from disassembly to recovery centres and from recovery centres to secondary markets are carried out by truck. The total net revenue is around \$2.06 million and the optimal network is depicted in Figure 3.5 (the complete result is shown in Appendix B).

Figure 3.5: The optimal network for single period problem

The larger capacity and lower holding cost at the new collection centre makes the cost of assigning used computers collection to the respective facility is lower than of assigning it to other feasible locations. On the other hand, a lower fixed cost and sufficient capacity at the assembly plant are the reasons for assigning the integrated disassembly and recovery to the respective facility resulting in lower costs. Hence, in this case, variable costs play an important role in determining the optimal location of

the collection centre, while fixed costs have a more important role in determining the optimal location of the disassembly and recovery centres.

Pertaining to the transportation mode, since all the locations reside in the same country and the distance from collection to disassembly centres are between eighty to one hundred fifty miles, the only available transportation mode is by truck. For the subsequent stages however, the available transportation modes are by truck or airplane as the distance from disassembly to recovery centres and from recovery centres to secondary markets are more than or equal to five hundred miles. Truck is chosen as the transportation mode as it leads to a lower transportation cost.

The optimal network for the multi-period problem is depicted in Figure 3.6. For all periods, a new collection centre is assigned for used computers collection and a new recovery centre is assigned as an integrated disassembly and recovery centre. In periods two and four, however, an outsourced collection centre is also assigned for used computer collection, whereas Retailer 1 is also assigned as a collection centre in period three. Transportations between facilities are also carried out by truck. The total net revenue is around \$4.6 million (the complete result is presented in Appendix C).

Figure 3.6: The optimal network for multi-period problem

When the supply is increased in the subsequent period, the capacity of the new collection centre is not sufficient to accommodate used products from primary market. Therefore, additional collection centres with low fixed costs are assigned in those time periods to handle the excess supply.

The new recovery centre, on the other hand, is assigned as integrated disassembly and recovery centre. Its larger capacity and lower holding and processing costs compensate the expensive fixed-opening cost.

Concerning the transportation mode, truck is chosen as the transportation mode to move goods in all stages and all periods for the same reason as in the case of single period.

3.1.2. Multi-objective, multi-period model

In the multi-objective model, besides maximising net revenue, minimising environmental impacts in the form of minimising energy consumption and minimising CO₂ emission are added as the objective functions. Energy consumption is also considered as environmental impact as the process of energy creation affects the environment significantly. Hence, reducing the energy consumption will lead to a better environment.

3.1.2.1. Additional assumptions

New assumptions that are related to new objective functions are added as follows:

1. Energy consumption considered in the model is from processes in disassembly and recovery centres and transportation between facilities.
2. CO₂ emission considered in the model is emission during transportation between facilities.

3. The rate of energy consumption and CO₂ emission are assumed to be constant across periods.

3.1.2.2. Additional model notation

The addition of new objective functions results in additional parameters and they are presented as follows.

Additional parameters

EC_{jko}	Energy consumption during transporting 1 kg of used product from collection centre j to disassembly centre k using transportation mode o
ED_{klo}	Energy consumption during transporting 1 kg of recovered modules from disassembly centre k to recovery centre l using transportation mode o
ES_{kno}	Energy consumption during transporting 1 kg of recovered modules from disassembly centre k to spare-part stock centre n using transportation mode o
ER_{lmo}	Energy consumption during transporting 1 kg of refurbished product from recovery centre l to secondary market m using transportation mode o
EUD_{qk}	Energy consumption during processing 1 unit of used product q at disassembly centre k
EUR_{ql}	Energy consumption during processing 1 unit of refurbished product q at recovery centre l
CC_{jko}	CO ₂ emission when transporting 1 kg of used product from collection centre j to disassembly centre k using transportation mode o
CD_{klo}	CO ₂ emission when transporting 1 kg of recovered modules from disassembly centre k to recovery centre l using transportation mode o
CS_{kno}	CO ₂ emission when transporting 1 kg of recovered modules from disassembly centre k to spare-part stock centre n using transportation mode o
CR_{lmo}	CO ₂ emission when transporting 1 kg of refurbished product from recovery centre l to secondary market m using transportation mode o

3.1.2.3. Model formulation

The additional objective functions are presented as follows.

Minimise energy consumption during refurbishment and transportation process

Energy consumption during transportation between locations:

$$\begin{aligned} & \sum_{r \in R_U} \sum_{j \in J} \sum_{k \in K} \sum_{o \in O_C} \sum_{t \in T} WP_r Q_{rjkt}^B EC_{jko} + \sum_{z \in Z_R} \sum_{k \in K} \sum_{l \in L} \sum_{o \in O_D} \sum_{t \in T} WM_z Q_{zklot}^C ED_{klo} \\ & + \sum_{z \in Z_R} \sum_{k \in K} \sum_{n \in N} \sum_{o \in O_S} \sum_{t \in T} WM_z Q_{zknott}^G ES_{kno} + \sum_{r \in R_R} \sum_{l \in L} \sum_{m \in M} \sum_{o \in O_R} \sum_{t \in T} WP_r Q_{rlmot}^E ER_{lmo} \end{aligned}$$

Energy consumption during disassembly and reassembly process:

$$+ \sum_{r \in R_U} \sum_{k \in K} \sum_{j \in J} \sum_{o \in O_C} \sum_{t \in T} Q_{rjkt}^B EUD_{qk} + \sum_{r \in R_R} \sum_{l \in L} \sum_{m \in M} \sum_{o \in O_R} \sum_{t \in T} Q_{rlmot}^E EUR_{ql} \quad (3.23)$$

Minimise CO₂ emission during transportation between locations

$$\begin{aligned} & \sum_{r \in R_U} \sum_{j \in J} \sum_{k \in K} \sum_{o \in O_C} \sum_{t \in T} WP_r Q_{rjkt}^B CC_{jko} + \sum_{z \in Z_R} \sum_{k \in K} \sum_{l \in L} \sum_{o \in O_D} \sum_{t \in T} WM_z Q_{zklot}^C CD_{klo} \\ & + \sum_{z \in Z_R} \sum_{k \in K} \sum_{n \in N} \sum_{o \in O_S} \sum_{t \in T} WM_z Q_{zknott}^G CS_{kno} + \sum_{r \in R_R} \sum_{l \in L} \sum_{m \in M} \sum_{o \in O_R} \sum_{t \in T} WP_r Q_{rlmot}^E CR_{lmo} \end{aligned} \quad (3.24)$$

Formulation (3.23) represents the second objective function, which is minimisation of energy consumption during transport and recovery processes (at disassembly and recovery centres), whereas Formulation (3.24) corresponds to the last objective function, namely, minimising CO₂ emissions during transportation.

Even though the objective functions are somewhat related to one another, optimising one does not necessarily lead to minimisation of the other two as each objective function has a different focus of optimisation. The first objective focuses on determining the network based on the net revenue. Therefore, network design with minimum total cost is chosen. The second objective, on the other hand, determines the network based on energy consumption in disassembly and recovery facilities, and during transportation between facilities (therefore, transportation mode used in each stage also plays an important part in determining the network). As the focus is on energy consumption, fixed and variable costs at facilities are not taken into account. The last objective determines the network solely based on emission during transportation between facilities. Therefore, the last objective solely decides the network based on the distance and the transportation modes used to move goods between locations.

The main driver of product recovery is environmental concern, thus by including environmental impact as business objectives, a green product recovery network is ensured.

3.1.2.4. Revisiting the numerical example

In the previous example, the model is solved for the first objective function (maximising net revenue). The model is now solved for the other two objective functions (additional data is presented in Appendix D). Results for minimising energy

consumption and minimising CO₂ emission are depicted in Figures 3.7 and 3.8 respectively (the complete results are presented in Appendix E).

Figure 3.7: The optimal network that minimises the energy consumption

Figure 3.7 indicates that when minimising energy consumption is considered as the sole objective function, locations with lower energy consumption (in this example, the new facilities) are chosen as recovery facilities. Since minimising costs is not a concern, besides assigning a new recovery centre as an integrated disassembly and recovery centre, a new disassembly centre is also assigned as a location for disassembly centre in all periods. Truck is chosen as the transportation mode to move goods from disassembly to recovery centres and from recovery centres to secondary market in all periods as it leads to lower energy consumption

Similar situation is also happened when the example is solved with minimising CO₂ emission as the sole objective function. Locations with the least total network distance are assigned. Truck is chosen as the transportation mode to move goods from disassembly to recovery centres and from recovery centres to secondary market in all periods as it leads to lower CO₂ emission.

Since minimising costs and minimising energy consumption at all facilities

are not a concern, four locations for disassembly centres and two locations for recovery centres are assigned in the optimal network.

Figure 3.8: Minimising CO₂ emission

The objective values when each objective function is solved individually and their payoff values are presented in Table 3.1. From this example, it can be concluded that all objectives are somewhat conflicting with each other.

Table 3.1: The payoff table

Objective Function	Net revenue (\$)	Energy consumption (GJ)	CO ₂ emission (KgCE)
Maximising net revenue	4,591,169	493,760	304,878
Minimising energy consumption	3,726,468	493,647	299,046
Minimising CO ₂ emission	743,932	538,229	298,046

3.1.2.5. Solving the multi-objective problem

In multi-objective optimisation, vectors of decision variables are regarded as optimal if their components cannot be improved without deterioration to at least one of the other components (Miettinen, 2001). This is usually called as Pareto optimality. There are usually a lot of Pareto optimal solutions (non-dominated/non-inferior/efficient solutions) and they form a set of Pareto optimal solutions or a

Pareto optimal set. Solving multi-objective problem means finding a solution that is Pareto optimal and satisfies the decision maker (DM).

In this research, the proposed approach adopted the “a posteriori” method. In this method, the set of non-dominated solutions are generated and then they are presented to the decision maker, who selects the most preferred solution. Therefore, in the first stage, the developed multi-objective, multi-period model is solved to find the set of non-dominated solutions using the spanning tree based genetic algorithm (st-GA). Thereafter in the second stage, the preferred solutions (selected by the decision maker) are re-evaluated under scenarios of uncertainty using simulation to determine the best preferred solution.

The st-GA is employed to find the set of non-dominated solutions for the developed multi-objective, multi-period model instead of using commercial software, for the following reasons:

1. The dimension of the model are:

a. Number of continue variables:

$$R_U IJT + R_U JK O_C T + Z_U KT + Z_R KL O_D T + Z_R KN O_S T + Z_R KT + Z_N LT + R_R LM O_R T$$

b. Number of binary variables: $JT + KT + LT + J_N T_O + K_N T_O + L_N T_O$

c. Number of constraints:

$$\begin{aligned} & Q_U IT + Q_U (J_E + J_N + J_O)T + Z_U KT + Z_R LT + Z_R NT + Z_R (K_E + K_N + K_O)T + Z_N LT \\ & + Q_R MT + Q_U (K_E + K_N)^2 T + Q_U (K_E + K_N)(K_O + L_N + L_E)T + Z_R (L_E + L_N)^2 T \\ & + Z_R (L_E + L_N) L_O T + Q_U JT + Q_U KT + Q_R LT + 2J_N T_O + 2K_N T_O + 2L_N T_O \end{aligned}$$

Network problem is considered as NP problem therefore increasing the size of problem in polynomial order will lead to an exponential increase of the computation time.

2. To find each non-dominated solutions using commercial software, such as Lingo, the problem must be converted so that there is only one objective function, which is usually carried out by using the weighting approach. Therefore, to find each solution, weights of all objective functions must be supplied every time. Using GA, on the other hand, a set of non-dominated solutions for the multi-objective optimisation problems can be found in a single run.

3.2. Summary

Modelling process of reverse logistics network for product recovery is presented. The model is developed in two stages. In the first stage, the developed basic model considers multi-product configurations with modular design, facility sharing in the forward chain facilities, the possibility of having integrated recovery facilities, multi-period planning horizon and the flexibility to open new facilities as needed at any period in the planning horizon. The model is applied to a case of computer refurbishment.

In the second stage, the model is modified to also consider multi-objective functions by the inclusion of minimising energy consumption and CO₂ emission. A method to find non-dominated solutions for the multi-objective model and determine the best preferred solution that has the best performance under uncertainty is presented in Chapter 4.

CHAPTER 4

THE METHODOLOGY

In this chapter, an approach to designing reverse logistics networks for discrete product recovery is presented and verified using a numerical example.

4.1. The methodology

Several approaches have been developed to solve multi-objective problems with uncertainty. The proposed approach consists of two major models namely MIP and simulation models. The MIP model (discussed in Chapter 3) is used to find non-dominated solutions when the multi-product configurations and modular product design, multi-objective function, and multiple periods are considered. The simulation models are used to re-evaluate the selected solutions (based on the preferences of decision makers) under uncertain conditions. Therefore, the developed approach reduces the complexity by only re-evaluating non-dominated solutions that are preferred by the DM.

The multi-objective MIP model is solved using genetic algorithms, especially spanning tree-based genetic algorithms (Gen & Cheng, 1997, 2000). GA is chosen as it is efficient in finding the set of non-dominated solutions for multi-objective optimisation problems in a single run (Coello Coello et al., 2002). The basic of multi-objective optimisation methods (the conventional and the evolutionary techniques) are explained in Appendix F

The simulation models of selected non-dominated solutions are developed and the models of the preferred solutions incorporate the characteristics that are considered in the MIP model and several scenarios of uncertain conditions are run

for each solution to determine the best-preferred solution. The approach is depicted in Figure 4.1 and explained in the following sections.

Figure 4.1: The Methodology

4.2. Spanning tree-based multi-objective GA for reverse logistics network design

This section explains the procedure to find non-dominated solutions of the multi-objective problem using spanning-tree based genetic algorithm (st-GA).

4.2.1. Genetic representation and chromosome feasibility

One way to represent a network problem is by using tree-based representation. There are three ways of encoding a tree, namely edge encoding, node (vertex) encoding and

edge and node encoding. In this research, a node encoding, using a combination of binary and Prüfer numbers, is employed to represent the network.

Cayley (1889) proved that for a complete graph with p nodes, there are $p^{(p-2)}$ distinct labelled trees and Prüfer (1918) presented the simplest proof of Cayley's formula by establishing a one-to-one correspondence between the set of spanning tree and a set of $p-2$ digits with an integer between 1 and p , inclusive.

The Prüfer number is chosen to represent the network because as Zhou and Gen (1997) observed, it is not only capable of equally and uniquely representing all the possible spanning trees for a graph, it also explicitly contains the information regarding the node's degree. The degree of a node is the number of edges incident to the node. In the Prüfer number, a node with degree d will appear exactly $d-1$ times. Therefore, if any solution violates the degree constraints, it is easy to modify the number of vertices appearing in its corresponding Prüfer number.

The solution in each period is represented by eight substrings. The first to third substrings are binary digits representing opened/closed condition of feasible locations of collection, disassembly and recovery centres, each with length equal to the number of locations in each stage. The fourth to eighth strings are the Prüfer numbers representing the distribution pattern of multi-product flows in each period, between primary market and collection centre, collection and disassembly centres, disassembly and recovery centres, disassembly centre and spare-part stock centres and lastly between recovery centre and secondary market.

An example of the representation scheme for a problem with 3 locations of primary market, 3 potential locations of collection centre, 2 potential locations of disassembly centres, 2 potential locations of recovery centre, 2 spare-part stock

centres and 3 locations of secondary market in a particular period is depicted in Figure 4.2.

Figure 4.2: Representation scheme

Since the binary substrings are generated, it is possible that the substrings may be infeasible. Three checking and repair procedures for binary substrings are developed to accommodate multi-product characteristics and certain constraints in the developed mathematical model. They are described next.

Procedure 1: Capacity feasibility checking and repair procedure

Check whether the total capacity for each product is sufficient to satisfy supply/demand; if it is not sufficient then step by step (product by product), open another facility with the least capacity.

For product number, $q = 1$ to Q :

- If the total capacity of opened-facilities for product q is less than the supply, open another facility with the least capacity.
- If the total capacity of opened-facilities is equal or greater than the supply, stop; otherwise repeat the step with the next product number.

Procedure 2: Integrated-facilities constraint repair procedure

If integrated facility is chosen in stage one (e.g., collection centre), it must also be chosen in the next stage (e.g., disassembly centre).

Procedure 3: New facility constraint repair procedure

Due to its high opening fixed cost, once a new facility is opened in a given period, it must remain open in all subsequent periods.

Infeasible Prüfer substrings may also be created in the generation process. Syarif et al. (2002) developed a procedure to check the feasibility of Prüfer substring to be decoded into a spanning tree and to repair it, which is described next.

Procedure 4: Check and repair for Prüfer number (Syarif et al., 2002)

Let R_i denote the number of appearances of node i in the Prüfer number Pr , and L_i denote the number of connections of the node i for all $i \in O \cup D$, where O is a set of origin nodes and D is a set of destination nodes.

Repeat the flowing steps until $\sum_{i \in O} L_i = \sum_{i \in D} L_i$ (feasibility condition)

Step 1: Determine R_i for $i \in O \cup D$ from $P(T)$

Step 2: $L_i = R_i + 1$

Step 3: If $\sum_{i \in O} L_i > \sum_{i \in D} L_i$, then select one digit in Pr which contains node i ($i \in O$) and replace it with the number j ($j \in D$). Otherwise, select one digit in Pr , which contains node i ($i \in D$) and replace it with the number j ($j \in O$). Then go to Step 1.

The encoding procedure of a network problem into a Prüfer number and the decoding procedure from a Prüfer number into a spanning tree (Gen and Cheng, 2000) are as follows:

Procedure 5: Encoding (Gen and Cheng, 2000)

Step 1: Let node i be the smallest labelled leaf vertex in a labelled tree $O-D$.

Step 2: Let j be the first digit in the encoding as the node j incident to node i is uniquely determined. The encoding is built and read from left to the right.

Chapter 4: The methodology

Step 3: Remove node i and the link from i to j ; thus a tree with $O + D - 1$ nodes is created.

Step 4: Repeat the above steps until only one edge is left, thus a Prüfer number or an encoding with $O + D - 2$ digits between 1 and $O + D$ inclusive is created.

Procedure 6: Decoding (Gen and Cheng, 2000)

Step 1: Let Pr be the original Prüfer number and Pr^- be the set of all nodes not included in Pr , which are designated as eligible nodes for consideration in building a tree.

Step 2: Repeat the process (2.1)-(2.5), until no digits are left in Pr .

(2.1) Let i be the smallest label node Pr^- and j be the leftmost digit of Pr .

(2.2) If i and j are not in the same set O or D , add the edge from i to j into the tree. Otherwise, select the next digit k from Pr that is not included in the same set with i . Exchange j with k and add the edge (i, k) to the tree.

(2.3) Remove j (or k) from Pr and i from Pr^- . If j (or k) does not appear anywhere in the remaining part of Pr , put it into P^- . Designate i as no longer eligible.

(2.4) Assign the available amount of units to $x_{ij} = \min\{a_i, b_j\}$, or $x_{ij} = \min\{a_i, b_k\}$, where a_i is the available supply of node i and b_j is the remaining demand at node j .

(2.5) Update the availability $a_i = a_i - x_{ij}$ and $b_j = b_j - x_{ij}$.

Step 3: If no digits remain in Pr , there are exactly two nodes, r and s , still eligible for consideration. Add link from r to s into the tree and form a tree with $(O + D - 1)$ links.

Chapter 4: The methodology

Step 4: If no units are available to assign, stop. Otherwise, there is remaining supply of node r and demand of node s , add edge (r,s) to tree and assign the available amount of units $x_{rs} = \min\{a_r, b_s\}$ to edge. If there exists a cycle, then remove the edge that has assigned zero flow. A new spanning tree is formed with $(O + D - 1)$ edges.

As an example, a step-by-step decoding process of the first binary and Prüfer 1 substrings in Figure 4.2 is presented next.

Figure 4.3: Illustration of Prüfer number decoding procedure

In this example, three primary markets and three collection centres are considered. Supply of goods from the three primary markets (a_i) are 1000, 700 and 1000 respectively, whereas the capacity of each collection centre (b_j) is 2000. The binary substring for collection centre is [0 1 1], indicating that the first collection centre is not opened (therefore its capacity is set to zero). The Prüfer substring between primary markets and collection centres, on the other hand, is $Pr = [1 2 5 4]$, therefore $Pr^- = [3 6]$.

Firstly, node 3 is the smallest number in Pr^- and node 1 is the leftmost digit in Pr . However, since these two nodes are in the same set, the next node in Pr which is not in the same set (node 5) is selected. Assign $x_{35} = \min \{a_3, b_5\} = 1000$, and update $a_3 = a_3 - x_{35}$, and $b_5 = b_5 - x_{35}$. Remove node 3 from Pr^- and node 5 from Pr , and since node 5 is no longer appear in the remaining of part of Pr , it is added to Pr^- .

Secondly, node 5 is smallest number is Pr^- and node 1 is the leftmost digit in Pr . Assign $x_{15} = \min \{a_1, b_5\} = 1000$, and update $a_1 = a_1 - x_{15}$, and $b_5 = b_5 - x_{15}$. Remove node 5 from Pr^- and node 1 from Pr , and since node 1 is no longer appear in the remaining of part of Pr , it is added to Pr^- . This process is repeated until Pr is empty and only nodes 2 and 6 remain in Pr^- . Since there is still available goods in node 2 and sufficient capacity in node 6, add edge (2,6) to the tree and assign $x_{26} = \min \{a_2, b_6\} = 700$. The decoding process is depicted in Figure 4.3.

4.2.2. Genetic operators

Crossover and mutation operators are employed as genetic operators. Crossover is done by exchanging the information of two parents to provide exploration capability. Crossover is performed with probability P_c with one-cut-point crossover as operator.

Figure 4.4: Crossover process

It is done by first selecting chromosomes that will undergo crossover operation. Random number r in the range of $[0, 1]$ is generated for each

chromosome, and chromosomes will undergo crossover operation if $r < P_c$. Secondly, a one-cut-point in each substring is randomly selected and the right parts of the two parents are exchanged to generate the offspring, as depicted in Figure 4.4. Thereafter, the above-mentioned checking and repair procedures (Procedure 1 to 4) are performed to ensure feasibility of the offspring.

Mutation, on the other hand, is done by modifying one or more of the gene values of an existing chromosome. Mutation creates new chromosomes that increase the variability of the population. Here, mutation is done with probability P_m and inversion and displacement mutations are used. First, chromosomes that undergo mutation are selected using the same process as in crossover operation. Secondly, mutation is performed on the selected chromosomes.

In inversion mutation, two points in each substring of selected chromosomes are randomly chosen, and the genes in between are inverted. In displacement mutation, genes in each substring are randomly chosen and inserted in a random position. Thereafter, Procedures 1 to 3 are performed, as the binary substrings of the offspring may not be feasible. The mutation processes are depicted in Figure 4.5.

Figure 4.5: Mutation Process

4.2.3. Evaluation and selection

Evaluation is done to measure the fitness of each chromosome while selection is the

process of selecting chromosomes for the next generation based on the fitness value. Here, evaluation is done by firstly, decoding the chromosomes into a spanning tree, calculating the objective values and then assigning the fitness value.

4.2.3.1. Decoding procedure

The decoding procedure of each chromosome is conducted period-by-period, stage-by-stage and product-by-product, as follows:

For each period:

Step 1: Decode the first Prüfer substring (product flows between first market locations and collection centres):

- For each used product:
 1. Determine supply from each origin i (a_i) and capacity at each destination-facility j (b_j).
 2. Change the capacity of the un-assigned destination-facilities to zero.
 3. New facility constraints (for subsequent periods): if there are allocations from origin i to new facility j in the previous period, assign the available amount of units $x_{ij} = \min \{a_i, b_j\}$ to those new facilities, and update the available supply at origin-facilities and the capacity at destination-facilities.
 4. Perform the original decoding procedure (Procedure 6) to the first Prüfer substring.
- If total inflows to new collection centres are zero, close the respective new collection centres by repairing the first binary substring.

Step 2: Decode the second Prüfer substring (product flows between collection centres and disassembly centres):

Chapter 4: The methodology

- For each used product:
 1. Determine supply from each origin-facility i (a_i) and capacity at destination-facility j (b_j).
 2. Change the capacity of the un-assigned destination-facilities to zero.
 3. Satisfy new facility constraints as in Step 1.
 4. Integrated facilities constraint: if integrated collection and disassembly centres are chosen (i.e., the collection centre is assigned to the distribution or disassembly centre), assign the amount of $x_{ij} = \min \{a_i, b_j\}$ to the respective integrated facilities (distribution or disassembly centre), and update the available supply at origin-facilities and the capacity at destination-facilities.
 5. Perform Procedure 6 on the second Prüfer substring.
- If total inflows to new disassembly centres are zero, close the respective new disassembly centres by repairing the second binary substring.

Step 3: Decode the fifth Prüfer substring (product flows between secondary market locations and recovery centres):

- For each refurbished product:
 1. Determine the demand from each origin i and capacity at each destination-facility j .
 2. Change the capacity of the un-assigned destination-facilities to zero.
 3. Satisfy new facility constraints as in Step 1.
 4. Satisfy integrated facilities constraints as in Step 2.
 5. Perform Procedure 6 on the fifth Prüfer substring.
- If total inflows to new recovery centres are zero, close the respective

new recovery centres by repairing the third binary substring.

Step 4: Decode the third Prüfer substring (recovered module flows between disassembly and recovery centres) and the fourth Prüfer substring (recovered module flows between disassembly centres and spare-part demand points):

- Determine the supply of used products from each disassembly centre (i), the demand for refurbished product from recovery centre (j) and the demand for recovered module spare-part demand point (k).
- Determine the used modules yielded at each disassembly centre, and recovered and new modules required at each recovery centre.
- For each used module that is not required in the recovery process, determine the recyclable and unrecyclable (waste) modules at each disassembly centre.
- For each used module that is required in the recovery process:
 1. Determine the recovered, recyclable, and unrecoverable and unrecyclable (waste) modules at each disassembly centre. Perform Procedure 6 on the third Prüfer substring where supply at each disassembly centre is the recovered modules and capacity at each recovery centre is the required recovered modules. Determine the excess supply from each disassembly centre.
 2. Perform Procedure 6 on the fourth Prüfer substring where supply at each disassembly centre is the excess recovered used modules and capacity at each spare-part demand point is the required recovered modules.
 3. The excess of supply from each disassembly centre is the surplus of

recovered modules that will be sold as spare-parts to the third party.

Step 5: Determine the transportation modes for stages 2 to 5 (the second to the fifth

Prüfer substrings):

- For each stage:

If there is only one mode of transportation available, then choose it as the transportation mode. Otherwise, check the last digit of the respective Prüfer number. If it is odd number, choose transportation mode that has either the minimum unit cost or the minimum unit CO₂ emission (choose it alternately across stages). If it is even number, choose transportation mode with the minimum unit energy consumption.

4.2.3.2. Fitness assignment methods

Two methods of fitness assignment are compared in this research, they are described next.

- a. A weighted sum approach (with randomly generated weights):

The fitness value of a chromosome in each generation is calculated using the following equations:

$$f = \sum_i w_i f'_i \quad (4.1)$$

$$f'_i = \frac{f_i - f_{i,\min}}{f_{i,\max} - f_{i,\min}} \quad (4.2)$$

where f_i is the i^{th} objective function value, $f_{i,\min} / f_{i,\max}$ is the minimum/maximum of f_i in the current population, and w_i is a randomly generated weight coefficient in each generation, such that $\sum_i w_i = 1$. Variable weight generation procedure based on the complete random solution search (Murata et al., 1995) enables the DM to search for the Pareto optimal solutions in

all directions along the Pareto frontier.

b. Pareto ranking method from Fonseca and Fleming (1993):

The procedure is as follows:

- Sort a population size M according to rank. The rank of a chromosome is given by the number of other chromosomes that dominate it in the current generation plus one.
- Assign fitness to chromosomes by interpolating from the best (rank 1) to the worst (rank $n \leq M$) according to some function, usually, but not necessarily, linear.
- Average the fitness of the chromosomes with the same rank, so that all of them are sampled at the same rate.

To avoid premature convergence, a niche formation method to distribute the population over the Pareto-optimal region is used and sharing is performed on the objective function values between pairs of solutions having the same rank (restricted sharing). Sharing is achieved by calculating the following sharing function value between two chromosomes:

$$Sh(d_{ij}) = \begin{cases} 1 - (d_{ij} / \sigma_{share})^2 & \text{if } d_{ij} < \sigma_{share} \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

where d_{ij} denotes the distance between chromosomes i and j and is calculated by using the Euclidean distance formula:

$$d_{ij} = \sqrt{\sum_{k=1}^{OF} (f_{k,i} - f_{k,j})^2} \quad (4.4)$$

where OF denotes the number of objective functions and $f_{k,i}$ and $f_{k,j}$ are the k^{th} objective function values of chromosomes i and j . Niche radius, σ_{share} , is

calculated in each generation by solving the following equation:

$$N_M \cdot \sigma_{share}^{OF-1} - \frac{\prod_{k=1}^{OF} (f_{k,max} - f_{k,min} + \sigma_{share}) - \prod_{k=1}^{OF} (f_{k,max} - f_{k,min})}{\sigma_{share}} = 0, \text{ and } \sigma_{share} > 0. \quad (4.5)$$

N_M indicates the number of chromosomes that can sample the area without interfering with each other, which in this case, it is the population size plus the number of offspring. $f_{k,min} / f_{k,max}$ denotes the minimum/maximum of the k^{th} objective function value in the current population. The shared fitness f_i'' for a chromosome is defined by dividing its average fitness (f_{avg}) by its niche counts:

$$f_i'' = \frac{f_{avg}}{\sum_{j=1}^N Sh(d_{ij})} \quad (4.6)$$

N denotes the number of chromosomes having the same rank.

The selection strategy is adopted from Gen and Cheng (2000), which combines $(\mu + \chi)$ with roulette wheel selection. The strategy is to select the μ number of best chromosomes available from population size μ and χ number from offspring. If μ number of different chromosomes are not available, the vacant pool is filled by using the roulette wheel selection strategy, in which the selection is done by using a selection probability which is proportional to the fitness value.

4.2.3.3. Preserving the non-dominated solutions

Non-dominated solutions found in each generation are preserved in a separate set. In each generation, this set is updated by the newly created offspring. The updating procedure (Gen and Cheng, 2000) is as follows:

Compare each offspring with all non-dominated solutions:

- If it is dominated by one non-dominated solution go to next offspring.

- If it dominates some non-dominated solutions, add it to the non-dominated solution set and delete the solutions dominated by it.
- If it is a new non-dominated solution and dominates none of the existing ones, simply add it into the non-dominated set.

4.2.4. Overall procedure

The overall procedure is as follows:

Step 1: Set GA parameter values.

Set the population size (*pop-size*), maximum number of generations (*max-gen*), crossover rate (P_c) and mutation rate (P_m).

Step 2: Generate initial population $P(g)$.

Generate each chromosome, check its feasibility, and repair it if infeasible.

Repeat this procedure *pop-size* times.

Step 3: Evaluate all chromosomes.

Decode each chromosome, and calculate its objective function values.

Step 4: Record all non-dominated solutions.

Find all non-dominated solutions that exist in the initial population and record it in a separate set $E(g)$.

Step 5: Perform crossover operation.

Perform one-point crossover operation on $P(g)$ with P_c , check the feasibility of the resulting offspring and repair them if infeasible.

Step 6: Perform mutation operation.

Perform displacement and inverse mutation operations (randomly chosen) on $P(g)$ with P_m , check the feasibility of the resulting offspring and repair them if infeasible.

Chapter 4: The methodology

Step 7: Evaluate all offspring $O(g)$.

Decode all offspring and calculate their objective function values.

Step 8: Update the non-dominated solution set $E(g)$.

Update $E(g)$ with $O(g)$.

Step 9: Assign fitness values to $P(g)$ and $O(g)$.

Step 10: Select new population $P(g+1)$ from $P(g)$ and $O(g)$.

Select the best *pop-size* chromosomes available from $P(g)$ and $O(g)$, if *pop-size* different chromosomes are not available, the vacant pool is filled by using roulette wheel selection.

Step 11: Terminating condition.

Increase the generation number. If it is less than or equal to *max-gen*, go to step 5, otherwise stop.

Step 12: Output the non-dominated solution set

The above procedure is depicted in Figure 4.6 and developed using C programming language.

Chapter 4: The methodology

Figure 4.6: The overall GA procedure

4.3. Selection of preferred solutions

In this method, only selected non-dominated solutions found by MIP model are tested under uncertainty using simulation models. The selection is done by determining the weights of all criteria, which is carried out by the decision makers (DMs) using paired-comparisons (Saaty, 1980). The nine-scale scoring is used, which is described in Table 4.1.

Table 4.1: Nine-scale scoring in paired-comparisons

Score	Description	Score	Description
1	Equal importance	7	Very strongly more important
3	Weakly more important	9	Absolutely more important
5	Strongly more important	2,4,6,8	Intermediate values

Thereafter, ranking and scoring of the solutions are done based on those criteria, and finally the solutions with the highest total score are selected. The total score for each solution is the weighted sum of the criteria's scores. The number of selected solutions is decided by the DMs according to the time and budget available for the project. The procedure is depicted in Figure 4.7.

Figure 4.7: Selecting preferred solutions

4.4. Simulation modelling

At this stage, the preferred solutions are re-evaluated using simulation models. The models are built using ARENA 8.01 (Rockwell Software, 2002) with the following assumptions:

1. Multi-period planning horizon, therefore the length of each replication is equal to the period of time considered in the preferred solutions. Supplies, demands, prices, costs, and the allocation of goods to each facility may change across periods.
2. Every used product is assumed to be stored at a collection centre for some period of time before being sent to the disassembly centre; also, every used product is assumed to be stored at a disassembly centre for some period of time before being disassembled into modules.
3. Inter-arrival time of the batches of used products at collection centre and demand at recovery centre are constant. Uncertainty in supply and demand are represented by the variation in batch sizes (the number of used products arriving at the primary markets and the demand of refurbished products at the recovery centres) with Normal distribution.
4. Total used modules required in a recovery centre are calculated based on the demand for refurbished products in the facility.
5. Allocation of entities to each facility is taken from the preferred solutions.
6. Costs (except for fixed costs) and revenues are calculated and accumulated during simulation.
7. Energy consumption in production and transportation processes and CO₂ emission during transportation process are calculated and accumulated during

simulation.

In building the simulation models using Arena, the reverse logistics network of product recovery is modelled as follows:

1. The storage duration is modelled by delaying the entity (delay type: non value added). The entity holding cost rate during the delay period is adjusted according to the entity's holding cost.
2. Transportation between facilities is modelled using *process* module. In the module, all entities are delayed (delay type: transfer), and the entity holding cost rate during the delay period is adjusted according to the entity's transportation cost.
3. Disassembly is modelled by using *duplicate* block. After duplicating, the original entity is disposed of.
4. New modules required to produce each refurbished product are considered as one entity for each refurbished product; therefore the number of required new modules is equal to demand.
5. Reassembly is modelled by using a combination of *match* module and *group* block. The *match* module is used to synchronise several types of entities that are going to be assembled, while the assembly process itself is modelled using the *group* block.
6. The use of *group* block and *match* module result in waiting time due to queuing effect. In this model, the waiting cost is ignored.
7. Capacity variability (across periods) at existing facilities is modelled using different resources with different capacity.

The simulation models are run for several scenarios of uncertainty. The

scenarios are formed by setting some parameters, such as supply and demand, to vary stochastically in each period. Thereafter, the simulation output is analysed to determine the best-preferred solution.

4.5. Determining the best-preferred solution

The selection is performed as follows: for each scenario, the averaged value of each criterion of the selected solutions is ranked and scored. The score for each solution in each scenario is the weighted sum of all the criteria's scores. The overall score for each solution is the sum of all the scenarios' scores. The solution with the highest overall score is the best-preferred network design. The process to select the best-preferred solution is depicted in Figure 4.8.

Figure 4.8: Determining best-preferred solution

4.6. A numerical example

To verify the approach, the method is applied to design a reverse logistics network

for computer refurbishment. The same example as in Chapter 3 is used; however, multi-objective functions and uncertainty are considered. Additional data are presented in Appendix F.

4.6.1. Problem description

The potential network is the same as in Chapter 3. It consists of 7 feasible locations for collection centres, 5 feasible locations for disassembly centres and 3 feasible locations for recovery centres. Four types of used PCs (with 17 types of modules) are refurbished into two types of refurbished PCs (5 new modules are required in the refurbishment process). There is no demand for spare-parts within the company, therefore, the surplus of recovered modules (if any) are sold to a third party. Three modes of transportation are considered (by car, truck or airplane) and the length of each planning period is one year with a planning horizon of 5 years.

It is assumed that DMs consider maximising net revenue (Objective 1) as weakly more important than minimising energy consumption (Objective 2), maximising net revenue as weakly more important than minimising CO_2 emission (Objective 3), and minimising energy consumption as important as minimising CO_2 emission. Therefore, the criteria weights of the three objective functions are, respectively, 0.6, 0.2, and 0.2. Furthermore, it is assumed that a score is given by reversing the order of the rank, and only three solutions are tested using simulation models.

4.6.2. Setting the GA parameters and determining fitness assignment method

The abovementioned st-GA procedure is used to find the non-dominated solutions. Since the quality of offspring is determined by the population size, fitness assignment methods, crossover and mutation rates, and maximum number of

generations, several experiments are done to determine those parameters. In the experiments, one or more parameters are changed and other parameters are kept constant. The base case parameters are:

- Fitness assignment method: Pareto-ranking based approach.
- Population size: According to Alander (1992) a value between n and $2n$ (n is the string length) is optimal for some problems. Since the string length of each chromosome for the above problem is 205, the population size is set to 500.
- Crossover and mutation rates: As mentioned, it is desirable to have a high crossover rate and a low mutation rate. Here, the crossover and mutation rates are set to 0.5 and 0.1, respectively.
- Maximum number of generations (max-gen) is set to 1000.

4.6.2.1. Comparing two fitness assignment methods and population sizes

As mentioned earlier, two methods of fitness assignment, namely weighted sum approach (with randomly generated weights) and Pareto-ranking are adopted. Firstly three different population sizes (300, 400, 500) using each fitness assignment method are compared. Thereafter, comparisons between the two methods with respect to each population size are done. The remaining GA parameters are similar to the base case parameters.

Table 4.2: Comparisons of population size and fitness assignment method

pop-size	Weighted sum approach			Time (min)	Pareto-ranking based approach			Time (min)
	Min	Max	average		min	max	average	
300	16	28	20.5	5	6	10	7.8	6
400	16	34	23.3	8	6	11	9.2	9
500	17	29	20.6	10	9	12	10.2	12

The minimum, maximum and average numbers of solutions found in ten runs are presented in Table 4.2 and the solutions (in criterion space) found in the best run

Chapter 4: The methodology

(the run with the maximum number of solutions found) for the three different population sizes and the fitness assignment methods are presented in Figure 4.9.

In the weighted-sum approach, on the average, more solutions are found when the population size is 400; however in the Pareto-ranking based approach more solutions are found when the population size is higher. The run time for the weighted sum approach is less than the other method and for both methods, it increases when the population size is increased.

Figure 4.9: Solutions found in different population sizes

Furthermore, as can be seen from the above figure, in both fitness assignment methods, solutions found at a higher population size have better quality. Comparing the two methods, more solutions are found using the weighted sum approach, however, in the best runs, all the solutions found using this method for all population sizes are dominated by the solutions found using the Pareto-ranking based approach, indicating that the latter is better in finding the set of non-dominated solutions. Comparison of both methods (population size = 500) is presented in Figure 4.10.

Figure 4.10: Comparison of fitness assignment methods

4.6.2.2. Comparing different crossover and mutation rates

In this section, three pairs of crossover and mutation rates are evaluated. The remaining GA parameters are the same as in the base case. The minimum, maximum and average number of solutions found in ten runs are presented in Table 4.3 and the solutions (in criterion space) found in the best run (the run with the maximum number of solutions found) are presented in Figure 4.11.

Comparing the three combinations of crossover and mutation rates, on the average, more solutions are found using the third combination ($P_c = 0.3$, $P_m=0.3$). However, as can be seen in Figure 4.11, the solutions found using the second combination ($P_c = 0.4$, $P_m=0.2$) have better quality.

Table 4.3: Comparison of crossover and mutation rates

Pc	Pm	Pareto-ranking based approach		
		Min	Max	Average
0.5	0.1	9	12	10.2
0.4	0.2	8	13	11
0.3	0.3	10	13	11.7

Figure 4.11: Comparison of different crossover and mutation rates

4.6.2.3. Comparing different numbers of generations

In this section, three different values of maximum generations (1000, 2500, and 5000) are evaluated. The remaining GA parameters are set to the same level as the base case. The minimum, maximum and average numbers of solutions found in ten runs are presented in Table 4.4 and the solutions (in criterion space) found in the best run (the run with the maximum number of solutions found) are presented in Figure 4.12.

Comparing the number of generations, on the average, as expected, more solutions are found, and the run time increases, when the number of generations is increased. As can be seen in Figure 4.12, the quality of the solutions found in the second (max-gen = 2500) and the third (max-gen 5000) runs are better than in the first run (max-gen = 1000). However, the quality of the solutions found in the second and the third runs are the same, while the run time of the third run is twice that of the second run, indicating that 2500 generations are sufficient for this problem.

Table 4.4: Comparisons of generations

No. of Generations	Pareto-ranking based approach			Time (min)
	min	max	average	
1000	9	12	10.2	12
2500	9	13	11.6	28
5000	12	13	12.8	56

Figure 4.12: Comparison of generations

4.6.3. Determination of the best-preferred network design

Based on the findings in the previous section, the following GA parameters are chosen:

- Population size = 500,

Chapter 4: The methodology

- Number of generations = 2500,
- Crossover rate = 0.4 and mutation rate = 0.2,
- Fitness assignment method used in Pareto-ranking based.

