EFFICIENT HEURISTICS FOR OPTIMALLY MATCHING BUYERS AND SELLERS IN E-MARKETPLACES

by

Bui Cong Giao

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science

Examination Committee:	Dr.Peter Haddawy (Chairman)
	Prof. Phan Minh Dung
	Dr. Sumanta Guha

Nationality:	Vietnamese
Previous degree(s):	Bachelor of Science
	HoChiMinh City University, Vietnam
Scholarship Donor:	HoChiMinh City Post and Telecommunications

Asian Institute of Technology
School of Advanced Technologies
Thailand
April 2003

ACKNOWLEDGEMENT

I would like to thank all who made this work possible. In particular, I would like to express gratitude to the following people:

To my advisor, Dr. Peter Haddawy who always spurred me to do better each day. I really enjoyed working under his supervision. His work ethics, rich knowledge, encouragement, and patience throughout the thesis were the important factors which helped me finish this work.

To Dr.Sumanta Guha for helpful recommendation. He has been very inspiring and friendly.

To Prof. Phan Minh Dung who provided me a lot of valuable knowledge in Artificial Intelligence so that I can do the work well.

Then, I wish to extend my special thanks to entire AIT community, who helped me during the unforgettable period of my life, which has started since I first attended AIT until the day I graduate.

I am thankful to HoChiMinh City Post and Telecommunications for providing scholarship, which enabled me to pursue the master program.

I am greatly grateful to my beloved father, my beloved mother, my sisters and my brother for taking care of my own family throughout my study in Thailand.

I am highly indebted to my wife who encourages me every day and devotes her time to look after my daughter.

Finally, it will be a shortcoming if I do not mention about my six-year-old daughter, who scrawled a letter to wish me good health to study well.

ABSTRACT

This thesis addresses the problem of matching buyers and sellers in barter trade exchange e-marketplaces. A barter trade exchange is a collection of businesses that buy and sell products among themselves. The collection of businesses is viewed as a micro-economy, so that matching is viewed from an economic perspective. An optimal matching seeks to maximize trade volume and to ensure that all companies share in the trade. The matching problem is given a formal representation and an efficient heuristic search algorithm is developed to solve it. The quality of solution of the heuristic search algorithm is evaluated by comparing it to the optimal solution obtained by exhaustive search on a large set of problems. The algorithm is shown to be fast enough to deal with very large real-world problems. The developed technique has the potential to greatly benefit the barter trade exchange industry as the size of trade exchanges grows.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Full Name
BTE	Barter Trade Exchange
BFS	Breadth First Search
CBB	Consumer Buying Behavior
DFS	Depth First Search
E-Commerce	Electronic Commerce
ES	Exhaustive Search
FIFS	First in, first served
IHCS	Improved Hill-Climbing Search
HCS	Hill-Climbing Search
NB	Network Broker
ODBC	Open Data Connectivity
OF	Objective function
QoS	Quality of Service

CHAPTER 1
INTRODUCTION

1.1 BACKGROUND

Barter was the original means of transaction commerce in ancient times. It involved the direct exchange of two products between two producers. This kind of direct swap is called "true barter". True barter has the limitation is that it requires corresponding demand on both sides. To overcome this limitation, currency was introduced. However, with money system, both sellers and buyers lose their benefits compared with bartering system. For instance, through barter people can get things that they want without paying money. They exchange their products or services instead. Hence, "modern barter" system is introduced to combine the advantages of bartering system over money system and solve the problem of "true barter.

Barter business has increasingly developed in recent years. It has been a successful industry for at least four decades in the USA and even for five decades in the USSR [8] because businesses can get many benefits through barter. It can help a business to conserve cash, generate new business partners, and can turn over stock or time-constrained products and services with full market price. People can sell their over-stock items through barter rather than remain unsold or they can sell their time-constrained items, e.g. perishable food, rooms, or airplane tickets, without discounting prices. According to the International Reciprocal Trade Association in 2001 there were a total of 1,596 barter trade exchanges, with a total of 470,960 businesses worldwide using their services. The total value of commercial worldwide barter transaction was $7.87 billion (the third consecutive year the industry has seen over 12% growth). In the USA, there are approximately 600 barter companies serving all parts of the nation and even overseas markets. As they are linked electronically in a national and international barter marketplace, their economic significance is growing [19].

1.2 PROBLEM STATEMENT

Trade brokers play a key role in managing modern barter business. A trade broker represents a set of client business, typically 150. However, with the rise of E-Commerce, the size of barter business has increased quickly and the broker tends to become overwhelmed by available information to process [1]. Surprisingly there is no complete software for automating the barter process so far. The existing barter software mainly focuses on accounting. Also, until now the model of agent-mediated E-Commerce for conducting business on the Internet is mainly explored from a very academic perspective, paying less attention to how proposed solutions to technical problems might fit into a practical business model, e.g. in barter business [1].

The human brokers do not have efficient tools to help them deal with customers' demands. They still contact customers by telephone or fax as main communication means. Also, they have to manage supply and demand of clients manually with the following constraints:

- Maximize trade volume over the long run.
- Fairly allocate transactions to each member business in the barter pool.

This becomes increasingly difficult as the size of a barter trade exchange grows.

1.3 OBJECTIVES

The thesis aims to implement a prototype application to optimize the barter process. In other words, the writer wants to implement the application for replacing the job of human brokers as much as possible. The following tasks will be implemented:

- Build a mathematic optimisation model of brokering in barter trade exchanges.
- Find a feasible solution for the mathematic model.
- Implement a recommendation scheduling engine.

1.4 SCOPE OF WORK

This thesis focuses on

- Design a heuristic to solve the optimization problem.
- Implement of a prototype system to represent the recommendation scheduling engine.

The engine optimizes the matching between supply and demand, and maintains the balance of clients' barter budget. It will receive predicted purchases, and then make recommendations which suppliers a business should purchase from.

1.5 STRUCTURE OF THE THESIS

The thesis is organized as follows:

- Chapter two reviews related literature. It starts from agent-mediated E-Commerce. Next, barter business, global optimization; hill climbing search, scheduling and matching problems are discussed. Finally, a problem similar to our problem is also introduced.
- Chapter three describes a system overview of barter trade exchanges. Functional components of the system are shown in detail. And, a broker agent is proposed to take charge of running this system.
- Chapter four depicts an optimization model for distribution of supply and demand in the barter trade exchange system. Many various methods are discussed to solve this optimization model, and then the evaluation of methods is also done.
- Chapter five depicts the extension of the proposed algorithm and a prototype application that represents the functionalities of the recommendation scheduling engine.
- Conclusions and recommendations for further research are presented in Chapter six.

CHAPTER 2
LITERATURE REVIEW

2.1 AGENT-MEDIATED E-COMMERCE

Over the last few years, a new kind of software application has appeared based on a synthesis of ideas from artificial intelligence, human computer interaction and electronic transaction: agents that help mediate E-Commerce activities. Agents differ from "traditional" software in that they are personalized, autonomous, proactive and adaptive. These qualities make agents particularly useful for the information-rich and process-rich environment of E-Commerce [3].

Agent-mediated E-Commerce uses software agents, semi-intelligent tools, to automate a variety of tasks including those involved in buying and selling products over the Internet. Software agents have been playing an important role in E-Commerce applications. There are hundreds of commerce agents: from customer to customer "smart" classified ads to merchant agents, from agents that facilitate expertise brokering to distributed reputation facilities. Guttman et al., 1998, classified these agents into six classes, which parallel a customer's six steps in purchasing decision-making shown in Figure 2.1 together with brief explanations [2]. Note that the process is cyclic and that the steps may overlap each other [5].