In ten runs, the minimum, maximum and average solutions found are 12, 14, and 12.7 respectively. The best run, in which 14 solutions are found, is chosen and depicted in Figure 4.13. The solutions are then ranked and scored (by reversing the order of the ranks) based on their net revenue, energy consumption and CO₂ emission. Using the aforementioned criteria weights, the total score for each network design is calculated and the three designs with the highest score are selected for testing in uncertain conditions using simulation models. In descending order, they are Solutions 1, 9, and 2 (see Figures 4.14 and 4.15). The complete results of the three solutions are presented in Appendix G.

Figure 4.13: The best run

The first two solutions have the same network structure, but different assignment of allocation centres in period 2. In Solution 1, Retailer 1 and a new

Chapter 4: The methodology

collection centre are chosen, whereas in Solution 9, new and outsourced collection centres are chosen. Both the network structures are almost similar except for outsourced collection centre in Solutions 1 and 9. In Solution 2, besides the new collection centre, only Retailer 1 is chosen as the collection centre for the excess supply of used product in periods 2 to 4.

In all the three solutions, truck is chosen as the transportation mode in all stages and periods as it leads to lower transportation costs and lower energy consumption and CO₂ emission during transportation.

Table 4.5: Three solutions with the highest scores

ND Solutions	Total Net Revenue (\$)	Energy Cons. (GJ)	CO ₂ emission (KgCE)	Score Obj 1	Score Obj 2	Score Obj 3	Total Score
Solution 1	4,590,164	493,758	304,756	13	13	3	11.00
Solution 9	4,593,188	493,759	304,878	14	11	1	10.80
Solution 2	4,586,539	493,756	304,634	11	14	4	10.20

Figure 4.14: A depiction of Solutions 1 and 9

Figure 4.15: A depiction of Solution 2

Three simulation models representing the preferred solutions are built using the assumptions mentioned in the previous section, and the models are run under scenarios of uncertainty. In this example, eight scenarios representing supply and

Chapter 4: The methodology

demand variability and variations in recoverability of used modules are considered and presented in Tables 4.6 and 4.7.

Each scenario is run for ten replications, and the results of each criterion are averaged. For each scenario, the average values of each criterion of the three designs are compared and then scored as the reverse order of the rank (the lower the rank, the higher the score). The calculation of total and overall score is presented in Table 4.8.

Table 4.6: Scenarios of uncertainty

Scenario	Supply variability	Demand variability	Recoverability
1	low	Low	Low
2	low	Low	high
3	low	High	low
4	low	High	high
5	high	Low	low
6	high	Low	high
7	high	High	low
8	high	High	high

Table 4.7: Description of each parameter variability

Uncertain Components	Variability	Descriptions
Supply	low	Normal distribution, standard deviation = 20% of yearly supply
	high	Normal distribution, standard deviation = 40% of yearly supply
Demand	low	Normal distribution, standard deviation = 20% of yearly demand
	high	Normal distribution, standard deviation = 40% of yearly demand
Recoverability Of used modules	low	50%
	high	80%

The result shows that Solution 2 has the highest overall score compared to Solutions 1 and 9. Even though Solutions 1 and 9 had the highest total score in the previous stage, Solution 2 has the highest overall score in the simulation stage, indicating that among the three solutions, under the DM preference, it has the best performance under uncertainty.

Table 4.8: Overall scores for preferred solutions

Solution	Scenario	Supply variability	Demand variability	β	Net revenue		Energy consumption		CO ₂ emission		Total score
					\$	Score	GJ	Score	kgCE	Score	
1	1	low	low	low	1,422,841	2	488,540	2	295,440	2	2
	2	low	low	high	6,782,433	2	493,696	2	304,269	3	2.2
	3	low	high	low	1,403,614	1	488,285	2	295,112	2	1.4
	4	low	high	high	6,781,193	1	493,496	2	304,388	2	1.4
	5	high	low	low	1,342,968	1	486,891	3	293,987	3	1.8
	6	high	low	high	6,774,501	2	493,608	2	304,420	2	2
	7	high	high	low	1,345,033	1	486,909	3	293,907	3	1.8
	8	high	high	high	6,629,788	1	491,005	2	302,513	3	1.6
Overall Score: 14.2											
9	1	low	low	low	1,461,386	3	489,279	1	295,923	1	2.2
	2	low	low	high	6,778,841	1	493,509	3	304,361	1	1.4
	3	low	high	low	1,462,448	3	489,327	1	295,805	1	2.2
	4	low	high	high	6,807,758	3	493,941	1	304,728	1	2.2
	5	high	low	low	1,388,106	2	487,985	1	295,047	1	1.6
	6	high	low	high	6,706,717	1	493,399	3	303,795	3	1.8
	7	high	high	low	1,425,666	3	489,205	1	295,902	1	2.2
	8	high	high	high	6,718,426	3	490,825	3	302,875	2	2.8
Overall Score: 16.4											
2	1	low	Low	low	1,416,762	1	488,478	3	295,094	3	1.8
	2	low	Low	high	6,801,550	3	494,134	1	304,338	2	2.4
	3	low	High	low	1,432,134	2	488,239	3	294,810	3	2.4
	4	low	High	high	6,807,047	2	493,213	3	304,346	3	2.4
	5	high	Low	low	1,412,629	3	487,846	2	294,970	2	2.6
	6	high	Low	high	6,821,508	3	494,880	1	305,192	1	2.2
	7	high	High	low	1,414,123	2	487,846	2	294,970	2	2
	8	high	High	high	6,681,487	2	492,394	1	303,219	1	1.6
Overall Score: 17.4											

4.7. Summary

An approach to the design of reverse logistics network for product recovery is presented in this chapter. The developed approach consists of two models namely MIP and simulation models. MIP is used to find non-dominated solutions when the product characteristics, multiple criteria, and multiple periods are considered. The multi-objective MIP model is solved using st-GA. Simulation models, on the other hand, are used to re-evaluate the selected solutions (based on preferences of DM) under uncertain conditions.

The approach is applied to a numerical example with data gathered from various sources. Several experiments are conducted to select the suitable parameters and method. The results of the numerical example show that the developed st-GA is stable enough in finding non-dominated solutions in each run as the number of solutions found in 10 runs are between 12 and 14. Considering the reverse logistics network problem, it is shown that solutions having higher scores in the deterministic stage do not necessarily have higher scores in the simulation stage, indicating that uncertainty is indeed an important characteristic that must be considered in designing the network.

CHAPTER 5

CASE STUDY AT COMPUCARE ASIA-PACIFIC

In this chapter, a case study conducted at Compucare* Asia-Pacific, a subsidiary of a multi-national computer manufacturer Computech*, is discussed. The methodology discussed in Chapter 4 is applied to explore the possibility of implementing part recovery at Compucare Asia-Pacific.

5.1. The company's background

Computech is a computer manufacturing company which designs, manufactures and markets personal computers and related personal computing and data communication solutions. The products include desktop and notebook personal computers, related devices and peripherals, networking and connectivity products and digital lifestyle products. The company distributes its products through wholesalers, resellers, national and regional retailers, and cataloguers. Its market includes America, Europe, Japan and Asia-Pacific. The company also offers after-sales service, including extended warranty contracts through its subsidiary company, Compucare.

Compucare Asia-Pacific is responsible for repair operations, spare-parts supply, and operations support of Compucare across the region. The network structure consists of service providers (SPs), local sub-hubs (LSHs), regional distribution centres (RDCs), and repair facilities that comprise original equipment manufacturers (OEMs) and third party repair vendors (RVs). Currently, Compucare Asia-Pacific repair network consists of one hundred SPs, three LSHs (in India, Korea and China), three RDCs (located in Japan, Australia, and Singapore) and seventy

* fictitious names

repair facilities. Their current network is depicted in Figure 5.1: Singapore RDC covers all of Asia-Pacific, except Japan, whereas Japan and Australia RDCs only cover their own countries.

The repair process begins with customers returning their faulty equipment (desktops/notebooks) to SPs. The SPs assist Compucare in identifying and repairing faulty equipment by swapping faulty modules (parts) with good modules (level one repair), whereas the faulty modules themselves are repaired by their original equipment manufacturers (OEMs) or RVs (level two repair).

Figure 5.1: Compucare's repair network

The process flow for most parts is as follows: Faulty modules identified as known bad boards (KBBs) are forwarded from SPs to their respective LSHs or RDCs. The local sub-hub serves as a transition point where parts from SPs are first consolidated in their respective countries before being shipped out to the respective

RDCs. The RDC is basically a warehouse facility, which handles the storage of computer components, and material handling and packaging operations. The LSHs and RDCs are outsourced to third party companies specialising in warehouse operations. KBBs are classified according to warranty at RDCs. KBBs under warranty are consolidated and shipped back to their OEMs, whereas KBBs without warranty are shipped either to their OEMs or repair vendors (RVs) depending on the costs. Recoverable KBBs are fixed, whereas unrecoverable KBBs are disposed of, and RDCs order new parts from OEMs to replace them. Fixed KBBs and new parts, now identified as known good boards (KGBs), are shipped back to RDCs that will forward them to LSHs or SPs, as requested.

5.2. Case study

In this case study, the objective is to explore the possibility of implementing part (module) recovery at Compucare by designing its reverse logistics network in the Asia-Pacific region using the existing spare-part repair network and other potential facilities. The recovered modules are used to replace unrecoverable, out-of-warranty faulty parts and the surplus will be sold to the spare-parts market. The designed network should consider economic and environmental aspects, should be able to handle uncertainty, and be valid for the next three years in the planning horizon.

5.2.1. The potential network for part recovery

In the proposed approach, the network for product recovery consists of three facilities, namely collection centres for collecting used equipment, disassembly centres for disassembly, testing and grading, and minor repair of used parts, and recovery centre for reassembly of used parts into refurbished products. Since in this case study the recovery activity focuses on part recovery, only the collection,

disassembly, testing and grading and minor repair activities are required. As a result, only the locations of the collection and disassembly centres are necessary to be determined.

Figure 5.2: The potential network for part recovery

Computcare's existing repair network actually consists of third party facilities. Therefore, only the third party facilities and the new dedicated recovery facilities (to be opened if chosen) are feasible for the module recovery network. Furthermore, the approach allows the possibility of having integrated facilities but only in the existing forward chain and at the new recovery facilities. Therefore, feasible locations for collection centres include SPs, new collection centres, LSHs, and new disassembly centres, whereas feasible locations for disassembly centres include RDCs, and new disassembly centres. The potential network is depicted in Figure 5.2.

The primary market locations are assumed to be in 8 countries (12 cities), namely China (Beijing and Shanghai, Hong Kong), South Korea (Seoul), Malaysia (Kuala Lumpur), Singapore, Thailand (Bangkok), Taiwan (Taipei), Japan (Tokyo and Osaka), and Australia (Sydney and Melbourne). The aforementioned countries

are selected based on each country's computer sales and their proximity to RDCs.

The spare-part repair network facilities considered in the study are those located in the aforementioned countries. Compucare indicates that it is considering opening another RDC in China to replace the existing LSHs, taking into account China's emerging market and the trend of OEMs and RVs shifting their facilities to that country.

Lee (2004) reported that Asia-Pacific dominates the global production of notebook PCs. In 2003, more than 76% of the global notebook PC production came from the region, and it is estimated that Taiwanese companies controlled as much as 57% of the global notebook production. Major PC vendors need their Taiwanese OEM partners to assemble notebook PCs in China due to its potential domestic market and lower manufacturing costs. Chinese mainland constituted 44.6% of the total PC market in Asia-Pacific, excluding Japan, in 2003. As a result, the top five notebook PC subsystem manufacturers in Taiwan (Quanta, Compal, ASUS, Winstron, and Inventec) had shifted more than 40% of their manufacturing capacity to Shanghai by 2003 (Lee, 2004). Furthermore, it is also reported that South Korea's leading notebook PC subsystem manufacturer, Samsung, also set up an assembly plant in Shanghai in mid 2003. The notebook PC manufacturing outsourcing reflects the PC manufacturing outsourcing in general.

Therefore, an additional RDC located in Shanghai is included in the case study replacing the two LSHs in Beijing and Shanghai. Hence, 4 RDCs (Singapore, Tokyo, Sydney, and Shanghai), 1 LSH (Seoul) and 12 SPs (Beijing, Shanghai, Hong Kong, Seoul, Kuala Lumpur, Singapore, Bangkok, Taipei, Tokyo, Osaka, Sydney, and Melbourne) are included as potential facilities.

Chapter 5: Case study at Computcare Asia-Pacific

Figure 5.3: Potential locations for recovery facilities

Besides the spare-part network, the potential reverse logistics network also consists of 8 new collection centres (Beijing, Shanghai, Bangkok, Taipei, Tokyo, Osaka, Sydney, and Melbourne) and 2 new disassembly centres (Tokyo and Shanghai). Determination of new facility locations are done based on computer sales in primary market locations, as higher sales volumes may lead to higher supply levels of used computers at collection centres. The potential locations for recovery facilities are presented in Figure 5.3. Lastly, three modes of transportation are considered, namely by car, truck and airplane.

5.2.2. Designing reverse logistics network for part recovery

5.2.2.1. Data

In order to apply the developed approach, a set of data is required. Data that is not provided by the company is based on published data taken from other sources.

The assumptions related to the data used in this part of the case study are as follows:

1. Products considered in the case study are classified into desktop and notebook. Only certain parts are taken into account in the study, which are the main logic board/main processing unit (MLB/MPU), display, mass storage, optical drive, card/modem, battery, power supply and memory (14 parts in total).
2. New product prices are assumed based on online sources and used product prices are assumed to be 20% of new prices.
3. Supplies of used products from the primary markets for the next three periods are assumed based on computer sales (shipment) data in 2001-2004 of the aforementioned countries (Gilliland, 2004). It is assumed that the growth rate of computer sales is equal to the growth rate of computer returns. Furthermore, it is

assumed that Computech has a 5% market-share for notebooks and 1-2.5% for desktops in the aforementioned countries with trade-in rate at 5% for notebooks and 1-5% for desktops. This assumption is based on a published article (Harris, 1998) and information provided by Compucare indicating that Computech has a 5% market share in the US.

4. Demands for spare-parts within the company (at RDCs) are determined based on data provided by Compucare. As mentioned earlier, recovered parts are used to replace unrecoverable and out-of-warranty parts. Therefore, demand is calculated by assuming that the warranty fraction is 0.5 and the recoverability fraction is 0.8. Furthermore, it is assumed that demand growth rate is the same as supply growth rate.
5. All prices and costs are assumed to increase by 5% each year. Fixed costs (\$/year) apply at all collection centres and all disassembly centres, except for RDCs Tokyo and Sydney, at which variable holding costs apply. Annual holding cost per unit is assumed to be 20% of used product value and every used product is assumed to be stored for an average of 7 days at collection centres and 7 days at disassembly centres.
6. Processing costs at disassembly centres are assumed to be 2% of the original product values at RDCs Tokyo and Sydney, 1.5% for Tokyo new DC, and 1% for the rest of the potential disassembly centres.
7. Costs, energy consumption and CO₂ emission during transportation between facilities are calculated based on the distances involved, and the consolidation volume at each facility. It is assumed that transportation between countries is carried out by airplane, whereas transportation within each country is carried out

either by car, truck or airplane depending on the distance.

8. No duty fees apply as all potential DCs are assumed to be located in the free trade zones (FTZ).
9. Energy consumption at disassembly centres is based on data by Luo et al. (2001).
10. Purchasing costs of the abovementioned parts are aggregated based on their volumes.
11. Disposal cost is assumed to be 1% of the new part price, except for MLB/MPU, which have a zero disposal cost, as suggested by Compucare personnel.
12. Recyclable module price is assumed to be 5% of the new part price.
13. Spare-part or good-condition used module price is assumed to be 20% of the new part price
14. Recyclability and recoverability fractions are determined based on interviews with Compucare personnel. As there are salvage values for certain parts, recyclable fractions for those parts are equal to 100%. Recyclable fractions for other parts are assumed to be 75%. On the other hand, since the demand for spare-parts within the company is small, and currently it is not known whether there is any demand for spare-parts from other parties, the recoverable fractions are assumed to be 10-25%.
15. Capacities of all potential facilities are assumed and the values are constant in all periods.
16. The complete set of data may be found in Appendix H.

5.2.2.2. Finding non-dominated solutions and selecting preferred solutions

Non-dominated solutions are found using spanning-tree based GA procedure. Based on previous experience in numerical examples, the following GA parameters are

chosen.

- Population size: 600 (the string length is 291),
- Number of generations: 2500 and 5000,
- Crossover rate: 0.4 and mutation rate: 0.2,
- Fitness assignment method: Pareto-ranking based.

In this case study, two values for the maximum number of generations are tried. The minimum, maximum and average numbers of solutions found in ten runs along with their running times are presented in Table 5.1, while the quality of the solutions found in the best runs (the runs with the maximum number of solutions) are compared in Figure 5.4.

Table 5.1: Comparisons of generations in the case study

No. of Generations	Min	Max	Average	Time (min)
2500	23	46	34.5	109
5000	26	69	53.2	218

Figure 5.4: The best runs

As expected, more generations require more time to solve, but they result in more solutions (on average). Comparing the quality of solutions, some solutions

found using max-gen = 2500 are dominated by solutions found using max-gen = 5000, indicating that the solutions found using the latter parameter have better quality. Therefore, the latter results are used for the case study.

In order to select preferred solutions, the criteria weights must be determined first. Since Compucare is still exploring the possibility of implementing part recovery, it is safe to assume that maximising net revenue is more important than the other two objectives. In paired comparisons, maximising net revenue is weakly more important than minimising energy consumption and CO₂ emission, while minimising energy consumptions is as important as minimising CO₂ emission. Therefore, criteria weights for maximising net revenue, minimising energy consumption and CO₂ emission are assumed to be 0.6, 0.2, and 0.2, respectively.

After determining the criteria weights, the non-dominated solutions found are ranked according to each criterion and scored. Scoring is assigned as the reverse of the rank and if two or more solutions have the same objective values, the scores for those solutions are averaged. Total score of each solution is calculated as the weighted sum of its objective values. Ten solutions with the highest total scores are presented in Table 5.2.

Table 5.2: Ten solutions with the highest score

No.	Solution No.	Net Revenue (\$)	Energy Consumption (GJ)	CO ₂ Emission (KgCE)	Score Obj. 1	Score Obj. 2	Score Obj. 3	Total Score
1	22	4,239,259	229,312	3,864,761	69	20	10	47.4
2	31	4,239,118	229,306	3,865,582	68	22	8	46.8
3	11	4,239,114	229,307	3,865,550	67	21	9	46.2
4	56	4,238,982	229,302	3,866,290	66	24	6	45.6
5	52	4,172,601	228,935	3,200,610	55	34	28	45.4
6	29	4,238,979	229,302	3,866,258	65	23	7	45
7	18	4,238,973	229,301	3,866,371	64	25	5	44.4
8	33	4,172,460	228,930	3,201,432	54	36	22	44
9	55	4,210,042	229,245	4,031,253	62	31	3	44
10	64	4,172,456	228,931	3,201,399	53	35	25	43.8

Figure 5.5: Network structure of Solutions 22 and 55

In this case study three network structures are selected to be re-evaluated using simulation models. Solutions 22, 31, 11, 56, 29, 18 and 55 have the same network structure, but different allocations in Stage 3 (between disassembly centres and spare-part demand points within the company), except for Solution 55 in which there is also different allocations in Stage 2 (between collection and disassembly centres). Comparing the objective values, the aforementioned solutions do not differ significantly, except for Solution 55. On the other hand, Solutions 52, 33 and 64 also have the same network structure but different allocations in Stage 3. These solutions

also do not differ significantly in their objective values. Based on these reasons, only Solutions 22, 52 and 55 are selected as the preferred solutions to be re-evaluated in the simulation stage. The network structures of the three solutions are depicted in Figures 5.5 and 5.6 and the complete results are presented in Appendix I.

As can be seen, in Solutions 22 and 55, nine collection centres (2, 3, 4, 6, 7, 11, 13, 14 and 21) and three disassembly centres are assigned, whereas in solution 52, ten collection centres (2, 3, 4, 6, 7, 11, 13, 14, 18 and 21) and four disassembly centres (2, 3, 4 and 5) are assigned. Both designs use existing facilities in the repair network and new facilities, and new collection centres are chosen for countries with relatively high supply of used PCs.

Figure 5.6: Network structure of Solutions 52

Regarding the assignment of DCs, in Solutions 22 and 55, one new disassembly centre at Shanghai (DC 2) and the existing RDCs at Singapore and Sydney (DC 3 and 5 respectively) are chosen to accommodate flows of used products. The new DC is assigned to serve most countries in the region. In Solution 52, on the other hand, besides the aforementioned DCs, the existing RDC at Tokyo is also chosen for periods 2 and 3, indicating that there are changes in the allocations of used products collected in Japan, from solely allocated to new DC (in period 1), to new DC and RDC Tokyo (in periods 2 and 3).

Pertaining to the transportation mode, in all the three solutions, airplane is used to move goods across countries in all periods as it is the only available transportation mode. However, to move goods within a country, truck is chosen in all periods as it leads to lower transportation costs and lower energy consumption and CO₂ emission during transportation.

Concerning the objective values of the preferred solutions, Solution 22 has the highest revenue, whereas Solution 52 has the lowest energy consumption and CO₂ emission.

5.2.2.3. Determination of the best-preferred solution

In order to determine the preferred solution that can perform well under uncertainty, the preferred solutions are simulated under scenarios of uncertainty. In this study, three parameters are varied to observe their effects on the objective functions. Those parameters are supply variability, recyclability fractions, and recoverability fractions as supply and quality of used products are important aspects in part recovery. Demand for spare-parts within the company is not explicitly varied as it is implicitly varied when recoverability fraction is varied. The scenarios and their descriptions are

listed in Tables 5.3 and 5.4.

Table 5.3: Scenarios of uncertainty in the case study

Scenario No.	Supply variability	Recyclability Fraction (α)	Recoverability Fraction (β)
1	Low	Low	Low
2	Low	Low	High
3	Low	High	Low
4	Low	High	High
5	High	Low	Low
6	High	Low	High
7	High	High	Low
8	High	High	High

Table 5.4: Descriptions of each parameter in scenarios of uncertainty

Parameters		Variability/Value	Descriptions
Supply variability		High	Normal distribution, mean = annual supply, standard deviation = 40% of annual supply
		Low	Normal distribution, mean = annual supply, standard deviation = 20% of annual supply
Recyclability fraction	α_1 (Part 1 and 2)	High	0.85
		Low	0.65
	α_2 (Part 3 to 8)	High	1
		Low	0.9
	α_3 (Part 9 to 14)	High	0.85
		Low	0.65
Recoverability fraction	β_1 (Part 1 and 2)	High	0.35
		Low	0.2
	β_2 (Part 3 to 8)	High	0.35
		Low	0.2
	β_3 (Part 9 to 14)	High	0.2
		Low	0.1

Simulation models of the preferred solutions are built using Arena 8.01 with the following assumptions:

1. Simulation length is 3 years, excluding a 1-year warming-up period. Parameters may change across periods.
2. Each batch of used products arrives at collection centres every 15 days. Batch size is calculated as the annual supply divided by the arrival frequency in each year (24). In uncertain scenarios, batch size is varied.

3. Allocations to each facility are taken from the preferred solutions.
4. Every used product is stored at the collection and disassembly centres for an average of 7 days.
5. Costs (except for fixed costs), revenues, energy consumption, and CO₂ emission are calculated and accumulated during simulation time.
6. Each run consists of 10 replications.

Figure 5.7: Simulation results (total net revenue)

The simulation results are presented in Figures 5.7 to 5.9. The results show that the parameter that has a great effect on net revenue is recoverability fraction as it determines the revenue coming from selling spare-part to other parties and savings from using spare-parts for internal purposes. A difference of 0.1 to 0.15 in the fraction recoverability results in a difference of 53.12% to 58.62% in the net revenue. Variation of recyclability fraction also affects the net revenue; however the effect is not as significant as the recoverability fraction as recyclability fractions only determine the revenue from selling recyclable modules to recyclers and the recyclable modules are sold for a much cheaper price than recovered parts. A difference of 0.1 to 0.2 in the fraction of recyclability results in a difference of 8.02%

to 8.44% in the net revenue. Furthermore, compared to recyclability fractions, supply variability has less impact on net revenue.

Figure 5.8: Simulation results (energy consumption)

Figure 5.9: Simulation results (CO₂ emission)

The parameters that affect the other two objective functions are supply variability and recoverability fractions, with the former having less impact than the latter, as recoverability fraction determines energy consumption and CO₂ emission between DCs and spare-part stock centres. A difference of 0.1 to 0.15 in fraction recoverability results in a difference of up to 10% difference in energy consumption and a difference of 1.74% to 2.57% in CO₂ emission. However, a difference in the fraction of recyclability results in no difference in the energy consumption and the

CO₂ emission as fraction of recyclability only determines the number of modules to be sold to recycler or to be disposed of.

The overall score for each solution is calculated by firstly ranking the three solutions based on their objective function values, assigning scores as the reverse of the rank, calculating the total score for each solution in each scenario as a weighted sum of the scores, and lastly summing the total scores for each solution. The overall scores of the preferred solutions are presented in Table 5.5. The results show that among the three solutions, Solution 22 has the highest net revenue in all the scenarios, the second lowest value of energy consumption in half of the scenarios, and the second lowest values of CO₂ emission in all the scenarios. Solution 52, on the other hand, has the lowest values of all objective functions in all the scenarios. Lastly, Solution 55 has the second highest net revenue in all the scenarios, the lowest energy consumption in half of the scenarios and highest CO₂ emission in all the scenarios. Therefore, overall, Solution 22 has the highest score compared to Solutions 52 and 55. Hence, just as in the deterministic stage, among the three preferred solutions, Solution 22 has the highest score under scenarios of uncertainty.

5.2.2.4. Summary

A case study is conducted at Compucare to design a reverse logistics network for part recovery by applying the proposed approach. During the stage of finding non-dominated solutions, sixty nine solutions are found. Under the assumed DM preferences, three network designs are chosen as preferred solutions and re-evaluated using simulation models. The simulation results indicate that, under the assumed preferences, Solution 22, where network structure consists of nine collection centres (five new) and three disassembly centres (one new), has the highest overall score.

Table 5.5: Overall score of preferred solutions

Solutions	Scenario	Supply variability	α	β	Net revenue		Energy Consumption		CO ₂ Emission		Total Score
					\$	Rank	Score	GJ	Rank	Score	
22	1	low	low	low	2,707,056	1	3	226,557	3	1	2
	2	low	low	high	6,431,323	1	3	226,773	2	2	2
	3	low	high	low	3,268,951	1	3	226,557	3	1	2
	4	low	high	high	6,991,464	1	3	226,773	2	2	2
	5	high	low	low	2,711,956	1	3	226,522	2	2	2
	6	high	low	high	6,421,612	1	3	226,814	3	1	2
	7	high	high	low	3,273,375	1	3	226,522	2	2	2
	8	high	high	high	6,981,750	1	3	226,814	3	1	2
Overall Score:											20
52	1	low	low	low	2,630,982	3	1	226,230	1	3	3
	2	low	low	high	6,357,533	3	1	226,430	1	3	3
	3	low	high	low	3,192,968	3	1	226,230	1	3	3
	4	low	high	high	6,917,421	3	1	226,430	1	3	3
	5	high	low	low	2,637,604	3	1	226,331	1	3	3
	6	high	low	high	6,362,768	3	1	226,396	1	3	3
	7	high	high	low	3,199,761	3	1	226,331	1	3	3
	8	high	high	high	6,922,785	3	1	226,396	1	3	3
Overall Score:											14.4
55	1	low	low	low	2,680,937	2	2	226,404	2	3	1
	2	low	low	high	6,403,599	2	2	226,806	3	1	1
	3	low	high	low	3,242,507	2	2	226,404	2	2	1
	4	low	high	high	6,963,735	2	2	226,806	3	1	1
	5	high	low	low	2,684,050	2	2	226,618	3	1	1
	6	high	low	high	6,383,976	2	2	226,555	2	2	1
	7	high	high	low	3,246,196	2	2	226,618	3	1	1
	8	high	high	high	6,943,489	2	2	226,555	2	2	1
Overall Score:											13.8

Therefore, if Compucare wants to implement part recovery in Asia-Pacific region, it needs to open five new collection centres and one new disassembly centre. Under the base case scenario, in which parameters are set to moderate values, implementing module recovery means having expected net revenue (including savings) of \$4.24 million in three years. Hence, besides aiding marketing efforts by promoting a “green” image, implementing part recovery can save costs and generate additional revenues to the company.

In general, even though currently product or part recovery is not considered as a manufacturer’s obligation in most countries in the Asia-Pacific region, government legislations may change in the near future. Therefore, preparing for the change by implementing it in a step-by-step approach is the best way to proceed.

CHAPTER 6

CONCLUSIONS, CONTRIBUTIONS AND FUTURE RESEARCH DIRECTIONS

6.1 Conclusions

In an environmentally-conscious world with limited resources and increasing levels of waste, it is a matter of time before product recovery becomes an essential activity in many countries. This research proposed an approach to designing a reverse logistics network for discrete product recovery that incorporates characteristics which exist in a real system. The decision making approach can be used by manufacturers to prepare themselves towards the changing environmental legislations.

The developed approach consists of mathematical and simulation models. A mixed integer programming model is developed to represent the reverse logistics network for discrete product recovery, and it incorporates:

- Product characteristics that exist in discrete products, such as modular product design and multiple product configurations.
- Multiple objective functions, namely maximising net revenue and minimising environmental impacts, i.e., minimising energy consumption and CO₂ emission.
- Multi-period planning horizon.
- Facility sharing between forward and reverse flows and integrated recovery facilities along with the flexibility of opening new facilities when required.

Spanning-tree based genetic algorithm is employed to find the set of non-dominated solutions for the multi-objective MIP model. The proposed approach uses

st- GA as it is efficient in finding a set of non-dominated solutions in a single run. The candidate solutions for multi-objective, multi-period network problem are represented as chromosomes, which are a combination of binary and Prüfer numbers.

One-point crossover, inverse and displacement mutation are employed as genetic operators, whereas the Pareto-based fitness assignment method with restricted sharing is adopted to determine an individual's fitness value. Infeasible solutions may be created in the generation and genetic operations. A repair strategy is adopted to handle infeasible solutions and three check-and-repair procedures for binary sub-strings related to the model constraints are developed. The decoding procedure, on the other hand, is modified to accommodate model characteristics such as multi-product configurations and modular design. The selection strategy is a combination of $(\mu+\chi)$ with roulette wheel selection. Furthermore, non-dominated solutions found in a given generation are recorded in a separate set, which is updated in each generation.

Preferred non-dominated solutions (selected by the decision makers) are re-evaluated under scenarios of uncertainty using simulation models. Therefore, the proposed approach reduces the complexity by only re-evaluating solutions preferred by the decision makers. Lastly, the best-preferred solution is selected based on its performance under uncertainty.

The approach is verified by using a numerical example with data collected from various resources. The results indicate that solutions with higher score (under DM's preferences) during the deterministic stage do not necessarily have higher score in the simulation stage also.

A case study is carried out at Compucare Asia-Pacific to validate the

approach. The proposed approach is applied to explore the possibility of implementing part recovery at Compucare by designing a reverse logistics for the purpose. Recovered parts from used computers are used (instead of newly purchased parts) to replace the unrecoverable out-of-warranty modules. The potential network consists of several facilities from the existing repair network facilities (SPs, LSHs and RDCs) and additional potential locations for new recovery facilities (new collection and disassembly centres). In the generation stage, sixty nine solutions are found and three different designs with the highest scores are selected as the preferred solutions. The solutions are simulated under scenarios of uncertainty; each scenario is a combination of variability in the supply of faulty parts, recyclability and recoverability fractions.

During this stage, Solution 22, whose network structure consists of nine collection centres (five new) and three disassembly centres (one new), has the highest overall score. Thus, in order to implement module recovery in the Asia Pacific region, Compucare needs to open five new collection centres and one new disassembly centre, besides the existing facilities. Under the base case scenario, in which the parameters are set to moderate values, implementing part recovery results expected in net revenues (including savings) of \$4.24 million in three years. Hence, besides complying with environmental legislations, implementing product recovery generates additional revenues and aids the company in marketing effort by promoting a “green” image.

6.2. Contributions

This research contributes in providing an improved decision making approach for the design of reverse logistics network for discrete product recovery by

considering important characteristics that exist in the real system. The approach can be used by the manufacturers or other independent parties to design reverse logistics network for product and/or part recovery. Besides maximising the net revenue, it also takes into account minimising environmental impacts as another objective functions. It also considers the possibilities of implementing the recovery activities in an economical way by taking into account facility sharing at forward chain facilities, integrated recovery facility and opening new recovery facility as required in any period in the planning horizon.

Regarding the st-GA, three new checking and repair procedures for binary substrings are developed and the basic decoding procedure is modified to take into account multi-period planning horizon, multi-product configurations, different transportation modes and also to accommodate new facility and integrated facility constraints. Since there are usually a lot of non-dominated solutions for the multi-objective problem, the approach reduces the complexity by only re-evaluating non-dominated solutions that are preferred by the DM.

6.3. Future research directions

The proposed approach considers only discrete product as the source for recovery activities. Furthermore, only part recovery and product refurbishment are considered as the available options for recovery activities. To make the approach more generic, the approach may include continuous products as the source of used products and recycling as one of the recovery options.

Secondly, the reverse logistics network may span across many countries. Even though the proposed model has already considered taxes for transporting goods between countries, complexities regarding currency exchange rate and environmental

regulations (such as the prohibition on import of used goods) in certain countries have not been considered in the model. The currency exchange factor can be introduced in the first objective function whereas environmental regulations can be modelled by introducing high penalty in the first objective function or by adding logical constraints.

As the network problem is a combinatorial problem, the time needed to generate non-dominated solutions increases when the number of potential locations for recovery facilities, product types and module types are increased. In proposed approach, the decoding stage takes most of the time as decoding each chromosome must be done on a period by period, stage by stage and product by product basis. A more efficient decoding procedure could accelerate the process significantly.

LIST OF PUBLICATIONS

- Kusumastuti, R.D., Piplani, R., Lim, G.H., 2003. A Reverse Logistics Network Model for Product Refurbishment, *Proceedings of 7th International Conference of the Decision Sciences Institute combined with 8th Annual Meeting of the Asia-Pacific Region of the DSI*, 4-8 July, Shanghai, China.
- Kusumastuti, R.D., Piplani, R., Lim, G.H., 2004. An Approach to Design Reverse Logistics Networks for Product Recovery, *Proceedings of IEEE International Engineering Management Conference*, 18-21 October 2004, Singapore, pp. 1239 – 1243.
- Lim G.H., Kusumastuti R.D. and Piplani R., 2005, Designing a Reverse Supply Chain Network for Product Refurbishment, *Proceedings of the International Conference on Simulation and Modelling*, 17-19 January 2005, Bangkok, Thailand, pp. 95 -100.

REFERENCES

- Alander, J.T., 1992. On optimal population size of genetic algorithms, *Proceedings of CompEuro 92*, pp. 65-70.
- Ayres, R., Ferrer, G., Van Leynseele, T., 1997. Eco-efficiency, asset recovery and remanufacturing, *European Management Journal*, Vol. 15, No. 5, pp. 557-574.
- Barros, A.I.R., Dekker, R., Scholten, V., 1998. A two level network for recycling sand: a case study, *European Journal of Operational Research*, Vol. 110, pp. 199-214.
- Bautista, J., Pereira J., 2005. Modelling problem of locating collection areas for urban waste management: an application to the metropolitan area of Barcelona, *Omega*, article in press.
- Beamon, B.M., Fernandes, C., 2004. Supply-chain network configuration for product recovery, *Production Planning & Control*, Vol. 15, No. 3, 270-281.
- Boon, J.E., Isaacs, J.A., Gupta, S.M., 2000. Economics of PC recycling, *Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing*, Vol. 4193, pp.13-20.
- Carter, C.R., Ellram, L.M., 1998. Reverse logistics: A review of the literature and framework for future investigation, *Journal of Business Logistics*, Vol. 19, No.1, pp. 85-102.
- Caruso, C., Colorni, A., Paruccini, M., 1993. The regional urban solid waste management system: a modelling approach, *European Journal of Operational Research*, Vol. 70, pp. 16-30.
- Cayley, A., 1889. A Theorem on Trees, *Quart. J. Math.*, Vol. 23, pp. 376-378.
- Clegg, A.J., Williams, D.J., Uzsoy, R., 1995. Production planning for companies with remanufacturing capability, *Proceedings of the 1995 International Symposium on Electronics and the Environment*, pp. 186-191.
- Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B., 2002. *Evolutionary Algorithms for Solving Multi-Objective Problems*, New York: Kluwer.
- Dekker, R., Fleischmann, M., Inderfurth, K., Van Wassenhove, L.N., 2004. *Reverse Logistics-Quantitative models for Closed-Loop Supply Chains*, Berlin, Springer.
- Dobos, I., 2003. Optimal production-inventory strategies for a HMMS-type reverse logistics system, *International Journal of Production Economics*, Vol. 81-82, pp. 351-360.
- Dowlatshahi, S., 2000. Developing a theory of reverse logistics, *Interfaces*, Vol. 30, No. 3, pp 143-155.
- Drucker, J., 2004. Old cell phones pile up by the millions, http://www.wirelessrecycling.com/home/press_releases/09242004.html.
- Ferguson, N., Browne, J., 2001. Issues in end-of-life product recovery and reverse logistics, *Production Planning and Control*, Vol. 12, No. 5, pp. 534-547.
- Ferrer, G., 1997. The economics of personal computer remanufacturing, *Resources, Conservation and Recycling*, Vol. 21, pp. 79-108.
- Ferrer, G., Ayres, R.U., 2000. The impact of remanufacturing in the economy, *Ecological Economics*, Vol. 32, pp. 413-429.
- Fleischmann, M., 2000. *Quantitative models for reverse logistics*, unpublished PhD Thesis, Erasmus University Rotterdam, Netherlands.

References

- Fleischmann, M., Beullens, P., Bloemhof-Ruwaard, J.M., Van Wassenhove, L.N., 2001. The impact of product recovery on logistics network design, *Production and Operations Management*, Vol. 10, No. 2, pp. 156-173.
- Fleischmann, M., Bloemhof-Ruwaard, J.M., Dekker, R., van der Laan, E., van Nunen, J.A.E.E., van Wassenhove, L.N., 1997. Quantitative models for reverse logistics: a review, *European Journal of Operational Research*, Vol. 103, pp. 1-17.
- Fleischmann, M., Krikke, H.R., Dekker, R., Flapper, S.D.P., 2000. A characterisation of logistics network for product recovery, *Omega*, Vol. 28, pp. 653-656.
- Fleischmann, M., Kuik, R., 2003. On optimal inventory control with independent stochastic item returns, *European Journal of Operational Research*, Vol. 151, pp. 25-37.
- Fleischmann, M., Kuik, R., Dekker, R., 2002. Controlling inventories with stochastic item returns: a basic model, *European Journal of Operational Research*, Vol. 138, pp. 63-75.
- Fleischmann, M., van Nunen, J. A.E.E., Grave, B., 2003. Integrating closed-loop supply chains and spare-parts management at IBM, *Interfaces*, Vol. 33, No. 6, pp. 44-56.
- Fonseca, C.M., Fleming, P.J., 1993. Multi-objective genetic algorithms, *Proceedings of IEE Colloquium on Genetic Algorithms for Control System Engineering*, pp. 6/1-6/5.
- Gen, M., Cheng, R., 1997. *Genetic Algorithms & Engineering Design*, New York: Wiley.
- Gen, M., Cheng, R., 2000. *Genetic Algorithms & Engineering Optimization*, New York: Wiley.
- Gilliland, M., 2004. Asia/Pacific PC market outlook 2004 (executive summary), *Gartner Dataquest*, 12 March 2004.
- Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.
- Gottinger, H.W., 1988. A computational model for solid waste management with application, *European Journal of Operational Research*, Vol. 35, pp. 350-364.
- Grenchus, E., Keene, R., Nobs, C., 1997. Demanufacturing of information technology equipment, *Proceedings of the International Symposium on Electronics and the Environment*, pp. 57-160.
- Grenchus, E., Johnson, S., McDonnell, D., 2001. Improving performance through reverse logistics at IBM, *Proceedings of the International Symposium on Electronics and the Environment*, pp. 236-240.
- Guide, V.D.R.Jr., 2000. Production planning and control for remanufacturing: industry practice and research needs, *Journal of Operations Management*, Volume 18, pp. 467-483.
- Guide, V.D.R.Jr, Harrison, T.P., Van Wassenhove, L.N., 2003. The challenge of closed-loop supply chain, *Interfaces*, Vol. 33, No. 6, pp. 3-6.
- Guide, V.D.R. Jr., Jayaraman, V., Linton, J.D., 2002. Building contingency planning for closed-loop supply chain with product recovery, *Journal of Operations Management*, Vol. 337, pp. 1-21.
- Guide, V.D.R.Jr, Jayaraman, V., Srivastava, R., Benton, W.C., 2000. Supply-chain management for recoverable manufacturing systems, *Interfaces*, Vol. 30, No. 3, pp. 125-142.
- Guide, V.D.R.Jr, Kraus, M., Srivastava, R., 1997. Scheduling policies for

References

- remanufacturing, *International Journal of Production Economics*, Vol. 48, pp. 187-204.
- Gungor, A., Gupta, S.M., 1998, Disassembly sequence planning for products with defective in product recovery, *Computers Industrial Engineering*, Vol. 35, No. 1-2, pp. 161-164.
- Gungor, A., Gupta, S.M., 1999. Issues in environmentally conscious manufacturing and product recovery: a survey, *Computers & Industrial Engineering*, Vol. 36, pp. 811-853.
- Gupta, S.M., Veerakamolmal, P., 2000. A bi-directional supply chain optimisation model for reverse logistics, *Proceedings of the IEEE International Symposium on Electronics and the Environment*, pp. 254-259.
- Hansen, U., 2000. Reverse logistics is the key for remanufacturing and a sustainable development, *Proceedings of the IEEE International Symposium on Electronics and the Environment*, pp. 238-242.
- Harris, M., 1998. Consumer demand for PCs bucks global trend, http://www.findarticles.com/p/articles/mi_m0WVI/is1998_Nov_2/ai_53171019.
- Hirsch, B.E., Kuhlmann, T., Schumacher, J., 1998. Logistics simulation of recycling networks, *Computers in Industry*, Vol. 36, pp. 31-38.
- Holland, J.H., 1975. *Adaptation in natural and artificial system*, Ann Arbor, The University of Michigan Press.
- Inderfurth, K., 2004. Optimal policies in hybrid manufacturing/remanufacturing systems with product substitution, *International Journal of Production Economics*, Vol. 90, pp. 325-343.
- Inderfurth, K., De Kok, A.G., Flapper, S.D.P., 2001. Product recovery in stochastic remanufacturing systems with multiple reuse options, *European Journal of Operational Research*, Vol. 133, pp.130-152.
- Istvan, Z., Garamvolgyi, E., 2000. Reverse logistics and management of end-of-life electric products, *Proceedings of the IEEE International Symposium on Electronics and the Environment*, pp.15-19.
- Jayaraman, V., Guide, V.D.R. Jr., Srivastava, R., 1999. A closed-loop logistics model for product recovery, *Journal of Operation Research Society*, Vol. 50, pp. 497-508.
- Jayaraman, V., Patterson, R.A., Rolland, E., 2003. The design of reverse distribution networks: models and solution procedures, *European Journal of Operational Research*, Vol. 150, pp. 128-149.
- Johnson, M.R., Wang, M.H., 1995. Planning product disassembly for material recovery opportunities, *International Journal of Production Research*, Vol. 33, No. 11, pp. 3119-3142.
- Kasmara, A., Muraki, M., Matsuoka, S., Sukoyo, Suryadi, K., 2001. Production planning in remanufacturing/manufacturing production system, *Proceedings of EcoDesign: Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing*, pp. 708-713.
- Kiesmuller, G.P., 2003a. A new approach for controlling a hybrid stochastic manufacturing/remanufacturing system with inventories and different lead times, *European Journal of Operational Research*, Vol. 147, pp. 62-71.
- Kiesmuller, G.P., 2003b. Optimal control of a one product recovery system with leadtimes, *International Journal of Production Economics*, Vol. 81-82, pp. 333-340.