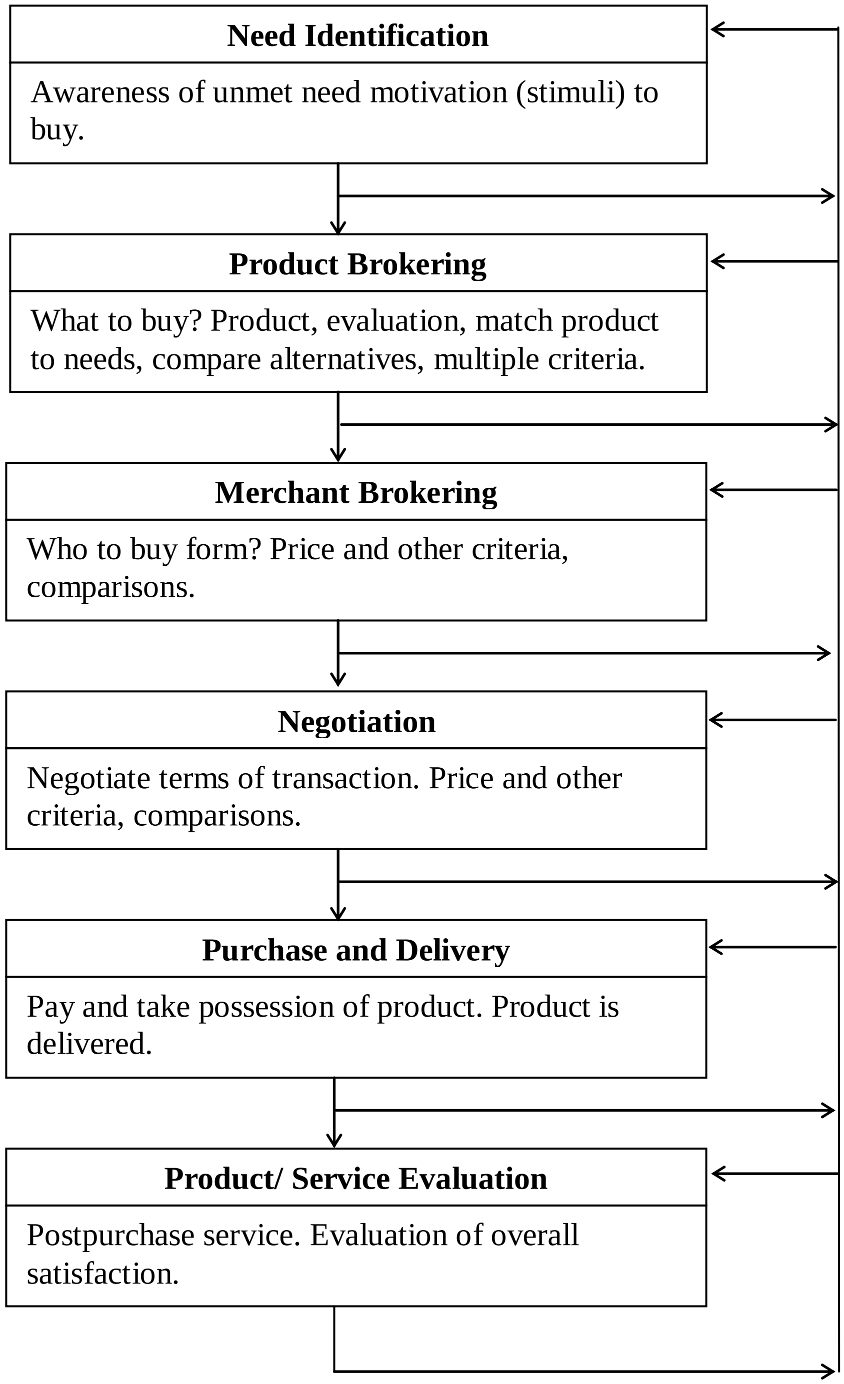


Figure 2.1: **The purchasing decision-making process** [5]

Guttman et al., 1998, also surveyed briefly several existing agent-mediated E-Commerce systems by describing their role in the context of Consumer Buying Behaviour (CBB) [2]. Table 2.1 lists the six CBB stages and shows some agent systems in the three agent-centric stages of the CBB model.

Table 2.1: **Roles and examples of agents as mediators in E-Commerce** [2]

	Personal Logic	Firefly	Bargain Finder	Jango	Kasbah	Auction Bot	Tete-a-Tete
Need Identification							
Product Brokering	X	X		X			X
Merchant Brokering			X	X	X		X
Negotiation					X	X	X
Purchase and Delivery							
Product Service & Evaluation							

a. Product Brokering

- ❖ PersonalLogic is a tool that enables consumers to narrow down the products that best meet their need by guiding them through a large product feature space.
- ❖ Firely recommends products via a "word of mouth" recommendation mechanism called automated collaborative filtering.

b. Merchant Brokering

- ❖ BargainFinder was the first shopping agent for price comparisons. Given a specific product, BargainFinder would request its price from each of nine different merchant Web sites. Value added services that merchants offered on their website were being bypassed by BargainFinder and therefore not considered in the consumer's buying decision. However, merchants don't want to be compared just in product price terms. As a result, a third of the on-line CD merchants accessed by BargainFinder blocked all of its price requests.
- ❖ Jango can be viewed as an advanced BargainFinder. Jango has product requests originate from each consumer's web browser instead of from a central site as in BargainFinder. For this reason, the merchants with a web presence will be forced to interoperate with agent [3]. This solves the merchant blocking of BargainFinder.
- ❖ MIT Media Lab's Kasbah is a multi-agent system for consumer-to-consumer E-Commerce. A user wanting to buy or sell a good creates an agent, gives it some strategic direction, and sends it off into a centralized agent marketplace. Kasbah agents pro-actively seek out potential buyers or sellers and negotiate with them on their owner's behalf.

c. Negotiation

- ❖ Kasbah enables buying agents to offer a bid to sellers with no restrictions on time or price.
- ❖ AuctionBot helps users create new auctions to sell products by choosing from a selection of auction types and then specifying parameters for that auction. Next, the seller's negotiation is completely automated by the system as defined by auctioneer protocols and parameters.
- ❖ Tete-a-Tete is a multi-agent, bilateral bargaining system. It integrates all three of the Product Brokering, Merchant Brokering, and Negotiation CBB stages. It negotiates across multi terms of a transaction.

Today' s first generation agent-mediated electronic commerce systems are already creating new markets (e.g., low cost consumer-to-consumer and refurbished goods) and beginning to reduce transaction costs in a variety of business process. The industries affected the earliest are those dealing with perishables (tickets, bandwidth availability, etc), surplus inventory and commodities (gas, books, electricity, etc).

Looking even further into the future, agents will explore new types of transactions in the form of dynamic relationships among previously unknown parties. At the speed of bits, agents will strategically form and reform coalitions to bid on contracts and leverage economies of scale – in essence, creating dynamic business partnerships that exist only as long as necessary. It is in this third-generation of agent-mediated electronic commerce where companies will be at their most agile and marketplaces will approach perfect efficiency [4].

2.2 BROKERING IN E-COMMERCE

2.2.1 The use of brokers

Most of the applications on the Internet that include some form of search make use of brokers. A broker is an intermediary between buyers and sellers. Using brokers can have a number of advantages like reducing search costs, maintaining privacy, information integration, reducing contracting risks, and pricing efficiency [11].

Brokers can help to reduce search cost in a number of ways. It may be expensive for providers and consumers to find each other. In the bazaar of the information superhighway, for example, thousands of products are exchanged among millions of people. Brokers can maintain a database of customer preferences, and reduce search costs by selectively routing information from providers to customers. Furthermore, producers may have trouble accurately gauging consumer demand for new products; many desirable items may never be produced simply because no one recognizes the demand for them. Brokers with access to customer preference data can predict demand.

A broker can guarantee the privacy of buyer and seller. As an intermediary, the broker can ensure that information is provided on a need-to-know basis only. The broker can search for products on behalf of a prospective buyer without giving personal information about that buyer to possible sellers. The other way around, the broker can also present information about an interesting product without revealing the source of the information. The broker can, for example, only provide the necessary personal information to buyer and seller when a match is found that is acceptable for both buyer and seller. It is possible to take this even further, the broker might handle the transactions as well, ensuring that buyer and seller need know each other's identity.

Information integration is yet another possible advantage of using brokers. The broker gathers product information from different sources, e.g., from different sellers, independent evaluators, and from other customers. Each source might use its own ontology, making it hard for an arbitrary participant of the market to understand all that information. A broker, being a specialized entity within the market, to knowledge to consult different databases and has knowledge of many different ontologies. By using that knowledge, the broker is capable of constructing one report that integrates the relevant parts of the obtained information.

Having a broker can reduce contracting risk. This can only be obtained if the broker has the means to enforce the market policies and regulation, e.g., the right to penalize offenders in terms of money and access to the market. The broker acts more or less like a policeman, thus providing a secure and reliable environment for people who do business in a fair way. For instance, Namo Kang et al., 2002, suggested a broker-based synchronous transaction algorithm that would guarantee a more fair and efficient transaction deal for both sellers and buyers [16].

At last, brokers help to avoid pricing inefficiencies. The balance is held here by broker to avoid any parties to attempt any free-riding strategy. Brokers can use pricing mechanisms that induce just the appropriate exchanges.

2.2.2 Broker agents

To illustrate the role of brokers, Marcel Albers et al., 1999, introduced a new architecture for virtual markets, GEMS (Global Electronic Market Stands) [23]. Fundamental in GEMS is that it aims to maintain all the good aspects of market places with tents (market categories) that are still held all over the world, and, at the same time, bring this market place to the world instead of being local to a specific town. GEMS globalises the market using the World Wide Web.

In Figure 2.2 the overall multi-agent architecture for the electronic market place is presented. The users/consumers of the market are represented and assisted by Interface Agents that, with the help of the Broker Agent, locate and contact the relevant Category Agents. Category Agents represent and assist providers/sellers in their business.

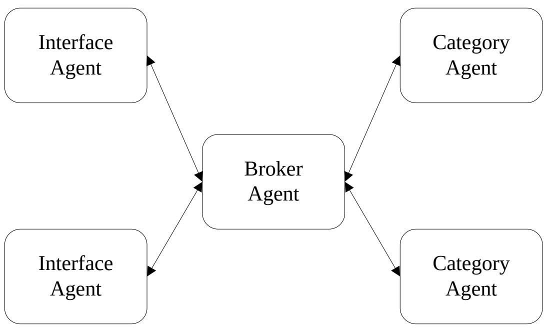


Figure 2.2: **The overall electronic market place architecture** [23]

The process of brokering involves a number of activities. For example, responding to buyer requests for products with certain properties, maintaining information on customers, building customer profiles on the basis of such customer information, maintaining information on products, maintaining provider profiles, matching buyer requests and product information (in a strict or soft manner), and responding to new offers of products by informing customers for whom these offers fit their profile. The generic broker agent architecture depicted in Figure 2.3 supports such activities by distinguishing different processes and having them work together in a coordinated manner.

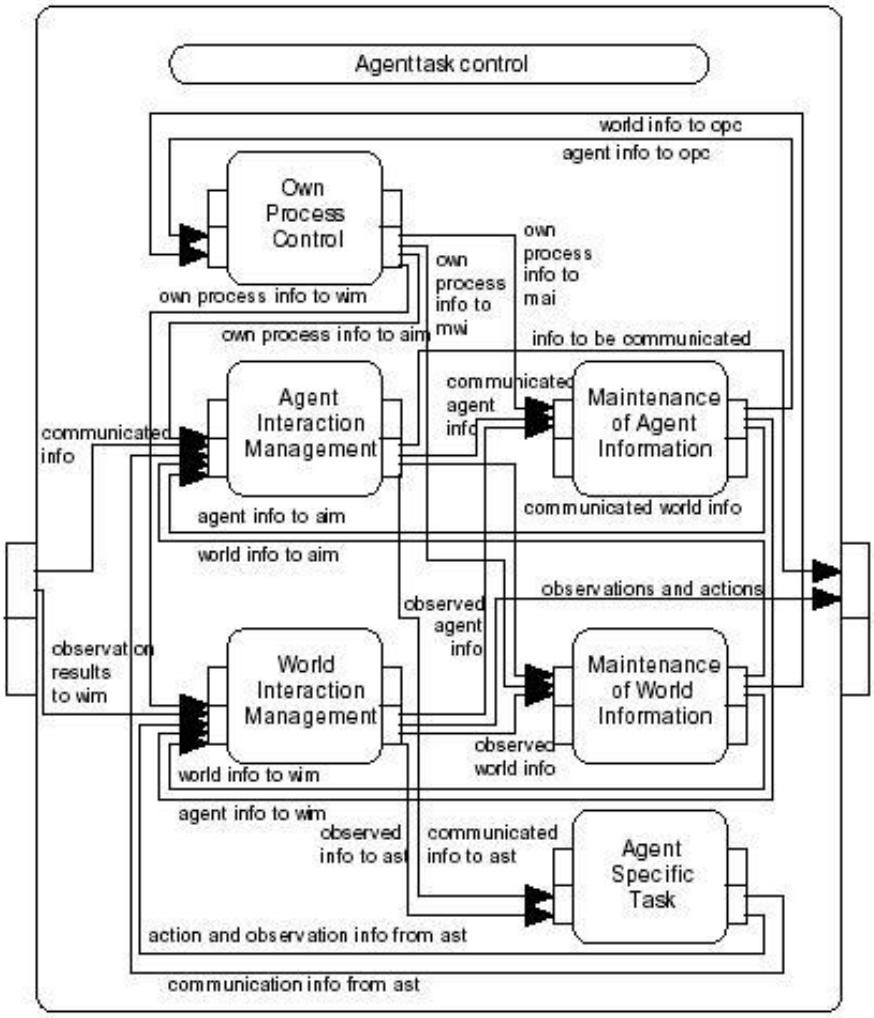


Figure 2.3: **Generic broker agent model** [25]

An optimisation model of brokering is the main objective that the thesis will present. The writer would like to construct a broker agent whose functionalities are similar to those of GEMS' broker agent. It will run the optimisation model and interface with client agents automatically. However, due to the scope of the thesis, maximal matching between supply and demand and maintaining the balance of clients' barter budget are foremost tasks.

2.2.3 Agent strategies for sellers, buyers, and brokers in E-Commerce

Electronic trade via the Web introduces agents- sellers and buyers- with multiple challenges. One of the challenges is that electronic sellers are faced with a lot of purchase-orders. The orders arrive from multiple anonymous buyers, and the sellers need to fulfil them in the face of limited stocks. At times, a seller may be unable to address all orders; nevertheless, he or she would like to maximize gains. To do so, they should have strategies that will maximize their gain given their partial fulfilment of the purchase-orders that they received. It is also necessary to study buyers' strategies for selecting sellers given their purchase-order satisfaction.

Goldman et al., 2002, evaluated RandS (Random Seller), a strategy for sellers, is the best [27]. The strategy is depicted as follows.

A seller chooses randomly which purchase-order to fulfil from those that were submitted to it. The seller serves completely all of the requests it can, constrained by the size of his stock. Assuming that the order of arrival of the buyers to sellers does not depend on any characteristic of the buyers, the RandS behaviour is equivalent to FIFS (first in, first served) behaviour.

Goldman et al., 2001, utilized a simulation tool for agent strategies to examine markets in two cases. The first is that buyers are willing to accept partial satisfaction and sellers' stocks are all the same [27]. The second is that sellers' stock are heterogeneous, and buyers suffer significant losses from partial satisfaction of their requests [28]. The experiments led to conclusions that sellers should behave randomly, i.e. should chose to supply the requests they are asked for in a random manner. RandS leads to a lower level of liquidity of the buyers in the market, the buyers remain rather static after being distributed among the sellers. The recommendation for the buyers is to punish the non-satisfying sellers as much as the stock size enables them, i.e., the severity of the punishment is inversely proportional to the amount of competition among the buyers.

In matching between supply and demand in barter trade exchanges, if demand is larger than supply and there are two buyers that identically satisfy constraints as mentioned in Section 1.3, brokers should match sellers to the buyers randomly. It is the similar in the opposite case, i.e. surplus.

As for brokers, their strategies are influenced by search costs. Some market spaces are very large and help in sorting through the options is required; others offer very complex product offerings and help in matching product offerings and business needs is useful. The search function of the online intermediary has been widely recognized by Bailey and Bakos, 1997, and has been called either "making searching easier" or reducing search costs" [32].

Segev and Beam, 1999, claimed that when search costs are very low, the broker should choose to charge medium-high fees and plan on attracting only a small subgroup of buyers and sellers [33]. As search costs increase to medium to medium-high, conditions for the broker improve. The broker can then afford to charge relatively high fees to buyers or sellers (but not to both) and still maintain a large volume of transactions and gather high revenues. When search costs become extremely high, the market space breaks down. The trading volume drastically decreases, and the broker's revenue with it.

The above making-money strategies of brokers can be applied in barter business that will be introduced in the following part.

2.3 BARTER BUSINESS

2.3.1 What is barter?

Barter is the exchange of goods and/ or services without the use of money.

Traditionally bartering is a one-to-one transaction, whereby the company or individual with a surplus in one resource and a need in another, has to locate a company or individual with an equal and opposite need for a reciprocal or "Contra" trade to take place [8]. Companies can advertise their surplus in a classified area and may find a partner, but usually they will have little success in finding an exact match. Thus, they ask an intermediary to help. The intermediary often uses a manual search-and-match approach to meet this task. Barter therefore offers a facility, which allows people to make all their assets tradable.