References

- Kiesmuller, G.P., van der Laan, E.A., 2001. An inventory model with dependent product demands and returns, *ERIM Report Series: Research in Management*, March 2001.
- Knemeyer, A.M., Ponzurick, T.G., Logar, C., M., 2002. A qualitative examination of factors affecting reverse logistics systems for end-of-life computers, *International Journal of Physical Distribution and Logistics Management*, Vol. 32, No. 6, pp. 455-479.
- Kokkinaki, A.I., Dekker, R., Lee, R., Pappis, C., 2001. Integrating a web-based system with business processes in closed loop supply chains, *Econometric Institute EI 2001-21*.
- Kopicki, R., Berg, M.J., Legg, L., Dasappa, V., Maggioni, C., 1993. *Reuse and Recycling – Reverse Logistics Opportunities*, Council of Logistics Management, Oak Brook, IL.
- Kouvelis, P., Gang, Y., 1997. *Robust discrete optimisation and its applications*, Dordrecht: Kluwer.
- Krikke, H.R., 1998. *Recovery strategies and reverse logistics network design*, Unpublished PhD Thesis, University of Twente, Netherlands.
- Krikke, H., Bloemhof-Ruwaard, J., Van Wassenhove, L.N., 2001a. Design of closed loop supply chains: production and return network for refrigerators, *ERIM Report Series: Research in Management*.
- Krikke, H.R., Van Harten, A., Schuur, P.C., 1998a. On medium term product recovery and disposal strategy for durable assembly products, *International Journal of Production Research*, Vol. 36, No.1, pp. 111-139.
- Krikke, H.R., Van Harten, A., Schuur, P.C., 1998b. Mixed policies for recovery and disposal of multiple-type consumer products, *Journal of Environmental Engineering*, pp. 368-379.
- Krikke, H.R., Van Harten, A., Schuur, P.C., 1999a. Business case Roteb: recovery strategies for monitors, *Computer and industrial Engineering*, Vol. 36, 739-757.
- Krikke, H., Kokkinaki, A., Van Nunen, J., 2001b. Information technology in closed-loop supply chains, Unpublished working paper.
- Krikke, H., Pappis, C.P., Tsoulfas, G.T., Bloemhof-Ruwaard, J., 2001c. Design principles for closed loop supply chains: optimising economic, logistics and environmental performance, *ERIM Report Series: Research in Management*.
- Krikke, H.R., van Harten, A., Schuur, P.C., 1999b. Business case Oco: reverse logistics network re-design for copiers, *OR-Spektrum*, Vol. 21, No. 3, pp. 381-409.
- Kroon, L., Vrijens, G., 1995. Returnable containers: an example of reverse logistics, *International Journal of Physical Distribution and Logistics Management*, Vol. 25 No. 2, pp. 56-68.
- Krumwiede, D.W., Sheu, C., 2002. A model for reverse logistics entry by third-party providers, *Omega*, Vol. 30, pp. 325-333.
- Lee, B., 2004. China set to take lead in global notebook production, *Gartner Dataquest*, 22 April 2004.
- Li, D., Yang, J., Biswal, M.P., 1999. Quantitative parametric connections between methods for generating noninferior solutions in multiobjective optimization, *European Journal of Operational Research*, Vol. 117, pp. 84-89.
- Lindo Systems, 2004. *Lingo: User's Guide*, Illinois, Lindo Systems Inc.
- Listes, O., Dekker, R., 2005. Stochastic approaches for product recovery network design: a case study, *European Journal of Operational Research*, Vol. 160, pp.

References

- 268-287.
- Louwers, D., Kip, B.J., Peters, E., Souren, F., Flapper, S.D.P., 1999. A facility location-allocation model for re-using carpet materials, *Computers and Industrial Engineering*, Vol. 36, No. 4, pp. 1-15.
- Lund, R., 1984. Remanufacturing, *Technology Review*, Vol. 87, No. 2, pp. 19-29.
- Lund, R., 1998. Remanufacturing: An American resource, *Proceedings of the fifth international congress for environmentally conscious design and manufacturing*, 16-17 June 1998, Rochester, NY: Rochester Institute of Technology.
- Luo, Y., Zhou, M., Caudill, R.J., 2001. An integrated e-supply chain model for agile and environmentally conscious manufacturing, *IEEE/ASME Transactions on Mechatronics*, Vol. 6, No. 4, pp. 377-386.
- Maede, L., Sarkis, J., 2002. A conceptual model for selecting and evaluating third-party reverse logistics providers, *Supply Chain Management: An International Journal*, Vol. 7, No. 5, pp. 283-295.
- Mahadevan, B., Pyke, D.F., Fleischmann, M., 2003. Periodic review, push inventory policies for remanufacturing, *European Journal of Operational Research*, Vol. 151, pp. 536
- Marx-Gomez, J., Rautenstrauch, C., Nurnberger, A., Kruse, R., 2002. Neuro-fuzzy approach to forecast returns of scrapped products to recycling and remanufacturing, *Knowledge-Based Systems*, Vol. 15, pp. 119-128.-551
- Michalewicz, Z., 1995. Heuristic methods for evolutionary computation techniques, *Journal of Heuristics*, Vol.1, No.2, pp.177-206.
- Miettinen, K., 2001. Some methods for nonlinear multi-objective optimization, *Proceedings of First International Conference on Evolutionary Multi-Criterion Optimization*, Zurich, Switzerland.
- Min, H., Ko, H.J., Ko, C.S., 2004. A genetic algorithm approach to developing the multi-echelon reverse logistics network for product returns, *Omega*, Article in press.
- Minner, S., 2001. Strategic safety stocks in reverse logistics supply chain, *International Journal of Production Economics*, Vol. 71, pp. 417-428.
- Murata T., Ishibuchi, H., 1995. MOGA: Multi-objective genetic algorithm, *Proceedings of IEEE International Conference on Evolutionary Computation*, 1995, Vol. 1, pp. 289-294.
- Murphy, 1986. A preliminary study of transportation and warehousing aspects of reverse distribution, *Transportation Journal*, Vol. 34, No. 1, pp. 48-56.
- Nagel, C., Meyer, P., 1998. Caught between ecology and economy: end-of-life aspects of environmentally conscious manufacturing, *Computers & Industrial Engineering*, Vol. 36, pp. 781-792.
- Nasr, N., Hughson, C., Varel, E., Bauer, R., 1998. State-of-the-art assessment of remanufacturing technology-draft document, National Centre for Resource Recovery and Remanufacturing, Rochester Institute of Technology, Rochester, New York.
- Penev, K.D., De Ron, A.J., 1996. Determination of a disassembly strategy, *International Journal of Production Research*, Vol. 34, No. 2, pp. 495-506.
- Pohlen, T., Farris, M.T., 1992. Reverse logistics in plastics recycling, *International Journal of Physical Distribution & Logistics Management*, Vol. 22, No. 7, pp. 35-47.
- Porter, M.E., van der Linde, C., 1995. Green and competitive, *Harvard Business*

References

- Review*, Vol. 73 No. 5, pp. 120-134.
- Prüfer, H., 1918. Neuer Beweis eines Satzes über Permutationen. *Arch. Math. Phys.*, Vol. 27, pp. 742-744.
- Realff, M.J., Ammons, J.C., Newton, D., 1999. Carpet recycling: determining the reverse production system design, *Polymer-Plastics Technology & Engineering*, 38/3, pp. 547-567.
- Realff, M.J., Ammons, J.C., Newton, D., 2000. Strategic design of reverse production systems, *Computers and Chemical Engineering*, Vol. 24, pp. 991-996.
- Richter, K., Sombrutzki, M., 2000. Remanufacturing planning for the reverse Wagner/Whitin models, *European Journal of Operational Research*, Vol. 121, pp. 304-315.
- Rockwell Software, 2002. *Arena Standard: User's Guide*, Rockwell Software Inc.
- Rogers, D.S., Tibben-Lembke, R.S., 1998. *Going backwards: reverse logistics trend and practices*, Reverse Logistics Executive Council.
- Saaty, T. L., 1980. *The Analytical Hierarchy Process*, McGraw-Hill, New York.
- Salkin, H.M., Mathur, K., Hass, R., 1989. *Foundations of Integer Programming*, North-Holland, New York.
- Schultmann, F., Engels, B., Rentz, O., 2003. Closed-loop supply chains for spent batteries, *Interfaces*, Vol. 33, No. 6, pp.57-71.
- Shapiro, J.F., 2001. *Modelling the Supply Chain*, Duxbury, USA.
- Sheu, J.B., Chou, Y.H., Hu, C.C., 2004. An integrated logistics operational model for green supply chain management, *Transportation Research Part E*, Vol. 41, No. 4, pp. 287-313.
- Shih, L., 2001. Reverse logistics system planning for recycling electrical appliances and computers in Taiwan, *Resources, Conservation and Recycling*, Vol. 32, pp. 55-72.
- Sim, E., Jung, S., Kim, H., Park, J., 2004. A generic network design for a closed-loop supply chain using genetic algorithms, *GECCO 2004, LNCS 3103*, pp. 1214-1225.
- Spengler, T.H., Puechart, H., Penkuhn, T., Rentz, O., 1997. Environmental integrated production and recycling management, *European Journal of Operational Research*, Vol. 97, pp. 308-326.
- Spicer, A.J., Johnson, M.R., 2004. Third party demanufacturing as a solution for extended producer responsibility, *Journal of Cleaner Production*, Vol. 12, pp. 37-45.
- Steinhilper, R., 2001. Recent trends and benefits of remanufacturing: from closed loop business to synergetic networks, *Proceedings of EcoDesign: Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing*, pp.481-488.
- Stough, R., Benson, B., 2000. *Mechanism for promoting recycling of electronics products*, Oak Ridge, TN, Community Reuse Organisation of East Tennessee.
- Syarif, A., Yun, Y., Gen, M., 2002. Study on multi-stage logistics chain network: a spanning tree based genetic algorithm approach, *Computers & Industrial Engineering*, Vol. 43, pp. 299-314.
- Tan, K.C., Lee, T.H., Khor, E.F., 2001. Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons, *Proceedings of the IEEE Congress on Evolutionary Computation*, Vol. 2, pp. 979-986, May 2001.
- Tang, O., Grubbstrom, R.W., 2005. Considering stochastic lead times in a manufacturing/remanufacturing system with deterministic demands and returns,

References

- International Journal of Production Economics*, Vol. 93-94, pp. 285-300.
- Teunter, R.H., van der Laan, E., Inderfurth, K., 2000. How to set the holding cost rates in average cost inventory models with reverse logistics, *Omega*, Vol. 28, pp. 409-415.
- Teunter, R.H., 2001. Economic order quantities for recoverable item inventory systems, *Naval Research Logistics*, Vol. 48, pp. 484-495.
- Teunter, R.H., 2004. Lot-sizing for inventory systems with product recovery, *Computers & Industrial Engineering*, Vol. 46, pp. 431-441.
- Thierry, M.C., Salomon, M., van Nunen, J., Wassenhove, L., 1995. Strategic issues in product recovery management, *California Management Review*, Vol. 37, No. 2, pp. 114-135.
- Tibben-Lembke, R.S., 2002. Life after death: reverse logistics and the product life cycle, *International Journal of Physical Distribution & Logistics Management*, Vol. 32, No. 3, pp. 223-244.
- Tibben-Lembke, R.S., Rogers, D.S., 2002. Differences between forward and reverse logistics in a retail environment, *Supply Chain Management: An International Journal*, Vol. 7, No. 5, pp. 271-282.
- Toktay, L.B., Wein, L.M., Zenios, S.A., 2000. Inventory management of remanufacturable products, *Management Science*, Vol. 46, No. 11, pp. 1412-1426.
- Van der Laan, E., 1997. *The effects of remanufacturing on inventory control*, Unpublished PhD Thesis, Erasmus University Rotterdam, Netherlands.
- Van der Laan, E., Solomon, M., 1997. Production planning and inventory control with remanufacturing and disposal, *European Journal of Operational Research*, Vol. 102, pp. 264-278.
- Van der Laan, E., Salomon, M., Dekker, R., Van Wassenhove, L., 1999. Inventory control in hybrid systems with remanufacturing, *Management Science*, Vol. 45, No. 5, pp. 733- 747.
- Van Hoek, R.I., 1999. From reversed logistics to green supply chains, *Supply Chain Management*, Vol. 4, No. 3, pp. 129-134.
- Vandermerwe, S., Oliff, M.D., 1990. Customers drive corporation green, *Long Range Planning*, Vol. 23, No. 6, pp. 10-16.
- White, C.D., Masanet, E., Rosen, C.M., Beckman, S.L., 2003. Product recovery with some byte: an overview of management challenges and environmental consequences in reverse manufacturing for the computer industry, *Journal of Cleaner Production*, Vol. 11, No. 4, pp. 445-458.
- Wong, L., 2001. Barriers to implementation of electronic end of life product program in the U.S., unpublished working paper, <http://ist-socrates.berkeley.edu/~es196/projects/2001final/wong.pdf>.
- Wu, H., Dunn, S.C., 1995. Environmentally responsible logistics systems, *International Journal of Physical Distribution & Logistics Management*, Vol. 25, No.2, pp. 20-38.
- Yamada, K., Shiffler, G., Smulders, C., 2004. 2Q04 update: PC Forecast scenarios 2004-2005, Gartner Dataquest, 18 May 2004.
- Zhou, G., Gen, M., 1997. A note on genetic algorithms approach to the degree-constrained spanning tree problem, *Networks*, Vol. 30, pp. 105-109.

Appendix A

Data for Numerical Example of Chapter 3

(Single Objective, Multi-Period Problem)

1. Product information

Product structures:

Figure A.1: Product configuration of PC1, PC2, and PC5 (Gupta & Veerakamolmal, 2000)

Figure A.2: Product configuration of PC3, PC4, and PC6 (Gupta & Veerakamolmal, 2000)

Table A.1: Used and new modules of PC1-PC6 (Gupta & Veerakamolmal, 2000)

Module		Module Name	Unit of modules yielded/required					
			Used Product				Ref. Product	
	No.		PC1	PC2	PC3	PC4	PC5	PC6
U S E D	1.	Housing Assembly (PC1, PC2)	1	1	-	-	-	-
	2.	Housing Assembly (PC3, PC4)	-	-	1	1	-	-
	3.	Memory module, 16 MB, SDRAM	2	-	-	-	-	-
	4.	Memory module, 32 MB, SDRAM	2	4	2	-	2	2
	5.	Memory module, 64 MB, SDRAM	-	-	2	4	2	2
	6.	Pentium II 350 MHz CPU and Heat Sink	1	-	-	-	-	-
	7.	Pentium II 400 MHz CPU and Heat Sink	-	1	1	-	1	-
	8.	Pentium II 450 MHz CPU and Heat Sink	-	-	-	2	-	2
	9.	Mother Board (PC1, PC2, PC5)	1	1	-	-	1	-
	10.	Mother Board (PC3, PC4, PC6)	-	-	1	1	-	1
	11.	Display and Sound Cards (PC1 – PC4)	1	1	1	1	-	-
	12.	4 GB Hard Drive	1	-	-	-	-	-
	13.	9.1 GB Hard Drive	-	1	2	-	2	-
	14.	12.6 GB Hard Drive	-	-	-	2	-	2
	15.	1.44 MB Diskette Drive	1	1	1	1	1	1
	16.	32x CD ROM Drive (PC1 – PC4)	1	1	1	1	-	-
	17.	Power Supply (PC1 – PC4)	1	1	1	2	-	-
N E W	18.	Housing Assembly (PC5)	-	-	-	-	1	-
	19.	Housing Assembly (PC6)	-	-	-	-	-	1
	20.	Display and Sound Cards (PC5, PC6)	-	-	-	-	1	1
	21.	DVD-ROM Drive (PC5, PC6)	-	-	-	-	1	1
	22.	Power Supply (PC5,PC6)	-	-	-	-	1	1

Table A.2: Weight of used and refurbished products

No.	Used Products	Weight (kg)
1.	PC1	5
2.	PC2	5
3.	PC3	5.4
4.	PC4	6.7
5.	PC5	5.4
6.	PC6	6.2

Table A.3: Weight of used modules

No.	Modules	Weight (kg)
1.	Housing Assembly (PC1, PC2)	1
2.	Housing Assembly (PC3, PC4)	1
3.	Memory module, 16 MB, SDRAM	0.1
4.	Memory module, 32 MB, SDRAM	0.1
5.	Memory module, 64 MB, SDRAM	0.1
6.	Pentium II 350 MHz CPU and Heat Sink	0.8
7.	Pentium II 400 MHz CPU and Heat Sink	0.8
8.	Pentium II 450 MHz CPU and Heat Sink	0.8
9.	Mother Board (PC1, PC2, PC5)	0.8
10.	Mother Board (PC3, PC4, PC6)	0.8
11.	Display and Sound Cards (PC1 – PC4)	0.1
12.	4 GB Hard Drive	0.4
13.	9.1 GB Hard Drive	0.4
14.	12.6 GB Hard Drive	0.4
15.	1.44 MB Diskette Drive	0.5
16.	32x CD ROM Drive (PC1 – PC4)	0.5
17.	Power Supply (PC1 – PC4)	0.5

2. Supply and demand data

Table A.4: Modified supply and demand data

Supply		Unit
PC1		4,350
PC2		6,450
PC3		7,350
PC4		9,750
Demand	From locations	Unit
PC5	2 nd market – 1	2,000
	2 nd market – 2	2,000
	2 nd market – 3	1,700
PC6	2 nd market – 1	2,000
	2 nd market – 2	2,000
	2 nd market – 3	2,300

3. Distance data

Table A.5: Distance data

No.	From	To	Distance (Mile)
1	1 st Market	Retailer 1	20
		Retailer 2	10
		Retailer 3	10
		New CC	20
		Outsourced CC	20
		Colletr (DC)	175
		Colletr (ND)	175
2	Retailer 1	Distr. Ctr	80
		New Diss Ctr	150
		Outsourced Diss. Ctr.	110
		Diss. Ctr. (AP)	110
		Diss. Ctr. (NR)	110
3	Retailer 2	Distr. Ctr	150
		New Diss. Ctr.	150
		Outsourced Diss. Ctr.	135
		Diss. Ctr. (AP)	135
		Diss. Ctr. (NR)	135
4	Retailer 3	Distr. Ctr.	115
		New Diss. Ctr.	150
		Outsourced Diss. Ctr.	125
		Diss. Ctr. (AP)	125
		Diss. Ctr. (NR)	125
5	New CC	Distr. Ctr	150
		New Diss. Ctr.	90
		Outsourced Diss. Ctr.	115
		Diss. Ctr. (AP)	115
		Diss. Ctr. (NR)	115
6	Outsourced CC	Distr. Ctr.	125
		New Diss. Ctr.	135
		Outsourced Diss. Ctr.	125
		Diss. Ctr. (AP)	125
		Diss. Ctr. (NR)	125
7	Colletr (DC)	Distr. Ctr.	0
		New Diss. Ctr.	
		Outsourced Diss. Ctr.	
		Diss. Ctr. (AP)	
8	Colletr (ND)	Distr. Ctr.	
		New Diss. Ctr.	0
		Outsourced Diss. Ctr.	
		Diss. Ctr. (AP)	
9	Distr. Ctr.	Assembly Plant	1000
		New Recov. Ctr	1000
		Outsourced RC	1000
10	New Diss. Ctr.	Assembly Plant	1000
		New Recov. Ctr	850
		Outsourced RC	925
11	Outsourced Diss. Ctr.	Assembly Plant	950
		New Recov. Ctr	900
		Outsourced RC	925
12	Diss. Ctr. (AP)	Assembly Plant	0
		New Recov. Ctr	
		Outsourced RC	

13	Diss. Ctr (NR)	Assembly Plant	
		New Recov. Ctr	0
		Outsourced RC	
14	Assembly Plant	2 nd Market 1	500
		2 nd Market 2	600
		2 nd Market 3	600
15	New Recov. Ctr	2 nd Market 1	500
		2 nd Market 2	500
		2 nd Market 3	550
16	Outsource d RC	2 nd Market 1	500
		2 nd Market 2	575
		2 nd Market 3	575

All potential locations are assumed to be located in the same country.

4. Other data

4.1. Transportation costs

Transportation cost is taken from Luo et al. (2001). Modes of transportation depend on the distance and whether the locations reside in the same country as follows:

1. Both locations are in the same country :
 - Distance < 20 miles, transportation modes: car or truck.
 - 20 miles $\leq$ Distance < 500 miles, transportation mode: truck.
 - Distance $\geq$ 500 miles, transportation modes: truck or air.
2. Both locations are not in the same country, transportation mode: air.

The cost between facilities is calculated under the assumption that the capacity of transportation by car, truck and airplane are 5, 35 and 35 units respectively with average weight per unit is 5.6 kg. Hence, transportation costs are calculated using equation (A.1) and data from Tables A.5 and A.6.

Table A.6: Transportation cost
(Luo et al., 2001)

Transportation Mode	Cost (\$/ton-mile)
Car	0.25
Truck	0.26
Airplane	0.80

$$\text{Transportation cost/kg} = \frac{(\text{unit cost} \times \text{distance})}{\text{Capacity} \times \text{weight/unit}} \quad (\text{A.1})$$

4.2. Fixed costs

Table A.7: Fixed costs to open new facilities or operate existing/outsourced facilities

Facilities	Annual cost (\$)
CC1 (Retailer 1)	10,000
CC2 (Retailer 2)	10,000
CC3 (Retailer 3)	10,000
CC4 (New CC)	30,000
CC5 (Outsourced Collection Centre)	5,000
CC6 (Distribution Centre)	5,000
CC7 (New Disassembly Centre)	5,000
DC1 (Distribution Centre)	100,000
DC2 (New Disassembly Centre)	300,000
DC3 (Outsourced Disassembly Centre)	50,000
DC4 (Assembly Plant)	50,000
DC5 (New Recovery Centre)	50,000
RC1 (Assembly Plant)	250,000
RC2 (New Recovery Centre)	750,000
RC3 (Outsourced Recovery Centre)	125,000

Table A.8: Fixed costs to operate new facilities

Facilities	Annual cost (\$)
CC4 (New Collection Centre)	11,000
DC2 (New Disassembly Centre)	110,000
RC2 (New Recovery Centre)	275,000

4.3. Holding costs

Holding costs are assumed to be the same for all product types.

Table A.9: Holding costs

Facilities	\$/unit
CC1	3.9
CC2	3.9
CC3	3.9
CC4	3.1
CC5	5.8
CC6	0.0
CC7	0.0
DC1	3.1
DC2	2.5
DC3	4.7
DC4	3.1
DC5	2.5

4.4. Processing costs

Table A.10: Processing costs

Facilities	\$ /unit					
	PC1	PC2	PC3	PC4	PC5	PC6
DC1	31.2	31.2	25	25		
DC2	25	25	20	20		
DC3	46.9	46.9	37.5	37.5		
DC4	31.2	31.2	25	25		
DC5	25	25	20	20		
RC1					31.2	25
RC2					25	20
RC3					46.9	37.5

4.5. Product prices

Price of recyclable modules is assumed to be the same for all module types.

Table A.11: Prices of refurbished products, recovered, new and recyclable modules

No.	Product	Price (\$/unit)
1.	PC5	400
2.	PC6	500
3.	Memory module, 32 MB, SDRAM	15
4.	Memory module, 64 MB, SDRAM	20
5.	Pentium II 400 MHz CPU and Heat Sink	25
6.	Pentium II 450 MHz CPU and Heat Sink	10
7.	Mother Board (PC1, PC2, PC5)	10
8.	Mother Board (PC3, PC4, PC6)	15
9.	9.1 GB Hard Drive	20
10.	12.6 GB Hard Drive	20
11.	1.44 MB Diskette Drive	10
12.	Housing Assembly (PC5)	25
13.	Housing Assembly (PC6)	30
14.	Display and Sound Cards (PC5, PC6)	75
15.	DVD-ROM Drive (PC5, PC6)	50
16.	Power Supply (PC5, PC6)	45
17.	Recyclable used modules per kg	2

4.6. Facility capacities

Table A.12: Capacity of all facilities

Facilities	\$/unit					
	PC1	PC2	PC3	PC4	PC5	PC6
CC1	1,500	2,500	2,500	3,500		
CC2	1,500	2,500	2,500	3,500		
CC3	1,500	2,500	2,500	3,500		
CC4	4,500	7,500	7,500	10,500		
CC5	9,000	15,000	15,000	21,000		
CC6	4,500	6,500	7,500	10,000		
CC7	13,500	19,500	22,500	30,000		
DC1	4,500	6,500	7,500	10,000		
DC2	13,500	19,500	22,500	30,000		
DC3	27,000	39,000	45,000	60,000		
DC4	4,500	6,500	7,500	10,000		
DC5	13,500	19,500	22,500	30,000		
RC1					6,000	7,000
RC2					18,000	21,000
RC3					36,000	42,000

4.7. Recyclability and recoverability fractions

Recyclability and recoverability fractions are assumed to be the same for all modules, which are 0.85 and 0.65 respectively.

5. Multi-period data

5.1. Supply and demand data

Supply and demand data for the first period is presented in Table A.4. Supply for the second and third period is increased by 10% and it is decreased by 10 % for the fourth and fifth period. Demand is decreased by 5% at the subsequent periods.

5.2. Costs

Costs are increased yearly by 10%.

5.3. Product prices

Prices of refurbished products and recovered modules are decreased by 10% yearly, whereas prices of new and recyclable modules are the same throughout all periods.

Price of recyclable modules is also assumed to be the same for all modules.

5.4. Capacity

Capacities at the forward chain facilities (CC1 to CC3, DC1 and RC1) are decreased by 5% yearly, while capacities at other facilities are constant across periods.

5.6. Recyclability and recoverability fractions

Recyclability and recoverability fractions are assumed to be the same for all modules and periods, which are 0.85 and 0.65 respectively.

Appendix B
Result for Numerical Example of Chapter 3
(Single Period, Maximising Net Revenue)

Net Revenue	\$ 2,057,681		
BINARY VARIABLES			
YC	j	t	
	1	0	
	2	0	
	3	0	
	4	1	
	5	0	
	6	0	
YD	k	t	
	1	0	
	2	0	
	3	0	
	4	1	
	5	0	
YR	l	t	
	1	1	
	2	0	
	3	0	

$Q^A_{r j}$	qjl	t	
	111	0	
	112	0	
	113	0	
	114	4350	
	115	0	
	116	0	
	117	0	
	211	0	
	212	0	
	213	0	
	214	6450	
	215	0	
	216	0	
	217	0	
	311	0	
	312	0	
$Q^B_{r jko}$	qjko	t	
	1122	0	
	1132	0	
	1142	0	
	1152	0	
	1212	0	
	1222	0	
	1232	0	
	1242	0	
	1252	0	
	1312	0	
	1322	0	
	1332	0	
	1342	0	
	1352	0	
	1412	0	
	1422	0	
$Q^C_{r klot}$	zklo	t	
	1112	0	
	1113	0	
	1122	0	
	1123	0	
	1132	0	
	1133	0	
	1212	0	
	1213	0	
	1222	0	
	1223	0	
	1232	0	
	1233	0	
	1312	0	
	1313	0	
	1322	0	
	1323	0	
$Q^E_{r lmot}$	qlmo	t	
	1112	2000	
	1113	0	
	1122	2000	
	1123	0	
	1132	1700	
	1133	0	
	1212	0	
	1213	0	
	1222	0	
	1223	0	
	1232	0	
	1233	0	
	1312	0	
	1313	0	
	1322	0	
	1323	0	
$Q^H_{r kl}$	zkl	t	
	1112	0	
	1113	0	
	1122	0	
	1123	0	
	1132	0	
	1133	0	
	1212	0	
	1213	0	
	1222	0	
	1223	0	
	1232	0	
	1233	0	
	1312	0	
	1313	0	
	1322	0	
	1323	0	
	1332	0	
	1333	0	
	1412	24000	
	1413	0	
	1422	0	
	1423	0	
	1432	0	
	1433	0	
	1512	0	
	1513	0	
	1522	0	
	1523	0	
	1532	0	
	1533	0	
	1612	0	
	1622	0	
	1632	0	
	1642	0	
	1652	0	
	1712	0	
	1722	0	
	1732	0	
	1742	0	
	1752	0	
	2112	0	
	2122	0	
	2132	0	
	2142	0	
	2152	0	
	2212	0	
	2222	0	
	2232	0	

z	k			
	t	t	t	t
1	0	0	0	7980
2	0	0	0	10905
3	0	0	0	3270
4	0	0	0	75
5	0	0	0	1320
6	0	0	0	4815
7	0	0	0	2347
8	0	0	0	75
9	0	0	0	6135

z	k			
	t	t	t	t
1	0	0	0	49200
2	0	0	0	53700
3	0	0	0	13800
4	0	0	0	19500
5	0	0	0	10800
6	0	0	0	17100
7	0	0	0	21150
8	0	0	0	19500
9	0	0	0	27900
10	0	0	0	10800
11	0	0	0	17100
12	0	0	0	8700
13	0	0	0	4350
14	0	0	0	27900
15	0	0	0	4350
16	0	0	0	27900
17	0	0	0	37650

z	t		t	
	t	t	t	t
1	5700	0	0	0
2	6300	0	0	0
3	12000	0	0	0
4	12000	0	0	0
5	12000	0	0	0

2233	0
2312	0
2313	0
2322	0
2323	0
2332	0
2333	0
2412	24000
2413	0
2422	0
2423	0
2432	0
2433	0
2512	0
2513	0
2522	0
2523	0
2532	0
2533	0
3112	0
3113	0
3122	0
3123	0
3132	0
3133	0
3212	0
3213	0
3222	0
3223	0
3232	0
3233	0
3312	0
3313	0
3322	0
3323	0
3332	0
3333	0
3412	5700
3413	0
3422	0
3423	0
3432	0
3433	0
3512	0
3513	0
3522	0
3523	0
3532	0
3533	0
4112	0
4113	0
4122	0
4123	0
4132	0

2222	0
2232	0
2242	0
2322	0
2312	0
2322	0
2332	0
2342	0
2352	0
2412	0
2422	0
2432	0
2442	6450
2452	0
2512	0
2522	0
2532	0
2542	0
2552	0
2612	0
2622	0
2632	0
2642	0
2652	0
2712	0
2722	0
2732	0
2742	0
2752	0
3112	0
3122	0
3132	0
3142	0
3152	0
3212	0
3222	0
3232	0
3242	0
3252	0
3312	0
3322	0
3332	0
3342	0
3352	0
3412	0
3422	0
3432	0
3442	7350
3452	0
3512	0
3522	0
3532	0
3542	0
3552	0

3612	0	4133	0
3622	0	4212	0
3632	0	4213	0
3642	0	4222	0
3652	0	4223	0
3712	0	4232	0
3722	0	4233	0
3732	0	4312	0
3742	0	4313	0
3752	0	4322	0
4112	0	4323	0
4122	0	4332	0
4132	0	4333	0
4142	0	4412	12600
4152	0	4413	0
4212	0	4422	0
4222	0	4423	0
4232	0	4432	0
4242	0	4433	0
4252	0	4512	0
4312	0	4513	0
4322	0	4522	0
4332	0	4523	0
4342	0	4532	0
4352	0	4533	0
4412	0	5112	0
4422	0	5113	0
4432	0	5122	0
4442	9750	5123	0
4452	0	5132	0
4512	0	5133	0
4522	0	5212	0
4532	0	5213	0
4542	0	5222	0
4552	0	5223	0
4612	0	5232	0
4622	0	5233	0
4632	0	5312	0
4642	0	5313	0
4652	0	5322	0
4712	0	5323	0
4722	0	5332	0
4732	0	5333	0
4742	0	5412	5700
4752	0	5413	0
		5422	0
		5423	0
		5432	0
		5433	0
		5512	0
		5513	0
		5522	0
		5523	0
		5532	0

5533	0
6112	0
6113	0
6122	0
6123	0
6132	0
6133	0
6212	0
6213	0
6222	0
6223	0
6232	0
6233	0
6312	0
6313	0
6322	0
6323	0
6332	0
6333	0
6412	6300
6413	0
6422	0
6423	0
6432	0
6433	0
6512	0
6513	0
6522	0
6523	0
6532	0
6533	0
7112	0
7113	0
7122	0
7123	0
7132	0
7133	0
7212	0
7213	0
7222	0
7223	0
7232	0
7233	0
7312	0
7313	0
7322	0
7323	0
7332	0
7333	0
7412	11400
7413	0
7422	0
7423	0
7432	0

7433	0
7512	0
7513	0
7522	0
7523	0
7532	0
7533	0
8112	0
8113	0
8122	0
8123	0
8132	0
8133	0
8212	0
8213	0
8222	0
8223	0
8232	0
8233	0
8312	0
8313	0
8322	0
8323	0
8332	0
8333	0
8412	12600
8413	0
8422	0
8423	0
8432	0
8433	0
8512	0
8513	0
8522	0
8523	0
8532	0
8533	0
9112	0
9113	0
9122	0
9123	0
9132	0
9133	0
9212	0
9213	0
9222	0
9223	0
9232	0
9233	0
9312	0
9313	0
9322	0
9323	0
9332	0

9333	0
9412	12000
9413	0
9422	0
9423	0
9432	0
9433	0
9512	0
9513	0
9522	0
9523	0
9532	0
9533	0

Appendix C

Result for Numerical Example of Chapter 3 (Multi-period, Maximising Net Revenue)

Net Rev. \$ 4,591,169

Net Rev. \$

[illegible]

Q ⁸ post	Q ⁸ analog				Q ⁸ post	Q ⁸ analog			
	gkko	t	gkko	t		gkko	t	gkko	t
1112	0	0	1112	0	1112	0	0	1112	0
1122	0	0	1122	0	1113	0	0	1113	0
1132	0	0	1132	0	1122	0	0	1122	0
1142	0	0	1142	0	1123	0	0	1123	0
1152	0	764	1152	0	1132	0	0	1132	0
1162	0	0	1162	0	1133	0	0	1133	0
1172	0	0	1172	0	1142	0	0	1142	0
1182	0	0	1182	0	1213	0	0	1213	0
1192	0	0	1192	0	1222	0	0	1222	0
1242	0	0	1242	0	1223	0	0	1223	0
1252	0	0	1252	0	1232	0	0	1232	0
1312	0	0	1312	0	1233	0	0	1233	0
1322	0	0	1322	0	1312	0	0	1312	0
1332	0	0	1332	0	1313	0	0	1313	0
1342	0	0	1342	0	1322	0	0	1322	0
1412	0	0	1412	0	1323	0	0	1323	0
1422	0	0	1422	0	1332	0	0	1332	0
1432	0	0	1432	0	1333	0	0	1333	0
1442	0	0	1442	0	1413	0	0	1413	0
1452	4350	4500	1452	4350	1413	0	0	1413	0
1512	0	0	1512	0	1422	0	0	1422	0
1522	0	0	1522	0	1423	0	0	1423	0
1532	0	0	1532	0	1432	0	0	1432	0
1542	0	0	1542	0	1433	0	0	1433	0
1552	0	285	1552	0	1512	0	0	1512	0
1612	0	0	1612	0	1513	0	0	1513	0
1622	0	0	1622	0	1522	24000	22600	1522	24000
1642	0	0	1642	0	1523	0	0	1523	21660
1652	0	0	1652	0	1532	0	0	1532	20580
1662	0	0	1662	0	1533	0	0	1533	0
1712	0	0	1712	0	2112	0	0	2112	0
1722	0	0	1722	0	2122	0	0	2122	0
1732	0	0	1732	0	2123	0	0	2123	0
1742	0	0	1742	0	2132	0	0	2132	0
1752	0	0	1752	0	2133	0	0	2133	0
2112	0	0	2112	0	2212	0	0	2212	0
2122	0	0	2122	0	2213	0	0	2213	0
2132	0	0	2132	0	2222	0	0	2222	0
2142	0	0	2142	0	2223	0	0	2223	0
2152	0	305	2152	0	2232	0	0	2232	0
2212	0	0	2212	0	2233	0	0	2233	0
2222	0	0	2222	0	2312	0	0	2312	0
2232	0	0	2232	0	2313	0	0	2313	0
2242	0	0	2242	0	2322	0	0	2322	0
2252	0	0	2252	0	2332	0	0	2332	0
2312	0	0	2312	0	2333	0	0	2333	0
2322	0	0	2322	0	2412	0	0	2412	0
2332	0	0	2332	0	2413	0	0	2413	0
2342	0	0	2342	0	2422	0	0	2422	0
2352	0	0	2352	0	2423	0	0	2423	0
2412	0	0	2412	0	2432	0	0	2432	0
2422	0	0	2422	0	2433	0	0	2433	0
2432	0	0	2432	0	2512	0	0	2512	0
2442	0	0	2442	0	2513	0	0	2513	0
2452	6450	7095	2452	6450	2522	24000	22600	2522	24000
2462	0	0	2462	0	2523	0	0	2523	20580
2512	0	0	2512	0	2532	0	0	2532	0
2522	0	0	2522	0	2533	0	0	2533	0
2532	0	0	2532	0	3112	0	0	3112	0
2542	0	0	2542	0	3113	0	0	3113	0
2552	0	0	2552	0	3122	0	0	3122	0
2612	0	0	2612	0					
2622	0	0	2622	0					
2632	0	0	2632	0					

[illegible]

4652	0	225	0	225	0
4612	0	0	0	0	0
4622	0	0	0	0	0
4632	0	0	0	0	0
4642	0	0	0	0	0
4652	0	0	0	0	0
4662	0	0	0	0	0
4672	0	0	0	0	0
4682	0	0	0	0	0
4692	0	0	0	0	0
4702	0	0	0	0	0
4712	0	0	0	0	0
4722	0	0	0	0	0
4732	0	0	0	0	0
4742	0	0	0	0	0
4752	0	0	0	0	0

5223	0	0	0	0	0
5232	0	0	0	0	0
5233	0	0	0	0	0
5312	0	0	0	0	0
5313	0	0	0	0	0
5322	0	0	0	0	0
5323	0	0	0	0	0
5332	0	0	0	0	0
5333	0	0	0	0	0
5412	0	0	0	0	0
5413	0	0	0	0	0
5422	0	0	0	0	0
5423	0	0	0	0	0
5432	0	0	0	0	0
5433	0	0	0	0	0
5512	0	0	0	0	0
5513	0	0	0	0	0
5522	5700	5415	5144	4683	4683
5523	0	0	0	0	0
5532	0	0	0	0	0
5533	0	0	0	0	0
6112	0	0	0	0	0
6113	0	0	0	0	0
6122	0	0	0	0	0
6123	0	0	0	0	0
6132	0	0	0	0	0
6133	0	0	0	0	0
6212	0	0	0	0	0
6213	0	0	0	0	0
6222	0	0	0	0	0
6223	0	0	0	0	0
6232	0	0	0	0	0
6233	0	0	0	0	0
6312	0	0	0	0	0
6313	0	0	0	0	0
6322	0	0	0	0	0
6323	0	0	0	0	0
6332	0	0	0	0	0
6333	0	0	0	0	0
6412	0	0	0	0	0
6413	0	0	0	0	0
6422	0	0	0	0	0
6423	0	0	0	0	0
6432	0	0	0	0	0
6433	0	0	0	0	0
6512	0	0	0	0	0
6513	0	0	0	0	0
6522	6300	5685	5686	5402	5131
6523	0	0	0	0	0
6532	0	0	0	0	0
6533	0	0	0	0	0
7112	0	0	0	0	0
7113	0	0	0	0	0
7122	0	0	0	0	0
7123	0	0	0	0	0
7132	0	0	0	0	0
7133	0	0	0	0	0
7212	0	0	0	0	0
7213	0	0	0	0	0
7222	0	0	0	0	0
7223	0	0	0	0	0
7232	0	0	0	0	0
7233	0	0	0	0	0
7312	0	0	0	0	0
7313	0	0	0	0	0
7322	0	0	0	0	0

[illegible]

9423	0	0	0	0	0	0	0
9432	0	0	0	0	0	0	0
9433	0	0	0	0	0	0	0
9512	0	0	0	0	0	0	0
9513	0	0	0	0	0	0	0
9522	1200	0	1080	1020	0	974	0
9523	0	1440	0	0	0	0	0
9532	0	0	0	0	0	0	0
9533	0	0	0	0	0	0	0

Q^H_{sk}															
z				t				k				t			
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q^H_{sk}															
z				t				k				t			
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Appendix D

Additional Data for Numerical Example of Chapter 3 and 4

(Multi-objective, Multi-period Problem)

1. Energy consumption and CO₂ during transportation

Energy consumption and CO₂ emission during transportation are taken from Luo et al. (2001).