Nowadays people can earn “barter dollars” or “trade dollars” by putting their products and services into a barter pool. Then they can use the “barter dollars” to buy products and services of other members in the pool. Barter dollars are units of account which denote the right to receive, or obligation to pay, in goods or services available from the participating members of an exchange. The use of barter dollars permits trades to be recorded and balances to be debited and credited, so the barter system can function smoothly.

Any kinds of product and services can work with barter. Popular bartering items are office space, idle facilities and labor, products and even banner ads. As a result, people can do business without direct swap any more through the intermediation, call “barter trade exchange”.

A barter trade exchange (BTE) functions primarily as the organizer of a marketplace where members buy and sell products and services among themselves either on a full barter basis, or a part-cash, part-barter basis. It may also act as a trading company by buying and selling for its own account from others [19].

The marketplace is used to create an opportunity for the trading of wasted resources, whilst simultaneously improving liquidity and cash flow. Also, through the barter pool, members sell their surplus at normal market prices [8]. However, the problem with manual matching done by a third party is that the commission is very high (30% or more) and it may take a long time to complete a transaction.

Electronic battering can improve the matching procedure by attracting more customers to the exchange. Also, the matching can be done faster. Consequently, better matches can be found and the commission is much lower (5 to 10%). Electronic battering may have tax implication that needs considering. Also, bartering sites must be financially secure. Otherwise users may not have a chance to use the points they accumulate. Some of bartering websites are www.intagio.com, www.ubarter.com, and whosbartering.com [6].

Barter exchanges make money by charging a commission on each barter transaction, signup fee, and renewal fee of members. All barter transactions are recorded, and barter income of each client is reported annually to the barter trade exchange. Corporate trade companies make money by negotiating favorable prices for media and other products and services, and exchanging these for the excess assets of their clients plus cash [19]. For example, BizXchange, a Berkeley-based business-to-business bartering firm charges clients a sign-up fee of either $295 or $595. The smaller fee will allow a company to make cash transaction fees of 7.5 percent on the gross value of all sales or on the gross value of all purchases. The larger sign-up fee will allow firms to make the same transactions and purchases with 6 percent on either side. Both types of packages also charge a monthly maintenance fee of $15 in cash and $15 in barter dollars. BizXchange clients are informed of existing and new participating companies in a weekly email newsletter [31].

2.3.2 Barter model

A barter pool as a network of inter-related business is modeled as follows. A link exists between two businesses if there is a potential sell/buy relationship between them (see Figure 2.4). The trade exchange can exercise control over the members in the pool only by making recommendation through its trade brokers. This means that it recommends to members that they purchase a product or service from another member. Note that providing recommendation can be viewed as passing messages among the links between potential sellers and buyers [1].

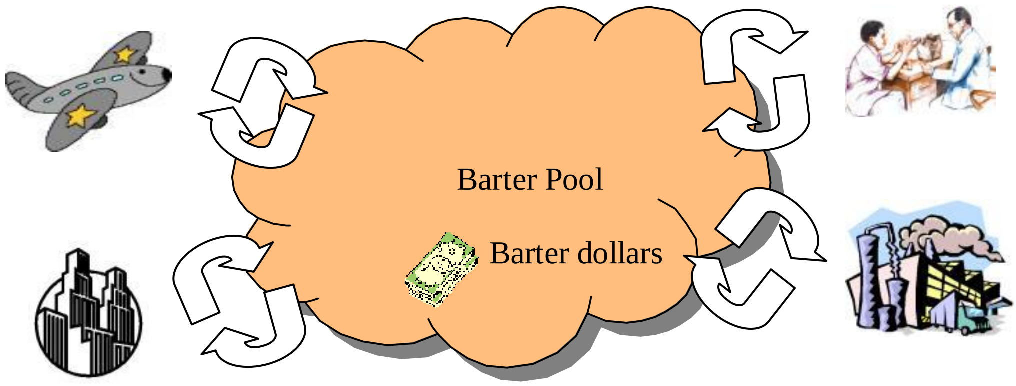


Figure 2.4: **Barter pool**

Each business has a financial operating range (see Figure 2.5). The lower bound is the credit limit extended to the business by the barter trade exchange, i.e. the maximum allowed negative trade balance. The upper bound is the maximum positive trade balance, set by the business itself. The financial operating point at which a business is maximally willing to able to buy and sell lies somewhere between these two bounds. This is a business optimal financial operating point. The trade exchange can function most effectively when each member is at its optimal financial operating point. The business must keep its credit balance because when the trade balance reaches the lower limit, the business can no longer buy and when it reaches the upper limit it can no longer sell [1].

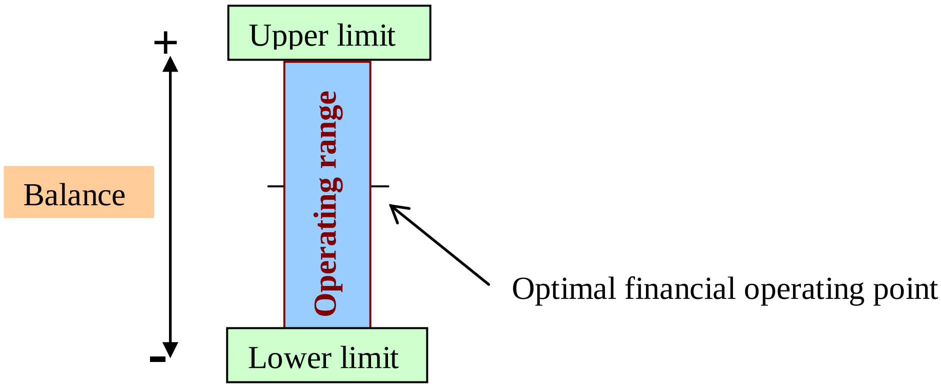


Figure 2.5: **Financial operating range**

The question is then how the trade exchange should best allocate messages in order maximize trade volume over the long run and keeping each member business at its optimal financial operating point. Let consider the following example. Suppose that business A has links from businesses B, C as shown in Figure 2.6.

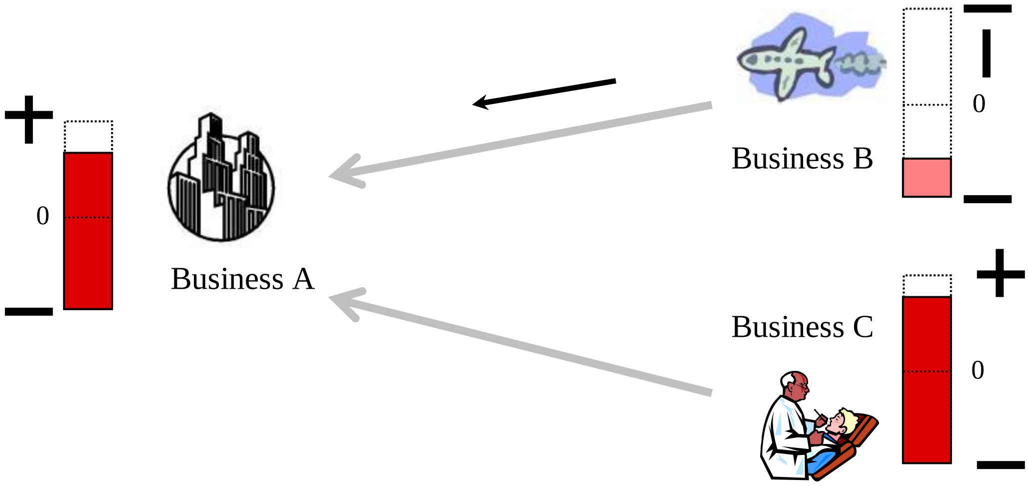


Figure 2.6: **Passing a message to maintain balance of trade**

Business A is close to the upper bound of its operating range. B is close to the lower bound of its operating range. C is close to the upper bound of its operating range. Barter managers should encourage A to spend its trade dollars and move B away from its credit. As a result, a message will be passed from B to A.

2.3.3 Benefits of doing business through barter [7]

a. *Converse cash*

Since cash flow is the important part of businesses, they try to keep cash as much as possible for unscheduled expenditures or unexpected business opportunities. Trading with barter without using money helps them to improve the situation.

b. *Create new customer base*

Trading introduces a member business to another who might never use the former’s products or services. In contrast, when the business needs a product or service, it asks the broker and then the broker direct providers to it. That thing is happening in the barter world!