Table D.1: Energy, and CO₂ emission for transportation
(Luo et al., 2001)

Transportation Mode	Energy Intensity (GJ/ton-mile)	CO ₂ Emission (KgCE/ton-mile)
Car	0.054	0.035
Truck	0.003	0.23
Airplane	0.021	8.4

Energy consumption and CO₂ emission during transportation are calculated using Equation (A.1) and data from Table D.1.

2. Energy consumption during processes

The following data is assumed based on Luo et al. (2001). It is assumed that energy consumption during process depends on product types and locations. More complex product requires more energy and it is assumed that energy consumptions in new facilities are lower than those existing or outsourced locations as new equipment more likely to be more energy-saving.

Table D.2: Energy consumption during processes (MJ)

Product Type	Locations							
	AP	New RC	Outsourcing RC	Distr. Ctr	New Diss. Ctr	Outsourcing DC	AP	NR
PC1				3,540	2,832	3,540	3,540	2,832
PC2				3,540	2,832	3,540	3,540	2,832
PC3				2,832	2,265.6	2,832	2,832	2,265.6
PC4				2,832	2,265.6	2,832	2,832	2,265.6
PC5	2950	2360	2950					
PC6	2360	1888	2360					

Appendix E

Result for Numerical Example of Chapter 3 (Multi Period, Minimising Environmental Impacts)

[illegible]

q	q ²	q ³	t	q ⁴	q ⁵	t
1112	0	0	0	0	0	0
1113	0	0	0	0	0	0
1122	0	0	0	0	0	0
1132	0	0	0	0	0	0
1133	0	0	0	0	0	0
1150	1236	764	425	682	0	0
1212	0	0	0	0	0	0
1222	0	0	0	0	0	0
1232	0	0	0	0	0	0
1233	0	0	0	0	0	0
1322	0	0	0	0	0	0
1342	0	0	0	0	0	0
1343	0	0	0	0	0	0
1412	0	0	0	0	0	0
1422	2031	3548	4500	4360	3857	0
1432	0	0	0	0	0	0
1433	0	0	0	0	0	0
1452	0	0	0	0	0	0
1453	819	0	0	0	0	0
1512	0	0	0	0	0	0
1522	0	0	0	0	0	0
1523	0	0	0	0	0	0
1542	0	0	0	0	0	0
1543	1542	0	0	0	0	0
1552	0	0	0	0	0	0
1612	0	0	0	0	0	0
1622	0	0	0	0	0	0
1632	0	0	0	0	0	0
1642	0	0	0	0	0	0
1652	0	0	0	0	0	0
1712	0	0	0	0	0	0
1722	0	0	0	0	0	0
1732	0	0	0	0	0	0
1742	0	0	0	0	0	0
1752	0	0	0	0	0	0
2112	0	0	0	0	0	0
2122	0	0	0	0	0	0
2132	0	0	0	0	0	0
2142	0	0	0	0	0	0
2152	2500	2375	2236	2143	2036	0
2212	0	0	0	0	0	0
2222	0	0	0	0	0	0
2232	0	0	0	0	0	0
2233	0	0	0	0	0	0
2332	0	0	0	0	0	0
2342	0	0	0	0	0	0
2352	0	0	0	0	0	0
2372	0	0	0	0	0	0
2382	0	0	0	0	0	0
2422	0	0	0	0	0	0
2432	0	0	0	0	0	0
2442	0	0	0	0	0	0
2452	3360	4720	4634	4562	4414	0
2512	0	0	0	0	0	0
2522	0	0	0	0	0	0
2532	0	0	0	0	0	0
2542	0	0	0	0	0	0
2552	0	0	0	0	0	0
2612	0	0	0	0	0	0
2622	0	0	0	0	0	0
2632	0	0	0	0	0	0
2645	0	0	0	0	0	0

[illegible]

4612	0	0	0	0	0	0	0	0	0
4622	0	0	0	0	0	0	0	0	0
4632	0	0	0	0	0	0	0	0	0
4642	0	0	0	0	0	0	0	0	0
4652	0	0	0	0	0	0	0	0	0
4712	0	0	0	0	0	0	0	0	0
4722	0	0	0	0	0	0	0	0	0
4732	0	0	0	0	0	0	0	0	0
4742	0	0	0	0	0	0	0	0	0
4752	0	0	0	0	0	0	0	0	0

5222	0	0	0	0	0	0	0	0	0
5233	0	0	0	0	0	0	0	0	0
5312	0	0	0	0	0	0	0	0	0
5313	0	0	0	0	0	0	0	0	0
5322	0	0	0	0	0	0	0	0	0
5323	0	0	0	0	0	0	0	0	0
5332	0	0	0	0	0	0	0	0	0
5333	0	0	0	0	0	0	0	0	0
5412	0	0	0	0	0	0	0	0	0
5413	0	0	0	0	0	0	0	0	0
5422	0	0	0	0	0	0	0	0	0
5423	0	0	0	0	0	0	0	0	0
5432	0	0	0	0	0	0	0	0	0
5433	0	0	0	0	0	0	0	0	0
5512	0	0	0	0	0	0	0	0	0
5513	0	0	0	0	0	0	0	0	0
5522	5700	5415	5144	4888	4643	4400	4155	3910	3665
5523	0	0	0	0	0	0	0	0	0
5532	0	0	0	0	0	0	0	0	0
5533	0	0	0	0	0	0	0	0	0
6112	0	0	0	0	0	0	0	0	0
6113	0	0	0	0	0	0	0	0	0
6122	0	0	0	0	0	0	0	0	0
6123	0	0	0	0	0	0	0	0	0
6132	0	0	0	0	0	0	0	0	0
6133	0	0	0	0	0	0	0	0	0
6212	0	0	0	0	0	0	0	0	0
6213	0	0	0	0	0	0	0	0	0
6222	0	0	0	0	0	0	0	0	0
6223	0	0	0	0	0	0	0	0	0
6232	0	0	0	0	0	0	0	0	0
6233	0	0	0	0	0	0	0	0	0
6312	0	0	0	0	0	0	0	0	0
6313	0	0	0	0	0	0	0	0	0
6322	0	0	0	0	0	0	0	0	0
6323	0	0	0	0	0	0	0	0	0
6332	0	0	0	0	0	0	0	0	0
6333	0	0	0	0	0	0	0	0	0
6412	0	0	0	0	0	0	0	0	0
6413	0	0	0	0	0	0	0	0	0
6422	0	0	0	0	0	0	0	0	0
6423	0	0	0	0	0	0	0	0	0
6432	0	0	0	0	0	0	0	0	0
6433	0	0	0	0	0	0	0	0	0
6512	0	0	0	0	0	0	0	0	0
6513	0	0	0	0	0	0	0	0	0
6522	6300	5985	5666	5402	5131	4860	4590	4320	4050
6523	0	0	0	0	0	0	0	0	0
6532	0	0	0	0	0	0	0	0	0
6533	0	0	0	0	0	0	0	0	0
7112	0	0	0	0	0	0	0	0	0
7113	0	0	0	0	0	0	0	0	0
7122	0	0	0	0	0	0	0	0	0
7123	0	0	0	0	0	0	0	0	0
7132	0	0	0	0	0	0	0	0	0
7133	0	0	0	0	0	0	0	0	0
7212	0	0	0	0	0	0	0	0	0
7213	0	0	0	0	0	0	0	0	0
7222	0	0	0	0	0	0	0	0	0
7223	0	0	0	0	0	0	0	0	0
7232	0	0	0	0	0	0	0	0	0
7233	0	0	0	0	0	0	0	0	0
7312	0	0	0	0	0	0	0	0	0
7313	0	0	0	0	0	0	0	0	0
7322	0	0	0	0	0	0	0	0	0
7323	0	0	0	0	0	0	0	0	0

7332	0	0	0	0	0	0	0	0	0
7333	0	0	0	0	0	0	0	0	0
7412	0	0	0	0	0	0	0	0	0
7413	0	0	0	0	0	0	0	0	0
7422	0	0	0	0	0	0	0	0	0
7423	0	0	0	0	0	0	0	0	0
7442	0	0	0	0	0	0	0	0	0
7453	0	0	0	0	0	0	0	0	0
7512	0	0	0	0	0	0	0	0	0
7513	0	0	0	0	0	0	0	0	0
7522	11400	10830	10288	9776	9286				
7523	0	0	0	0	0	0	0	0	0
7532	0	0	0	0	0	0	0	0	0
7533	0	0	0	0	0	0	0	0	0
8112	0	0	0	0	0	0	0	0	0
8113	0	0	0	0	0	0	0	0	0
8122	0	0	0	0	0	0	0	0	0
8123	0	0	0	0	0	0	0	0	0
8142	0	0	0	0	0	0	0	0	0
8143	0	0	0	0	0	0	0	0	0
8212	0	0	0	0	0	0	0	0	0
8213	0	0	0	0	0	0	0	0	0
8222	0	0	0	0	0	0	0	0	0
8223	0	0	0	0	0	0	0	0	0
8232	0	0	0	0	0	0	0	0	0
8233	0	0	0	0	0	0	0	0	0
8312	0	0	0	0	0	0	0	0	0
8313	0	0	0	0	0	0	0	0	0
8322	0	0	0	0	0	0	0	0	0
8323	0	0	0	0	0	0	0	0	0
8332	0	0	0	0	0	0	0	0	0
8333	0	0	0	0	0	0	0	0	0
8412	0	0	0	0	0	0	0	0	0
8413	0	0	0	0	0	0	0	0	0
8422	0	0	0	0	0	0	0	0	0
8423	0	0	0	0	0	0	0	0	0
8432	0	0	0	0	0	0	0	0	0
8433	0	0	0	0	0	0	0	0	0
8512	0	0	0	0	0	0	0	0	0
8513	0	0	0	0	0	0	0	0	0
8522	12600	11970	11372	10804	10262				
8523	0	0	0	0	0	0	0	0	0
8532	0	0	0	0	0	0	0	0	0
8533	0	0	0	0	0	0	0	0	0
9112	0	0	0	0	0	0	0	0	0
9113	0	0	0	0	0	0	0	0	0
9122	0	0	0	0	0	0	0	0	0
9123	0	0	0	0	0	0	0	0	0
9132	0	0	0	0	0	0	0	0	0
9133	0	0	0	0	0	0	0	0	0
9212	0	0	0	0	0	0	0	0	0
9213	0	0	0	0	0	0	0	0	0
9222	0	0	0	0	0	0	0	0	0
9223	0	0	0	0	0	0	0	0	0
9232	0	0	0	0	0	0	0	0	0
9233	0	0	0	0	0	0	0	0	0
9312	0	0	0	0	0	0	0	0	0
9313	0	0	0	0	0	0	0	0	0
9322	0	0	0	0	0	0	0	0	0
9323	0	0	0	0	0	0	0	0	0
9332	0	0	0	0	0	0	0	0	0
9333	0	0	0	0	0	0	0	0	0
9412	0	0	0	0	0	0	0	0	0
9413	0	0	0	0	0	0	0	0	0
9422	0	0	0	0	0	0	0	0	0
9423	0	0	0	0	0	0	0	0	0

9432	0	0	0	0	0	0	0	0	0
9433	0	0	0	0	0	0	0	0	0
9512	0	0	0	0	0	0	0	0	0
9513	0	0	0	0	0	0	0	0	0
9522	1200	11400	10830	10290	9774	9258	8742	8226	7710
9523	0	0	0	0	0	0	0	0	0
9524	0	0	0	0	0	0	0	0	0
9533	0	0	0	0	0	0	0	0	0

α^{st}_{st}											
α^{st}_{st}				k							
z	t	t	t	t	t	t	t	t	t	t	t
2	0	0	0	0	0	0	0	0	0	8407	7358
3	0	0	0	0	0	0	0	0	0	2096	2324
4	0	0	0	0	0	0	0	0	0	0	2306
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	3604	3109
7	0	0	0	0	0	0	0	0	0	2820	2582
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	3604	3109
										2582	2547

α^{st}_{st}											
α^{st}_{st}				k							
z	t	t	t	t	t	t	t	t	t	t	t
2	0	0	0	0	0	0	0	0	0	4152	4943
3	0	0	0	0	0	0	0	0	0	4522	4605
4	0	0	0	0	0	0	0	0	0	1193	1178
5	0	0	0	0	0	0	0	0	0	1335	1376
6	0	0	0	0	0	0	0	0	0	8769	8331
7	0	0	0	0	0	0	0	0	0	15237	13991
8	0	0	0	0	0	0	0	0	0	17538	16662
9	0	0	0	0	0	0	0	0	0	19385	18415
10	0	0	0	0	0	0	0	0	0	24006	22322
11	0	0	0	0	0	0	0	0	0	8769	8331
12	0	0	0	0	0	0	0	0	0	15237	13991
13	0	0	0	0	0	0	0	0	0	4638	2472
14	0	0	0	0	0	0	0	0	0	2319	2238
15	0	0	0	0	0	0	0	0	0	2319	2238
16	0	0	0	0	0	0	0	0	0	24006	22322
17	0	0	0	0	0	0	0	0	0	33698	31529

CO2 Emission kgCE 298,046

BINARY VARIABLES

	t						
YC	1	1	1	1	1	1	1
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0
	6	0	0	0	0	0	0
	7	0	0	0	0	0	0

	t						
YD	1	1	1	1	1	1	1
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	1	1	1	1	1	1
	5	1	1	1	1	1	1

	t						
YR	1	1	1	1	1	1	1
	2	1	0	1	1	1	1
	3	0	0	0	0	0	0

	t						
YGO	1	0	0	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	1	1	1	1	1
	5	0	0	0	0	0	0
	6	0	0	0	0	0	0
	7	0	0	0	0	0	0

	t						
YDO	1	0	0	0	0	0	0
	2	0	0	0	0	0	0
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0

	t						
YRO	1	0	0	0	0	0	0
	2	0	0	1	1	1	1
	3	0	0	0	0	0	0

Q^E₂₈

z	t						
1	2000	1524	1805	1715	1629	3700	3881
2	2000	1900	867	1715	635	4300	4085
3	4000	3434	2672	3430	2464	8000	7966
4	4000	3434	2672	3430	2464	8000	7966
5	4000	3434	2672	3430	2464	8000	7966

Q ^E ₂₈	t						
111	1500	1425	1354	1286	1222	1222	1222
112	0	0	0	0	0	0	0
113	0	0	0	0	0	0	0
114	2850	3360	3910	3499	3128	3128	3128
115	0	0	0	0	0	0	0
116	0	0	0	0	0	0	0
117	0	0	0	0	0	0	0
211	2500	2375	2295	2143	2036	2036	2036
212	0	0	0	0	0	0	0
213	0	0	0	0	0	0	0
214	3950	4720	5549	4952	4414	4414	4414
215	0	0	0	0	0	0	0
216	0	0	0	0	0	0	0
217	0	0	0	0	0	0	0
311	2500	2375	2295	2143	2036	2036	2036
312	0	0	0	0	0	0	0
313	0	0	0	0	0	0	0
314	4850	5710	6638	5942	5314	5314	5314
315	0	0	0	0	0	0	0
316	0	0	0	0	0	0	0
317	0	0	0	0	0	0	0
411	3500	3325	3169	3001	2851	2851	2851
412	0	0	0	0	0	0	0
413	0	0	0	0	0	0	0
414	6250	7400	8639	7724	6899	6899	6899
415	0	0	0	0	0	0	0
416	0	0	0	0	0	0	0
417	0	0	0	0	0	0	0

Q ^E ₂₈	t						
1112	2000	1051	0	1715	1629	1629	1629
1113	0	0	0	0	0	0	0
1122	0	0	0	0	0	0	0
1123	0	0	0	0	0	0	0
1132	0	0	0	0	0	0	0
1133	0	0	0	0	0	0	0
1212	0	849	1805	0	0	0	0
1213	0	0	0	0	0	0	0
1222	2000	1900	1805	1715	1629	1629	1629
1223	0	0	0	0	0	0	0
1232	1700	1615	1534	1458	1385	1385	1385
1233	0	0	0	0	0	0	0
1312	0	0	0	0	0	0	0
1313	0	0	0	0	0	0	0
1322	0	0	0	0	0	0	0
1323	0	0	0	0	0	0	0
1332	0	0	0	0	0	0	0
1333	2000	1175	0	719	1629	1629	1629
2112	0	0	0	0	0	0	0
2123	0	0	0	0	0	0	0
2132	0	0	0	0	0	0	0
2133	0	0	0	0	0	0	0
2212	0	725	1805	996	0	0	0
2213	0	0	0	0	0	0	0
2222	2000	1900	1805	1715	1629	1629	1629
2223	0	0	0	0	0	0	0
2232	2300	2185	2076	1972	1873	1873	1873
2233	0	0	0	0	0	0	0
2312	0	0	0	0	0	0	0
2313	0	0	0	0	0	0	0
2322	0	0	0	0	0	0	0
2323	0	0	0	0	0	0	0
2332	0	0	0	0	0	0	0
2333	0	0	0	0	0	0	0

σ^0_{part}				σ^0_{area}			
qkso	0	0	0	zks0	0	0	t
1112	0	0	0	1112	0	0	0
1122	0	0	0	1113	0	0	0
1132	0	0	0	1122	0	0	0
1142	0	0	0	1123	0	0	0
1152	0	0	0	1132	0	0	0
1162	0	0	0	1133	0	0	0
1172	0	0	0	1142	0	0	0
1182	0	0	0	1152	0	0	0
1192	0	0	0	1162	0	0	0
1202	0	0	0	1172	0	0	0
1212	0	0	0	1182	0	0	0
1222	0	0	0	1192	0	0	0
1232	0	0	0	1202	0	0	0
1242	0	0	0	1212	0	0	0
1252	0	0	0	1222	0	0	0
1262	0	0	0	1232	0	0	0
1272	0	0	0	1242	0	0	0
1282	0	0	0	1252	0	0	0
1292	0	0	0	1262	0	0	0
1302	0	0	0	1272	0	0	0
1312	0	0	0	1282	0	0	0
1322	0	0	0	1292	0	0	0
1332	0	0	0	1302	0	0	0
1342	0	0	0	1312	0	0	0
1352	0	0	0	1322	0	0	0
1362	0	0	0	1332	0	0	0
1372	0	0	0	1342	0	0	0
1382	0	0	0	1352	0	0	0
1392	0	0	0	1362	0	0	0
1402	0	0	0	1372	0	0	0
1412	531	2724	3801	1382	0	0	0
1422	0	0	0	1392	0	0	0
1432	2319	1236	0	1402	0	0	0
1442	0	0	0	1412	0	0	0
1452	0	0	0	1422	0	0	0
1462	0	0	0	1432	8000	4452	0
1472	0	0	0	1442	0	0	0
1482	0	0	0	1452	0	0	0
1492	0	0	0	1462	0	0	0
1502	0	0	0	1472	0	0	0
1512	0	0	0	1482	0	0	0
1522	0	0	0	1492	0	0	0
1532	0	0	0	1502	0	0	0
1542	0	0	0	1512	0	0	0
1552	0	0	0	1522	16000	18348	21660
1562	0	0	0	1532	0	0	0
1572	0	0	0	1542	0	0	0
1582	0	0	0	1552	0	0	0
1592	0	0	0	1562	0	0	0
1602	0	0	0	1572	0	0	0
1612	0	0	0	1582	0	0	0
1622	0	0	0	1592	0	0	0
1632	0	0	0	1602	0	0	0
1642	0	0	0	1612	0	0	0
1652	0	0	0	1622	0	0	0
1662	0	0	0	1632	0	0	0
1672	0	0	0	1642	0	0	0
1682	0	0	0	1652	0	0	0
1692	0	0	0	1662	0	0	0
1702	0	0	0	1672	0	0	0
1712	0	0	0	1682	0	0	0
1722	0	0	0	1692	0	0	0
1732	0	0	0	1702	0	0	0
1742	0	0	0	1712	0	0	0
1752	0	0	0	1722	0	0	0
1762	0	0	0	1732	0	0	0
1772	0	0	0	1742	0	0	0
1782	0	0	0	1752	0	0	0
1792	0	0	0	1762	0	0	0
1802	0	0	0	1772	0	0	0
1812	0	0	0	1782	0	0	0
1822	0	0	0	1792	0	0	0
1832	0	0	0	1802	0	0	0
1842	0	0	0	1812	0	0	0
1852	0	0	0	1822	0	0	0
1862	0	0	0	1832	0	0	0
1872	0	0	0	1842	0	0	0
1882	0	0	0	1852	0	0	0
1892	0	0	0	1862	0	0	0
1902	0	0	0	1872	0	0	0
1912	0	0	0	1882	0	0	0
1922	0	0	0	1892	0	0	0
1932	0	0	0	1902	0	0	0
1942	0	0	0	1912	0	0	0
1952	0	0	0	1922	0	0	0
1962	0	0	0	1932	0	0	0
1972	0	0	0	1942	0	0	0
1982	0	0	0	1952	0	0	0
1992	0	0	0	1962	0	0	0
2002	0	0	0	1972	0	0	0
2012	0	0	0	1982	0	0	0
2022	0	0	0	1992	0	0	0
2032	0	0	0	2002	0	0	0
2042	0	0	0	2012	0	0	0
2052	0	0	0	2022	0	0	0
2062	0	0	0	2032	0	0	0
2072	0	0	0	2042	0	0	0
2082	0	0	0	2052	0	0	0
2092	0	0	0	2062	0	0	0
2102	0	0	0	2072	0	0	0
2112	0	0	0	2082	0	0	0
2122	0	0	0	2092	0	0	0
2132	0	0	0	2102	0	0	0
2142	1742	2375	2256	2112	0	0	0
2152	0	0	0	2122	0	0	0
2162	0	0	0	2132	0	0	0
2172	0	0	0	2142	0	0	0
2182	0	0	0	2152	0	0	0
2192	0	0	0	2162	0	0	0
2202	0	0	0	2172	0	0	0
2212	0	0	0	2182	0	0	0
2222	0	0	0	2192	0	0	0
2232	0	0	0	2202	0	0	0
2242	0	0	0	2212	0	0	0
2252	0	0	0	2222	0	0	0
2262	0	0	0	2232	0	0	0
2272	0	0	0	2242	0	0	0
2282	0	0	0	2252	0	0	0
2292	0	0	0	2262	0	0	0
2302	0	0	0	2272	0	0	0
2312	0	0	0	2282	0	0	0
2322	0	0	0	2292	0	0	0
2332	0	0	0	2302	0	0	0
2342	0	0	0	2312	0	0	0
2352	0	0	0	2322	0	0	0
2362	0	0	0	2332	0	0	0
2372	0	0	0	2342	0	0	0
2382	0	0	0	2352	0	0	0
2392	0	0	0	2362	0	0	0
2402	0	0	0	2372	0	0	0
2412	0	0	0	2382	0	0	0
2422	0	0	0	2392	0	0	0
2432	0	0	0	2402	0	0	0
2442	0	0	0	2412	0	0	0
2452	0	0	0	2422	0	0	0
2462	0	0	0	2432	0	0	0
2472	0	0	0	2442	0	0	0
2482	0	0	0	2452	0	0	0
2492	0	0	0	2462	0	0	0
2502	0	0	0	2472	0	0	0
2512	0	0	0	2482	0	0	0
2522	0	0	0	2492	0	0	0
2532	0	0	0	2502	0	0	0
2542	0	0	0	2512	0	0	0
2552	0	0	0	2522	0	0	0
2562	0	0	0	2532	0	0	0
2572	0	0	0	2542	0	0	0
2582	0	0	0	2552	0	0	0
2592	0	0	0	2562	0	0	0
2602	0	0	0	2572	0	0	0
2612	0	0	0	2582	0	0	0
2622	0	0	0	2592	0	0	0
2632	0	0	0	2602	0	0	0
2642	0	0	0	2612	0	0	0
2652	0	0	0	2622	0	0	0
2662	0	0	0	2632	0	0	0
2672	0	0	0	2642	0	0	0
2682	0	0	0	2652	0	0	0
2692	0	0	0	2662	0	0	0
2702	0	0	0	2672	0	0	0
2712	0	0	0	2682	0	0	0
2722	0	0	0	2692	0	0	0
2732	0	0	0	2702	0	0	0
2742	0	0	0	2712	0	0	0
2752	0	0	0	2722	0	0	0
2762	0	0	0	2732	0	0	0
2772	0	0	0	2742	0	0	0
2782	0	0	0	2752	0	0	0
2792	0	0	0	2762	0	0	0
2802	0	0	0	2772	0	0	0
2812	0	0	0	2782	0	0	0
2822	0	0	0	2792	0	0	0
2832	0	0	0	2802	0	0	0
2842	0	0	0	2812	0	0	0
2852	0	0	0	2822	0	0	0
2862	0	0	0	2832	0	0	0
2872	0	0	0	2842	0	0	0
2882	0	0	0	2852	0	0	0
2892	0	0	0	2862	0	0	0
2902	0	0	0	2872	0	0	0
2912	0	0	0	2882	0	0	0
2922	0	0	0	2892	0	0	0
2932	0	0	0	2902	0	0	0
2942	0	0	0	2912	0	0	0
2952	0	0	0	2922	0	0	0
2962	0	0	0	2932	0	0	0
2972	0	0	0	2942	0	0	0
2982	0	0	0	2952	0	0	0
2992	0	0	0	2962	0	0	0
3002	0	0	0	2972	0	0	0
3012	0	0	0	2982	0	0	0
3022	0	0	0	2992	0	0	0
3032	0	0	0	3002	0	0	0
3042	0	0	0	3012	0	0	0
3052	0	0	0	3022	0	0	0
3062	0	0	0	3032	0	0	0
3072	0	0	0	3042	0	0	0
3082	0	0	0	3052	0	0	0
3092	0	0	0	3062	0	0	0
3102	0	0	0	3072	0	0	0
3112	0	0	0	3082	0	0	0
3122	0	0	0	3092	0	0	0
3132	0	0	0	3102	0	0	0
3142	0	0	0	3112	0	0	0
3152	0	0	0	3122	0	0	0
3162	0	0	0	3132	0	0	0
3172	0	0	0	3142	0	0	0
3182	0	0	0	3152	0	0	0
3192	0	0	0	3162			

[illegible]

4562	0	0	0	0	0	0	0	0	0
4612	0	0	0	0	0	0	0	0	0
4622	0	0	0	0	0	0	0	0	0
4632	0	0	0	0	0	0	0	0	0
4642	0	0	0	0	0	0	0	0	0
4652	0	0	0	0	0	0	0	0	0
4662	0	0	0	0	0	0	0	0	0
4672	0	0	0	0	0	0	0	0	0
4682	0	0	0	0	0	0	0	0	0
4692	0	0	0	0	0	0	0	0	0
4702	0	0	0	0	0	0	0	0	0
4712	0	0	0	0	0	0	0	0	0
4722	0	0	0	0	0	0	0	0	0
4732	0	0	0	0	0	0	0	0	0
4742	0	0	0	0	0	0	0	0	0
4752	0	0	0	0	0	0	0	0	0

5223	0	0	0	0	0	0	0	0	0
5232	0	0	0	0	0	0	0	0	0
5242	0	0	0	0	0	0	0	0	0
5252	0	0	0	0	0	0	0	0	0
5312	0	0	0	0	0	0	0	0	0
5313	0	0	0	0	0	0	0	0	0
5322	0	0	0	0	0	0	0	0	0
5323	0	0	0	0	0	0	0	0	0
5332	0	0	0	0	0	0	0	0	0
5333	0	0	0	0	0	0	0	0	0
5412	2000	1051	0	0	1715	1629	0	0	0
5413	0	0	0	0	0	0	0	0	0
5422	0	0	0	0	0	0	0	0	0
5423	0	0	0	0	0	0	0	0	0
5432	0	0	0	0	0	0	0	0	0
5433	0	0	0	0	0	0	0	0	0
5512	0	0	0	0	0	0	0	0	0
5513	0	0	0	0	0	0	0	0	0
5522	3700	4384	5144	3173	3014	0	0	0	0
5523	0	0	0	0	0	0	0	0	0
5532	0	0	0	0	0	0	0	0	0
5533	0	0	0	0	0	0	0	0	0
6112	0	0	0	0	0	0	0	0	0
6113	0	0	0	0	0	0	0	0	0
6122	0	0	0	0	0	0	0	0	0
6123	0	0	0	0	0	0	0	0	0
6132	0	0	0	0	0	0	0	0	0
6133	0	0	0	0	0	0	0	0	0
6212	0	0	0	0	0	0	0	0	0
6213	0	0	0	0	0	0	0	0	0
6222	0	0	0	0	0	0	0	0	0
6223	0	0	0	0	0	0	0	0	0
6232	0	0	0	0	0	0	0	0	0
6233	0	0	0	0	0	0	0	0	0
6312	0	0	0	0	0	0	0	0	0
6313	0	0	0	0	0	0	0	0	0
6322	0	0	0	0	0	0	0	0	0
6323	0	0	0	0	0	0	0	0	0
6332	0	0	0	0	0	0	0	0	0
6333	0	0	0	0	0	0	0	0	0
6412	2000	1175	0	719	1629	0	0	0	0
6413	0	0	0	0	0	0	0	0	0
6422	0	0	0	0	0	0	0	0	0
6423	0	0	0	0	0	0	0	0	0
6432	0	0	0	0	0	0	0	0	0
6433	0	0	0	0	0	0	0	0	0
6512	0	0	0	0	0	0	0	0	0
6513	0	0	0	0	0	0	0	0	0
6522	4300	4810	5686	4683	3502	0	0	0	0
6523	0	0	0	0	0	0	0	0	0
6532	0	0	0	0	0	0	0	0	0
6533	0	0	0	0	0	0	0	0	0
7112	0	0	0	0	0	0	0	0	0
7113	0	0	0	0	0	0	0	0	0
7122	0	0	0	0	0	0	0	0	0
7123	0	0	0	0	0	0	0	0	0
7132	0	0	0	0	0	0	0	0	0
7133	0	0	0	0	0	0	0	0	0
7212	0	0	0	0	0	0	0	0	0
7213	0	0	0	0	0	0	0	0	0
7222	0	0	0	0	0	0	0	0	0
7223	0	0	0	0	0	0	0	0	0
7232	0	0	0	0	0	0	0	0	0
7233	0	0	0	0	0	0	0	0	0
7312	0	0	0	0	0	0	0	0	0
7313	0	0	0	0	0	0	0	0	0
7322	0	0	0	0	0	0	0	0	0

7323	0	0	0	0	0	0	0	0	0
7332	0	0	0	0	0	0	0	0	0
7333	0	0	0	0	0	0	0	0	0
7412	4000	2102	0	3430	3258	0	0	0	0
7413	0	0	0	0	0	0	0	0	0
7422	0	0	0	0	0	0	0	0	0
7423	0	0	0	0	0	0	0	0	0
7432	0	0	0	0	0	0	0	0	0
7433	0	0	0	0	0	0	0	0	0
7512	0	0	0	0	0	0	0	0	0
7513	0	0	0	0	0	0	0	0	0
7522	7400	8728	10288	6346	6028	0	0	0	0
7523	0	0	0	0	0	0	0	0	0
7532	0	0	0	0	0	0	0	0	0
7533	0	0	0	0	0	0	0	0	0
8112	0	0	0	0	0	0	0	0	0
8113	0	0	0	0	0	0	0	0	0
8122	0	0	0	0	0	0	0	0	0
8123	0	0	0	0	0	0	0	0	0
8132	0	0	0	0	0	0	0	0	0
8133	0	0	0	0	0	0	0	0	0
8212	0	0	0	0	0	0	0	0	0
8213	0	0	0	0	0	0	0	0	0
8222	0	0	0	0	0	0	0	0	0
8223	0	0	0	0	0	0	0	0	0
8232	0	0	0	0	0	0	0	0	0
8233	0	0	0	0	0	0	0	0	0
8312	0	0	0	0	0	0	0	0	0
8313	0	0	0	0	0	0	0	0	0
8322	0	0	0	0	0	0	0	0	0
8323	0	0	0	0	0	0	0	0	0
8422	0	0	0	0	0	0	0	0	0
8423	4000	2350	0	1438	3258	0	0	0	0
8424	0	0	0	0	0	0	0	0	0
8425	0	0	0	0	0	0	0	0	0
8426	0	0	0	0	0	0	0	0	0
8427	0	0	0	0	0	0	0	0	0
8428	0	0	0	0	0	0	0	0	0
8429	0	0	0	0	0	0	0	0	0
8430	0	0	0	0	0	0	0	0	0
8431	0	0	0	0	0	0	0	0	0
8432	0	0	0	0	0	0	0	0	0
8433	0	0	0	0	0	0	0	0	0
8512	0	0	0	0	0	0	0	0	0
8513	0	0	0	0	0	0	0	0	0
8522	8600	9620	11372	9385	7004	0	0	0	0
8523	0	0	0	0	0	0	0	0	0
8532	0	0	0	0	0	0	0	0	0
8533	0	0	0	0	0	0	0	0	0
8542	0	0	0	0	0	0	0	0	0
8543	0	0	0	0	0	0	0	0	0
9112	0	0	0	0	0	0	0	0	0
9113	0	0	0	0	0	0	0	0	0
9122	0	0	0	0	0	0	0	0	0
9123	0	0	0	0	0	0	0	0	0
9132	0	0	0	0	0	0	0	0	0
9133	0	0	0	0	0	0	0	0	0
9212	0	0	0	0	0	0	0	0	0
9213	0	0	0	0	0	0	0	0	0
9222	0	0	0	0	0	0	0	0	0
9223	0	0	0	0	0	0	0	0	0
9232	0	0	0	0	0	0	0	0	0
9233	0	0	0	0	0	0	0	0	0
9312	0	0	0	0	0	0	0	0	0
9313	0	0	0	0	0	0	0	0	0
9322	0	0	0	0	0	0	0	0	0
9323	0	0	0	0	0	0	0	0	0
9332	0	0	0	0	0	0	0	0	0
9333	0	0	0	0	0	0	0	0	0
9412	4000	2226	0	2434	3258	0	0	0	0
9413	0	0	0	0	0	0	0	0	0
9422	0	0	0	0	0	0	0	0	0

9423	0	0	0	0	0	0	0
9432	0	0	0	0	0	0	0
9433	0	0	0	0	0	0	0
9512	0	0	0	0	0	0	0
9513	0	0	0	0	0	0	0
9522	8000	9174	0	10850	7850	850	850
9532	0	0	0	0	0	0	0
9533	0	0	0	0	0	0	0

z	t										k														
	t		t		t		t		t		t		t		t		t		t						
1	4208	4940	4893	4458	4235	690	3966	8356	6557	4890	0	0	0	0	0	1625	0	3670	1715	3217	1368	3472	379	1869	0
2	2498	7033	10864	9063	7473	0	1205	3415	2560	1815	0	0	0	0	0	2375	3069	0	1715	41	6033	4289	6299	4477	6028
3	1174	1544	1466	1393	1323	0	602	1707	1280	907	0	0	0	0	0	813	366	867	858	815	1284	1940	1670	1448	1282
4	75	1973	3965	3139	2413	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	975	926	980	836	794	345	1381	2471	1998	1583	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1211	2530	3448	2962	2530	0	602	1707	1280	907	0	0	0	0	0	1188	1169	938	858	815	2416	1940	1670	1725	1732
7	2348	3068	2933	2786	2647	0	1205	3415	2560	1815	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	75	1973	3965	3139	2413	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	2186	3466	4328	3798	3324	345	1983	4178	3278	2490	0	0	0	0	0	1188	1169	938	858	815	2416	1940	1670	1725	1732

z	t										k														
	6612	7600	7220	6858	6516	1062	6102	12856	10087	7662	0	0	0	0	0	14908	10567	13776	13192	12531	26719	29851	25685	23883	22492
1	3842	10819	16713	13943	11497	0	1853	5253	3939	2792	0	0	0	0	0	15962	15289	8222	13192	7646	33896	31109	34792	27996	31765
2	1806	2375	2256	2143	2036	0	927	2627	1970	1396	0	0	0	0	0	4327	2923	4111	3958	3769	7667	8955	7705	7110	6609
3	115	3035	6101	4828	3712	0	0	0	0	0	0	0	0	0	0	6154	5846	2668	5277	2570	13231	12569	14827	11345	13218
4	1500	1425	1354	1286	1222	531	2124	3801	3074	2435	0	0	0	0	0	3077	2361	2777	2638	2506	5692	5970	5137	4882	4637
5	1883	3982	5306	4557	3892	0	927	2627	1970	1396	0	0	0	0	0	4904	4721	2777	3958	2538	10333	9270	9982	8326	9274
6	3612	4750	4512	4286	4072	0	1853	5253	3939	2792	0	0	0	0	0	6154	4721	5564	5277	5012	11385	11940	10274	9763	9274
7	345	1973	3965	3138	2413	0	0	0	0	0	0	0	0	0	0	6154	5846	2668	5277	2570	13231	12569	14827	11345	13218
8	3383	5317	6660	5843	5114	531	3051	6428	5044	3831	0	0	0	0	0	7981	7082	5564	6596	5044	16025	15240	15119	13207	13911
9	1500	1425	1354	1286	1222	531	2124	3801	3074	2435	0	0	0	0	0	3077	2361	2777	2638	2506	5692	5970	5137	4882	4637
10	1883	3982	5306	4557	3892	0	927	2627	1970	1396	0	0	0	0	0	4904	4721	2777	3958	2538	10333	9270	9982	8326	9274
11	3000	2850	2708	2572	2444	1062	4248	7602	6148	4870	0	0	0	0	0	1154	2472	218	0	0	3485	0	0	850	1386
12	1500	1425	1354	1286	1222	531	2124	3801	3074	2435	0	0	0	0	0	577	1236	109	0	0	1742	0	0	425	693
13	3383	5317	6660	5843	5114	531	3051	6428	5044	3831	0	0	0	0	0	7981	7082	5564	6596	5044	16025	15240	15119	13207	13911
14	1500	1425	1354	1286	1222	531	2124	3801	3074	2435	0	0	0	0	0	577	1236	109	0	0	1742	0	0	425	693
15	3383	5317	6660	5843	5114	531	3051	6428	5044	3831	0	0	0	0	0	7981	7082	5564	6596	5044	16025	15240	15119	13207	13911
16	3383	5317	6660	5843	5114	531	3051	6428	5044	3831	0	0	0	0	0	7981	7082	5564	6596	5044	16025	15240	15119	13207	13911
17	3421	6835	9711	8257	6970	531	3051	6428	5044	3831	0	0	0	0	0	11058	10005	6888	9235	6329	22640	21524	22533	18879	20520

Appendix F

Multi-objective Optimisation Methods

As can be seen from the previous section, several reverse logistics network models have multiple objective functions to be optimised. This section briefly explains multi-objective optimisation methods, conventional and evolutionary techniques and the approaches in solving multi-objective problems with uncertainties.

F.1. Conventional techniques

In multi-objective optimisation, vectors of decision variables are regarded as optimal if their components cannot be improved without deterioration to at least one of the other components (Miettinen, 2001). This is usually called Pareto optimality. There are usually a large number of Pareto optimal solutions (non-dominated/non-inferior/efficient solutions) and they form a set of Pareto optimal solutions or a Pareto optimal set.

Solving a multi-objective optimisation problem means finding a feasible decision vector such that it is Pareto optimal and satisfies the decision maker (DM). Li et al. (1999) classified multi-objective optimisation methods into two broad categories, namely non-interactive methodologies and interactive methodologies. A non-interactive methodology requires decision maker's preference either before or after generating non-dominated solutions. An interactive methodology that searches for the most preferred solution in the set of non-dominated solutions requires on-line progressive determination of decision maker's preference during the solution process.

Table F.1: Conventional multi-objective optimisation methods (Miettinen, 2001)

Class	Method	Descriptions
No-preference methods	Single objective approach	Optimising one objective (the most important one) and convert the remaining objectives as constraints.
	Minimum deviation method	Aiming at finding the best compromise solution which minimises the sum of individual objective's fractional deviation.
A posteriori methods	Weighting method	The objective is to minimise the weighted sum of all objectives. The decision variables belong to the feasible region, and the sum of all criteria weights equals to one.
	ϵ -constraint method	The objective is to minimise one of the objectives. Each of the remaining objectives must be less than or equal to its upper bound (ϵ) and the decision variables belong to the feasible region.
	Method of weighted metrics	The objective is to minimise the distance between some reference point and the feasible objective region.
A priori methods	Value function method	Value function (U) is a function that represents the preferences of (DM). The function is assumed to be strongly decreasing and in this case, the maximum of U is Pareto optimal.
	Lexicographic ordering	DM must arrange the objectives according to their absolute importance. The most important one is minimised subject to the original constraints. If it has a unique solution, the process stops, otherwise it continues by minimising the 2 nd most important one with added new constraints to preserve its optimal value.
	Goal programming	The DM is asked to specify aspiration levels for the objectives and deviations from the aspiration levels are minimised. An objective jointly with an aspiration level forms a goal.
Interactive methods	Geoffrion-Dyer-Feinberg method	The method is based on the maximisation of the underlying (implicitly known) value function. The objective is assumed to be continuously differentiable and the feasible region must be compact and convex.
	Tchebycheff method	It is an interactive weighting vector space reduction method where value functions are not used. It is assumed that the objective functions are bounded.
	GUESS method	In this method, the DM specifies a reference point (or a guess) below the nadir objective vector and the minimum weighted deviation from the nadir objective is maximised. Then DM specifies a new reference point and the iteration continues until DM is satisfied with the solution produced.
	STEM (STEP Method)	In this method, preference information from DM is used to reduce solution space successively. The general optimisation problem is reformulated as min-max formulation with bounded and weighted objectives.

A more detailed classification of multi-objective optimisation methods was carried out by Miettinen (2001) who divided the methodologies into four classes (non-interactive methodologies were further classified into three classes):

- No-preference methods, where the opinions of DM are not taken into consideration.
- A posteriori methods, also known as methods for generating Pareto optimal solutions. After Pareto optimal set has been generated, it is presented to the DM, who selects the most preferred solution.
- A priori methods, where the DM must specify his/her preferences, hopes and opinions before the solution process.
- Interactive methods, where the DM works together with an analyst or an interactive computer program; the analyst tries to determine the preference structure of the DM in an interactive way.

Several methods (and their descriptions) belonging to each class can be found in Table F.1.

F.2. Evolutionary multi-objective optimisation techniques

Evolutionary techniques are stochastic algorithms whose search methods model some natural phenomena, namely genetic inheritance and Darwinian strife for survival (Michalewicz, 1995). Evolutionary techniques for multi-objective optimisation are gaining attention due to their effectiveness and robustness in searching for a set of global trade-off solutions (Tan et al., 2001). The most well known class of evolutionary techniques is GA, developed by Holland (1975). GA combines elements of directed and stochastic searches which can make a good balance between exploration and exploitation of the search space. GA differs from

conventional optimisation and search procedures in several fundamental ways (Goldberg, 1989):

- It works with a coding of solution set, not the solution itself.
- It searches from a population of solutions, not a single solution.
- It uses payoff information (fitness function), not derivatives or other auxiliary knowledge.
- It uses probabilistic transition rules, not deterministic rules.

Because of its evolutionary nature, GA can be used to search for solutions without regard to the specific inner workings of the problem.

In general, GA has five basic components (Michalewicz, 1995):

- A genetic representation of solutions to the problem,
- A way to create an initial population of solutions,
- An evaluation function rating solutions in terms of their fitness,
- Genetic operators that alter the genetic composition of offspring during reproduction (crossover and mutation operators),
- Values for the parameters of GA.

GA starts with an initial set of solutions called a population. Each individual in the population is called a chromosome, and represents a candidate solution to the problem. A chromosome consists of a string of symbol which are usually, but not necessarily, binary numbers. The chromosomes evolve through successive iterations called generations. During each generation, the chromosomes are evaluated using some measures of fitness.

To create the next generation, new chromosomes called offspring, are formed using genetic operators. Crossover operates on two chromosomes at a time and

generates offspring by combining features of both chromosomes, while mutation produces spontaneous random changes in various chromosomes. A new generation is formed by selecting from the parents and offspring based on their fitness values. After several generations, the best solution converges and hopefully represents the optimal or sub optimal solution to the problem. The GA procedure is presented in Figure F.1.

Figure F.1: GA procedure (Gen and Cheng, 2000)

The basic issue in GA is to set of the GA parameters, such as population size and crossover and mutation rates. Experimental work by Alander (1992) suggested that a value for population size, which is between n and $2n$, where n is the string length, is optimal for some problems. Concerning crossover and mutation rates, according to De Jong (in Goldberg, 1989), good GA performance requires the choice of high crossover rates and low mutation rates (inversely proportional to population) and a moderate population size. Gen and Cheng (1997) further stated that:

- A higher crossover rate allows exploration of more of the solution space and reduces the chances of settling for a false optimum; but if it is too high, it results in the wastage of computational time in exploring unpromising regions of solution space.

- If mutation rate is too low, many chromosomes that would have been useful are never tried out; but if it is too high, there will be much random perturbation, and the offspring start losing their resemblance to the parents and the algorithm will lose the ability to learn from history of search.

From the viewpoint of methodology, there are two basic approaches to multi-objective optimisation, namely generation and preference-based. Generation approaches are used to identify an entire set of Pareto solutions or an approximation thereof, whereas preference-based approaches attempt to obtain a compromised or preferred solution.

Issues in genetic multi-objective optimisation are determining the fitness value of individuals according to multi-objective functions and handling solutions found in a given generation. Concerning the first issue, methods to determine fitness value can be categorised into five classes, namely vector evaluation, weighted-sum, Pareto based, compromise and goal-programming (Gen and Cheng, 2000). Conceptually, vector evaluation, Pareto ranking-based and random-weighting approaches are designated as generation methods, whereas compromise, adaptive weighting and goal programming approaches are designated as preference-based approaches. Concerning the solutions in a given generation, infeasible solutions may possibly be found. Two major techniques used in handling this problem are repairing and penalising. Also in a given generation, a Pareto solution obtained may or may not be a true Pareto solution to the problem. There are two different ways to handle Pareto solutions, which are either to preserve Pareto solutions separately from the population pool or not to preserve them.

Appendix G

The Best Three Solutions of Numerical Example of Chapter 4

For all solutions:

$$\begin{aligned}\text{alloc_1}[t,r,i,j] &= Q_{rijt}^A \\ \text{alloc_3}[t,z,k,l,o] &= Q_{zklot}^C \\ \text{spare_part}[t,z,k] &= Q_{zkt}^H\end{aligned}$$

$$\begin{aligned}\text{alloc_2}[t,r,j,k,o] &= Q_{rjkot}^B \\ \text{alloc_4}[t,r,m,l,o] &= Q_{rlmot}^E \\ \text{new_part}[t,z,l] &= Q_{zlt}^R\end{aligned}$$

$\text{recyclable}[t,z,k]$ = portion of used module z obtained at disassembly centre k that are sold to recycler in period t .

$\text{waste}[t,z,k]$ = portion of used module z obtained at disassembly centre k that are disposed of in period t .

Total net revenue is in \$, energy consumption is in MJ and CO₂ emission is in KgCE..

Solution 1

collection centres

period 1
facilities = [0 0 0 1 0 0 0]
total capacity for product 1 = 4500
total capacity for product 2 = 7500
total capacity for product 3 = 7500
total capacity for product 4 = 10500

period 2
facilities = [1 0 0 1 0 0 0]
total capacity for product 1 = 5925
total capacity for product 2 = 9875
total capacity for product 3 = 9875
total capacity for product 4 = 13825

period 3
facilities = [1 0 0 1 0 0 0]
total capacity for product 1 = 5854
total capacity for product 2 = 9756
total capacity for product 3 = 9756
total capacity for product 4 = 13659

period 4
facilities = [0 0 0 1 1 0 0]
total capacity for product 1 = 13500
total capacity for product 2 = 22500
total capacity for product 3 = 22500
total capacity for product 4 = 31500

period 5
facilities = [0 0 0 1 0 0 0]
total capacity for product 1 = 4500
total capacity for product 2 = 7500
total capacity for product 3 = 7500
total capacity for product 4 = 10500

disassembly centres

period 1
facilities = [0 0 0 0 1]

total capacity for product 1 = 13500
total capacity for product 2 = 19500
total capacity for product 3 = 22500
total capacity for product 4 = 30000

period 2
facilities = [0 0 0 0 1]
total capacity for product 1 = 13500
total capacity for product 2 = 19500
total capacity for product 3 = 22500
total capacity for product 4 = 30000

period 3
facilities = [0 0 0 0 1]
total capacity for product 1 = 13500
total capacity for product 2 = 19500
total capacity for product 3 = 22500
total capacity for product 4 = 30000

period 4
facilities = [0 0 0 0 1]
total capacity for product 1 = 13500
total capacity for product 2 = 19500
total capacity for product 3 = 22500
total capacity for product 4 = 30000

period 5
facilities = [0 0 0 0 1]
total capacity for product 1 = 13500
total capacity for product 2 = 19500
total capacity for product 3 = 22500
total capacity for product 4 = 30000

recovery centres

period 1
facilities = [0 1 0]
total capacity for product 1 = 18000
total capacity for product 2 = 21000

period 2	[1 6 2 5]
facilities = [0 1 0]	
total capacity for product 1 = 18000	[6 2 5 2]
total capacity for product 2 = 21000	
	[6 3 4 3]
period 3	
facilities = [0 1 0]	[4 2 4 2]
total capacity for product 1 = 18000	
total capacity for product 2 = 21000	[3 4 1 5]
period 4	
facilities = [0 1 0]	
total capacity for product 1 = 18000	alloc_1[1,1,1,4] = 4350
total capacity for product 2 = 21000	alloc_1[1,2,1,4] = 6450
	alloc_1[1,3,1,4] = 7350
	alloc_1[1,4,1,4] = 9750
period 5	
facilities = [0 1 0]	alloc_2[1,1,4,5,2] = 4350
total capacity for product 1 = 18000	alloc_2[1,2,4,5,2] = 6450
total capacity for product 2 = 21000	alloc_2[1,3,4,5,2] = 7350
	alloc_2[1,4,4,5,2] = 9750
prufer number-stage 1	
[1 1 1 1 1 1]	alloc_4[1,1,1,2,2] = 2000
	alloc_4[1,1,2,2,2] = 2000
[1 1 1 1 1 1]	alloc_4[1,1,3,2,2] = 1700
	alloc_4[1,2,1,2,2] = 2000
[1 1 1 1 1 1]	alloc_4[1,2,2,2,2] = 2000
	alloc_4[1,2,3,2,2] = 2300
[1 1 1 1 1 1]	alloc_3[1,1,5,2,2] = 24000
	alloc_3[1,2,5,2,2] = 24000
[1 1 1 1 1 1]	alloc_3[1,3,5,2,2] = 5700
	alloc_3[1,4,5,2,2] = 12600
prufer number-stage 2	alloc_3[1,5,5,2,2] = 5700
	alloc_3[1,6,5,2,2] = 6300
[9 4 10 11 11 2 11 6 10 6]	alloc_3[1,7,5,2,2] = 11400
	alloc_3[1,8,5,2,2] = 12600
[11 10 10 5 10 2 12 6 4 9]	alloc_3[1,9,5,2,2] = 12000
[9 10 8 8 6 10 6 7 11 3]	spare_part[1][1][5] = 7980
	spare_part[1][2][5] = 10905
[10 11 6 9 9 6 10 6 11 4]	spare_part[1][3][5] = 3270
	spare_part[1][4][5] = 75
[9 10 5 9 10 5 2 12 10 6]	spare_part[1][5][5] = 1320
	spare_part[1][6][5] = 4815
prufer number-stage 3	spare_part[1][7][5] = 2347
	spare_part[1][8][5] = 75
[6 7 5 2 7 6]	spare_part[1][9][5] = 6135
[5 7 5 7 7 8]	recyclable[1][1][5] = 9839
	waste[1][1][5] = 7381
[4 6 7 7 3 6]	recyclable[1][2][5] = 10739
	waste[1][2][5] = 8056
[7 5 8 2 7 6]	recyclable[1][3][5] = 2759
	waste[1][3][5] = 2071
[5 7 7 3 6 7]	recyclable[1][4][5] = 3899
	waste[1][4][5] = 2926
prufer number-stage 4	recyclable[1][5][5] = 2159

```

recyclable[1][6][5] = 3419
recyclable[1][7][5] = 4229
waste[1][7][5] = 3174
recyclable[1][8][5] = 3899
waste[1][8][5] = 2926
recyclable[1][9][5] = 5579
waste[1][9][5] = 4186
recyclable[1][10][5] = 9180
waste[1][10][5] = 1620
recyclable[1][11][5] = 14535
waste[1][11][5] = 2565
recyclable[1][12][5] = 7395
waste[1][12][5] = 1305
recyclable[1][13][5] = 3697
waste[1][13][5] = 653
recyclable[1][14][5] = 23715
waste[1][14][5] = 4185
recyclable[1][15][5] = 3697
waste[1][15][5] = 653
recyclable[1][16][5] = 23715
waste[1][16][5] = 4185
recyclable[1][17][5] = 32002
waste[1][17][5] = 5648

new_part[1][1][2] = 5700
new_part[1][2][2] = 6300
new_part[1][3][2] = 12000
new_part[1][4][2] = 12000
new_part[1][5][2] = 12000

alloc_1[2,1,1,1] = 285
alloc_1[2,1,1,4] = 4500
alloc_1[2,2,1,4] = 7095
alloc_1[2,3,1,1] = 585
alloc_1[2,3,1,4] = 7500
alloc_1[2,4,1,1] = 225
alloc_1[2,4,1,4] = 10500

alloc_2[2,1,1,5,2] = 285
alloc_2[2,1,4,5,2] = 4500
alloc_2[2,2,4,5,2] = 7095
alloc_2[2,3,1,5,2] = 585
alloc_2[2,3,4,5,2] = 7500
alloc_2[2,4,1,5,2] = 225
alloc_2[2,4,4,5,2] = 10500

alloc_4[2,1,1,2,2] = 1900
alloc_4[2,1,2,2,2] = 1900
alloc_4[2,1,3,2,2] = 1615
alloc_4[2,2,1,2,2] = 1900
alloc_4[2,2,2,2,2] = 1900
alloc_4[2,2,3,2,2] = 2185

alloc_3[2,1,5,2,2] = 22800
alloc_3[2,2,5,2,2] = 22800
alloc_3[2,3,5,2,2] = 5415
alloc_3[2,4,5,2,2] = 11970
alloc_3[2,5,5,2,2] = 5415

alloc_3[2,6,5,2,2] = 5985
alloc_3[2,7,5,2,2] = 10830
alloc_3[2,8,5,2,2] = 11970
alloc_3[2,9,5,2,2] = 11400

spare_part[2][1][5] = 12378
spare_part[2][2][5] = 15595
spare_part[2][3][5] = 4452
spare_part[2][4][5] = 1972
spare_part[2][5][5] = 2307
spare_part[2][6][5] = 6241
spare_part[2][7][5] = 4292
spare_part[2][8][5] = 1972
spare_part[2][9][5] = 8548

recyclable[2][1][5] = 10823
waste[2][1][5] = 8119
recyclable[2][2][5] = 11813
waste[2][2][5] = 8862
recyclable[2][3][5] = 3035
waste[2][3][5] = 2278
recyclable[2][4][5] = 4289
waste[2][4][5] = 3219
recyclable[2][5][5] = 2375
waste[2][5][5] = 1783
recyclable[2][6][5] = 3761
waste[2][6][5] = 2823
recyclable[2][7][5] = 4652
waste[2][7][5] = 3491
recyclable[2][8][5] = 4289
waste[2][8][5] = 3219
recyclable[2][9][5] = 6137
waste[2][9][5] = 4605
recyclable[2][10][5] = 10098
waste[2][10][5] = 1782
recyclable[2][11][5] = 15988
waste[2][11][5] = 2822
recyclable[2][12][5] = 8134
waste[2][12][5] = 1436
recyclable[2][13][5] = 4067
waste[2][13][5] = 718
recyclable[2][14][5] = 26086
waste[2][14][5] = 4604
recyclable[2][15][5] = 4067
waste[2][15][5] = 718
recyclable[2][16][5] = 26086
waste[2][16][5] = 4604
recyclable[2][17][5] = 35202
waste[2][17][5] = 6213

new_part[2][1][2] = 5415
new_part[2][2][2] = 5985
new_part[2][3][2] = 11400
new_part[2][4][2] = 11400
new_part[2][5][2] = 11400

alloc_1[3,1,1,1] = 764
alloc_1[3,1,1,4] = 4500

```



```

alloc_1[3,2,1,1] = 305
alloc_1[3,2,1,4] = 7500
alloc_1[3,3,1,1] = 1394
alloc_1[3,3,1,4] = 7500
alloc_1[3,4,1,1] = 1298
alloc_1[3,4,1,4] = 10500

alloc_2[3,1,1,5,2] = 764
alloc_2[3,1,4,5,2] = 4500
alloc_2[3,2,1,5,2] = 305
alloc_2[3,2,4,5,2] = 7500
alloc_2[3,3,1,5,2] = 1394
alloc_2[3,3,4,5,2] = 7500
alloc_2[3,4,1,5,2] = 1298
alloc_2[3,4,4,5,2] = 10500

alloc_4[3,1,1,2,2] = 1805
alloc_4[3,1,2,2,2] = 1805
alloc_4[3,1,3,2,2] = 1534
alloc_4[3,2,1,2,2] = 1805
alloc_4[3,2,2,2,2] = 1805
alloc_4[3,2,3,2,2] = 2076

alloc_3[3,1,5,2,2] = 21660
alloc_3[3,2,5,2,2] = 21660
alloc_3[3,3,5,2,2] = 5144
alloc_3[3,4,5,2,2] = 11372
alloc_3[3,5,5,2,2] = 5144
alloc_3[3,6,5,2,2] = 5686
alloc_3[3,7,5,2,2] = 10288
alloc_3[3,8,5,2,2] = 11372
alloc_3[3,9,5,2,2] = 10830

spare_part[3][1][5] = 17038
spare_part[3][2][5] = 20577
spare_part[3][3][5] = 5710
spare_part[3][4][5] = 3965
spare_part[3][5][5] = 3350
spare_part[3][6][5] = 7763
spare_part[3][7][5] = 6347
spare_part[3][8][5] = 3965
spare_part[3][9][5] = 11114

recyclable[3][1][5] = 11907
waste[3][1][5] = 8931
recyclable[3][2][5] = 12995
waste[3][2][5] = 9748
recyclable[3][3][5] = 3339
waste[3][3][5] = 2506
recyclable[3][4][5] = 4719
waste[3][4][5] = 3540
recyclable[3][5][5] = 2613
waste[3][5][5] = 1962
recyclable[3][6][5] = 4138
waste[3][6][5] = 3105
recyclable[3][7][5] = 5118
waste[3][7][5] = 3840
recyclable[3][8][5] = 4719

waste[3][8][5] = 3540
recyclable[3][9][5] = 6752
waste[3][9][5] = 5065
recyclable[3][10][5] = 11108
waste[3][10][5] = 1961
recyclable[3][11][5] = 17588
waste[3][11][5] = 3104
recyclable[3][12][5] = 8948
waste[3][12][5] = 1580
recyclable[3][13][5] = 4474
waste[3][13][5] = 790
recyclable[3][14][5] = 28696
waste[3][14][5] = 5065
recyclable[3][15][5] = 4474
waste[3][15][5] = 790
recyclable[3][16][5] = 28696
waste[3][16][5] = 5065
recyclable[3][17][5] = 38725
waste[3][17][5] = 6834

new_part[3][1][2] = 5144
new_part[3][2][2] = 5686
new_part[3][3][2] = 10830
new_part[3][4][2] = 10830
new_part[3][5][2] = 10830

alloc_1[4,1,1,4] = 4500
alloc_1[4,1,1,5] = 285
alloc_1[4,2,1,4] = 7095
alloc_1[4,3,1,4] = 7500
alloc_1[4,3,1,5] = 585
alloc_1[4,4,1,4] = 10500
alloc_1[4,4,1,5] = 225

alloc_2[4,1,4,5,2] = 4500
alloc_2[4,1,5,5,2] = 285
alloc_2[4,2,4,5,2] = 7095
alloc_2[4,3,4,5,2] = 7500
alloc_2[4,3,5,5,2] = 585
alloc_2[4,4,4,5,2] = 10500
alloc_2[4,4,5,5,2] = 225

alloc_4[4,1,1,2,2] = 1715
alloc_4[4,1,2,2,2] = 1715
alloc_4[4,1,3,2,2] = 1458
alloc_4[4,2,1,2,2] = 1715
alloc_4[4,2,2,2,2] = 1715
alloc_4[4,2,3,2,2] = 1972

alloc_3[4,1,5,2,2] = 20580
alloc_3[4,2,5,2,2] = 20580
alloc_3[4,3,5,2,2] = 4888
alloc_3[4,4,5,2,2] = 10804
alloc_3[4,5,5,2,2] = 4888
alloc_3[4,6,5,2,2] = 5402
alloc_3[4,7,5,2,2] = 9776
alloc_3[4,8,5,2,2] = 10804
alloc_3[4,9,5,2,2] = 10290

```

spare_part[4][1][5] = 14598
 spare_part[4][2][5] = 17815
 spare_part[4][3][5] = 4979
 spare_part[4][4][5] = 3138
 spare_part[4][5][5] = 2834
 spare_part[4][6][5] = 6824
 spare_part[4][7][5] = 5346
 spare_part[4][8][5] = 3138
 spare_part[4][9][5] = 9658

recyclable[4][1][5] = 10823
 waste[4][1][5] = 8119
 recyclable[4][2][5] = 11813
 waste[4][2][5] = 8862
 recyclable[4][3][5] = 3035
 waste[4][3][5] = 2278
 recyclable[4][4][5] = 4289
 waste[4][4][5] = 3219
 recyclable[4][5][5] = 2375
 waste[4][5][5] = 1783
 recyclable[4][6][5] = 3761
 waste[4][6][5] = 2823
 recyclable[4][7][5] = 4652
 waste[4][7][5] = 3491
 recyclable[4][8][5] = 4289
 waste[4][8][5] = 3219
 recyclable[4][10][5] = 10098
 waste[4][10][5] = 1782
 recyclable[4][11][5] = 15988
 waste[4][11][5] = 2822
 recyclable[4][12][5] = 8134
 waste[4][12][5] = 1436
 recyclable[4][13][5] = 4067
 waste[4][13][5] = 718
 recyclable[4][14][5] = 26086
 waste[4][14][5] = 4604
 recyclable[4][15][5] = 4067
 waste[4][15][5] = 718
 recyclable[4][16][5] = 26086
 waste[4][16][5] = 4604
 recyclable[4][17][5] = 35202
 waste[4][17][5] = 6213

new_part[4][1][2] = 4888
 new_part[4][2][2] = 5402
 new_part[4][3][2] = 10290
 new_part[4][4][2] = 10290
 new_part[4][5][2] = 10290

alloc_1[5,1,1,4] = 4350
 alloc_1[5,2,1,4] = 6450
 alloc_1[5,3,1,4] = 7350
 alloc_1[5,4,1,4] = 9750

alloc_2[5,1,4,5,2] = 4350
 alloc_2[5,2,4,5,2] = 6450
 alloc_2[5,3,4,5,2] = 7350

alloc_2[5,4,4,5,2] = 9750

alloc_4[5,1,1,2,2] = 1629
 alloc_4[5,1,2,2,2] = 1629
 alloc_4[5,1,3,2,2] = 1385
 alloc_4[5,2,1,2,2] = 1629
 alloc_4[5,2,2,2,2] = 1629
 alloc_4[5,2,3,2,2] = 1873

alloc_3[5,1,5,2,2] = 19548
 alloc_3[5,2,5,2,2] = 19548
 alloc_3[5,3,5,2,2] = 4643
 alloc_3[5,4,5,2,2] = 10262
 alloc_3[5,5,5,2,2] = 4643
 alloc_3[5,6,5,2,2] = 5131
 alloc_3[5,7,5,2,2] = 9286
 alloc_3[5,8,5,2,2] = 10262
 alloc_3[5,9,5,2,2] = 9774

spare_part[5][1][5] = 12432
 spare_part[5][2][5] = 15357
 spare_part[5][3][5] = 4327
 spare_part[5][4][5] = 2413
 spare_part[5][5][5] = 2377
 spare_part[5][6][5] = 5984
 spare_part[5][7][5] = 4461
 spare_part[5][8][5] = 2413
 spare_part[5][9][5] = 8361

recyclable[5][1][5] = 9839
 waste[5][1][5] = 7381
 recyclable[5][2][5] = 10739
 waste[5][2][5] = 8056
 recyclable[5][3][5] = 2759
 waste[5][3][5] = 2071
 recyclable[5][4][5] = 3899
 waste[5][4][5] = 2926
 recyclable[5][5][5] = 2159
 waste[5][5][5] = 1621
 recyclable[5][6][5] = 3419
 waste[5][6][5] = 2566
 recyclable[5][7][5] = 4229
 waste[5][7][5] = 3174
 recyclable[5][8][5] = 3899
 waste[5][8][5] = 2926
 recyclable[5][9][5] = 5579
 waste[5][9][5] = 4186
 recyclable[5][10][5] = 9180
 waste[5][10][5] = 1620
 recyclable[5][11][5] = 14535
 waste[5][11][5] = 2565
 recyclable[5][12][5] = 7395
 waste[5][12][5] = 1305
 recyclable[5][13][5] = 3697
 waste[5][13][5] = 653
 recyclable[5][14][5] = 23715
 waste[5][14][5] = 4185
 recyclable[5][15][5] = 3697

waste[5][15][5] = 653
 recyclable[5][16][5] = 23715
 recyclable[5][17][5] = 32002
 waste[5][17][5] = 5648

new_part[5][1][2] = 4643
 new_part[5][2][2] = 5131
 new_part[5][3][2] = 9774
 new_part[5][4][2] = 9774
 new_part[5][5][2] = 9774

total net revenue = 4590164.77
total energy consumption = 493757855.94
total co2 emission = 304756.33

Solution 2

collection centres

period 1
 facilities = [0 0 0 1 0 0 0]
 total capacity for product 1 = 4500
 total capacity for product 2 = 7500
 total capacity for product 3 = 7500
 total capacity for product 4 = 10500

period 2
 facilities = [1 0 0 1 0 0 0]
 total capacity for product 1 = 5925
 total capacity for product 2 = 9875
 total capacity for product 3 = 9875
 total capacity for product 4 = 13825

period 3
 facilities = [1 0 0 1 0 0 0]
 total capacity for product 1 = 5854
 total capacity for product 2 = 9756
 total capacity for product 3 = 9756
 total capacity for product 4 = 13659

period 4
 facilities = [1 0 0 1 0 0 0]
 total capacity for product 1 = 5786
 total capacity for product 2 = 9643
 total capacity for product 3 = 9643
 total capacity for product 4 = 13501

period 5
 facilities = [0 0 0 1 0 0 0]
 total capacity for product 1 = 4500
 total capacity for product 2 = 7500
 total capacity for product 3 = 7500
 total capacity for product 4 = 10500

disassembly centres

period 1

facilities = [0 0 0 0 1]
 total capacity for product 1 = 13500
 total capacity for product 2 = 19500
 total capacity for product 3 = 22500
 total capacity for product 4 = 30000

period 2
 facilities = [0 0 0 0 1]
 total capacity for product 1 = 13500
 total capacity for product 2 = 19500
 total capacity for product 3 = 22500
 total capacity for product 4 = 30000

period 3
 facilities = [0 0 0 0 1]
 total capacity for product 1 = 13500
 total capacity for product 2 = 19500
 total capacity for product 3 = 22500
 total capacity for product 4 = 30000

period 4
 facilities = [0 0 0 0 1]
 total capacity for product 1 = 13500
 total capacity for product 2 = 19500
 total capacity for product 3 = 22500
 total capacity for product 4 = 30000

period 5
 facilities = [0 0 0 0 1]
 total capacity for product 1 = 13500
 total capacity for product 2 = 19500
 total capacity for product 3 = 22500
 total capacity for product 4 = 30000

recovery centres

period 1
 facilities = [0 1 0]
 total capacity for product 1 = 18000
 total capacity for product 2 = 21000

period 2
 facilities = [0 1 0]
 total capacity for product 1 = 18000
 total capacity for product 2 = 21000

period 3
 facilities = [0 1 0]
 total capacity for product 1 = 18000
 total capacity for product 2 = 21000

period 4
 facilities = [0 1 0]
 total capacity for product 1 = 18000
 total capacity for product 2 = 21000

period 5
 facilities = [0 1 0]

total capacity for product 1 = 18000	alloc_2[1,2,4,5,2] = 6450
total capacity for product 2 = 21000	alloc_2[1,3,4,5,2] = 7350
	alloc_2[1,4,4,5,2] = 9750
prufer number-stage 1	
[1 1 1 1 1 1]	alloc_4[1,1,1,2,2] = 2000
	alloc_4[1,1,2,2,2] = 2000
[1 1 1 1 1 1]	alloc_4[1,1,3,2,2] = 1700
	alloc_4[1,2,1,2,2] = 2000
[1 1 1 1 1 1]	alloc_4[1,2,2,2,2] = 2000
	alloc_4[1,2,3,2,2] = 2300
[1 1 1 1 1 1]	
	alloc_3[1,1,5,2,2] = 24000
[1 1 1 1 1 1]	alloc_3[1,2,5,2,2] = 24000
	alloc_3[1,3,5,2,2] = 5700
prufer number-stage 2	alloc_3[1,4,5,2,2] = 12600
	alloc_3[1,5,5,2,2] = 5700
[11 8 10 7 11 9 7 4 10 5]	alloc_3[1,6,5,2,2] = 6300
	alloc_3[1,7,5,2,2] = 11400
[10 9 7 10 10 11 7 7 7 9]	alloc_3[1,8,5,2,2] = 12600
	alloc_3[1,9,5,2,2] = 12000
[9 10 11 3 11 4 6 9 4 9]	
	spare_part[1][1][5] = 7980
[9 9 8 4 3 3 8 9 7 12]	spare_part[1][2][5] = 10905
	spare_part[1][3][5] = 3270
[2 10 11 10 9 11 7 4 7 11]	spare_part[1][4][5] = 75
	spare_part[1][5][5] = 1320
prufer number-stage 3	spare_part[1][6][5] = 4815
	spare_part[1][7][5] = 2347
[6 7 6 1 8 3]	spare_part[1][9][5] = 6135
[4 6 7 7 4 6]	recyclable[1][1][5] = 9839
	waste[1][1][5] = 7381
[7 5 7 2 7 8]	recyclable[1][2][5] = 10739
	waste[1][2][5] = 8056
[8 2 6 7 8 5]	recyclable[1][3][5] = 2759
	waste[1][3][5] = 2071
[4 6 7 7 6 5]	recyclable[1][4][5] = 3899
	waste[1][4][5] = 2926
prufer number-stage 4	recyclable[1][5][5] = 2159
	waste[1][5][5] = 1621
[4 2 5 3]	recyclable[1][6][5] = 3419
	waste[1][6][5] = 2566
[5 2 5 3]	recyclable[1][7][5] = 4229
	waste[1][7][5] = 3174
[6 1 2 4]	recyclable[1][8][5] = 3899
	waste[1][8][5] = 2926
[4 2 6 3]	recyclable[1][9][5] = 5579
	waste[1][9][5] = 4186
[2 5 3 4]	recyclable[1][10][5] = 9180
	waste[1][10][5] = 1620
	recyclable[1][11][5] = 14535
	waste[1][11][5] = 2565
alloc_1[1,1,1,4] = 4350	recyclable[1][12][5] = 7395
alloc_1[1,2,1,4] = 6450	waste[1][12][5] = 1305
alloc_1[1,3,1,4] = 7350	recyclable[1][13][5] = 3697
alloc_1[1,4,1,4] = 9750	waste[1][13][5] = 653
	recyclable[1][14][5] = 23715
alloc_2[1,1,4,5,2] = 4350	waste[1][14][5] = 4185

```

recyclable[1][15][5] = 3697
waste[1][15][5] = 653
recyclable[1][16][5] = 23715
waste[1][16][5] = 4185
recyclable[1][17][5] = 32002
waste[1][17][5] = 5648

new_part[1][1][2] = 5700
new_part[1][2][2] = 6300
new_part[1][3][2] = 12000
new_part[1][4][2] = 12000
new_part[1][5][2] = 12000

alloc_1[2,1,1,1] = 285
alloc_1[2,1,1,4] = 4500
alloc_1[2,2,1,4] = 7095
alloc_1[2,3,1,1] = 585
alloc_1[2,3,1,4] = 7500
alloc_1[2,4,1,1] = 225
alloc_1[2,4,1,4] = 10500

alloc_2[2,1,1,5,2] = 285
alloc_2[2,1,4,5,2] = 4500
alloc_2[2,2,4,5,2] = 7095
alloc_2[2,3,1,5,2] = 585
alloc_2[2,3,4,5,2] = 7500
alloc_2[2,4,1,5,2] = 225
alloc_2[2,4,4,5,2] = 10500

alloc_4[2,1,1,2,2] = 1900
alloc_4[2,1,2,2,2] = 1900
alloc_4[2,1,3,2,2] = 1615
alloc_4[2,2,1,2,2] = 1900
alloc_4[2,2,2,2,2] = 1900
alloc_4[2,2,3,2,2] = 2185

alloc_3[2,1,5,2,2] = 22800
alloc_3[2,2,5,2,2] = 22800
alloc_3[2,3,5,2,2] = 5415
alloc_3[2,4,5,2,2] = 11970
alloc_3[2,5,5,2,2] = 5415
alloc_3[2,6,5,2,2] = 5985
alloc_3[2,7,5,2,2] = 10830
alloc_3[2,8,5,2,2] = 11970
alloc_3[2,9,5,2,2] = 11400

spare_part[2][1][5] = 12378
spare_part[2][2][5] = 15595
spare_part[2][3][5] = 4452
spare_part[2][4][5] = 1972
spare_part[2][5][5] = 2307
spare_part[2][6][5] = 6241
spare_part[2][7][5] = 4292
spare_part[2][8][5] = 1972
spare_part[2][9][5] = 8548

recyclable[2][1][5] = 10823
waste[2][1][5] = 8119

recyclable[2][2][5] = 11813
waste[2][2][5] = 8862
recyclable[2][3][5] = 3035
waste[2][3][5] = 2278
recyclable[2][4][5] = 4289
waste[2][4][5] = 3219
recyclable[2][5][5] = 2375
waste[2][5][5] = 1783
recyclable[2][6][5] = 3761
waste[2][6][5] = 2823
recyclable[2][7][5] = 4652
waste[2][7][5] = 3491
recyclable[2][8][5] = 4289
waste[2][8][5] = 3219
recyclable[2][9][5] = 6137
waste[2][9][5] = 4605
recyclable[2][10][5] = 10098
waste[2][10][5] = 1782
recyclable[2][11][5] = 15988
waste[2][11][5] = 2822
recyclable[2][12][5] = 8134
waste[2][12][5] = 1436
recyclable[2][13][5] = 4067
waste[2][13][5] = 718
recyclable[2][14][5] = 26086
waste[2][14][5] = 4604
recyclable[2][15][5] = 4067
waste[2][15][5] = 718
recyclable[2][16][5] = 26086
waste[2][16][5] = 4604
recyclable[2][17][5] = 35202
waste[2][17][5] = 6213

new_part[2][1][2] = 5415
new_part[2][2][2] = 5985
new_part[2][3][2] = 11400
new_part[2][4][2] = 11400
new_part[2][5][2] = 11400

alloc_1[3,1,1,1] = 764
alloc_1[3,1,1,4] = 4500
alloc_1[3,2,1,1] = 305
alloc_1[3,2,1,4] = 7500
alloc_1[3,3,1,1] = 1394
alloc_1[3,3,1,4] = 7500
alloc_1[3,4,1,1] = 1298
alloc_1[3,4,1,4] = 10500

alloc_2[3,1,1,5,2] = 764
alloc_2[3,1,4,5,2] = 4500
alloc_2[3,2,1,5,2] = 305
alloc_2[3,2,4,5,2] = 7500
alloc_2[3,3,1,5,2] = 1394
alloc_2[3,3,4,5,2] = 7500
alloc_2[3,4,1,5,2] = 1298
alloc_2[3,4,4,5,2] = 10500

alloc_4[3,1,1,2,2] = 1805

```

```

alloc_4[3,1,2,2,2] = 1805
alloc_4[3,1,3,2,2] = 1534
alloc_4[3,2,1,2,2] = 1805
alloc_4[3,2,2,2,2] = 1805
alloc_4[3,2,3,2,2] = 2076

alloc_3[3,1,5,2,2] = 21660
alloc_3[3,2,5,2,2] = 21660
alloc_3[3,3,5,2,2] = 5144
alloc_3[3,4,5,2,2] = 11372
alloc_3[3,5,5,2,2] = 5144
alloc_3[3,6,5,2,2] = 5686
alloc_3[3,7,5,2,2] = 10288
alloc_3[3,8,5,2,2] = 11372
alloc_3[3,9,5,2,2] = 10830

spare_part[3][1][5] = 17038
spare_part[3][2][5] = 20577
spare_part[3][3][5] = 5710
spare_part[3][4][5] = 3965
spare_part[3][5][5] = 3350
spare_part[3][6][5] = 7763
spare_part[3][7][5] = 6347
spare_part[3][8][5] = 3965
spare_part[3][9][5] = 11114

recyclable[3][1][5] = 11907
waste[3][1][5] = 8931
recyclable[3][2][5] = 12995
waste[3][2][5] = 9748
recyclable[3][3][5] = 3339
waste[3][3][5] = 2506
recyclable[3][4][5] = 4719
waste[3][4][5] = 3540
recyclable[3][5][5] = 2613
waste[3][5][5] = 1962
recyclable[3][6][5] = 4138
waste[3][6][5] = 3105
recyclable[3][7][5] = 5118
waste[3][7][5] = 3840
recyclable[3][8][5] = 4719
waste[3][8][5] = 3540
recyclable[3][9][5] = 6752
waste[3][9][5] = 5065
recyclable[3][10][5] = 11108
waste[3][10][5] = 1961
recyclable[3][11][5] = 17588
waste[3][11][5] = 3104
recyclable[3][12][5] = 8948
waste[3][12][5] = 1580
recyclable[3][13][5] = 4474
waste[3][13][5] = 790
recyclable[3][14][5] = 28696
waste[3][14][5] = 5065
recyclable[3][15][5] = 4474
waste[3][15][5] = 790
recyclable[3][16][5] = 28696
waste[3][16][5] = 5065

recyclable[3][17][5] = 38725
waste[3][17][5] = 6834

new_part[3][1][2] = 5144
new_part[3][2][2] = 5686
new_part[3][3][2] = 10830
new_part[3][4][2] = 10830
new_part[3][5][2] = 10830

alloc_1[4,1,1,1] = 285
alloc_1[4,1,1,4] = 4500
alloc_1[4,2,1,4] = 7095
alloc_1[4,3,1,1] = 585
alloc_1[4,3,1,4] = 7500
alloc_1[4,4,1,1] = 225
alloc_1[4,4,1,4] = 10500

alloc_2[4,1,1,5,2] = 285
alloc_2[4,1,4,5,2] = 4500
alloc_2[4,2,4,5,2] = 7095
alloc_2[4,3,1,5,2] = 585
alloc_2[4,3,4,5,2] = 7500
alloc_2[4,4,1,5,2] = 225
alloc_2[4,4,4,5,2] = 10500

alloc_4[4,1,1,2,2] = 1715
alloc_4[4,1,2,2,2] = 1715
alloc_4[4,1,3,2,2] = 1458
alloc_4[4,2,1,2,2] = 1715
alloc_4[4,2,2,2,2] = 1715
alloc_4[4,2,3,2,2] = 1972

alloc_3[4,1,5,2,2] = 20580
alloc_3[4,2,5,2,2] = 20580
alloc_3[4,3,5,2,2] = 4888
alloc_3[4,4,5,2,2] = 10804
alloc_3[4,5,5,2,2] = 4888
alloc_3[4,6,5,2,2] = 5402
alloc_3[4,7,5,2,2] = 9776
alloc_3[4,8,5,2,2] = 10804
alloc_3[4,9,5,2,2] = 10290

spare_part[4][1][5] = 14598
spare_part[4][2][5] = 17815
spare_part[4][3][5] = 4979
spare_part[4][4][5] = 3138
spare_part[4][5][5] = 2834
spare_part[4][6][5] = 6824
spare_part[4][7][5] = 5346
spare_part[4][8][5] = 3138
spare_part[4][9][5] = 9658

recyclable[4][1][5] = 10823
waste[4][1][5] = 8119
recyclable[4][4][5] = 4289
waste[4][4][5] = 3219
recyclable[4][6][5] = 3761
waste[4][6][5] = 2823