Members of exchange are not restricted in using barter dollars only. They can buy and sell with cash at the rest of the world. Some exchanges are flexible in allowing members to trade half cash, half trade dollars.

c. *Solve slow or old inventory*

When a client has excess capacity of products or service, especially when these items are time-constraint, the broker will try to sell this kind of stuffs. In other words, the broker will help the client extent their market to sell their goods to new customers. Then the client can use barter dollar that they earn to purchase other products or services that their business is currently spending cash on.

d. *Improve marketing/ advertising effort*

A member business can substantially reduce its advertising expenditures through the use of media that is readily available through barter networks. Hence joining a good barter exchange with members spreading out nationwide or even worldwide, the member business can assume that its business existence, to some extent, is automatically known by other members.

e. *Improve the quality of working and living*

Companies often provide bonus and/ or incentives for their staff moral and productivity. Barter provides the staff with the means of cost-effectively incentive or promotion program. Also, spending the barter dollars the staff earns through their business for items that they might need at home is one pleasure of bartering. They can buy any products available in the barter pool.

2.3.4 Job of a barter broker

This thesis aims to create a recommendation scheduling engine for the optimisation of matching supply and demand in the barter pool. So, the important thing is that the job of human broker should be understood. Then, after getting the list of broker jobs, the writer can think about which part can be automated.

Besides job of brokers mentioned in Section 2.2, barter brokers must assume the following tasks [7]:

Before a client joins

- Get "Needs and Wants" list of customer: at least 10 items that customer would like to get through barter.
- Ask customer what items they can put in the pool ("What you have" list).
- Explain rules about part barter/ part cash trades.
- Explain what lines of credit are offered.

After a client joins

- Confirm the "Needs and Wants" list.
- Confirm "What You Have" list.
- Help the client to develop a barter budget.
- Educate the client: explain to them how they should spend barter credits.
- Keep customer credit balance: help to reduce large positive or negative trade balance. This is a task that can be automated.

In the following parts, the writer would like to introduce mathematic aspects related to the construction of an optimisation model of brokering in barter trade exchanges. Firstly, global optimisation is mentioned since it is a general picture of the problem in the thesis, whose ultimate purpose is an approach to global minimum as close as possible. Next, necessary techniques that aim to solve the problem efficiently, not only optimal degree but also acceptable runtime, are also studied. They are hill climbing search and job-shop scheduling.

2.3 GLOBAL OPTIMISATION

The aim of global optimisation is to find the solution for which the objective function obtains its smallest value, the global minimum. In contrast to local optimisation for which the attainment of the local minimum is decidable (gradient equal to zero) no such general criterion exists in global optimisation for asserting the global minimum has been reached [21].

Generalized Descent Method: One way to prevent unnecessary local searches is to continue random sampling in A until a point with function value smaller than the smallest minimum currently known is found. Starting a local search from this point guarantees that a lower minimum is found. This technique can be seen as a generalization of local search methods for the case of global optimisation.

Formulation of Global Optimization Problem. Global minima that are attained in isolated points are of course generally not possible to determine. Ruling out such problem is necessary but not sufficient for obtaining solvable problems.

There are two ways to deal which the unsolvable problem. The first is to consider only problems for which efficient solution techniques exist. The second approach is to reformulate the global optimisation problem by relaxing the solvability requirement in order to be able to consider a larger class of problems.

Heuristic in Global Optimization. The traditional mathematical approach is not very relevant because the global optimisation problem has proved to be unsolvable in general and thus constitutes an intractable mathematical problem. In order to be able to treat this problem some heuristics must be introduced.

Arguments for introducing heuristics in optimisation are presented in [34]. The authors of the paper discussed why and when to use heuristics, features of good heuristics and how to use them. The writer quotes the following from the conclusion in their paper: "*The need for and good heuristics in both academia and business will continue increasing fast. When confronted with real world problems, a researcher in academia experiences at least once the painful disappointment of seeing his product, a theoretically sound and mathematically "respectable" procedure, not used by its ultimate user. This has encouraged researchers to develop new improved heuristics and rigorously evaluate their performance, thus spreading further their usage in practice, where heuristics have been advocated for a long tim*e."

In methods using heuristics normally values of some parameters must be chosen by the user. A clear relation between these parameter values and the problem characteristics does not exist. Therefore some trial and error technique may be needed in a real situation.

Evaluation of Methods. A treatment of methods is not complete without some comparisons or evaluation of their performance. Evaluate of methods are based on the three criteria below:

- Evaluating Performance. The main reason for evaluating the performance of methods is to be able to choose the most efficient method for solving some class of problems. This includes the calibration of a given method by determining the optimal values of some parameters of the method.
- Cost Estimates. In a study where all algorithms are run on the same computer the computer cost (CPU time) could be used. In this case it would possible to take into account such factors as ease of use, memory required and so on.

- Empirical Comparison of Algorithms. The probability that an algorithm will find the global minimum can for most algorithms be determined empirically as m_1/m by apply the algorithm several times m to the same random problem using different random numbers and recording the number of times m_1 the global minimum is found. By calibrating the methods so that the ratio $m_1/m = \gamma$ for two methods (e.g. $\gamma = 0.9$) it would be possible to make a fair comparison between the methods by recording the time needed. By comparing methods in this way for different problems, operating characteristics permitting the choice of optimal methods for some class of problems could be obtained.

2.4 HILL-CLIMBING SEARCH (HCS)

Hill Climbing Search is based on Depth First Search (DFS). A **heuristic** is used to improve the search efficiency. It is a complete search and gives **non-optimal** solutions, i.e. it will not necessarily find the most efficient route through the state space [18].

This is a kind of *Iterative Improvement Algorithms* [12]. The general ideal is to start with a complete configuration and to make modifications to improve its quality. The best way to understand the method is to consider all the states laid out on the surface of a landscape. The height of any point on the landscape corresponds to the evaluation function of the state of that point (see Figure 2.7). The idea of the method is to move around the landscape trying to find the highest peaks.

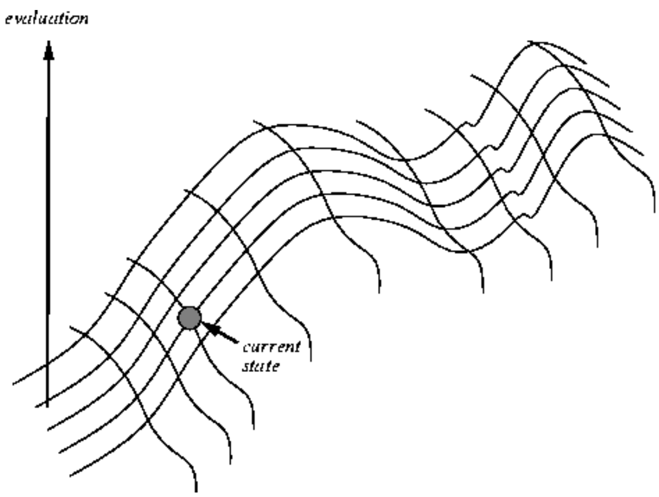


Figure 2.7: **Iterative improvement algorithms try to find peaks on a surface of states where height is defined by evaluation function**

The hill-climbing search algorithm is shown as follows

function HILL-CLIMBING (*problem*) **returns** a solution state
    inputs: *problem*, a problem
    static: *current*, a node
            next, a node

    current ⟵ MAKE-NODE(INITIAL-STATE[problem])
    loop do
        next ⟵ a highest-valued successor of current
        if VALUE[next] < VALUE[current] **then return** current
        current ⟵ next
    end

At first, an initial state is established. After that the algorithm is simply a loop that continually moves in the direction of increasing value. The algorithm does not maintain a search tree. It keeps track of only the current state and do not look ahead beyond the immediate neighbor of that state. An important refinement is that when there is more than one best successor to choose from, the algorithm can select among them at random.

The average runtime for the hill-climbing program is very close to linear. For example, a hill-climbing program solved the million-queen problem in less than four minutes on a SPARCstation1 [2].