```

```
recyclable[4][7][5] = 4652
waste[4][7][5] = 3491
recyclable[4][8][5] = 4289
waste[4][8][5] = 3219
recyclable[4][9][5] = 6137
waste[4][9][5] = 4605
recyclable[4][10][5] = 10098
waste[4][10][5] = 1782
recyclable[4][11][5] = 15988
waste[4][11][5] = 2822
recyclable[4][12][5] = 8134
waste[4][12][5] = 1436
recyclable[4][13][5] = 4067
waste[4][13][5] = 718
recyclable[4][14][5] = 26086
waste[4][14][5] = 4604
recyclable[4][15][5] = 4067
waste[4][15][5] = 718
recyclable[4][16][5] = 26086
waste[4][16][5] = 4604
recyclable[4][17][5] = 35202
waste[4][17][5] = 6213
```

```
new_part[4][1][2] = 4888
new_part[4][2][2] = 5402
new_part[4][3][2] = 10290
new_part[4][4][2] = 10290
new_part[4][5][2] = 10290
```

```
alloc_1[5,1,1,4] = 4350
alloc_1[5,2,1,4] = 6450
alloc_1[5,3,1,4] = 7350
alloc_1[5,4,1,4] = 9750
```

```
alloc_2[5,1,4,5,2] = 4350
alloc_2[5,2,4,5,2] = 6450
alloc_2[5,3,4,5,2] = 7350
alloc_2[5,4,4,5,2] = 9750
```

```
alloc_4[5,1,1,2,2] = 1629
alloc_4[5,1,2,2,2] = 1629
alloc_4[5,1,3,2,2] = 1385
alloc_4[5,2,1,2,2] = 1629
alloc_4[5,2,2,2,2] = 1629
alloc_4[5,2,3,2,2] = 1873
```

```
alloc_3[5,1,5,2,2] = 19548
alloc_3[5,2,5,2,2] = 19548
alloc_3[5,3,5,2,2] = 4643
alloc_3[5,4,5,2,2] = 10262
alloc_3[5,5,5,2,2] = 4643
alloc_3[5,6,5,2,2] = 5131
alloc_3[5,7,5,2,2] = 9286
alloc_3[5,8,5,2,2] = 10262
alloc_3[5,9,5,2,2] = 9774
```

```
spare_part[5][1][5] = 12432
spare_part[5][2][5] = 15357
```

```
spare_part[5][3][5] = 4327
spare_part[5][4][5] = 2413
spare_part[5][5][5] = 2377
spare_part[5][6][5] = 5984
spare_part[5][7][5] = 4461
spare_part[5][8][5] = 2413
spare_part[5][9][5] = 8361
```

```
recyclable[5][1][5] = 9839
waste[5][1][5] = 7381
recyclable[5][2][5] = 10739
waste[5][2][5] = 8056
recyclable[5][3][5] = 2759
waste[5][3][5] = 2071
recyclable[5][4][5] = 3899
waste[5][4][5] = 2926
recyclable[5][5][5] = 2159
waste[5][5][5] = 1621
recyclable[5][6][5] = 3419
waste[5][6][5] = 2566
recyclable[5][7][5] = 4229
waste[5][7][5] = 3174
recyclable[5][8][5] = 3899
waste[5][8][5] = 2926
recyclable[5][9][5] = 5579
waste[5][9][5] = 4186
recyclable[5][10][5] = 9180
waste[5][10][5] = 1620
recyclable[5][11][5] = 14535
waste[5][11][5] = 2565
recyclable[5][12][5] = 7395
waste[5][12][5] = 1305
recyclable[5][13][5] = 3697
waste[5][13][5] = 653
recyclable[5][14][5] = 23715
waste[5][14][5] = 4185
recyclable[5][15][5] = 3697
waste[5][15][5] = 653
recyclable[5][16][5] = 23715
waste[5][16][5] = 4185
recyclable[5][17][5] = 32002
waste[5][17][5] = 5648
```

```
new_part[5][1][2] = 4643
new_part[5][2][2] = 5131
new_part[5][3][2] = 9774
new_part[5][4][2] = 9774
new_part[5][5][2] = 9774
```

total net revenue = 4586539.52
total energy consumption = 493756454.90
total co2 emission = 304634.50

Solution 9

collection centres

period 1

facilities = [0 0 0 1 0 0 0]
 total capacity for product 1 = 4500
 total capacity for product 2 = 7500
 total capacity for product 3 = 7500
 total capacity for product 4 = 10500

period 2
 facilities = [0 0 0 1 1 0 0]
 total capacity for product 1 = 13500
 total capacity for product 2 = 22500
 total capacity for product 3 = 22500
 total capacity for product 4 = 31500

period 3
 facilities = [1 0 0 1 0 0 0]
 total capacity for product 1 = 5854
 total capacity for product 2 = 9756
 total capacity for product 3 = 9756
 total capacity for product 4 = 13659

period 4
 facilities = [0 0 0 1 1 0 0]
 total capacity for product 1 = 13500
 total capacity for product 2 = 22500
 total capacity for product 3 = 22500
 total capacity for product 4 = 31500

period 5
 facilities = [0 0 0 1 0 0 0]
 total capacity for product 1 = 4500
 total capacity for product 2 = 7500
 total capacity for product 3 = 7500
 total capacity for product 4 = 10500

disassembly centres

period 1
 facilities = [0 0 0 0 1]
 total capacity for product 1 = 13500
 total capacity for product 2 = 19500
 total capacity for product 3 = 22500
 total capacity for product 4 = 30000

period 2
 facilities = [0 0 0 0 1]
 total capacity for product 1 = 13500
 total capacity for product 2 = 19500
 total capacity for product 3 = 22500
 total capacity for product 4 = 30000

period 3
 facilities = [0 0 0 0 1]
 total capacity for product 1 = 13500
 total capacity for product 2 = 19500
 total capacity for product 3 = 22500
 total capacity for product 4 = 30000

period 4

facilities = [0 0 0 0 1]
 total capacity for product 1 = 13500
 total capacity for product 2 = 19500
 total capacity for product 3 = 22500
 total capacity for product 4 = 30000

period 5
 facilities = [0 0 0 0 1]
 total capacity for product 1 = 13500
 total capacity for product 2 = 19500
 total capacity for product 3 = 22500
 total capacity for product 4 = 30000

recovery centres

period 1
 facilities = [0 1 0]
 total capacity for product 1 = 18000
 total capacity for product 2 = 21000

period 2
 facilities = [0 1 0]
 total capacity for product 1 = 18000
 total capacity for product 2 = 21000

period 3
 facilities = [0 1 0]
 total capacity for product 1 = 18000
 total capacity for product 2 = 21000

period 4
 facilities = [0 1 0]
 total capacity for product 1 = 18000
 total capacity for product 2 = 21000

period 5
 facilities = [0 1 0]
 total capacity for product 1 = 18000
 total capacity for product 2 = 21000

prufer number-stage 1

[1 1 1 1 1 1]

[1 1 1 1 1 1]

[1 1 1 1 1 1]

[1 1 1 1 1 1]

[1 1 1 1 1 1]

prufer number-stage 2

[3 8 8 5 10 4 9 9 12 4]

[11 11 1 1 9 11 5 9 4 12]

[10 4 11 9 5 9 10 4 11 4]	spare_part[1][1][5] = 7980
[9 7 10 12 5 2 8 9 5 9]	spare_part[1][2][5] = 10905
[6 3 12 10 10 11 4 9 12 4]	spare_part[1][3][5] = 3270
prufer number-stage 3	spare_part[1][4][5] = 75
[1 7 7 1 8 7]	spare_part[1][5][5] = 1320
[2 6 7 7 4 7]	spare_part[1][6][5] = 4815
[6 3 8 6 6 5]	spare_part[1][7][5] = 2347
[6 4 7 6 3 6]	spare_part[1][8][5] = 75
[6 7 3 5 6 7]	spare_part[1][9][5] = 6135
prufer number-stage 4	
[6 2 1 5]	recyclable[1][1][5] = 9839
[2 6 5 1]	waste[1][1][5] = 7381
[1 5 3 4]	recyclable[1][2][5] = 10739
[2 4 6 2]	waste[1][2][5] = 8056
[4 1 6 3]	recyclable[1][3][5] = 2759
	waste[1][3][5] = 2071
alloc_1[1,1,1,4] = 4350	recyclable[1][4][5] = 3899
alloc_1[1,2,1,4] = 6450	waste[1][4][5] = 2926
alloc_1[1,3,1,4] = 7350	recyclable[1][5][5] = 2159
alloc_1[1,4,1,4] = 9750	waste[1][5][5] = 1621
	recyclable[1][6][5] = 3419
alloc_2[1,1,4,5,2] = 4350	waste[1][6][5] = 2566
alloc_2[1,2,4,5,2] = 6450	recyclable[1][7][5] = 4229
alloc_2[1,3,4,5,2] = 7350	waste[1][7][5] = 3174
alloc_2[1,4,4,5,2] = 9750	recyclable[1][8][5] = 3899
	waste[1][8][5] = 2926
alloc_4[1,1,1,2,2] = 2000	recyclable[1][9][5] = 5579
alloc_4[1,1,2,2,2] = 2000	waste[1][9][5] = 4186
alloc_4[1,1,3,2,2] = 1700	recyclable[1][10][5] = 9180
alloc_4[1,2,1,2,2] = 2000	waste[1][10][5] = 1620
alloc_4[1,2,2,2,2] = 2000	recyclable[1][11][5] = 14535
alloc_4[1,2,3,2,2] = 2300	waste[1][11][5] = 2565
	recyclable[1][12][5] = 7395
alloc_3[1,1,5,2,2] = 24000	waste[1][12][5] = 1305
alloc_3[1,2,5,2,2] = 24000	recyclable[1][13][5] = 3697
alloc_3[1,3,5,2,2] = 5700	waste[1][13][5] = 653
alloc_3[1,4,5,2,2] = 12600	recyclable[1][14][5] = 23715
alloc_3[1,5,5,2,2] = 5700	waste[1][14][5] = 4185
alloc_3[1,6,5,2,2] = 6300	recyclable[1][15][5] = 3697
alloc_3[1,7,5,2,2] = 11400	waste[1][15][5] = 653
alloc_3[1,8,5,2,2] = 12600	recyclable[1][16][5] = 23715
alloc_3[1,9,5,2,2] = 12000	waste[1][16][5] = 4185
	recyclable[1][17][5] = 32002
	waste[1][17][5] = 5648
	new_part[1][1][2] = 5700
	new_part[1][2][2] = 6300
	new_part[1][3][2] = 12000
	new_part[1][4][2] = 12000
	new_part[1][5][2] = 12000
	alloc_1[2,1,1,4] = 4500
	alloc_1[2,1,1,5] = 285
	alloc_1[2,2,1,4] = 7095
	alloc_1[2,3,1,4] = 7500
	alloc_1[2,3,1,5] = 585
	alloc_1[2,4,1,4] = 10500
	alloc_1[2,4,1,5] = 225

```

alloc_2[2,1,4,5,2] = 4500
alloc_2[2,1,5,5,2] = 285
alloc_2[2,2,4,5,2] = 7095
alloc_2[2,3,4,5,2] = 7500
alloc_2[2,3,5,5,2] = 585
alloc_2[2,4,4,5,2] = 10500
alloc_2[2,4,5,5,2] = 225

alloc_4[2,1,1,2,2] = 1900
alloc_4[2,1,2,2,2] = 1900
alloc_4[2,1,3,2,2] = 1615
alloc_4[2,2,1,2,2] = 1900
alloc_4[2,2,2,2,2] = 1900
alloc_4[2,2,3,2,2] = 2185

alloc_3[2,1,5,2,2] = 22800
alloc_3[2,2,5,2,2] = 22800
alloc_3[2,3,5,2,2] = 5415
alloc_3[2,4,5,2,2] = 11970
alloc_3[2,5,5,2,2] = 5415
alloc_3[2,6,5,2,2] = 5985
alloc_3[2,7,5,2,2] = 10830
alloc_3[2,8,5,2,2] = 11970
alloc_3[2,9,5,2,2] = 11400

spare_part[2][1][5] = 12378
spare_part[2][2][5] = 15595
spare_part[2][3][5] = 4452
spare_part[2][4][5] = 1972
spare_part[2][5][5] = 2307
spare_part[2][6][5] = 6241
spare_part[2][7][5] = 4292
spare_part[2][8][5] = 1972
spare_part[2][9][5] = 8548

recyclable[2][1][5] = 10823
waste[2][1][5] = 8119
recyclable[2][2][5] = 11813
waste[2][2][5] = 8862
recyclable[2][3][5] = 3035
waste[2][3][5] = 2278
recyclable[2][5][5] = 2375
waste[2][5][5] = 1783
recyclable[2][6][5] = 3761
waste[2][6][5] = 2823
recyclable[2][7][5] = 4652
waste[2][7][5] = 3491
recyclable[2][8][5] = 4289
waste[2][8][5] = 3219
recyclable[2][10][5] = 10098
waste[2][10][5] = 1782
recyclable[2][12][5] = 8134
waste[2][12][5] = 1436
recyclable[2][13][5] = 4067
waste[2][13][5] = 718
recyclable[2][14][5] = 26086
waste[2][14][5] = 4604

recyclable[2][15][5] = 4067
waste[2][15][5] = 718
recyclable[2][16][5] = 26086
waste[2][16][5] = 4604
recyclable[2][17][5] = 35202
waste[2][17][5] = 6213

new_part[2][1][2] = 5415
new_part[2][2][2] = 5985
new_part[2][3][2] = 11400
new_part[2][4][2] = 11400
new_part[2][5][2] = 11400

alloc_1[3,1,1,1] = 764
alloc_1[3,1,1,4] = 4500
alloc_1[3,2,1,1] = 305
alloc_1[3,2,1,4] = 7500
alloc_1[3,3,1,1] = 1394
alloc_1[3,3,1,4] = 7500
alloc_1[3,4,1,1] = 1298
alloc_1[3,4,1,4] = 10500

alloc_2[3,1,1,5,2] = 764
alloc_2[3,1,4,5,2] = 4500
alloc_2[3,2,1,5,2] = 305
alloc_2[3,2,4,5,2] = 7500
alloc_2[3,3,1,5,2] = 1394
alloc_2[3,3,4,5,2] = 7500
alloc_2[3,4,1,5,2] = 1298
alloc_2[3,4,4,5,2] = 10500

alloc_4[3,1,1,2,2] = 1805
alloc_4[3,1,2,2,2] = 1805
alloc_4[3,1,3,2,2] = 1534
alloc_4[3,2,1,2,2] = 1805
alloc_4[3,2,2,2,2] = 1805
alloc_4[3,2,3,2,2] = 2076

alloc_3[3,1,5,2,2] = 21660
alloc_3[3,2,5,2,2] = 21660
alloc_3[3,3,5,2,2] = 5144
alloc_3[3,4,5,2,2] = 11372
alloc_3[3,5,5,2,2] = 5144
alloc_3[3,6,5,2,2] = 5686
alloc_3[3,7,5,2,2] = 10288
alloc_3[3,8,5,2,2] = 11372
alloc_3[3,9,5,2,2] = 10830

spare_part[3][1][5] = 17038
spare_part[3][2][5] = 20577
spare_part[3][3][5] = 5710
spare_part[3][4][5] = 3965
spare_part[3][5][5] = 3350
spare_part[3][6][5] = 7763
spare_part[3][7][5] = 6347
spare_part[3][8][5] = 3965
spare_part[3][9][5] = 11114

```

```

recyclable[3][1][5] = 11907
waste[3][1][5] = 8931
recyclable[3][2][5] = 12995
waste[3][2][5] = 9748
recyclable[3][3][5] = 3339
waste[3][3][5] = 2506
recyclable[3][4][5] = 4719
waste[3][4][5] = 3540
recyclable[3][5][5] = 2613
waste[3][5][5] = 1962
recyclable[3][6][5] = 4138
waste[3][6][5] = 3105
recyclable[3][7][5] = 5118
waste[3][7][5] = 3840
recyclable[3][8][5] = 4719
waste[3][8][5] = 3540
recyclable[3][9][5] = 6752
waste[3][9][5] = 5065
recyclable[3][10][5] = 11108
waste[3][10][5] = 1961
recyclable[3][11][5] = 17588
waste[3][11][5] = 3104
recyclable[3][12][5] = 8948
waste[3][12][5] = 1580
recyclable[3][13][5] = 4474
waste[3][13][5] = 790
recyclable[3][14][5] = 28696
waste[3][14][5] = 5065
recyclable[3][15][5] = 4474
waste[3][15][5] = 790
recyclable[3][16][5] = 28696
waste[3][16][5] = 5065
recyclable[3][17][5] = 38725
waste[3][17][5] = 6834

new_part[3][1][2] = 5144
new_part[3][2][2] = 5686
new_part[3][3][2] = 10830
new_part[3][4][2] = 10830
new_part[3][5][2] = 10830

alloc_1[4,1,1,4] = 4500
alloc_1[4,1,1,5] = 285
alloc_1[4,2,1,4] = 7095
alloc_1[4,3,1,4] = 7500
alloc_1[4,3,1,5] = 585
alloc_1[4,4,1,4] = 10500
alloc_1[4,4,1,5] = 225

alloc_2[4,1,4,5,2] = 4500
alloc_2[4,1,5,5,2] = 285
alloc_2[4,2,4,5,2] = 7095
alloc_2[4,3,4,5,2] = 7500
alloc_2[4,3,5,5,2] = 585
alloc_2[4,4,4,5,2] = 10500
alloc_2[4,4,5,5,2] = 225

alloc_4[4,1,1,2,2] = 1715

alloc_4[4,1,2,2,2] = 1715
alloc_4[4,1,3,2,2] = 1458
alloc_4[4,2,1,2,2] = 1715
alloc_4[4,2,2,2,2] = 1715
alloc_4[4,2,3,2,2] = 1972

alloc_3[4,1,5,2,2] = 20580
alloc_3[4,2,5,2,2] = 20580
alloc_3[4,3,5,2,2] = 4888
alloc_3[4,4,5,2,2] = 10804
alloc_3[4,5,5,2,2] = 4888
alloc_3[4,6,5,2,2] = 5402
alloc_3[4,7,5,2,2] = 9776
alloc_3[4,8,5,2,2] = 10804
alloc_3[4,9,5,2,2] = 10290

spare_part[4][1][5] = 14598
spare_part[4][2][5] = 17815
spare_part[4][3][5] = 4979
spare_part[4][4][5] = 3138
spare_part[4][5][5] = 2834
spare_part[4][6][5] = 6824
spare_part[4][7][5] = 5346
spare_part[4][8][5] = 3138
spare_part[4][9][5] = 9658

recyclable[4][1][5] = 10823
waste[4][1][5] = 8119
recyclable[4][2][5] = 11813
waste[4][2][5] = 8862
recyclable[4][3][5] = 3035
waste[4][3][5] = 2278
recyclable[4][4][5] = 4289
waste[4][4][5] = 3219
recyclable[4][5][5] = 2375
waste[4][5][5] = 1783
recyclable[4][6][5] = 3761
waste[4][6][5] = 2823
recyclable[4][7][5] = 4652
waste[4][7][5] = 3491
recyclable[4][8][5] = 4289
waste[4][8][5] = 3219
recyclable[4][9][5] = 6137
waste[4][9][5] = 4605
recyclable[4][10][5] = 10098
waste[4][10][5] = 1782
recyclable[4][11][5] = 15988
waste[4][11][5] = 2822
recyclable[4][12][5] = 8134
waste[4][12][5] = 1436
recyclable[4][13][5] = 4067
waste[4][13][5] = 718
recyclable[4][14][5] = 26086
waste[4][14][5] = 4604
recyclable[4][15][5] = 4067
waste[4][15][5] = 718
recyclable[4][16][5] = 26086
waste[4][16][5] = 4604

```

recyclable[4][17][5] = 35202
waste[4][17][5] = 6213

new_part[4][1][2] = 4888
new_part[4][2][2] = 5402
new_part[4][3][2] = 10290
new_part[4][4][2] = 10290
new_part[4][5][2] = 10290

alloc_1[5,1,1,4] = 4350
alloc_1[5,2,1,4] = 6450
alloc_1[5,3,1,4] = 7350
alloc_1[5,4,1,4] = 9750

alloc_2[5,1,4,5,2] = 4350
alloc_2[5,2,4,5,2] = 6450
alloc_2[5,3,4,5,2] = 7350
alloc_2[5,4,4,5,2] = 9750

alloc_4[5,1,1,2,2] = 1629
alloc_4[5,1,2,2,2] = 1629
alloc_4[5,1,3,2,2] = 1385
alloc_4[5,2,1,2,2] = 1629
alloc_4[5,2,2,2,2] = 1629
alloc_4[5,2,3,2,2] = 1873

alloc_3[5,1,5,2,2] = 19548
alloc_3[5,2,5,2,2] = 19548
alloc_3[5,3,5,2,2] = 4643
alloc_3[5,4,5,2,2] = 10262
alloc_3[5,5,5,2,2] = 4643
alloc_3[5,6,5,2,2] = 5131
alloc_3[5,7,5,2,2] = 9286
alloc_3[5,8,5,2,2] = 10262
alloc_3[5,9,5,2,2] = 9774

spare_part[5][1][5] = 12432
spare_part[5][2][5] = 15357
spare_part[5][3][5] = 4327
spare_part[5][4][5] = 2413
spare_part[5][5][5] = 2377
spare_part[5][6][5] = 5984
spare_part[5][7][5] = 4461
spare_part[5][8][5] = 2413
spare_part[5][9][5] = 8361

recyclable[5][1][5] = 9839
waste[5][1][5] = 7381
recyclable[5][2][5] = 10739
waste[5][2][5] = 8056
recyclable[5][3][5] = 2759
waste[5][3][5] = 2071
recyclable[5][4][5] = 3899
waste[5][4][5] = 2926
recyclable[5][5][5] = 2159
waste[5][5][5] = 1621
recyclable[5][6][5] = 3419
waste[5][6][5] = 2566

recyclable[5][7][5] = 4229
waste[5][7][5] = 3174
recyclable[5][8][5] = 3899
waste[5][8][5] = 2926
recyclable[5][9][5] = 5579
waste[5][9][5] = 4186
recyclable[5][10][5] = 9180
waste[5][10][5] = 1620
recyclable[5][11][5] = 14535
waste[5][11][5] = 2565
recyclable[5][12][5] = 7395
waste[5][12][5] = 1305
recyclable[5][13][5] = 3697
waste[5][13][5] = 653
recyclable[5][14][5] = 23715
waste[5][14][5] = 4185
recyclable[5][15][5] = 3697
waste[5][15][5] = 653
recyclable[5][16][5] = 23715
waste[5][16][5] = 4185
recyclable[5][17][5] = 32002
waste[5][17][5] = 5648

new_part[5][1][2] = 4643
new_part[5][2][2] = 5131
new_part[5][3][2] = 9774
new_part[5][4][2] = 9774
new_part[5][5][2] = 9774

total net revenue = 4593188.69
total energy consumption = 493759256.99
total co2 emission = 304878.16

Appendix H

Data for Case Study of Chapter 5

1. Product data

Products considered in the case study are classified into desktop and notebook. New product prices are assumed based on online sources and used product values are assumed to be 20% of new product prices. Only certain parts are taken into account in the study. Product information is presented in the following table (volumetric weights are assumed based on information from Compucare and online sources).

Table H.1: Product data

No.	Product				Module (Part)	
	Type	Average original price (\$)	Average used value (\$)	Volumetric Weight (kg)	Type	Volumetric Weight (kg)
1.	Laptop	3000	600	2.5	1. Main logic board/main processing unit (MLB/MPU)	0.38
					2. Display	2.19
					3. Mass storage	0.25
					4. Optical drive	0.38
					5. Card/Modem	0.1
					6. Battery	0.5
					7. Memory	0.1
2.	Desktop	2300	460	12	8. Main logic board/main processing unit (MLB/MPU)	1.5
					9. Display	8.75
					10. Mass storage	1
					11. Optical drive	1.5
					12. Card/Modem	0.1
					13. Power Supply	0.5
					14. Memory	0.1

2. Supply and demand data

Supply and demand data for the first periods are presented in Tables H.2 and H.3. Data for the subsequent periods are determined using growth rates of computer sales data presented in Table H.4. Data for 2005-2007 is a moving average with

three-period window of the growth rate of computer sales in each country in 2002-2004, taken from Gilliland (2004).

Table H.2: Supply at each primary market

Primary Market (PM) #	Description	Supply at Period 1	
		Laptop	Desktop
1	Beijing	1137	818
2	Shanghai	1137	818
3	Hong Kong	370	605
4	Seoul	984	1993
5	Kuala Lumpur	347	568
6	Singapore	162	688
7	Bangkok	745	1218
8	Taipei	2037	1667
9	Tokyo	1956	1168
10	Osaka	1956	1168
11	Sydney	320	1048
12	Melbourne	320	1048

Table H.3: Demand at each Compucare's RDCs

Part Number	Description	Period 1			
		Singapore	Tokyo	Sydney	Shanghai
1	MLB/MPU laptop	307	109	451	449
2	MLB/MPU desktop	302	539	555	574
3	Display laptop	114	268	312	186
4	Display desktop	152	375	91	88
5	Mass storage laptop	56	105	122	71
6	Mass storage desktop	88	42	40	185
7	Optical drive laptop	108	284	332	151
8	Optical drive desktop	96	97	121	298
9	Card/modem laptop	24	109	126	17
10	Card/modem desktop	22	74	92	37
11	Power supply	74	40	28	106
12	Battery	117	117	117	187
13	Memory laptop	25	25	25	25
14	Memory desktop	25	25	25	25

Table H.4: Growth rate of computer sales

Country	2002	2003	2004	2005	2006	2007
China	1.38	1.10	1.11	1.20	1.14	1.15
ASPAC(guam)	0.99	1.09	1.03	1.03	1.05	1.04
Hong Kong	0.96	1.04	1.05	1.02	1.04	1.03
India	1.15	1.19	1.19	1.18	1.19	1.18
Indonesia	1.16	1.20	1.09	1.15	1.15	1.13
South Korea	1.26	0.88	1.03	1.06	0.99	1.03
Malaysia	1.32	1.06	1.10	1.16	1.10	1.12
Philippines	1.20	1.06	1.05	1.10	1.07	1.08

Singapore	1.07	1.01	1.06	1.05	1.04	1.05
Thailand	1.61	1.52	1.35	1.49	1.46	1.43
Taiwan	1.95	1.07	1.09	1.37	1.18	1.21
Vietnam	1.25	1.17	1.19	1.20	1.19	1.19
Japan	0.92	1.03	1.04	1.00	1.02	1.02
Australia	1.29	1.16	1.07	1.17	1.13	1.13

3. Capacity

Annual capacity of all potential collection and disassembly centres are assumed.

They are assumed to be constant in all periods.

Table H.5: Capacity at each potential collection centre

Collection Centre	Description	Laptop (unit)	Desktop (unit)
CC 1	Beijing (new CC)	6000	6000
CC 2	Shanghai (new CC)	6000	6000
CC 3	Bangkok (new CC)	6000	6000
CC 4	Taipei (new CC)	6000	6000
CC 5	Tokyo (new CC)	6000	6000
CC 6	Osaka (new CC)	6000	6000
CC 7	Sydney (new CC)	6000	6000
CC 8	Melbourne (new CC)	6000	6000
CC 9	Beijing (SP)	3000	3000
CC 10	Shanghai (SP)	3000	3000
CC 11	Hong Kong (SP)	3000	3000
CC 12	Seoul (SP)	3000	3000
CC 13	Kuala Lumpur (SP)	3000	3000
CC 14	Singapore (SP)	3000	3000
CC 15	Bangkok (SP)	3000	3000
CC 16	Taipei (SP)	3000	3000
CC 17	Tokyo (SP)	3000	3000
CC 18	Osaka (SP)	3000	3000
CC 19	Sydney (SP)	3000	3000
CC 20	Melbourne (SP)	3000	3000
CC 21	Seoul (LSH)	6000	6000
CC 22	Tokyo (new DC)	12000	12000
CC 23	Shanghai (new DC)	12000	12000

Table H.6: Capacity at each potential disassembly centre

Disassembly Centre	Description	Laptop (unit)	Desktop (unit)
DC 1	Tokyo (new DC)	12000	12000
DC 2	Shanghai (new DC)	12000	12000
DC 3	Singapore (RDC)	9000	9000
DC 4	Tokyo (RDC)	9000	9000

DC 5	Sydney (RDC)	9000	9000
DC 6	Shanghai (RDC)	9000	9000

4. Other data

4.1. Costs, energy consumption and CO₂ emission during transportation

Transportation costs are calculated based on the distance between the cities, taken from <http://www.geobytes.com>, whereas data for cost, energy consumption and CO₂ emission during transportation and options of transportation modes are the same as in numerical example of Chapter 4. The transportation cost is calculated using Formula (H.1). Energy consumption and CO₂ emission during transportation are also calculated in the similar manner.

Table H.7: Distance data (miles)

Cities	Singapore	Tokyo	Sydney	Shanghai (PRC)
Beijing	2,784	1,304	5,566	668
Shanghai	2,359	1,106	4,898	0
Hong Kong	1,608	1,797	4,857	755
Seoul	2,909	719	5,181	550
Kuala Lumpur	197	3,312	4,116	2,319
Singapore	0	3,311	3,918	2,359
Bangkok	891	2,866	4,687	1,787
Taipei	2,018	1,315	4,519	421
Tokyo	3,311	0	4,869	1,106
Osaka	3,080	251	4,849	856
Sydney	3,918	4,869	0	4,898
Melbourne	3,767	5,095	443	5,005

Table H.8: Consolidation volumes

Facilities	Consolidation Volume (kg)
SPs	400
LSH	750
RDCs	1,500
New CCs	500
New DCs	1,500

$$\text{Transportation cost (\$/kg)} = \frac{\text{distance} \times \text{unit transportation cost}}{\text{consolidation volume}} \quad (\text{H.1})$$

4.2. Cost and price data

All costs and prices are assumed to increase by 5% yearly. Costs and price data for the first period are as follows.

a. Fixed and holding costs

Fixed costs for outsourced facilities consists of only yearly operating costs, while fixed costs for new facilities consist of opening and operating costs.

Table H.9: Fixed costs at new CCs

Facilities	Opening (\$)	Operating (\$)
CC 1	32,000	0
CC 2	32,000	0
CC 3	32,000	0
CC 4	40,000	0
CC 5	40,000	0
CC 6	40,000	0
CC 7	40,000	0
CC 8	40,000	0

Table H.10: Fixed costs at new DCs

Facilities	Opening (\$)	Operating (\$)
DC 1	250,000	0
DC 2	200,000	0

Table H.11: Fixed costs at other CCs

Facilities	Operating cost (\$)
CC 9	16,000
CC 10	16,000
CC 11	20,000
CC 12	20,000
CC 13	16,000
CC 14	20,000
CC 15	16,000
CC 16	20,000
CC 17	20,000
CC 18	20,000
CC 19	20,000
CC 20	20,000
CC 21	30,000
CC 22	0
CC 23	0

Table H.12: Fixed costs at other DCs

Facilities	Operating cost (\$)
DC 5	125,000
DC 6	0
DC 7	0
DC 8	100,000

Based on information from Compucare, variable holding costs only apply to certain facilities as presented in Table H.13.

Table H.13: Variable holding costs at potential DCs

Facilities	Laptop (\$/unit/year)	Desktop (\$/unit/year)
DC 1	0	0
DC 2	0	0
DC 3	0	0
DC 4	12.00	9.20
DC 5	12.00	9.20
DC 6	0	0

b. Purchasing costs

Purchasing costs of new parts are aggregated based on their volumes.

Table H.14: Purchasing cost

Part types	\$/unit
Part1	234.82
Part2	241.06
Part3	289.62
Part4	179.40
Part5	73.26
Part6	53.27
Part7	79.45
Part8	79.45
Part9	38.05
Part10	133.54
Part11	57.31
Part12	54.90
Part13	50.00
Part14	50.00

4.3. Energy consumption at disassembly centres

Energy consumptions at disassembly centres are assumed based on Luo et al. (2001)

and energy consumption for each product type and location is assumed just as in Appendix D.

Table H.15: Energy consumption at potential disassembly centres

Product	(MJ/unit)					
	DC 1	DC 2	DC 3	DC 4	DC 5	DC 6
Laptop	3186	3186	3540	3540	3540	3540
Desktop	2548.8	2548.8	2832	2832	2832	2832

4.4. Recyclability and recoverability fractions

Determined based on information from Compucare and the values are assumed to be constant across periods.

Table H.16: Recyclability & recoverability fractions

Part	Recyclability fraction	Recoverability fraction
Part1	0.75	0.25
Part2	0.75	0.25
Part3	1	0.25
Part4	1	0.25
Part5	1	0.25
Part6	1	0.25
Part7	1	0.25
Part8	1	0.25
Part9	0.75	0.1
Part10	0.75	0.1
Part11	0.75	0.1
Part12	0.75	0.1
Part13	0.75	0.1
Part14	0.75	0.1

Appendix I

The Best Three Solutions of Case Study of Chapter 5

For all solutions:

$$\text{alloc_1}[t,r,i,j] = Q_{rij}^A$$

$$\text{alloc_3}[t,z,k,n,o] = Q_{zkn}^G$$

$$\text{alloc_2}[t,r,j,k,o] = Q_{rjkot}^B$$

$$\text{spare_part}[t,z,k] = Q_{zkt}^H$$

$\text{recyclable}[t,z,k]$ = portion of used module z obtained at disassembly centre k that are sold to recycler in period t .

$\text{waste}[t,z,k]$ = portion of used module z obtained at disassembly centre k that are disposed of in period t .

Total net revenue is in \$, energy consumption is in MJ and CO₂ emission is in KgCE.

Solution 22

collection centres

[14 26 9 4 9 11 4 2 31 6 14 7 11 9 20 2 10 4
12 7 3 31 27 17 4 5 4 15 18 10 7 9 30]

period 1

facilities = [0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 0
0 0 1 0 0]

total capacity for product 1 = 45000

total capacity for product 2 = 45000

[13 4 6 34 4 18 25 11 13 3 10 9 12 6 33 12 14
7 3 10 10 6 35 2 5 10 17 8 4 25 11 6 30]

[16 2 7 6 11 5 25 21 30 3 33 11 13 24 16 5 3 9
16 10 3 2 6 22 3 6 13 2 3 3 8 11]

period 2

facilities = [0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 0
0 0 1 0 0]

total capacity for product 1 = 45000

total capacity for product 2 = 45000

prufer number-stage 2

[25 25 26 25 27 24 27 25 29 25 25 13 26 25
13 22 24 27 28 25 29 25 27 18 25 7 29]

[11 29 29 26 25 29 27 29 28 20 28 27 26 26
25 27 27 22 29 25 29 7 24 27 28 26 23]

period 3

facilities = [0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 0
0 0 1 0 0]

total capacity for product 1 = 45000

total capacity for product 2 = 45000

[28 29 26 17 28 27 28 28 25 4 26 26 26 27 15
24 24 29 25 25 26 27 27 7 28 12 26]

prufer number-stage 3

disassembly centres

[8 10 9 4 9 3 7 3]

period 1

facilities = [0 1 1 0 1 0]

total capacity for product 1 = 30000

total capacity for product 2 = 30000

[8 3 10 9 10 9 2 2]

[3 8 9 7 1 9 8 4]

period 2

facilities = [0 1 1 0 1 0]

total capacity for product 1 = 30000

total capacity for product 2 = 30000

$\text{alloc_1}[1,1,1,2] = 1137$

$\text{alloc_1}[1,1,2,2] = 1137$

$\text{alloc_1}[1,1,3,11] = 370$

$\text{alloc_1}[1,1,4,21] = 984$

$\text{alloc_1}[1,1,5,13] = 347$

$\text{alloc_1}[1,1,6,14] = 162$

$\text{alloc_1}[1,1,7,3] = 745$

$\text{alloc_1}[1,1,8,4] = 2037$

$\text{alloc_1}[1,1,9,6] = 1956$

$\text{alloc_1}[1,1,10,6] = 1956$

$\text{alloc_1}[1,1,11,7] = 320$

$\text{alloc_1}[1,1,12,7] = 320$

$\text{alloc_1}[1,2,1,2] = 818$

$\text{alloc_1}[1,2,2,2] = 818$

period 3

facilities = [0 1 1 0 1 0]

total capacity for product 1 = 30000

total capacity for product 2 = 30000

prufer number-stage 1

```

alloc_1[1,2,3,11] = 605
alloc_1[1,2,4,21] = 1993
alloc_1[1,2,5,13] = 568
alloc_1[1,2,6,14] = 688
alloc_1[1,2,7,3] = 1218
alloc_1[1,2,8,4] = 1667
alloc_1[1,2,9,6] = 1168
alloc_1[1,2,10,6] = 1168
alloc_1[1,2,11,7] = 1048
alloc_1[1,2,12,7] = 1048

alloc_2[1,1,2,2,2] = 2274
alloc_2[1,1,3,3,3] = 745
alloc_2[1,1,4,2,3] = 2037
alloc_2[1,1,6,2,3] = 3912
alloc_2[1,1,7,5,2] = 640
alloc_2[1,1,11,2,3] = 370
alloc_2[1,1,13,3,3] = 347
alloc_2[1,1,14,3,2] = 162
alloc_2[1,1,21,2,3] = 984
alloc_2[1,2,2,2,2] = 1636
alloc_2[1,2,3,3,3] = 1218
alloc_2[1,2,4,2,3] = 1667
alloc_2[1,2,6,2,3] = 2336
alloc_2[1,2,7,5,2] = 2096
alloc_2[1,2,11,2,3] = 605
alloc_2[1,2,13,3,3] = 568
alloc_2[1,2,14,3,2] = 688
alloc_2[1,2,21,2,2] = 1993

alloc_3[1,1,2,2,3] = 109
alloc_3[1,1,2,3,3] = 285
alloc_3[1,1,2,4,2] = 449
alloc_3[1,1,3,1,2] = 307
alloc_3[1,1,3,3,3] = 6
alloc_3[1,1,5,3,2] = 160
alloc_3[1,2,2,2,3] = 539
alloc_3[1,2,2,4,2] = 574
alloc_3[1,2,3,1,2] = 302
alloc_3[1,2,3,3,3] = 31
alloc_3[1,2,5,3,2] = 524
alloc_3[1,3,2,2,3] = 268
alloc_3[1,3,2,4,2] = 186
alloc_3[1,3,3,1,2] = 114
alloc_3[1,3,3,3,3] = 152
alloc_3[1,3,5,3,2] = 160
alloc_3[1,4,2,2,3] = 375
alloc_3[1,4,2,4,2] = 88
alloc_3[1,4,3,1,2] = 152
alloc_3[1,4,5,3,2] = 91
alloc_3[1,5,2,2,3] = 105
alloc_3[1,5,2,4,2] = 71
alloc_3[1,5,3,1,2] = 56
alloc_3[1,5,5,3,2] = 122
alloc_3[1,6,2,2,3] = 42
alloc_3[1,6,2,4,2] = 185
alloc_3[1,6,3,1,2] = 88
alloc_3[1,6,5,3,2] = 40

alloc_3[1,7,2,2,3] = 284
alloc_3[1,7,2,4,2] = 151
alloc_3[1,7,3,1,2] = 108
alloc_3[1,7,3,3,3] = 172
alloc_3[1,7,5,3,2] = 160
alloc_3[1,8,2,2,3] = 97
alloc_3[1,8,2,4,2] = 298
alloc_3[1,8,3,1,2] = 96
alloc_3[1,8,5,3,2] = 121
alloc_3[1,9,2,2,3] = 109
alloc_3[1,9,2,4,2] = 17
alloc_3[1,9,3,1,2] = 24
alloc_3[1,9,3,3,3] = 62
alloc_3[1,9,5,3,2] = 64
alloc_3[1,10,2,2,3] = 74
alloc_3[1,10,2,4,2] = 37
alloc_3[1,10,3,1,2] = 22
alloc_3[1,10,5,3,2] = 92
alloc_3[1,11,2,2,3] = 40
alloc_3[1,11,2,4,2] = 106
alloc_3[1,11,3,1,2] = 74
alloc_3[1,11,5,3,2] = 28
alloc_3[1,12,2,2,3] = 117
alloc_3[1,12,2,3,3] = 45
alloc_3[1,12,2,4,2] = 187
alloc_3[1,12,3,1,2] = 117
alloc_3[1,12,3,3,3] = 8
alloc_3[1,12,5,3,2] = 64
alloc_3[1,13,2,2,3] = 25
alloc_3[1,13,2,4,2] = 25
alloc_3[1,13,3,1,2] = 25
alloc_3[1,13,5,3,2] = 25
alloc_3[1,14,2,2,3] = 25
alloc_3[1,14,2,4,2] = 25
alloc_3[1,14,3,1,2] = 25
alloc_3[1,14,5,3,2] = 25

spare_part[1][1][2] = 1551
spare_part[1][2][2] = 946
spare_part[1][2][3] = 285
spare_part[1][3][2] = 1940
spare_part[1][3][3] = 47
spare_part[1][4][2] = 1596
spare_part[1][4][3] = 466
spare_part[1][4][5] = 433
spare_part[1][5][2] = 2218
spare_part[1][5][3] = 257
spare_part[1][5][5] = 38
spare_part[1][6][2] = 1832
spare_part[1][6][3] = 530
spare_part[1][6][5] = 484
spare_part[1][7][2] = 1959
spare_part[1][7][3] = 33
spare_part[1][8][2] = 1664
spare_part[1][8][3] = 522
spare_part[1][8][5] = 403
spare_part[1][9][2] = 831
spare_part[1][9][3] = 39

```

```

spare_part[1][10][2] = 712
spare_part[1][10][3] = 225
spare_part[1][10][5] = 117
spare_part[1][11][2] = 677
spare_part[1][11][3] = 173
spare_part[1][11][5] = 181
spare_part[1][12][2] = 608
spare_part[1][13][2] = 907
spare_part[1][13][3] = 100
spare_part[1][13][5] = 39
spare_part[1][14][2] = 773
spare_part[1][14][3] = 222
spare_part[1][14][5] = 184