However, the local search has drawbacks:

- Local maxima: a local maximum, as opposed to a global maximum, is a peak that is lower than the highest peak in the state space. Once on a local maximum, the algorithm will halt even though the solution may be far from satisfactory.
- Plateau: a plateau is an area of the state where the evaluation function is essential flat. The search will conduct a random walk.
- Ridge: A ridge is a special kind of local maximum. It is an area of the search space that is higher that the surrounding areas and that it itself has a slope. But the orientation of the high region, compared to the set of available moves and the directions in which they move, makes it impossible to traverse a ridge by single moves. Any point on a ridge can look like peak because movements in all probe directions is downward.

2.5 SCHEDULING

Scheduling is allocating shared resource over time to competing activities. It has been the subject of a significant amount of literature in the operations research area.

As mentioned earlier, the main task of human brokers in a barter trade exchange is to manage supply and demand of clients such that all clients feel that they are fairly behaved and the exchange gets a maximum benefits over the long run with a minimum management cost. Hence, principles of scheduling and methods for repair based scheduling [9] are discussed herein so that the knowledge is a cue for finding an efficient solution of barter brokers' problem.

Any resource allocation and scheduling policy must consider three factors [24]:

- **Fairness**: All processes that are competing for the use of a particular resource to be given approximately equal and fair access to that resource. This is especially so for jobs of the same class, that is, jobs of similar demands.
- **Differential responsiveness**: On the other hand, the scheduler may need to discriminate among different classes of jobs with different service requirement. The scheduler should also view these decisions dynamically. For instance, if a process is waiting for the use of a resource, the scheduler may wish to schedule that process for execution as soon as possible to free up the device for later demands from other processes.
- **Efficiency**: The scheduler should attempt to maximize throughput, minimize response time, accommodate as many users as possible; finding the right balance for a particular situation is an ongoing problem for operating research.

Most scheduling applications involve optimal allotment between supply and demand. One of these applications is job-shop scheduling.

A job-shop scheduling problem [35] consists of a set of machines and a collection of jobs to be scheduled. Operation precedence constraints give the order in which the operations that comprise each job must be processed. The job shop scheduling problem thus can be defined as the allocation of machines over time to perform a collection of jobs to minimize/maximize a performance measure while satisfying the operation precedence constraints, machine capacity constraints, processing time requirements, and ready time requirements. Since the job-shop scheduling problem is NP-hard, i.e., the computational requirement grows exponentially as a function of the problem size, it is unlikely that a practical approach to this scheduling problem can yield an optimal solution. Therefore, one could even use an exact method to find an optimal solution for small problem instances. But for lager problem instances it is more appropriate to use heuristics or approximation algorithms, such as local-search-based algorithms, to find a good solution that is not necessarily the optimum one.

Local-search-based algorithms include *local search*, *simulated annealing*, or *tabu search*. These search techniques are very efficient in solving combinatorial optimisation problems.

1. *Local Search*: An example of this kind is hill-climbing search (see section 2.4). The drawback of local search is that it has a tendency of getting stuck at a local optimum (or a cycle).
2. *Simulated Annealing:* Simulated annealing can be regarded as a variation of local search. The main difference is that instead of starting again randomly when stuck on a local maximum, the search to take some downhill steps is allowed to escape the local maximum. Its drawbacks include the following: its performance is heavily influenced by the initial temperature and the decrement ratio of the temperature, context-sensitive search behaviour, and it could be potentially time consuming when applied to complex problem instances.
3. *Tabu Search:* The method is to forbid some search directions (moves) at a present iteration in order to avoid cycling and escape from a local optimal point. This strategy can make use of any local improvement technique.

2.6 MATCHING PROBLEMS

This thesis aims to create an automatic recommendation scheduling engine for the optimisation of matching supply and demand in the barter pool. The main method to build the optimisation model is local repair heuristics, which will be introduced in chapter 4. However, the writer thinks that matching problems in the theory of network flows need surveying so that this helps to have another insight for this problem.

A matching in a graph G = (N, A) is a set of arcs with the property that every node is incident to at most an arc in the set; thus a matching induces a pairing of (some of) the nodes in the graph using the arcs in A. In a matching, each node is matched with at most one other node, and some nodes might not be matched with any other node. The matching problem seeks a matching that optimises some criteria. Matching problems on bipartite graphs (i.e., those with two sets of nodes and with arcs that join only nodes between the two sets, as in the assignment and transportation problems) are called bipartite matching problems, and those on nonbipartite matching problems. There are two additional ways of categorizing matching problems: cardinality matching problems, which maximize the number of pairs of nodes matched, and weighted matching problems, which maximize or minimize the weight of the matching. The weighted matching problem on a bipartite graph is also known as the assignment problem.

Application of matching problems arise in matching roommates to hostels, matching pilots to compatible airplanes, scheduling airline crews for available flight legs, and assigning duties to bus drivers.

Here the bipartite weighted matching problem is focused because it can solve the problem of the thesis. There are many algorithms for the problem: successive shortest algorithm, Hungarian algorithm, relaxation algorithm, and cost scaling algorithm [15]. The writer would like to introduce one of them: Hungarian algorithm.

Given a weighted bipartite network G= (N_1 U $N_{2,}$ A) with $|N_1| = |N_2| = n$ and arc weights c_{ij}, find a perfect matching of minimum weight. The algorithm is a direct implementation of primal-dual algorithm [4] for the minimum cost flow problem. The primal-dual algorithm transforms the minimum cost flow problem with a single supply node *s* and a single demand node *t*. At every iteration, the primal-dual algorithm computes shortest path distance from s to all other nodes, update node potentials, and then solves a maximum flow problem that sends the maximum possible flow from node *s* to node *t* over arcs with zero reduced costs. When applied to the assignment problem, this algorithm terminates within n iterations since each iteration sends at least 1 unit of flow, and hence assigns at least one additional node in N_1.

2.7 A SIMILAR PROBLEM – PRICING NETWORK RESOURCES

The writer would like to introduce the problem because it is fairly similar to our problem. Although the algorithm, *the Optimal Distributed Algorithm for Pricing Network Resources* [26], to solve this problem is inapplicable for the optimisation model of brokering in barter business, the way of showing the issue in the round and reasoning to solve it deserve an attention. Fulp et al., 1998, gave us a brief survey on control theory of economy as follows [26].

Advances in computer network technology have resulted in complex networks that must accommodate a variety of network applications. These applications transmit a range of information, from simple text and graphics to complex interactive voice and video. Each application requires a certain Quality of Service (QoS), which may include bounds on the packet: delay, variation and loss. These service guarantees can be provided if the network resources are available, such as link bandwidth, buffer space and processor time. Since the amount of resources is finite, contention may occur. For this reason, networks need a method of flow control to manage resources in a fair and efficient manner. A distributed microeconomic flow control technique that models the network as competitive markets is introduced. In these markets switches price their link bandwidth based on supply and demand, and users purchase bandwidth so as to maximize their individual QoS.

Users can only enter the network economy through a network broker (NB). This entity is an agent for the user and is located between the user and the edge of the network. Representing the user in the economy the NB performs the following tasks: connection admission control, policing, and purchase decisions. Although the NB works as an agent for the user (making purchasing decisions), it is assumed that the NB operates honestly in regards to both the switches and the user. The NB controls network admission by initially requiring the user to have enough wealth to afford at least an acceptable QoS; otherwise, the user is denied access. The purpose of this requirement is to be certain all users are viable consumers in the market and to prevent overloading the economy. It is believed that the social welfare of the economy is better when it consists of fewer users each receiving a good QoS, instead of many users each receiving a poor QoS. Hence, there are attempts to maximize the number of users in the economy, where each user can afford an acceptable QoS.

There are two goals associated with flow control, fairness among applications and the balance between throughput and QoS [29, 30]. Defining fairness is difficult because of the various types of applications and their desired QoS. The balance between throughput and QoS is the concept that the network should seek high resource utilization, but not at the expense of poor QoS (and vice versa). Hence, due to heterogeneous networks, diverse resource requirements and the goals associated with flow control, proper flow control is a challenging problem. Several different methods of flow control have been proposed, some specifically for certain types of networks. The flow control methods based on economy are the following:

- An economic flow control method models the network as an economy, and then applies microeconomic principles for resource allocation. A simple network economy consists of two types of agents: consumers (network applications) and producers (switches). Consumers require resources to satisfy their QoS. Producers own the resources sought by consumers, and seek to maximize their satisfaction by selling or renting their resources. Using this framework, microeconomics can be used to define how network resources are allocated.