```

```

recyclable[1][1][2] = 4788
waste[1][1][2] = 2395
recyclable[1][1][3] = 627
waste[1][1][3] = 314
recyclable[1][1][5] = 320
waste[1][1][5] = 160
recyclable[1][2][2] = 4118
waste[1][2][2] = 2060
recyclable[1][2][3] = 1237
waste[1][2][3] = 619
recyclable[1][2][5] = 1048
waste[1][2][5] = 524
recyclable[1][3][2] = 7182
waste[1][3][2] = 1
recyclable[1][3][3] = 940
waste[1][3][3] = 1
recyclable[1][3][5] = 480
waste[1][3][5] = 0
recyclable[1][4][2] = 6177
waste[1][4][2] = 1
recyclable[1][4][3] = 1855
waste[1][4][3] = 1
recyclable[1][4][5] = 1572
waste[1][4][5] = 0
recyclable[1][5][2] = 7182
waste[1][5][2] = 1
recyclable[1][5][3] = 940
waste[1][5][3] = 1
recyclable[1][5][5] = 480
waste[1][5][5] = 0
recyclable[1][6][2] = 6177
waste[1][6][2] = 1
recyclable[1][6][3] = 1855
waste[1][6][3] = 1
recyclable[1][6][5] = 1572
waste[1][6][5] = 0
recyclable[1][7][2] = 7182
waste[1][7][2] = 1
recyclable[1][7][3] = 940
waste[1][7][3] = 1
recyclable[1][7][5] = 480
waste[1][7][5] = 0
recyclable[1][8][2] = 6177
waste[1][8][2] = 1

```

```

recyclable[1][8][3] = 1855
waste[1][8][3] = 1
recyclable[1][8][5] = 1572
waste[1][8][5] = 0
recyclable[1][9][2] = 6225
waste[1][9][2] = 2395
recyclable[1][9][3] = 815
waste[1][9][3] = 314
recyclable[1][9][5] = 416
waste[1][9][5] = 160
recyclable[1][10][2] = 5354
waste[1][10][2] = 2060
recyclable[1][10][3] = 1608
waste[1][10][3] = 619
recyclable[1][10][5] = 1362
waste[1][10][5] = 525
recyclable[1][11][2] = 5354
waste[1][11][2] = 2060
recyclable[1][11][3] = 1608
waste[1][11][3] = 619
recyclable[1][11][5] = 1362
waste[1][11][5] = 525
recyclable[1][12][2] = 6225
waste[1][12][2] = 2395
recyclable[1][12][3] = 815
waste[1][12][3] = 314
recyclable[1][12][5] = 416
waste[1][12][5] = 160
recyclable[1][13][2] = 6225
waste[1][13][2] = 2395
recyclable[1][13][3] = 815
waste[1][13][3] = 314
recyclable[1][13][5] = 416
waste[1][13][5] = 160
recyclable[1][14][2] = 5354
waste[1][14][2] = 2060
recyclable[1][14][3] = 1608
waste[1][14][3] = 619
recyclable[1][14][5] = 1362
waste[1][14][5] = 525

```

```

alloc_1[2,1,1,2] = 1292
alloc_1[2,1,2,2] = 1292
alloc_1[2,1,3,11] = 383
alloc_1[2,1,4,21] = 974
alloc_1[2,1,5,13] = 383
alloc_1[2,1,6,14] = 168
alloc_1[2,1,7,3] = 1083
alloc_1[2,1,8,4] = 2398
alloc_1[2,1,9,6] = 1966
alloc_1[2,1,10,6] = 1966
alloc_1[2,1,11,7] = 363
alloc_1[2,1,12,7] = 363
alloc_1[2,2,1,2] = 768
alloc_1[2,2,2,2] = 768
alloc_1[2,2,3,11] = 612
alloc_1[2,2,4,21] = 1853
alloc_1[2,2,5,13] = 516

```

```

alloc_1[2,2,6,14] = 682
alloc_1[2,2,7,3] = 1007
alloc_1[2,2,8,4] = 1269
alloc_1[2,2,9,6] = 1111
alloc_1[2,2,10,6] = 1111
alloc_1[2,2,11,7] = 1001
alloc_1[2,2,12,7] = 1001

alloc_2[2,1,2,2,2] = 2584
alloc_2[2,1,3,3,3] = 1083
alloc_2[2,1,4,2,3] = 2398
alloc_2[2,1,6,2,3] = 3932
alloc_2[2,1,7,5,2] = 726
alloc_2[2,1,11,2,3] = 383
alloc_2[2,1,13,3,3] = 383
alloc_2[2,1,14,3,2] = 168
alloc_2[2,1,21,2,3] = 974
alloc_2[2,2,2,2,2] = 1536
alloc_2[2,2,3,3,3] = 1007
alloc_2[2,2,4,2,3] = 1269
alloc_2[2,2,6,2,3] = 2222
alloc_2[2,2,7,5,2] = 2002
alloc_2[2,2,11,2,3] = 612
alloc_2[2,2,13,3,3] = 516
alloc_2[2,2,14,3,2] = 682
alloc_2[2,2,21,2,3] = 1853

alloc_3[2,1,2,2,3] = 110
alloc_3[2,1,2,3,3] = 282
alloc_3[2,1,2,4,2] = 492
alloc_3[2,1,3,1,2] = 359
alloc_3[2,1,3,3,3] = 49
alloc_3[2,1,5,3,2] = 181
alloc_3[2,2,2,2,3] = 513
alloc_3[2,2,2,4,2] = 521
alloc_3[2,2,3,1,2] = 287
alloc_3[2,2,3,3,3] = 30
alloc_3[2,2,5,3,2] = 500
alloc_3[2,3,2,2,3] = 269
alloc_3[2,3,2,4,2] = 203
alloc_3[2,3,3,1,2] = 33
alloc_3[2,3,3,3,3] = 173
alloc_3[2,3,5,3,2] = 181
alloc_3[2,4,2,2,3] = 357
alloc_3[2,4,2,4,2] = 79
alloc_3[2,4,3,1,2] = 138
alloc_3[2,4,5,3,2] = 87
alloc_3[2,5,2,2,3] = 105
alloc_3[2,5,2,4,2] = 78
alloc_3[2,5,3,1,2] = 64
alloc_3[2,5,5,3,2] = 138
alloc_3[2,6,2,2,3] = 40
alloc_3[2,6,2,4,2] = 169
alloc_3[2,6,3,1,2] = 83
alloc_3[2,6,5,3,2] = 39
alloc_3[2,7,2,2,3] = 286
alloc_3[2,7,2,4,2] = 166
alloc_3[2,7,3,1,2] = 129

alloc_3[2,7,3,3,3] = 195
alloc_3[2,7,5,3,2] = 181
alloc_3[2,8,2,2,3] = 92
alloc_3[2,8,2,4,2] = 275
alloc_3[2,8,3,1,2] = 90
alloc_3[2,8,5,3,2] = 116
alloc_3[2,9,2,2,3] = 109
alloc_3[2,9,2,4,2] = 19
alloc_3[2,9,3,1,2] = 28
alloc_3[2,9,3,3,3] = 71
alloc_3[2,9,5,3,2] = 72
alloc_3[2,10,2,2,3] = 71
alloc_3[2,10,2,4,2] = 33
alloc_3[2,10,3,1,2] = 21
alloc_3[2,10,5,3,2] = 88
alloc_3[2,11,2,2,3] = 38
alloc_3[2,11,2,4,2] = 97
alloc_3[2,11,3,1,2] = 69
alloc_3[2,11,5,3,2] = 27
alloc_3[2,12,2,2,3] = 136
alloc_3[2,12,2,3,3] = 37
alloc_3[2,12,2,4,2] = 205
alloc_3[2,12,3,1,2] = 136
alloc_3[2,12,3,3,3] = 27
alloc_3[2,12,5,3,2] = 72
alloc_3[2,13,2,2,3] = 23
alloc_3[2,13,2,4,2] = 23
alloc_3[2,13,3,1,2] = 23
alloc_3[2,13,5,3,2] = 23
alloc_3[2,14,2,2,3] = 23
alloc_3[2,14,2,4,2] = 23
alloc_3[2,14,3,1,2] = 23
alloc_3[2,14,5,3,2] = 23

spare_part[2][1][2] = 1683
spare_part[2][2][2] = 839
spare_part[2][2][3] = 234
spare_part[2][3][2] = 2095
spare_part[2][3][3] = 202
spare_part[2][4][2] = 1437
spare_part[2][4][3] = 413
spare_part[2][4][5] = 413
spare_part[2][5][2] = 2384
spare_part[2][5][3] = 344
spare_part[2][5][5] = 43
spare_part[2][6][2] = 1664
spare_part[2][6][3] = 468
spare_part[2][6][5] = 461
spare_part[2][7][2] = 2115
spare_part[2][7][3] = 84
spare_part[2][8][2] = 1506
spare_part[2][8][3] = 461
spare_part[2][8][5] = 384
spare_part[2][9][2] = 899
spare_part[2][9][3] = 64
spare_part[2][10][2] = 645
spare_part[2][10][3] = 199
spare_part[2][10][5] = 112

```

spare_part[2][11][2] = 614
 spare_part[2][11][3] = 151
 spare_part[2][11][5] = 173
 spare_part[2][12][2] = 649
 spare_part[2][13][2] = 981
 spare_part[2][13][3] = 140
 spare_part[2][13][5] = 49
 spare_part[2][14][2] = 703
 spare_part[2][14][3] = 197
 spare_part[2][14][5] = 177

recyclable[2][1][2] = 5135
 waste[2][1][2] = 2569
 recyclable[2][1][3] = 817
 waste[2][1][3] = 409
 recyclable[2][1][5] = 363
 waste[2][1][5] = 182
 recyclable[2][2][2] = 3746
 waste[2][2][2] = 1873
 recyclable[2][2][3] = 1102
 waste[2][2][3] = 552
 recyclable[2][2][5] = 1001
 waste[2][2][5] = 501
 recyclable[2][3][2] = 7703
 waste[2][3][2] = 1
 recyclable[2][3][3] = 1225
 waste[2][3][3] = 1
 recyclable[2][3][5] = 544
 waste[2][3][5] = 1
 recyclable[2][4][2] = 5619
 waste[2][4][2] = 0
 recyclable[2][4][3] = 1653
 waste[2][4][3] = 1
 recyclable[2][4][5] = 1501
 waste[2][4][5] = 1
 recyclable[2][5][2] = 7703
 waste[2][5][2] = 1
 recyclable[2][5][3] = 1225
 waste[2][5][3] = 1
 recyclable[2][5][5] = 544
 waste[2][5][5] = 1
 recyclable[2][6][2] = 5619
 waste[2][6][2] = 0
 recyclable[2][6][3] = 1653
 waste[2][6][3] = 1
 recyclable[2][6][5] = 1501
 waste[2][6][5] = 1
 recyclable[2][7][2] = 7703
 waste[2][7][2] = 1
 recyclable[2][7][3] = 1225
 waste[2][7][3] = 1
 recyclable[2][7][5] = 544
 waste[2][7][5] = 1
 recyclable[2][8][2] = 5619
 waste[2][8][2] = 0
 recyclable[2][8][3] = 1653
 waste[2][8][3] = 1
 recyclable[2][8][5] = 1501

waste[2][8][5] = 1
 recyclable[2][9][2] = 6676
 waste[2][9][2] = 2568
 recyclable[2][9][3] = 1062
 waste[2][9][3] = 409
 recyclable[2][9][5] = 471
 waste[2][9][5] = 183
 recyclable[2][10][2] = 4869
 waste[2][10][2] = 1874
 recyclable[2][10][3] = 1433
 waste[2][10][3] = 552
 recyclable[2][10][5] = 1301
 waste[2][10][5] = 501
 recyclable[2][11][2] = 4869
 waste[2][11][2] = 1874
 recyclable[2][11][3] = 1433
 waste[2][11][3] = 552
 recyclable[2][11][5] = 1301
 waste[2][11][5] = 501
 recyclable[2][12][2] = 6676
 waste[2][12][2] = 2568
 recyclable[2][12][3] = 1062
 waste[2][12][3] = 409
 recyclable[2][12][5] = 471
 waste[2][12][5] = 183
 recyclable[2][13][2] = 6676
 waste[2][13][2] = 2568
 recyclable[2][13][3] = 1062
 waste[2][13][3] = 409
 recyclable[2][13][5] = 471
 waste[2][13][5] = 183
 recyclable[2][14][2] = 4869
 waste[2][14][2] = 1874
 recyclable[2][14][3] = 1433
 waste[2][14][3] = 552
 recyclable[2][14][5] = 1301
 waste[2][14][5] = 501

alloc_1[3,1,1,2] = 1482
 alloc_1[3,1,2,2] = 1482
 alloc_1[3,1,3,1] = 396
 alloc_1[3,1,4,2] = 999
 alloc_1[3,1,5,1] = 429
 alloc_1[3,1,6,1] = 177
 alloc_1[3,1,7,3] = 1554
 alloc_1[3,1,8,4] = 2910
 alloc_1[3,1,9,6] = 1975
 alloc_1[3,1,10,6] = 1975
 alloc_1[3,1,11,7] = 408
 alloc_1[3,1,12,7] = 408
 alloc_1[3,2,1,2] = 768
 alloc_1[3,2,2,2] = 768
 alloc_1[3,2,3,1] = 605
 alloc_1[3,2,4,2] = 1923
 alloc_1[3,2,5,1] = 503
 alloc_1[3,2,6,1] = 686
 alloc_1[3,2,7,3] = 750
 alloc_1[3,2,8,4] = 1198


```

alloc_1[3,2,9,6] = 1036
alloc_1[3,2,10,6] = 1036
alloc_1[3,2,11,7] = 982
alloc_1[3,2,12,7] = 982

alloc_2[3,1,2,2,2] = 2964
alloc_2[3,1,3,3,3] = 1554
alloc_2[3,1,4,2,3] = 2910
alloc_2[3,1,6,2,3] = 3950
alloc_2[3,1,7,5,2] = 816
alloc_2[3,1,11,2,3] = 396
alloc_2[3,1,13,3,3] = 429
alloc_2[3,1,14,3,2] = 177
alloc_2[3,1,21,2,3] = 999
alloc_2[3,2,2,2,2] = 1536
alloc_2[3,2,3,3,3] = 750
alloc_2[3,2,4,2,3] = 1198
alloc_2[3,2,6,2,3] = 2072
alloc_2[3,2,7,5,2] = 1964
alloc_2[3,2,11,2,3] = 605
alloc_2[3,2,13,3,3] = 503
alloc_2[3,2,14,3,2] = 686
alloc_2[3,2,21,2,3] = 1923

alloc_3[3,1,2,2,3] = 110
alloc_3[3,1,2,3,3] = 261
alloc_3[3,1,2,4,2] = 551
alloc_3[3,1,3,1,2] = 428
alloc_3[3,1,3,3,3] = 112
alloc_3[3,1,5,3,2] = 204
alloc_3[3,2,2,2,3] = 478
alloc_3[3,2,2,4,2] = 520
alloc_3[3,2,3,1,2] = 272
alloc_3[3,2,3,3,3] = 29
alloc_3[3,2,5,3,2] = 491
alloc_3[3,3,2,2,3] = 271
alloc_3[3,3,2,4,2] = 226
alloc_3[3,3,3,1,2] = 157
alloc_3[3,3,3,3,3] = 195
alloc_3[3,3,5,3,2] = 204
alloc_3[3,4,2,2,3] = 334
alloc_3[3,4,2,4,2] = 78
alloc_3[3,4,3,1,2] = 128
alloc_3[3,4,5,3,2] = 86
alloc_3[3,5,2,2,3] = 106
alloc_3[3,5,2,4,2] = 88
alloc_3[3,5,3,1,2] = 75
alloc_3[3,5,5,3,2] = 155
alloc_3[3,6,2,2,3] = 38
alloc_3[3,6,2,4,2] = 169
alloc_3[3,6,3,1,2] = 78
alloc_3[3,6,5,3,2] = 38
alloc_3[3,7,2,2,3] = 288
alloc_3[3,7,2,4,2] = 187
alloc_3[3,7,3,1,2] = 155
alloc_3[3,7,3,3,3] = 220
alloc_3[3,7,5,3,2] = 204
alloc_3[3,8,2,2,3] = 86

alloc_3[3,8,2,4,2] = 276
alloc_3[3,8,3,1,2] = 83
alloc_3[3,8,5,3,2] = 113
alloc_3[3,9,2,2,3] = 110
alloc_3[3,9,2,4,2] = 22
alloc_3[3,9,3,1,2] = 32
alloc_3[3,9,3,3,3] = 80
alloc_3[3,9,5,3,2] = 81
alloc_3[3,10,2,2,3] = 66
alloc_3[3,10,2,4,2] = 33
alloc_3[3,10,3,1,2] = 19
alloc_3[3,10,5,3,2] = 86
alloc_3[3,11,2,2,3] = 36
alloc_3[3,11,2,4,2] = 97
alloc_3[3,11,3,1,2] = 65
alloc_3[3,11,5,3,2] = 26
alloc_3[3,12,2,2,3] = 162
alloc_3[3,12,2,3,3] = 27
alloc_3[3,12,2,4,2] = 229
alloc_3[3,12,3,1,2] = 162
alloc_3[3,12,3,3,3] = 54
alloc_3[3,12,5,3,2] = 81
alloc_3[3,13,2,2,3] = 23
alloc_3[3,13,2,4,2] = 23
alloc_3[3,13,3,1,2] = 23
alloc_3[3,13,5,3,2] = 23
alloc_3[3,14,2,2,3] = 23
alloc_3[3,14,2,4,2] = 23
alloc_3[3,14,3,1,2] = 23
alloc_3[3,14,5,3,2] = 23

spare_part[3][1][2] = 1882
spare_part[3][2][2] = 835
spare_part[3][2][3] = 183
spare_part[3][3][2] = 2307
spare_part[3][3][3] = 188
spare_part[3][4][2] = 1421
spare_part[3][4][3] = 356
spare_part[3][4][5] = 405
spare_part[3][5][2] = 2610
spare_part[3][5][3] = 465
spare_part[3][5][5] = 49
spare_part[3][6][2] = 1626
spare_part[3][6][3] = 406
spare_part[3][6][5] = 453
spare_part[3][7][2] = 2329
spare_part[3][7][3] = 165
spare_part[3][8][2] = 1471
spare_part[3][8][3] = 401
spare_part[3][8][5] = 378
spare_part[3][9][2] = 989
spare_part[3][9][3] = 104
spare_part[3][10][2] = 634
spare_part[3][10][3] = 174
spare_part[3][10][5] = 110
spare_part[3][11][2] = 600
spare_part[3][11][3] = 128
spare_part[3][11][5] = 170

```

spare_part[3][12][2] = 703
 spare_part[3][13][2] = 1075
 spare_part[3][13][3] = 193
 spare_part[3][13][5] = 58
 spare_part[3][14][2] = 687
 spare_part[3][14][3] = 170
 spare_part[3][14][5] = 173

 recyclable[3][1][2] = 5609
 waste[3][1][2] = 2806
 recyclable[3][1][3] = 1080
 waste[3][1][3] = 540
 recyclable[3][1][5] = 408
 waste[3][1][5] = 204
 recyclable[3][2][2] = 3667
 waste[3][2][2] = 1834
 recyclable[3][2][3] = 969
 waste[3][2][3] = 486
 recyclable[3][2][5] = 982
 waste[3][2][5] = 491
 recyclable[3][3][2] = 8414
 waste[3][3][2] = 1
 recyclable[3][3][3] = 1620
 waste[3][3][3] = 0
 recyclable[3][3][5] = 612
 waste[3][3][5] = 0
 recyclable[3][4][2] = 5500
 waste[3][4][2] = 1
 recyclable[3][4][3] = 1454
 waste[3][4][3] = 1
 recyclable[3][4][5] = 1473
 waste[3][4][5] = 0
 recyclable[3][5][2] = 8414
 waste[3][5][2] = 1
 recyclable[3][5][3] = 1620
 waste[3][5][3] = 0
 recyclable[3][5][5] = 612
 waste[3][5][5] = 0
 recyclable[3][6][2] = 5500
 waste[3][6][2] = 1
 recyclable[3][6][3] = 1454
 waste[3][6][3] = 1
 recyclable[3][6][5] = 1473
 waste[3][6][5] = 0
 recyclable[3][7][2] = 8414
 waste[3][7][2] = 1
 recyclable[3][7][3] = 1620
 waste[3][7][3] = 0
 recyclable[3][7][5] = 612
 waste[3][7][5] = 0
 recyclable[3][8][2] = 5500
 waste[3][8][2] = 1
 recyclable[3][8][3] = 1454
 waste[3][8][3] = 1
 recyclable[3][8][5] = 1473
 waste[3][8][5] = 0
 recyclable[3][9][2] = 7292
 waste[3][9][2] = 2806

recyclable[3][9][3] = 1404
 waste[3][9][3] = 540
 recyclable[3][9][5] = 530
 waste[3][9][5] = 205
 recyclable[3][10][2] = 4767
 waste[3][10][2] = 1834
 recyclable[3][10][3] = 1260
 waste[3][10][3] = 486
 recyclable[3][10][5] = 1276
 waste[3][10][5] = 492
 recyclable[3][11][2] = 4767
 waste[3][11][2] = 1834
 recyclable[3][11][3] = 1260
 waste[3][11][3] = 486
 recyclable[3][11][5] = 1276
 waste[3][11][5] = 492
 recyclable[3][12][2] = 7292
 waste[3][12][2] = 2806
 recyclable[3][12][3] = 1404
 waste[3][12][3] = 540
 recyclable[3][12][5] = 530
 waste[3][12][5] = 205
 recyclable[3][13][2] = 7292
 waste[3][13][2] = 2806
 recyclable[3][13][3] = 1404
 waste[3][13][3] = 540
 recyclable[3][13][5] = 530
 waste[3][13][5] = 205
 recyclable[3][14][2] = 4767
 waste[3][14][2] = 1834
 recyclable[3][14][3] = 1260
 waste[3][14][3] = 486
 recyclable[3][14][5] = 1276
 waste[3][14][5] = 492

total net revenue = 4239259.00
total energy consumption = 229311518.00
total co2 emission = 3864761.00

Solution 52

collection centres

period 1
 facilities = [0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1
 0 0 1 0 0]
 total capacity for product 1 = 48000
 total capacity for product 2 = 48000

period 2
 facilities = [0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 0
 0 0 1 0 0]
 total capacity for product 1 = 45000
 total capacity for product 2 = 45000

period 3
 facilities = [0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 0
 0 0 1 0 0]

total capacity for product 1 = 45000
total capacity for product 2 = 45000

disassembly centres

period 1

facilities = [0 1 1 0 1 0]

total capacity for product 1 = 30000

total capacity for product 2 = 30000

period 2

facilities = [0 1 1 1 1 0]

total capacity for product 1 = 39000

total capacity for product 2 = 39000

period 3

facilities = [0 1 1 1 1 0]

total capacity for product 1 = 39000

total capacity for product 2 = 39000

prufer number-stage 1

[33 5 2 31 4 4 12 25 32 8 11 7 3 2 9 25 8 3 29
8 2 9 8 17 6 3 15 4 26 10 24 9 30]

[4 23 31 34 10 33 5 10 12 1 8 4 7 10 4 34 1 35
2 14 34 2 8 10 2 20 21 24 2 7 3 5 1]

[10 28 12 6 9 27 3 6 9 12 15 5 5 3 21 1 7 9 25
7 30 4 10 4 14 28 3 8 31 20 12 25 8]

prufer number-stage 2

[25 25 26 25 27 29 5 25 29 25 25 13 26 25 28
22 24 25 28 25 17 25 27 25 25 7 29]

[11 29 27 26 25 29 27 29 28 20 12 27 26 26
25 29 29 29 25 25 7 24 27 28 26 23]

[27 29 26 5 25 27 23 28 26 25 26 26 26 28
15 24 25 27 25 26 15 27 24 28 12 26]

prufer number-stage 3

[8 10 9 9 3 9 4 6]

[8 3 4 9 8 9 8 2]

[3 8 9 9 4 9 2 9]

alloc_1[1,1,1,2] = 1137

alloc_1[1,1,2,2] = 1137

alloc_1[1,1,3,11] = 370

alloc_1[1,1,4,21] = 984

alloc_1[1,1,5,13] = 347

alloc_1[1,1,6,14] = 162

alloc_1[1,1,7,3] = 745

alloc_1[1,1,8,4] = 2037

alloc_1[1,1,9,6] = 912

alloc_1[1,1,9,18] = 1044

alloc_1[1,1,10,18] = 1956

alloc_1[1,1,11,7] = 320

alloc_1[1,1,12,7] = 320

alloc_1[1,2,1,2] = 818

alloc_1[1,2,2,2] = 818

alloc_1[1,2,3,11] = 605

alloc_1[1,2,4,21] = 1993

alloc_1[1,2,5,13] = 568

alloc_1[1,2,6,14] = 688

alloc_1[1,2,7,3] = 1218

alloc_1[1,2,8,4] = 1667

alloc_1[1,2,9,18] = 1168

alloc_1[1,2,10,18] = 1168

alloc_1[1,2,11,7] = 1048

alloc_1[1,2,12,7] = 1048

alloc_2[1,1,2,2,2] = 2274

alloc_2[1,1,3,3,3] = 745

alloc_2[1,1,4,2,3] = 2037

alloc_2[1,1,6,2,3] = 912

alloc_2[1,1,7,5,2] = 640

alloc_2[1,1,11,2,3] = 370

alloc_2[1,1,13,3,3] = 347

alloc_2[1,1,14,3,2] = 162

alloc_2[1,1,18,2,3] = 3000

alloc_2[1,1,21,2,3] = 984

alloc_2[1,2,2,2,2] = 1636

alloc_2[1,2,3,3,3] = 1218

alloc_2[1,2,4,2,3] = 1667

alloc_2[1,2,7,5,2] = 2096

alloc_2[1,2,11,2,3] = 605

alloc_2[1,2,13,3,3] = 568

alloc_2[1,2,14,3,2] = 688

alloc_2[1,2,18,2,3] = 2336

alloc_2[1,2,21,2,3] = 1993

alloc_3[1,1,2,2,3] = 109

alloc_3[1,1,2,3,3] = 285

alloc_3[1,1,2,4,2] = 449

alloc_3[1,1,3,1,2] = 307

alloc_3[1,1,3,3,3] = 6

alloc_3[1,1,5,3,2] = 160

alloc_3[1,2,2,2,3] = 539

alloc_3[1,2,2,4,2] = 574

alloc_3[1,2,3,1,2] = 302

alloc_3[1,2,3,3,3] = 31

alloc_3[1,2,5,3,2] = 524

alloc_3[1,3,2,2,3] = 268

alloc_3[1,3,2,4,2] = 186

alloc_3[1,3,3,1,2] = 114

alloc_3[1,3,3,3,3] = 152

alloc_3[1,3,5,3,2] = 160

alloc_3[1,4,2,2,3] = 375

alloc_3[1,4,2,4,2] = 88

alloc_3[1,4,3,1,2] = 152

```

alloc_3[1,4,5,3,2] = 91
alloc_3[1,5,2,2,3] = 105
alloc_3[1,5,2,4,2] = 71
alloc_3[1,5,3,1,2] = 56
alloc_3[1,5,5,3,2] = 122
alloc_3[1,6,2,2,3] = 42
alloc_3[1,6,2,4,2] = 185
alloc_3[1,6,3,1,2] = 88
alloc_3[1,6,5,3,2] = 40
alloc_3[1,7,2,2,3] = 284
alloc_3[1,7,2,4,2] = 151
alloc_3[1,7,3,1,2] = 108
alloc_3[1,7,3,3,3] = 172
alloc_3[1,7,5,3,2] = 160
alloc_3[1,8,2,2,3] = 97
alloc_3[1,8,2,4,2] = 298
alloc_3[1,8,3,1,2] = 96
alloc_3[1,8,5,3,2] = 121
alloc_3[1,9,2,2,3] = 109
alloc_3[1,9,2,4,2] = 17
alloc_3[1,9,3,1,2] = 24
alloc_3[1,9,3,3,3] = 62
alloc_3[1,9,5,3,2] = 64
alloc_3[1,10,2,2,3] = 74
alloc_3[1,10,2,4,2] = 37
alloc_3[1,10,3,1,2] = 22
alloc_3[1,10,5,3,2] = 92
alloc_3[1,11,2,2,3] = 40
alloc_3[1,11,2,4,2] = 106
alloc_3[1,11,3,1,2] = 74
alloc_3[1,11,5,3,2] = 28
alloc_3[1,12,2,2,3] = 117
alloc_3[1,12,2,3,3] = 45
alloc_3[1,12,2,4,2] = 187
alloc_3[1,12,3,1,2] = 117
alloc_3[1,12,3,3,3] = 8
alloc_3[1,12,5,3,2] = 64
alloc_3[1,13,2,2,3] = 25
alloc_3[1,13,2,4,2] = 25
alloc_3[1,13,3,1,2] = 25
alloc_3[1,13,5,3,2] = 25
alloc_3[1,14,2,2,3] = 25
alloc_3[1,14,2,4,2] = 25
alloc_3[1,14,3,1,2] = 25
alloc_3[1,14,5,3,2] = 25

spare_part[1][1][2] = 1551
spare_part[1][2][2] = 946
spare_part[1][2][3] = 285
spare_part[1][3][2] = 1940
spare_part[1][3][3] = 47
spare_part[1][4][2] = 1596
spare_part[1][4][3] = 466
spare_part[1][4][5] = 433
spare_part[1][5][2] = 2218
spare_part[1][5][3] = 257
spare_part[1][5][5] = 38
spare_part[1][6][2] = 1832

spare_part[1][6][3] = 530
spare_part[1][6][5] = 484
spare_part[1][7][2] = 1959
spare_part[1][7][3] = 33
spare_part[1][8][2] = 1664
spare_part[1][8][3] = 522
spare_part[1][8][5] = 403
spare_part[1][9][2] = 831
spare_part[1][9][3] = 39
spare_part[1][10][2] = 712
spare_part[1][10][3] = 225
spare_part[1][10][5] = 117
spare_part[1][11][2] = 677
spare_part[1][11][3] = 173
spare_part[1][11][5] = 181
spare_part[1][12][2] = 608
spare_part[1][13][2] = 907
spare_part[1][13][3] = 100
spare_part[1][13][5] = 39
spare_part[1][14][2] = 773
spare_part[1][14][3] = 222
spare_part[1][14][5] = 184

recyclable[1][1][2] = 4788
waste[1][1][2] = 2395
recyclable[1][1][3] = 627
waste[1][1][3] = 314
recyclable[1][1][5] = 320
waste[1][1][5] = 160
recyclable[1][2][2] = 4118
waste[1][2][2] = 2060
recyclable[1][2][3] = 1237
waste[1][2][3] = 619
recyclable[1][2][5] = 1048
waste[1][2][5] = 524
recyclable[1][3][2] = 7182
waste[1][3][2] = 1
recyclable[1][3][3] = 940
waste[1][3][3] = 1
recyclable[1][3][5] = 480
waste[1][3][5] = 0
recyclable[1][4][2] = 6177
waste[1][4][2] = 1
recyclable[1][4][3] = 1855
waste[1][4][3] = 1
recyclable[1][4][5] = 1572
waste[1][4][5] = 0
recyclable[1][5][2] = 7182
waste[1][5][2] = 1
recyclable[1][5][3] = 940
waste[1][5][3] = 1
recyclable[1][5][5] = 480
waste[1][5][5] = 0
recyclable[1][6][2] = 6177
waste[1][6][2] = 1
recyclable[1][6][3] = 1855
waste[1][6][3] = 1
recyclable[1][6][5] = 1572

```

```

waste[1][6][5] = 0
recyclable[1][7][2] = 7182
waste[1][7][2] = 1
recyclable[1][7][3] = 940
waste[1][7][3] = 1
recyclable[1][7][5] = 480
waste[1][7][5] = 0
recyclable[1][8][2] = 6177
waste[1][8][2] = 1
recyclable[1][8][3] = 1855
waste[1][8][3] = 1
recyclable[1][8][5] = 1572
waste[1][8][5] = 0
recyclable[1][9][2] = 6225
waste[1][9][2] = 2395
recyclable[1][9][3] = 815
waste[1][9][3] = 314
recyclable[1][9][5] = 416
waste[1][9][5] = 160
recyclable[1][10][2] = 5354
waste[1][10][2] = 2060
recyclable[1][10][3] = 1608
waste[1][10][3] = 619
recyclable[1][10][5] = 1362
waste[1][10][5] = 525
recyclable[1][11][2] = 5354
waste[1][11][2] = 2060
recyclable[1][11][3] = 1608
waste[1][11][3] = 619
recyclable[1][11][5] = 1362
waste[1][11][5] = 525
recyclable[1][12][2] = 6225
waste[1][12][2] = 2395
recyclable[1][12][3] = 815
waste[1][12][3] = 314
recyclable[1][12][5] = 416
waste[1][12][5] = 160
recyclable[1][13][2] = 6225
waste[1][13][2] = 2395
recyclable[1][13][3] = 815
waste[1][13][3] = 314
recyclable[1][13][5] = 416
waste[1][13][5] = 160
recyclable[1][14][2] = 5354
waste[1][14][2] = 2060
recyclable[1][14][3] = 1608
waste[1][14][3] = 619
recyclable[1][14][5] = 1362
waste[1][14][5] = 525

alloc_1[2,1,1,2] = 1292
alloc_1[2,1,2,2] = 1292
alloc_1[2,1,3,11] = 383
alloc_1[2,1,4,21] = 974
alloc_1[2,1,5,13] = 383
alloc_1[2,1,6,14] = 168
alloc_1[2,1,7,3] = 1083
alloc_1[2,1,8,4] = 2398

alloc_1[2,1,9,6] = 1966
alloc_1[2,1,10,6] = 1966
alloc_1[2,1,11,7] = 363
alloc_1[2,1,12,7] = 363
alloc_1[2,2,1,2] = 768
alloc_1[2,2,2,2] = 768
alloc_1[2,2,3,11] = 612
alloc_1[2,2,4,21] = 1853
alloc_1[2,2,5,13] = 516
alloc_1[2,2,6,14] = 682
alloc_1[2,2,7,3] = 1007
alloc_1[2,2,8,4] = 1269
alloc_1[2,2,9,6] = 1111
alloc_1[2,2,10,6] = 1111
alloc_1[2,2,11,7] = 1001
alloc_1[2,2,12,7] = 1001

alloc_2[2,1,2,2,2] = 2584
alloc_2[2,1,3,3,3] = 1083
alloc_2[2,1,4,2,3] = 2398
alloc_2[2,1,6,2,3] = 3932
alloc_2[2,1,7,5,2] = 726
alloc_2[2,1,11,2,3] = 383
alloc_2[2,1,13,3,3] = 383
alloc_2[2,1,14,3,2] = 168
alloc_2[2,1,21,2,3] = 974
alloc_2[2,2,2,2,2] = 1536
alloc_2[2,2,3,3,3] = 1007
alloc_2[2,2,4,2,3] = 1269
alloc_2[2,2,6,4,2] = 2222
alloc_2[2,2,7,5,2] = 2002
alloc_2[2,2,11,2,3] = 612
alloc_2[2,2,13,3,3] = 516
alloc_2[2,2,14,3,2] = 682
alloc_2[2,2,21,2,3] = 1853

alloc_3[2,1,2,2,3] = 110
alloc_3[2,1,2,3,3] = 282
alloc_3[2,1,2,4,2] = 492
alloc_3[2,1,3,1,2] = 359
alloc_3[2,1,3,3,3] = 49
alloc_3[2,1,5,3,2] = 181
alloc_3[2,2,2,4,2] = 521
alloc_3[2,2,3,1,2] = 287
alloc_3[2,2,3,3,3] = 30
alloc_3[2,2,4,2,2] = 513
alloc_3[2,2,5,3,2] = 500
alloc_3[2,3,2,2,3] = 269
alloc_3[2,3,2,4,2] = 203
alloc_3[2,3,3,1,2] = 33
alloc_3[2,3,3,3,3] = 173
alloc_3[2,3,5,3,2] = 181
alloc_3[2,4,2,4,2] = 79
alloc_3[2,4,3,1,2] = 138
alloc_3[2,4,4,2,2] = 357
alloc_3[2,4,5,3,2] = 87
alloc_3[2,5,2,2,3] = 105
alloc_3[2,5,2,4,2] = 78

```

```

alloc_3[2,5,3,1,2] = 64
alloc_3[2,5,5,3,2] = 138
alloc_3[2,6,2,4,2] = 169
alloc_3[2,6,3,1,2] = 83
alloc_3[2,6,4,2,2] = 40
alloc_3[2,6,5,3,2] = 39
alloc_3[2,7,2,2,3] = 286
alloc_3[2,7,2,4,2] = 166
alloc_3[2,7,3,1,2] = 129
alloc_3[2,7,3,3,3] = 195
alloc_3[2,7,5,3,2] = 181
alloc_3[2,8,2,4,2] = 275
alloc_3[2,8,3,1,2] = 90
alloc_3[2,8,4,2,2] = 92
alloc_3[2,8,5,3,2] = 116
alloc_3[2,9,2,2,3] = 109
alloc_3[2,9,2,4,2] = 19
alloc_3[2,9,3,1,2] = 28
alloc_3[2,9,3,3,3] = 71
alloc_3[2,9,5,3,2] = 72
alloc_3[2,10,2,4,2] = 33
alloc_3[2,10,3,1,2] = 21
alloc_3[2,10,4,2,2] = 71
alloc_3[2,10,5,3,2] = 88
alloc_3[2,11,2,4,2] = 97
alloc_3[2,11,3,1,2] = 69
alloc_3[2,11,4,2,2] = 38
alloc_3[2,11,5,3,2] = 27
alloc_3[2,12,2,2,3] = 136
alloc_3[2,12,2,3,3] = 37
alloc_3[2,12,2,4,2] = 205
alloc_3[2,12,3,1,2] = 136
alloc_3[2,12,3,3,3] = 27
alloc_3[2,12,5,3,2] = 72
alloc_3[2,13,2,2,3] = 23
alloc_3[2,13,2,4,2] = 23
alloc_3[2,13,3,1,2] = 23
alloc_3[2,13,5,3,2] = 23
alloc_3[2,14,2,4,2] = 23
alloc_3[2,14,3,1,2] = 23
alloc_3[2,14,4,2,2] = 23
alloc_3[2,14,5,3,2] = 23

spare_part[2][1][2] = 1683
spare_part[2][2][2] = 796
spare_part[2][2][3] = 234
spare_part[2][2][4] = 42
spare_part[2][3][2] = 2095
spare_part[2][3][3] = 202
spare_part[2][4][2] = 1238
spare_part[2][4][3] = 413
spare_part[2][4][4] = 198
spare_part[2][4][5] = 413
spare_part[2][5][2] = 2384
spare_part[2][5][3] = 344
spare_part[2][5][5] = 43
spare_part[2][6][2] = 1148
spare_part[2][6][3] = 468

spare_part[2][6][4] = 515
spare_part[2][6][5] = 461
spare_part[2][7][2] = 2115
spare_part[2][7][3] = 84
spare_part[2][8][2] = 1042
spare_part[2][8][3] = 461
spare_part[2][8][4] = 463
spare_part[2][8][5] = 384
spare_part[2][9][2] = 899
spare_part[2][9][3] = 64
spare_part[2][10][2] = 494
spare_part[2][10][3] = 199
spare_part[2][10][4] = 151
spare_part[2][10][5] = 112
spare_part[2][11][2] = 430
spare_part[2][11][3] = 151
spare_part[2][11][4] = 184
spare_part[2][11][5] = 173
spare_part[2][12][2] = 649
spare_part[2][13][2] = 981
spare_part[2][13][3] = 140
spare_part[2][13][5] = 49
spare_part[2][14][2] = 504
spare_part[2][14][3] = 197
spare_part[2][14][4] = 199
spare_part[2][14][5] = 177

recyclable[2][1][2] = 5135
waste[2][1][2] = 2569
recyclable[2][1][3] = 817
waste[2][1][3] = 409
recyclable[2][1][5] = 363
waste[2][1][5] = 182
recyclable[2][2][2] = 2635
waste[2][2][2] = 1318
recyclable[2][2][3] = 1102
waste[2][2][3] = 552
recyclable[2][2][4] = 1111
waste[2][2][4] = 556
recyclable[2][2][5] = 1001
waste[2][2][5] = 501
recyclable[2][3][2] = 7703
waste[2][3][2] = 1
recyclable[2][3][3] = 1225
waste[2][3][3] = 1
recyclable[2][3][5] = 544
waste[2][3][5] = 1
recyclable[2][4][2] = 3952
waste[2][4][2] = 1
recyclable[2][4][3] = 1653
waste[2][4][3] = 1
recyclable[2][4][4] = 1666
waste[2][4][4] = 1
recyclable[2][4][5] = 1501
waste[2][4][5] = 1
recyclable[2][5][2] = 7703
waste[2][5][2] = 1
recyclable[2][5][3] = 1225