- One approach of applying microeconomics to computer networks involves a maximization of utility functions. A utility function maps a resource amount to a satisfaction value. Using this function, one can compare the satisfaction levels of different resource amounts. The maximization process determines the optimal resource allocation such that the utility of a group of users is maximized subject to budget and resource availability constraints. Accurately grouping users together may be problematic due to the wide variety of applications and their diverse resource requirements. Another problem is that these approaches generally require a centralized entity to determine the optimal allocation amount. This is undesirable because the economy relies on one entity, which is not reliable or fault tolerant.

- Another microeconomic approach, congestion pricing, charges users for their consumption of resources and resources are priced to reflect supply and demand. With such a model, prices can be set to encourage high utilization of network resources as well as a fair distribution. Users act independently, attempting to maximize their own utility and prices are set based on local resource conditions. It has been shown that pricing based on supply and demand results in higher utilization than traditional at (single) pricing. Prices of links in the system were iteratively adjusted until equilibrium of supply and demand was reached. The approach uses congestion pricing in a competitive market. Similar to other microeconomic flow control methods, the approach is decentralized, seeks an equilibrium price and achieves a Pareto optimal distribution (Pareto optimality is the allocation of finite resources such that no sub-set of users can improve on their allocation without lowering the utility of another). In addition, the approach maximizes individual QoS, adapts to network dynamics and is scalable to heterogeneous networks.

CHAPTER 3
SYSTEM ARCHITECTURE

3.1 SYSTEM DESCRIPTION OF BARTER TRADE EXCHANGES

The writer would like to introduce a system, which runs barter trade exchanges (see Figure 3.1). The activities and organization of system are depicted as follows.

When a client joins the barter trade exchange (BTE) system, he or she will access the Web site of this exchange to register identify and needs and wants list. The system will store such information in database Product and Service. The database is organized on the type of products and services of Universal Standard Products and Service Classification code (UNSPSC). The business profile is classified on the type of Standard Industrial Classification (SIC) or North American Industrial Classification Standard (NAICS). Currently UNSPSC has 14,306 items classified according to catalogs which have the structure below [13]:

XX Segment
The logical aggregation of families for analytical purpose

> XX Family
> A commonly recognized group of interrelated commodity categories
>
> > XX Class
> > A group of commodities sharing a common use or function
> >
> > > XX Commodity
> > > A group of substitutable products or service

NAICS has 1,810 items and its format is:

First two digit – sector
Third digit – subsector
Fourth digit – industry group
Fifth digit – NAICS industry
Sixth digit – US industry

5	1					Information
5	1	3				Broadcasting and Telecommunication
5	1	3	3			Telecommunications
5	1	3	3	2		Wireless Telecommunications Carriers (except Satellite)
5	1	3	3	2	1	Paging
5	1	3	3	2	2	Cellular and other Wireless Telecommunication

In the barter pool, all transactions must be recorded in database transaction history. This is necessary for the following prediction activity because thanks to recordkeeping, the broker can calculate commission and the government can impose tax on transactions

The system not only helps members barter products or services, but also generate new trade ideas to them since more transactions occur, more money the broker gets. Hence, forecast on potential buying and selling in near future of members is essential to maximize trade volume. The task should be periodically done. Kaewpitakkun, 2002, build a prototype of the prediction engine to carry out this task [14]. Note that clients in the BTE system can purchase for both personal and business case. Kaewpitakkun, 2002, focused on making prediction for business purchases only [14]. Since the business purchases are quite systematic, the personalization technique (i.e. collaborative filtering) alone may not suitable for this task. The probability model like Bayesian Network is used for this task.

However, besides to encourage member businesses in order to spend barter dollars by forecasting needs, the BTE system must pay due attention to trade balance of each member. The recommendation scheduling engine has to optimise predicted purchases to give the best recommendations to the businesses. This helps they spend or get barter dollars as much as possible as well as keeping their current balance near optimal balance. After that human brokers check the recommendations once more before they are sent to clients. This guarantees that the recommendations are feasible because in some cases human brokers need readjust them a little to be suitable for real situations. The BTE system also needs receive feedback from clients to improve the quality of recommendations. The human broker can change operational parameters of the recommendation scheduling engine to meet clients’ needs and benefits of the barter trade exchange.

Traditionally, such recommendations are sent to clients by telephone or fax; however, to make the system more automatic the writer suggests that the messages should be sent by email as well as the two conventional communication means.

When clients receive the recommendation, they can follow the recommendation, or ignore it and spend barter dollars till they use up. To address the undesired action, the task - Educate the client- of the broker should be paid due attention.

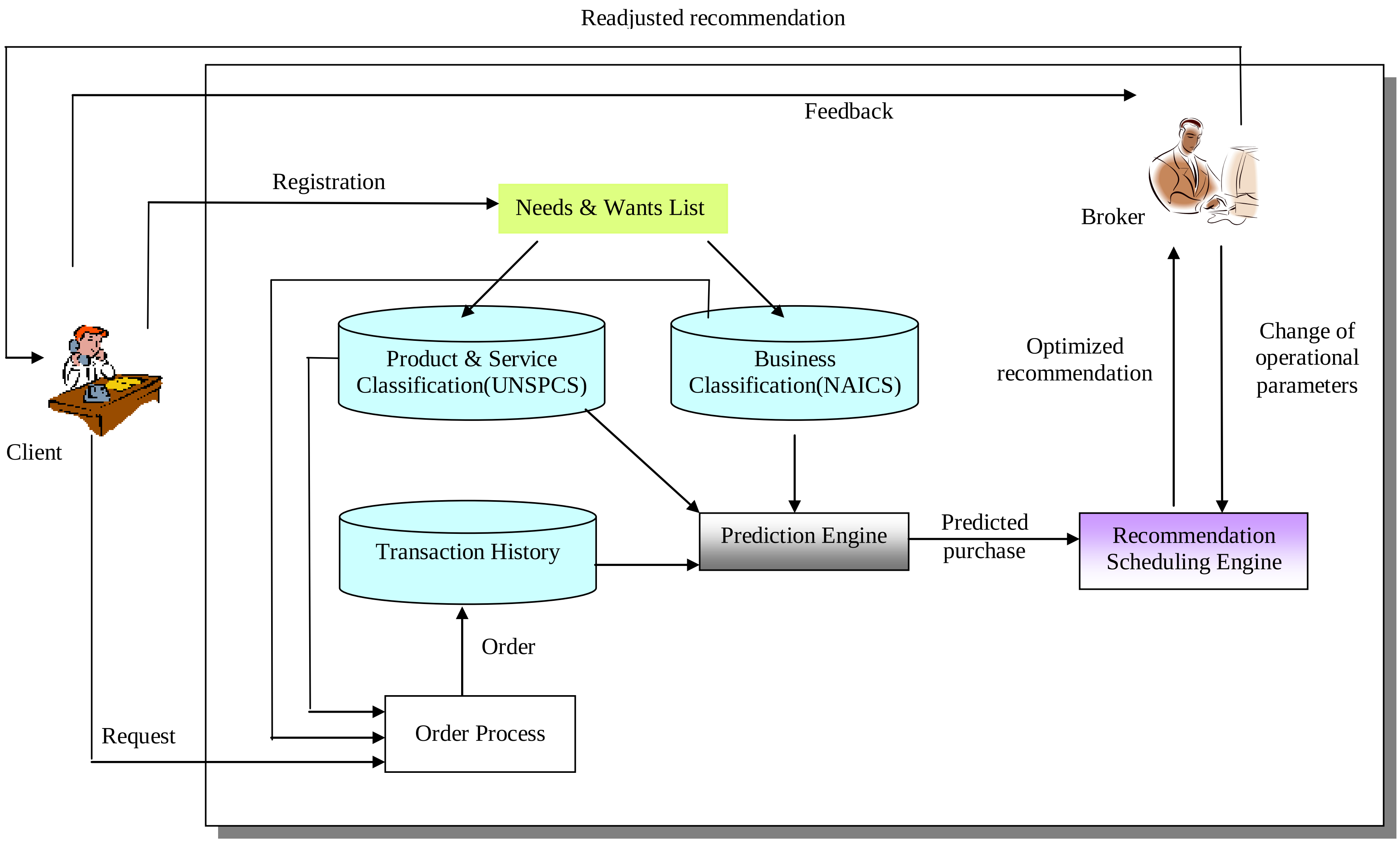


Figure 3.1: **The barter trade exchange system**

3.2 CORE TASKS OF THE BTE SYSTEM' S BROKER AGENT

The intermediary role of the broker provides a broad domain of tasks varying from setting the interface configurations to triggering a match on offers and demands. A broker agent is designed to meet these various tasks. The agent helps to run the barter system automatically by assuming the following tasks.

a) **Entrance control**: The task involves the first interaction activities between users and the BTE system. Users of the system are not only the human users, but also the agents interacting on behalf of their users. So far, the entrance protocols have been divided over the following subtasks:

   - Entrance controls concerning "New Users". This task concerns users who are interacting with the BTE system for the first time. The system requires some primary information from new users about their identity, needs and wants before entering the marketplace. In case a new membership is created, information such as login, password, and id-number are some of the primary attributes assigned to a user. The user profiles database is also informed about the new user and updated.