```

```

waste[2][5][3] = 1
recyclable[2][5][5] = 544
waste[2][5][5] = 1
recyclable[2][6][2] = 3952
waste[2][6][2] = 1
recyclable[2][6][3] = 1653
waste[2][6][3] = 1
recyclable[2][6][4] = 1666
waste[2][6][4] = 1
recyclable[2][6][5] = 1501
waste[2][6][5] = 1
recyclable[2][7][2] = 7703
waste[2][7][2] = 1
recyclable[2][7][3] = 1225
waste[2][7][3] = 1
recyclable[2][7][5] = 544
waste[2][7][5] = 1
recyclable[2][8][2] = 3952
waste[2][8][2] = 1
recyclable[2][8][3] = 1653
waste[2][8][3] = 1
recyclable[2][8][4] = 1666
waste[2][8][4] = 1
recyclable[2][8][5] = 1501
waste[2][8][5] = 1
recyclable[2][9][2] = 6676
waste[2][9][2] = 2568
recyclable[2][9][3] = 1062
waste[2][9][3] = 409
recyclable[2][9][5] = 471
waste[2][9][5] = 183
recyclable[2][10][2] = 3425
waste[2][10][2] = 1318
recyclable[2][10][3] = 1433
waste[2][10][3] = 552
recyclable[2][10][4] = 1444
waste[2][10][4] = 556
recyclable[2][10][5] = 1301
waste[2][10][5] = 501
recyclable[2][11][2] = 3425
waste[2][11][2] = 1318
recyclable[2][11][3] = 1433
waste[2][11][3] = 552
recyclable[2][11][4] = 1444
waste[2][11][4] = 556
recyclable[2][11][5] = 1301
waste[2][11][5] = 501
recyclable[2][12][2] = 6676
waste[2][12][2] = 2568
recyclable[2][12][3] = 1062
waste[2][12][3] = 409
recyclable[2][12][5] = 471
waste[2][12][5] = 183
recyclable[2][13][2] = 6676
waste[2][13][2] = 2568
recyclable[2][13][3] = 1062
waste[2][13][3] = 409
recyclable[2][13][5] = 471
waste[2][13][5] = 183
recyclable[2][14][2] = 3425
waste[2][14][2] = 1318
recyclable[2][14][3] = 1433
waste[2][14][3] = 552
recyclable[2][14][4] = 1444
waste[2][14][4] = 556
recyclable[2][14][5] = 1301
waste[2][14][5] = 501

alloc_1[3,1,1,2] = 1482
alloc_1[3,1,2,2] = 1482
alloc_1[3,1,3,1] = 396
alloc_1[3,1,4,2] = 999
alloc_1[3,1,5,1] = 429
alloc_1[3,1,6,1] = 177
alloc_1[3,1,7,3] = 1554
alloc_1[3,1,8,4] = 2910
alloc_1[3,1,9,6] = 1975
alloc_1[3,1,10,6] = 1975
alloc_1[3,1,11,7] = 408
alloc_1[3,1,12,7] = 408
alloc_1[3,2,1,2] = 768
alloc_1[3,2,2,2] = 768
alloc_1[3,2,3,1] = 605
alloc_1[3,2,4,2] = 1923
alloc_1[3,2,5,1] = 503
alloc_1[3,2,6,1] = 686
alloc_1[3,2,7,3] = 750
alloc_1[3,2,8,4] = 1198
alloc_1[3,2,9,6] = 1036
alloc_1[3,2,10,6] = 1036
alloc_1[3,2,11,7] = 982
alloc_1[3,2,12,7] = 982

alloc_2[3,1,2,2,2] = 2964
alloc_2[3,1,3,3,3] = 1554
alloc_2[3,1,4,2,3] = 2910
alloc_2[3,1,6,2,3] = 3950
alloc_2[3,1,7,5,2] = 816
alloc_2[3,1,11,2,3] = 396
alloc_2[3,1,13,3,3] = 429
alloc_2[3,1,14,3,2] = 177
alloc_2[3,1,21,2,3] = 999
alloc_2[3,2,2,2,2] = 1536
alloc_2[3,2,3,3,3] = 750
alloc_2[3,2,4,2,3] = 1198
alloc_2[3,2,6,4,2] = 2072
alloc_2[3,2,7,5,2] = 1964
alloc_2[3,2,11,2,3] = 605
alloc_2[3,2,13,3,3] = 503
alloc_2[3,2,14,3,2] = 686
alloc_2[3,2,21,2,3] = 1923

alloc_3[3,1,2,2,3] = 110
alloc_3[3,1,2,3,3] = 261
alloc_3[3,1,2,4,2] = 551
alloc_3[3,1,3,1,2] = 428

```



```

alloc_3[3,1,3,3,3] = 112
alloc_3[3,1,5,3,2] = 204
alloc_3[3,2,2,4,2] = 520
alloc_3[3,2,3,1,2] = 272
alloc_3[3,2,3,3,3] = 29
alloc_3[3,2,4,2,2] = 478
alloc_3[3,2,5,3,2] = 491
alloc_3[3,3,2,2,3] = 271
alloc_3[3,3,2,4,2] = 226
alloc_3[3,3,3,1,2] = 157
alloc_3[3,3,3,3,3] = 195
alloc_3[3,3,5,3,2] = 204
alloc_3[3,4,2,4,2] = 78
alloc_3[3,4,3,1,2] = 128
alloc_3[3,4,4,2,2] = 334
alloc_3[3,4,5,3,2] = 86
alloc_3[3,5,2,2,3] = 106
alloc_3[3,5,2,4,2] = 88
alloc_3[3,5,3,1,2] = 75
alloc_3[3,5,5,3,2] = 155
alloc_3[3,6,2,4,2] = 169
alloc_3[3,6,3,1,2] = 78
alloc_3[3,6,4,2,2] = 38
alloc_3[3,6,5,3,2] = 38
alloc_3[3,7,2,2,3] = 288
alloc_3[3,7,2,4,2] = 187
alloc_3[3,7,3,1,2] = 155
alloc_3[3,7,3,3,3] = 220
alloc_3[3,7,5,3,2] = 204
alloc_3[3,8,2,4,2] = 276
alloc_3[3,8,3,1,2] = 83
alloc_3[3,8,4,2,2] = 86
alloc_3[3,8,5,3,2] = 113
alloc_3[3,9,2,2,3] = 110
alloc_3[3,9,2,4,2] = 22
alloc_3[3,9,3,1,2] = 32
alloc_3[3,9,3,3,3] = 80
alloc_3[3,9,5,3,2] = 81
alloc_3[3,10,2,4,2] = 33
alloc_3[3,10,3,1,2] = 19
alloc_3[3,10,4,2,2] = 66
alloc_3[3,10,5,3,2] = 86
alloc_3[3,11,2,4,2] = 97
alloc_3[3,11,3,1,2] = 65
alloc_3[3,11,4,2,2] = 36
alloc_3[3,11,5,3,2] = 26
alloc_3[3,12,2,2,3] = 162
alloc_3[3,12,2,3,3] = 27
alloc_3[3,12,2,4,2] = 229
alloc_3[3,12,3,1,2] = 162
alloc_3[3,12,3,3,3] = 54
alloc_3[3,12,5,3,2] = 81
alloc_3[3,13,2,2,3] = 23
alloc_3[3,13,2,4,2] = 23
alloc_3[3,13,3,1,2] = 23
alloc_3[3,13,5,3,2] = 23
alloc_3[3,14,2,4,2] = 23
alloc_3[3,14,3,1,2] = 23
alloc_3[3,14,4,2,2] = 23
alloc_3[3,14,5,3,2] = 23

spare_part[3][1][2] = 1882
spare_part[3][2][2] = 795
spare_part[3][2][3] = 183
spare_part[3][2][4] = 40
spare_part[3][3][2] = 2307
spare_part[3][3][3] = 188
spare_part[3][4][2] = 1237
spare_part[3][4][3] = 356
spare_part[3][4][4] = 184
spare_part[3][4][5] = 405
spare_part[3][5][2] = 2610
spare_part[3][5][3] = 465
spare_part[3][5][5] = 49
spare_part[3][6][2] = 1146
spare_part[3][6][3] = 406
spare_part[3][6][4] = 480
spare_part[3][6][5] = 453
spare_part[3][7][2] = 2329
spare_part[3][7][3] = 165
spare_part[3][8][2] = 1039
spare_part[3][8][3] = 401
spare_part[3][8][4] = 432
spare_part[3][8][5] = 378
spare_part[3][9][2] = 989
spare_part[3][9][3] = 104
spare_part[3][10][2] = 493
spare_part[3][10][3] = 174
spare_part[3][10][4] = 141
spare_part[3][10][5] = 110
spare_part[3][11][2] = 429
spare_part[3][11][3] = 128
spare_part[3][11][4] = 171
spare_part[3][11][5] = 170
spare_part[3][12][2] = 703
spare_part[3][13][2] = 1075
spare_part[3][13][3] = 193
spare_part[3][13][5] = 58
spare_part[3][14][2] = 503
spare_part[3][14][3] = 170
spare_part[3][14][4] = 184
spare_part[3][14][5] = 173

recyclable[3][1][2] = 5609
waste[3][1][2] = 2806
recyclable[3][1][3] = 1080
waste[3][1][3] = 540
recyclable[3][1][5] = 408
waste[3][1][5] = 204
recyclable[3][2][2] = 2631
waste[3][2][2] = 1316
recyclable[3][2][3] = 969
waste[3][2][3] = 486
recyclable[3][2][4] = 1036
waste[3][2][4] = 518
recyclable[3][2][5] = 982

```


waste[3][2][5] = 491
 recyclable[3][3][2] = 8414
 waste[3][3][2] = 1
 recyclable[3][3][3] = 1620
 waste[3][3][3] = 0
 recyclable[3][3][5] = 612
 waste[3][3][5] = 0
 recyclable[3][4][2] = 3946
 waste[3][4][2] = 1
 recyclable[3][4][3] = 1454
 waste[3][4][3] = 1
 recyclable[3][4][4] = 1554
 waste[3][4][4] = 0
 recyclable[3][4][5] = 1473
 waste[3][4][5] = 0
 recyclable[3][5][2] = 8414
 waste[3][5][2] = 1
 recyclable[3][5][3] = 1620
 waste[3][5][3] = 0
 recyclable[3][5][5] = 612
 waste[3][5][5] = 0
 recyclable[3][6][2] = 3946
 waste[3][6][2] = 1
 recyclable[3][6][3] = 1454
 waste[3][6][3] = 1
 recyclable[3][6][4] = 1554
 waste[3][6][4] = 0
 recyclable[3][6][5] = 1473
 waste[3][6][5] = 0
 recyclable[3][7][2] = 8414
 waste[3][7][2] = 1
 recyclable[3][7][3] = 1620
 waste[3][7][3] = 0
 recyclable[3][7][5] = 612
 waste[3][7][5] = 0
 recyclable[3][8][2] = 3946
 waste[3][8][2] = 1
 recyclable[3][8][3] = 1454
 waste[3][8][3] = 1
 recyclable[3][8][4] = 1554
 waste[3][8][4] = 0
 recyclable[3][8][5] = 1473
 waste[3][8][5] = 0
 recyclable[3][9][2] = 7292
 waste[3][9][2] = 2806
 recyclable[3][9][3] = 1404
 waste[3][9][3] = 540
 recyclable[3][9][5] = 530
 waste[3][9][5] = 205
 recyclable[3][10][2] = 3420
 waste[3][10][2] = 1316
 recyclable[3][10][3] = 1260
 waste[3][10][3] = 486
 recyclable[3][10][4] = 1346
 waste[3][10][4] = 519
 recyclable[3][10][5] = 1276
 waste[3][10][5] = 492
 recyclable[3][11][2] = 3420

waste[3][11][2] = 1316
 recyclable[3][11][3] = 1260
 waste[3][11][3] = 486
 recyclable[3][11][4] = 1346
 waste[3][11][4] = 519
 recyclable[3][11][5] = 1276
 waste[3][11][5] = 492
 recyclable[3][12][2] = 7292
 waste[3][12][2] = 2806
 recyclable[3][12][3] = 1404
 waste[3][12][3] = 540
 recyclable[3][12][5] = 530
 waste[3][12][5] = 205
 recyclable[3][13][2] = 7292
 waste[3][13][2] = 2806
 recyclable[3][13][3] = 1404
 waste[3][13][3] = 540
 recyclable[3][13][5] = 530
 waste[3][13][5] = 205
 recyclable[3][14][2] = 3420
 waste[3][14][2] = 1316
 recyclable[3][14][3] = 1260
 waste[3][14][3] = 486
 recyclable[3][14][4] = 1346
 waste[3][14][4] = 519
 recyclable[3][14][5] = 1276
 waste[3][14][5] = 492

total net revenue = 4172601.00
total energy consumption = 228935463.00
total co2 emission = 3200610.00

Solution 55

collection centres

period 1
 facilities = [0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 0
 0 0 1 0 0]
 total capacity for product 1 = 45000
 total capacity for product 2 = 45000

period 2
 facilities = [0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 0
 0 0 1 0 0]
 total capacity for product 1 = 45000
 total capacity for product 2 = 45000

period 3
 facilities = [0 1 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 0
 0 0 1 0 0]
 total capacity for product 1 = 45000
 total capacity for product 2 = 45000

disassembly centres

period 1
 facilities = [0 1 1 0 1 0]

total capacity for product 1 = 30000
total capacity for product 2 = 30000

period 2
facilities = [0 1 1 0 1 0]
total capacity for product 1 = 30000
total capacity for product 2 = 30000

period 3
facilities = [0 1 1 0 1 0]
total capacity for product 1 = 30000
total capacity for product 2 = 30000

prufer number-stage 1

[12 5 12 11 30 4 15 3 7 27 5 1 10 29 11 6 19 6
25 10 9 7 8 6 5 18 10 20 4 15 25 9 30]

[26 2 6 15 2 7 6 22 29 1 7 3 26 10 9 33 8 6 27
11 5 10 6 3 32 26 35 5 11 11 8 32 1]

[2 3 5 11 18 6 29 8 33 10 12 12 31 8 21 5 25 9
5 3 21 5 6 19 3 2 30 5 7 7 24 8 22]

prufer number-stage 2

[25 25 26 25 27 26 5 25 29 25 25 13 26 25 28
22 24 25 28 25 17 25 27 29 25 7 29]

[11 29 27 26 25 29 27 29 28 20 12 27 26 26
25 29 29 29 25 25 7 24 27 28 26 23]

[28 29 7 29 26 27 28 28 25 8 26 26 26 26 24
15 27 29 25 25 26 27 27 7 28 12 26]

prufer number-stage 3

[8 10 8 9 3 9 6 6]

[10 3 8 9 4 8 9 4]

[3 8 9 2 9 2 8 9]

alloc_1[1,1,1,2] = 1137
alloc_1[1,1,2,2] = 1137
alloc_1[1,1,3,11] = 370
alloc_1[1,1,4,21] = 984
alloc_1[1,1,5,13] = 347
alloc_1[1,1,6,14] = 162
alloc_1[1,1,7,3] = 745
alloc_1[1,1,8,4] = 2037
alloc_1[1,1,9,6] = 1956
alloc_1[1,1,10,6] = 1956
alloc_1[1,1,11,7] = 320
alloc_1[1,1,12,7] = 320
alloc_1[1,2,1,2] = 818
alloc_1[1,2,2,2] = 818

alloc_1[1,2,3,11] = 605
alloc_1[1,2,4,21] = 1993
alloc_1[1,2,5,13] = 568
alloc_1[1,2,6,14] = 688
alloc_1[1,2,7,3] = 1218
alloc_1[1,2,8,4] = 1667
alloc_1[1,2,9,6] = 1168
alloc_1[1,2,10,6] = 1168
alloc_1[1,2,11,7] = 1048
alloc_1[1,2,12,7] = 1048

alloc_2[1,1,2,2,2] = 2274
alloc_2[1,1,3,3,3] = 745
alloc_2[1,1,4,2,3] = 2037
alloc_2[1,1,6,2,3] = 3912
alloc_2[1,1,7,5,2] = 640
alloc_2[1,1,11,2,3] = 370
alloc_2[1,1,13,3,3] = 347
alloc_2[1,1,14,3,2] = 162
alloc_2[1,1,21,2,3] = 984
alloc_2[1,2,2,2,2] = 1636
alloc_2[1,2,3,3,3] = 1218
alloc_2[1,2,4,2,3] = 1667
alloc_2[1,2,6,2,3] = 2336
alloc_2[1,2,7,5,2] = 2096
alloc_2[1,2,11,2,3] = 605
alloc_2[1,2,13,3,3] = 568
alloc_2[1,2,14,3,2] = 688
alloc_2[1,2,21,2,3] = 1993

alloc_3[1,1,2,2,3] = 109
alloc_3[1,1,2,3,3] = 285
alloc_3[1,1,2,4,2] = 449
alloc_3[1,1,3,1,2] = 307
alloc_3[1,1,3,3,3] = 6
alloc_3[1,1,5,3,2] = 160
alloc_3[1,2,2,2,3] = 539
alloc_3[1,2,2,4,2] = 574
alloc_3[1,2,3,1,2] = 302
alloc_3[1,2,3,3,3] = 31
alloc_3[1,2,5,3,2] = 524
alloc_3[1,3,2,2,3] = 268
alloc_3[1,3,2,4,2] = 186
alloc_3[1,3,3,1,2] = 114
alloc_3[1,3,3,3,3] = 152
alloc_3[1,3,5,3,2] = 160
alloc_3[1,4,2,2,3] = 375
alloc_3[1,4,2,4,2] = 88
alloc_3[1,4,3,1,2] = 152
alloc_3[1,4,5,3,2] = 91
alloc_3[1,5,2,2,3] = 105
alloc_3[1,5,2,4,2] = 71
alloc_3[1,5,3,1,2] = 56
alloc_3[1,5,5,3,2] = 122
alloc_3[1,6,2,2,3] = 42
alloc_3[1,6,2,4,2] = 185
alloc_3[1,6,3,1,2] = 88
alloc_3[1,6,5,3,2] = 40

```

alloc_3[1,7,2,2,3] = 284
alloc_3[1,7,2,4,2] = 151
alloc_3[1,7,3,1,2] = 108
alloc_3[1,7,3,3,3] = 172
alloc_3[1,7,5,3,2] = 160
alloc_3[1,8,2,2,3] = 97
alloc_3[1,8,2,4,2] = 298
alloc_3[1,8,3,1,2] = 96
alloc_3[1,8,5,3,2] = 121
alloc_3[1,9,2,2,3] = 109
alloc_3[1,9,2,4,2] = 17
alloc_3[1,9,3,1,2] = 24
alloc_3[1,9,3,3,3] = 62
alloc_3[1,9,5,3,2] = 64
alloc_3[1,10,2,2,3] = 74
alloc_3[1,10,2,4,2] = 37
alloc_3[1,10,3,1,2] = 22
alloc_3[1,10,5,3,2] = 92
alloc_3[1,11,2,2,3] = 40
alloc_3[1,11,2,4,2] = 106
alloc_3[1,11,3,1,2] = 74
alloc_3[1,11,5,3,2] = 28
alloc_3[1,12,2,2,3] = 117
alloc_3[1,12,2,3,3] = 45
alloc_3[1,12,2,4,2] = 187
alloc_3[1,12,3,1,2] = 117
alloc_3[1,12,3,3,3] = 8
alloc_3[1,12,5,3,2] = 64
alloc_3[1,13,2,2,3] = 25
alloc_3[1,13,2,4,2] = 25
alloc_3[1,13,3,1,2] = 25
alloc_3[1,13,5,3,2] = 25
alloc_3[1,14,2,2,3] = 25
alloc_3[1,14,2,4,2] = 25
alloc_3[1,14,3,1,2] = 25
alloc_3[1,14,5,3,2] = 25

spare_part[1][1][2] = 1551
spare_part[1][2][2] = 946
spare_part[1][2][3] = 285
spare_part[1][3][2] = 1940
spare_part[1][3][3] = 47
spare_part[1][4][2] = 1596
spare_part[1][4][3] = 466
spare_part[1][4][5] = 433
spare_part[1][5][2] = 2218
spare_part[1][5][3] = 257
spare_part[1][5][5] = 38
spare_part[1][6][2] = 1832
spare_part[1][6][3] = 530
spare_part[1][6][5] = 484
spare_part[1][7][2] = 1959
spare_part[1][7][3] = 33
spare_part[1][8][2] = 1664
spare_part[1][8][3] = 522
spare_part[1][8][5] = 403
spare_part[1][9][2] = 831
spare_part[1][9][3] = 39

spare_part[1][10][2] = 712
spare_part[1][10][3] = 225
spare_part[1][10][5] = 117
spare_part[1][11][2] = 677
spare_part[1][11][3] = 173
spare_part[1][11][5] = 181
spare_part[1][12][2] = 608
spare_part[1][13][2] = 907
spare_part[1][13][3] = 100
spare_part[1][13][5] = 39
spare_part[1][14][2] = 773
spare_part[1][14][3] = 222
spare_part[1][14][5] = 184

recyclable[1][1][2] = 4788
waste[1][1][2] = 2395
recyclable[1][1][3] = 627
waste[1][1][3] = 314
recyclable[1][1][5] = 320
waste[1][1][5] = 160
recyclable[1][2][2] = 4118
waste[1][2][2] = 2060
recyclable[1][2][3] = 1237
waste[1][2][3] = 619
recyclable[1][2][5] = 1048
waste[1][2][5] = 524
recyclable[1][3][2] = 7182
waste[1][3][2] = 1
recyclable[1][3][3] = 940
waste[1][3][3] = 1
recyclable[1][3][5] = 480
waste[1][3][5] = 0
recyclable[1][4][2] = 6177
waste[1][4][2] = 1
recyclable[1][4][3] = 1855
waste[1][4][3] = 1
recyclable[1][4][5] = 1572
waste[1][4][5] = 0
recyclable[1][5][2] = 7182
waste[1][5][2] = 1
recyclable[1][5][3] = 940
waste[1][5][3] = 1
recyclable[1][5][5] = 480
waste[1][5][5] = 0
recyclable[1][6][2] = 6177
waste[1][6][2] = 1
recyclable[1][6][3] = 1855
waste[1][6][3] = 1
recyclable[1][6][5] = 1572
waste[1][6][5] = 0
recyclable[1][7][2] = 7182
waste[1][7][2] = 1
recyclable[1][7][3] = 940
waste[1][7][3] = 1
recyclable[1][7][5] = 480
waste[1][7][5] = 0
recyclable[1][8][2] = 6177
waste[1][8][2] = 1

```

```

recyclable[1][8][3] = 1855
waste[1][8][3] = 1
recyclable[1][8][5] = 1572
waste[1][8][5] = 0
recyclable[1][9][2] = 6225
waste[1][9][2] = 2395
recyclable[1][9][3] = 815
waste[1][9][3] = 314
recyclable[1][9][5] = 416
waste[1][9][5] = 160
recyclable[1][10][2] = 5354
waste[1][10][2] = 2060
recyclable[1][10][3] = 1608
waste[1][10][3] = 619
recyclable[1][10][5] = 1362
waste[1][10][5] = 525
recyclable[1][11][2] = 5354
waste[1][11][2] = 2060
recyclable[1][11][3] = 1608
waste[1][11][3] = 619
recyclable[1][11][5] = 1362
waste[1][11][5] = 525
recyclable[1][12][2] = 6225
waste[1][12][2] = 2395
recyclable[1][12][3] = 815
waste[1][12][3] = 314
recyclable[1][12][5] = 416
waste[1][12][5] = 160
recyclable[1][13][2] = 6225
waste[1][13][2] = 2395
recyclable[1][13][3] = 815
waste[1][13][3] = 314
recyclable[1][13][5] = 416
waste[1][13][5] = 160
recyclable[1][14][2] = 5354
waste[1][14][2] = 2060
recyclable[1][14][3] = 1608
waste[1][14][3] = 619
recyclable[1][14][5] = 1362
waste[1][14][5] = 525

alloc_1[2,1,1,2] = 1292
alloc_1[2,1,2,2] = 1292
alloc_1[2,1,3,11] = 383
alloc_1[2,1,4,21] = 974
alloc_1[2,1,5,13] = 383
alloc_1[2,1,6,14] = 168
alloc_1[2,1,7,3] = 1083
alloc_1[2,1,8,4] = 2398
alloc_1[2,1,9,6] = 1966
alloc_1[2,1,10,6] = 1966
alloc_1[2,1,11,7] = 363
alloc_1[2,1,12,7] = 363
alloc_1[2,2,1,2] = 768
alloc_1[2,2,2,2] = 768
alloc_1[2,2,3,11] = 612
alloc_1[2,2,4,21] = 1853
alloc_1[2,2,5,13] = 516

alloc_1[2,2,6,14] = 682
alloc_1[2,2,7,3] = 1007
alloc_1[2,2,8,4] = 1269
alloc_1[2,2,9,6] = 1111
alloc_1[2,2,10,6] = 1111
alloc_1[2,2,11,7] = 1001
alloc_1[2,2,12,7] = 1001

alloc_2[2,1,2,2,2] = 2584
alloc_2[2,1,3,3,3] = 1083
alloc_2[2,1,4,2,3] = 2398
alloc_2[2,1,6,2,3] = 3932
alloc_2[2,1,7,5,2] = 726
alloc_2[2,1,11,2,3] = 383
alloc_2[2,1,13,3,3] = 383
alloc_2[2,1,14,3,2] = 168
alloc_2[2,1,21,2,3] = 974
alloc_2[2,2,2,2,2] = 1536
alloc_2[2,2,3,3,3] = 1007
alloc_2[2,2,4,2,3] = 1269
alloc_2[2,2,6,2,3] = 2222
alloc_2[2,2,7,5,2] = 2002
alloc_2[2,2,11,2,3] = 612
alloc_2[2,2,13,3,3] = 516
alloc_2[2,2,14,3,2] = 682
alloc_2[2,2,21,2,3] = 1853

alloc_3[2,1,2,2,3] = 110
alloc_3[2,1,2,3,3] = 282
alloc_3[2,1,2,4,2] = 492
alloc_3[2,1,3,1,2] = 359
alloc_3[2,1,3,3,3] = 49
alloc_3[2,1,5,3,2] = 181
alloc_3[2,2,2,2,3] = 513
alloc_3[2,2,2,4,2] = 521
alloc_3[2,2,3,1,2] = 287
alloc_3[2,2,3,3,3] = 30
alloc_3[2,2,5,3,2] = 500
alloc_3[2,3,2,2,3] = 269
alloc_3[2,3,2,4,2] = 203
alloc_3[2,3,3,1,2] = 33
alloc_3[2,3,3,3,3] = 173
alloc_3[2,3,5,3,2] = 181
alloc_3[2,4,2,2,3] = 357
alloc_3[2,4,2,4,2] = 79
alloc_3[2,4,3,1,2] = 138
alloc_3[2,4,5,3,2] = 87
alloc_3[2,5,2,2,3] = 105
alloc_3[2,5,2,4,2] = 78
alloc_3[2,5,3,1,2] = 64
alloc_3[2,5,5,3,2] = 138
alloc_3[2,6,2,2,3] = 40
alloc_3[2,6,2,4,2] = 169
alloc_3[2,6,3,1,2] = 83
alloc_3[2,6,5,3,2] = 39
alloc_3[2,7,2,2,3] = 286
alloc_3[2,7,2,4,2] = 166
alloc_3[2,7,3,1,2] = 129

```

```

alloc_3[2,7,3,3,3] = 195
alloc_3[2,7,5,3,2] = 181
alloc_3[2,8,2,2,3] = 92
alloc_3[2,8,2,4,2] = 275
alloc_3[2,8,3,1,2] = 90
alloc_3[2,8,5,3,2] = 116
alloc_3[2,9,2,2,3] = 109
alloc_3[2,9,2,4,2] = 19
alloc_3[2,9,3,1,2] = 28
alloc_3[2,9,3,3,3] = 71
alloc_3[2,9,5,3,2] = 72
alloc_3[2,10,2,2,3] = 71
alloc_3[2,10,2,4,2] = 33
alloc_3[2,10,3,1,2] = 21
alloc_3[2,10,5,3,2] = 88
alloc_3[2,11,2,2,3] = 38
alloc_3[2,11,2,4,2] = 97
alloc_3[2,11,3,1,2] = 69
alloc_3[2,11,5,3,2] = 27
alloc_3[2,12,2,2,3] = 136
alloc_3[2,12,2,3,3] = 37
alloc_3[2,12,2,4,2] = 205
alloc_3[2,12,3,1,2] = 136
alloc_3[2,12,3,3,3] = 27
alloc_3[2,12,5,3,2] = 72
alloc_3[2,13,2,2,3] = 23
alloc_3[2,13,2,4,2] = 23
alloc_3[2,13,3,1,2] = 23
alloc_3[2,13,5,3,2] = 23
alloc_3[2,14,2,2,3] = 23
alloc_3[2,14,2,4,2] = 23
alloc_3[2,14,3,1,2] = 23
alloc_3[2,14,5,3,2] = 23

spare_part[2][1][2] = 1683
spare_part[2][2][2] = 839
spare_part[2][2][3] = 234
spare_part[2][3][2] = 2095
spare_part[2][3][3] = 202
spare_part[2][4][2] = 1437
spare_part[2][4][3] = 413
spare_part[2][4][5] = 413
spare_part[2][5][2] = 2384
spare_part[2][5][3] = 344
spare_part[2][5][5] = 43
spare_part[2][6][2] = 1664
spare_part[2][6][3] = 468
spare_part[2][6][5] = 461
spare_part[2][7][2] = 2115
spare_part[2][7][3] = 84
spare_part[2][8][2] = 1506
spare_part[2][8][3] = 461
spare_part[2][8][5] = 384
spare_part[2][9][2] = 899
spare_part[2][9][3] = 64
spare_part[2][10][2] = 645
spare_part[2][10][3] = 199
spare_part[2][10][5] = 112

spare_part[2][11][2] = 614
spare_part[2][11][3] = 151
spare_part[2][11][5] = 173
spare_part[2][12][2] = 649
spare_part[2][13][2] = 981
spare_part[2][13][3] = 140
spare_part[2][13][5] = 49
spare_part[2][14][2] = 703
spare_part[2][14][3] = 197
spare_part[2][14][5] = 177

recyclable[2][1][2] = 5135
waste[2][1][2] = 2569
recyclable[2][1][3] = 817
waste[2][1][3] = 409
recyclable[2][1][5] = 363
waste[2][1][5] = 182
recyclable[2][2][2] = 3746
waste[2][2][2] = 1873
recyclable[2][2][3] = 1102
waste[2][2][3] = 552
recyclable[2][2][5] = 1001
waste[2][2][5] = 501
recyclable[2][3][2] = 7703
waste[2][3][2] = 1
recyclable[2][3][3] = 1225
waste[2][3][3] = 1
recyclable[2][3][5] = 544
waste[2][3][5] = 1
recyclable[2][4][2] = 5619
waste[2][4][2] = 0
recyclable[2][4][3] = 1653
waste[2][4][3] = 1
recyclable[2][4][5] = 1501
waste[2][4][5] = 1
recyclable[2][5][2] = 7703
waste[2][5][2] = 1
recyclable[2][5][3] = 1225
waste[2][5][3] = 1
recyclable[2][5][5] = 544
waste[2][5][5] = 1
recyclable[2][6][2] = 5619
waste[2][6][2] = 0
recyclable[2][6][3] = 1653
waste[2][6][3] = 1
recyclable[2][6][5] = 1501
waste[2][6][5] = 1
recyclable[2][7][2] = 7703
waste[2][7][2] = 1
recyclable[2][7][3] = 1225
waste[2][7][3] = 1
recyclable[2][7][5] = 544
waste[2][7][5] = 1
recyclable[2][8][2] = 5619
waste[2][8][2] = 0
recyclable[2][8][3] = 1653
waste[2][8][3] = 1
recyclable[2][8][5] = 1501

```

```

waste[2][8][5] = 1
recyclable[2][9][2] = 6676
waste[2][9][2] = 2568
recyclable[2][9][3] = 1062
waste[2][9][3] = 409
recyclable[2][9][5] = 471
waste[2][9][5] = 183
recyclable[2][10][2] = 4869
waste[2][10][2] = 1874
recyclable[2][10][3] = 1433
waste[2][10][3] = 552
recyclable[2][10][5] = 1301
waste[2][10][5] = 501
recyclable[2][11][2] = 4869
waste[2][11][2] = 1874
recyclable[2][11][3] = 1433
waste[2][11][3] = 552
recyclable[2][11][5] = 1301
waste[2][11][5] = 501
recyclable[2][12][2] = 6676
waste[2][12][2] = 2568
recyclable[2][12][3] = 1062
waste[2][12][3] = 409
recyclable[2][12][5] = 471
waste[2][12][5] = 183
recyclable[2][13][2] = 6676
waste[2][13][2] = 2568
recyclable[2][13][3] = 1062
waste[2][13][3] = 409
recyclable[2][13][5] = 471
waste[2][13][5] = 183
recyclable[2][14][2] = 4869
waste[2][14][2] = 1874
recyclable[2][14][3] = 1433
waste[2][14][3] = 552
recyclable[2][14][5] = 1301
waste[2][14][5] = 501

alloc_1[3,1,1,2] = 1482
alloc_1[3,1,2,2] = 1482
alloc_1[3,1,3,11] = 396
alloc_1[3,1,4,21] = 999
alloc_1[3,1,5,13] = 429
alloc_1[3,1,6,14] = 177
alloc_1[3,1,7,3] = 1554
alloc_1[3,1,8,4] = 2910
alloc_1[3,1,9,6] = 1975
alloc_1[3,1,10,6] = 1975
alloc_1[3,1,11,7] = 408
alloc_1[3,1,12,7] = 408
alloc_1[3,2,1,2] = 768
alloc_1[3,2,2,2] = 768
alloc_1[3,2,3,11] = 605
alloc_1[3,2,4,21] = 1923
alloc_1[3,2,5,13] = 503
alloc_1[3,2,6,14] = 686
alloc_1[3,2,7,3] = 750
alloc_1[3,2,8,4] = 1198

alloc_1[3,2,9,6] = 1036
alloc_1[3,2,10,6] = 1036
alloc_1[3,2,11,7] = 982
alloc_1[3,2,12,7] = 982

alloc_2[3,1,2,2,2] = 2964
alloc_2[3,1,3,2,3] = 781
alloc_2[3,1,3,3,3] = 773
alloc_2[3,1,4,2,3] = 2910
alloc_2[3,1,6,2,3] = 3950
alloc_2[3,1,7,5,2] = 816
alloc_2[3,1,11,2,3] = 396
alloc_2[3,1,13,3,3] = 429
alloc_2[3,1,14,3,2] = 177
alloc_2[3,1,21,2,3] = 999
alloc_2[3,2,2,2,2] = 1536
alloc_2[3,2,3,2,3] = 750
alloc_2[3,2,4,2,3] = 1198
alloc_2[3,2,6,2,3] = 2072
alloc_2[3,2,7,5,2] = 1964
alloc_2[3,2,11,2,3] = 605
alloc_2[3,2,13,3,3] = 503
alloc_2[3,2,14,3,2] = 686
alloc_2[3,2,21,2,3] = 1923

alloc_3[3,1,2,1,3] = 84
alloc_3[3,1,2,2,3] = 110
alloc_3[3,1,2,3,3] = 373
alloc_3[3,1,2,4,2] = 551
alloc_3[3,1,3,1,2] = 344
alloc_3[3,1,5,3,2] = 204
alloc_3[3,2,2,2,3] = 478
alloc_3[3,2,2,3,3] = 4
alloc_3[3,2,2,4,2] = 520
alloc_3[3,2,3,1,2] = 272
alloc_3[3,2,3,3,3] = 25
alloc_3[3,2,5,3,2] = 491
alloc_3[3,3,2,2,3] = 271
alloc_3[3,3,2,3,3] = 8
alloc_3[3,3,2,4,2] = 226
alloc_3[3,3,3,1,2] = 157
alloc_3[3,3,3,3,3] = 187
alloc_3[3,3,5,3,2] = 204
alloc_3[3,4,2,2,3] = 334
alloc_3[3,4,2,4,2] = 78
alloc_3[3,4,3,1,2] = 128
alloc_3[3,4,5,3,2] = 86
alloc_3[3,5,2,2,3] = 106
alloc_3[3,5,2,4,2] = 88
alloc_3[3,5,3,1,2] = 75
alloc_3[3,5,5,3,2] = 155
alloc_3[3,6,2,2,3] = 38
alloc_3[3,6,2,4,2] = 169
alloc_3[3,6,3,1,2] = 78
alloc_3[3,6,5,3,2] = 38
alloc_3[3,7,2,2,3] = 288
alloc_3[3,7,2,3,3] = 31
alloc_3[3,7,2,4,2] = 187

```

```

alloc_3[3,7,3,1,2] = 155
alloc_3[3,7,3,3,3] = 189
alloc_3[3,7,5,3,2] = 204
alloc_3[3,8,2,2,3] = 86
alloc_3[3,8,2,4,2] = 276
alloc_3[3,8,3,1,2] = 83
alloc_3[3,8,5,3,2] = 113
alloc_3[3,9,2,2,3] = 110
alloc_3[3,9,2,4,2] = 22
alloc_3[3,9,3,1,2] = 32
alloc_3[3,9,3,3,3] = 80
alloc_3[3,9,5,3,2] = 81
alloc_3[3,10,2,2,3] = 66
alloc_3[3,10,2,4,2] = 33
alloc_3[3,10,3,1,2] = 19
alloc_3[3,10,5,3,2] = 86
alloc_3[3,11,2,2,3] = 36
alloc_3[3,11,2,4,2] = 97
alloc_3[3,11,3,1,2] = 65
alloc_3[3,11,5,3,2] = 26
alloc_3[3,12,2,1,3] = 25
alloc_3[3,12,2,2,2] = 162
alloc_3[3,12,2,3,3] = 81
alloc_3[3,12,2,4,2] = 229
alloc_3[3,12,3,1,2] = 137
alloc_3[3,12,5,3,2] = 81
alloc_3[3,13,2,2,3] = 23
alloc_3[3,13,2,4,2] = 23
alloc_3[3,13,3,1,2] = 23
alloc_3[3,13,5,3,2] = 23
alloc_3[3,14,2,2,3] = 23
alloc_3[3,14,2,4,2] = 23
alloc_3[3,14,3,1,2] = 23
alloc_3[3,14,5,3,2] = 23

spare_part[3][1][2] = 1882
spare_part[3][2][2] = 1019
spare_part[3][3][2] = 2495
spare_part[3][4][2] = 1609
spare_part[3][4][3] = 169
spare_part[3][4][5] = 405
spare_part[3][5][2] = 2806
spare_part[3][5][3] = 269
spare_part[3][5][5] = 49
spare_part[3][6][2] = 1814
spare_part[3][6][3] = 219
spare_part[3][6][5] = 453
spare_part[3][7][2] = 2494
spare_part[3][8][2] = 1659
spare_part[3][8][3] = 214
spare_part[3][8][5] = 378
spare_part[3][9][2] = 1068
spare_part[3][9][3] = 25
spare_part[3][10][2] = 709
spare_part[3][10][3] = 99
spare_part[3][10][5] = 110
spare_part[3][11][2] = 675
spare_part[3][11][3] = 53

spare_part[3][11][5] = 170
spare_part[3][12][2] = 703
spare_part[3][13][2] = 1154
spare_part[3][13][3] = 114
spare_part[3][13][5] = 58
spare_part[3][14][2] = 762
spare_part[3][14][3] = 95
spare_part[3][14][5] = 173

recyclable[3][1][2] = 6000
waste[3][1][2] = 3000
recyclable[3][1][3] = 689
waste[3][1][3] = 346
recyclable[3][1][5] = 408
waste[3][1][5] = 204
recyclable[3][2][2] = 4042
waste[3][2][2] = 2021
recyclable[3][2][3] = 594
waste[3][2][3] = 298
recyclable[3][2][5] = 982
waste[3][2][5] = 491
recyclable[3][3][2] = 9000
waste[3][3][2] = 0
recyclable[3][3][3] = 1034
waste[3][3][3] = 1
recyclable[3][3][5] = 612
waste[3][3][5] = 0
recyclable[3][4][2] = 6063
waste[3][4][2] = 0
recyclable[3][4][3] = 891
waste[3][4][3] = 1
recyclable[3][4][5] = 1473
waste[3][4][5] = 0
recyclable[3][5][2] = 9000
waste[3][5][2] = 0
recyclable[3][5][3] = 1034
waste[3][5][3] = 1
recyclable[3][5][5] = 612
waste[3][5][5] = 0
recyclable[3][6][2] = 6063
waste[3][6][2] = 0
recyclable[3][6][3] = 891
waste[3][6][3] = 1
recyclable[3][6][5] = 1473
waste[3][6][5] = 0
recyclable[3][7][2] = 9000
waste[3][7][2] = 0
recyclable[3][7][3] = 1034
waste[3][7][3] = 1
recyclable[3][7][5] = 612
waste[3][7][5] = 0
recyclable[3][8][2] = 6063
waste[3][8][2] = 0
recyclable[3][8][3] = 891
waste[3][8][3] = 1
recyclable[3][8][5] = 1473
waste[3][8][5] = 0
recyclable[3][9][2] = 7800

```

waste[3][9][2] = 3000
recyclable[3][9][3] = 896
waste[3][9][3] = 346
recyclable[3][9][5] = 530
waste[3][9][5] = 205
recyclable[3][10][2] = 5254
waste[3][10][2] = 2022
recyclable[3][10][3] = 772
waste[3][10][3] = 299
recyclable[3][10][5] = 1276
waste[3][10][5] = 492
recyclable[3][11][2] = 5254
waste[3][11][2] = 2022
recyclable[3][11][3] = 772
waste[3][11][3] = 299
recyclable[3][11][5] = 1276
waste[3][11][5] = 492
recyclable[3][12][2] = 7800
waste[3][12][2] = 3000
recyclable[3][12][3] = 896
waste[3][12][3] = 346
recyclable[3][12][5] = 530
waste[3][12][5] = 205
recyclable[3][13][2] = 7800
waste[3][13][2] = 3000
recyclable[3][13][3] = 896
waste[3][13][3] = 346
recyclable[3][13][5] = 530
waste[3][13][5] = 205
recyclable[3][14][2] = 5254
waste[3][14][2] = 2022
recyclable[3][14][3] = 772
waste[3][14][3] = 299
recyclable[3][14][5] = 1276
waste[3][14][5] = 492

total net revenue = 4210042.00
total energy consumption = 229244555.00
total co2 emission = 4031253.00