   - Entrance controls concerning "Known Users" (members security protocols). Here, the focus is on the virtual market members attempting to access the market place. In order to enter the BTE system, members need to identify themselves. Through authentication/authorization protocols, the members are given permission to perform activities in the system. Accessing product or service catalogs, buying, selling, or just looking around are examples of such user activities.

   - Entrance controls concerning "Subscription Activities". New subscriptions are created for interested users and the necessary procedures related to this new creation, such as payment for these subscriptions, generating new access rights or informing users about prices or other changes are included. Adding, removal, or updating subscriptions belongs to this task category.

b) **Demand/Offer generation**: The information received from customers (consumers and providers) concerning consumers' preferences and providers' specifications is being used in order to generate demands or offers compatible with the market place standards. The broker agent carries out the task as a prediction engine.

c) **Matching algorithms**: Information about available offers is used in order to find suitable matches for the demands in focus. The degree of (mis)match is used here to select the final (best) results acceptable for both consumers and providers. The agent performs the task as a recommendation scheduling engine. This task will be presented in details in Chapter 4 and 5.

d) **Transaction protocols**: After the consumer and provider confirm an agreement in order to exchange merchandise, the actual transaction must take place. Rules and regulations are used in order to execute the desired transaction protocols. In case a transaction takes place on a network (Internet), special secure and safe transaction protocols are applied in order to protect this process against unauthorized and illegal access. Different security methods such as encryption or public keys can be used.

e) **Internal safeguard** is guaranteed by taking care of the following subtasks.

- *Fraud inspection*: Based on the observations of the activities taken place in the market place, an inspection is performed in order to check for any kind of infringement or breach. Guidelines and standards are used to make the judgments. In case any violation occurs, the trespassers are warned and if necessary actions are taken by the market place against these violations (blocking user access rights, warning other parties about the user illegal activities and bad reputations…). By taking serious actions against breaches and violations, the virtual market indirectly provides some insurance against illegal activities and trespassers.
- *Guard transaction*: Guarding transaction involves validating and verifying the protocols used during transaction activities. Using more secure transaction protocols and monitoring the related activities within the market place for any unauthorized access are included here.
- *Quality determination*: Product or service categories are evaluated in order to specify the quality of merchandise provided in the barter trade exchange. For each product category, rules and standards are based on which products are evaluated. It is also possible to use information from different resources in order to achieve a fair level of product evaluation. Upon these resources, an average level of quality determination is reached.

f) **Information support**

Clients of the barter trade exchange can require and receive information about the market place, concerning general information (where & what) or more specific information, such as product categories. General and specific information about the marketplace facilities and activities are included within this category of tasks.

Clients can request information about different characteristics of the barter trade exchange. General market information, product categories information, or other facilities within the market place are some examples of the required information. This task is comparable to the task of an information (help) desk where people lost, or interested in other aspects of the market can get answer to their questions. Notice here that information support to known customers of the market place is also based on their past activities and their characteristics known to the barter trade exchange. This is where learning from/about customer becomes important in order to reach them in a more efficient way (narrow casting).

g) **Marketing**: Two types of marketing have been distinguished. From the first point of view, products are classified based on the customer's interests and activities. New strategies are derived in order to approach the customers in a more effective way. From the second point of view, the marketing strategies are derived within each product category. Within every product category, customers are classified in classes based on their interests on the product category. Applying marketing strategies like direct marketing [22], narrow casting can be enhanced and performed more effectively and user-friendly as well. Only people who are interested are approached by the marketing strategies and uninterested customers would be saved from tiresome and boring (advertising) information.

h) **Creation/removal of categories**: A category is added to/removed from the BTE system. Adding a new category requires designing a new ontology or reusing an existing one for the specific product type, specification of all corresponding product-related knowledge, defining evaluation protocols, and product specific match knowledge structures are some of the tasks which must be performed when defining a new category. In short, all the characteristics are identified and specified.

i) **Creation/removal of users**: This task helps to manipulate user information. When a customer subscription is expired, or the customer explicitly cancels a subscription, the corresponding user profile is modified and updated. For a new user the same type of procedures is applied in order to add the user to the user profile.

CHAPTER 4
OPTIMIZATION MODEL

4.1 MATHEMATICAL MODEL

In order to find a suitable model for this problem, firstly a model is started up as simple as possible. After finding such a model, it will be developed to satisfy real transactions.

In a barter pool, let the set of companies C= $\{C_i \mid i = 1\ldots m\}$ and the set of products be P = $\{P_j \mid j = 1\ldots n\}$. For simplicity, it is assumed that companies wish to trade in single units of a good. The requirement matrix is R = $\{r_{ij}\}$. Each cell of the matrix is +, -, or 0. These symbols are interpreted as follows: r_{ij} = + indicates company C_i has the product P_j available to supply, r_{ij} = - indicates company C_i has a need for product, while r_{ij} = 0 indicates company C_i has no interest in product P_j.

A directed link between a cell + and a cell – in the same column is the movement of a product from a supplying company to one needing the product.

Let T be a trade set of directed links in the requirement matrix. A maximal trade set T is a trade set of largest possible cardinality. This cardinality is determined as follows. For each product P_j, let s_j be the number of +, corresponding to the number of supplying companies and n_j be the number of -, corresponding to the number of needing the product. Then the largest cardinality that a trade set may have is

$$\sum_{j=1}^{n} \min(s_j, n_j)$$

Let t_j be $\min(s_j, n_j)$. Then obviously, the maximal trade set T has the number of links is

$$Z_T = \sum_{j=1}^{n} t_j$$

Let x_i be the number of products that C_i provides and y_i be the number of products that C_i consumes. The balance of company C_i is defined to be $x_i - y_i$. The absolute balance of C_i, $|x_i - y_i|$, is concerned because it represents the trade ability of the company in the future. The ideal credit balance is close to zero since at this status the company can participate in future trade as much as possible (see Section 2.3.2 for the explanation). The absolute balance of the maximal trade set T is defined as

$$ab_T=\sum_{i=1}^{m}|x_i-y_i| \quad \text{where m is the number of companies}$$

Also, the fair distribution of transactions for barter members is taken into account. It is most desirable that a client who has much trade ability will be allocated many recommendations of transactions. The physiology shows fairness in a marketplace.

The fairness distribution of transactions per company is given as follows:

For company C_i

Let x_i be the number of products available to supply

Let y_i be the number of products that the company needs

Trade ability of C_i in the barter pool is represented by the ratio of $\dfrac{x_i + y_i}{\sum_{i=1}^{m} x_i + y_i}$

With Z_T is maximal trade links as defined earlier, the desirable average number of links of C_i is $AL_i = \frac{\mathit{x}_i + \mathit{y}_i}{\sum_{i=1}^{m} \mathit{x}_i + \mathit{y}_i} * 2 * Z_T$

The fairness distribution of transactions of C_i is evaluated by the formula

$$| x_i + y_i - AL_i |$$

Therefore, the fairness of the maximal trade set T is $f_T = \sum_{i=1}^{n} | x_i + y_i - AL_i |$

Our goals are to minimize ab_T to ensure that each company reaches to zero balance in order to provide maximum future ability to participate in trade, and to minimize f_T to ensure fair participation in trade by companies.

The above discussion can be concisely represented as follows:

Consider matrix m x n

	P_1	P_2	….	P_n
C_1	+	-	….	0
C_2	+	+	….	-
.				
.				
.				
C_m	0	-	….	+

Consider row C_i

Let

x_i : the number of cell +

y_i : the number of cell -

x_i : the number of cells + which have a link to a cell - in another row; $x_i \leq \mathit{x}_i$

y_i : the number of cells -which have a link to a cell + in another row; $y_i \leq \mathit{y}_i$

$x_i + y_i$: the total number of + and - which have a link to a cell whose sign is opposite in another row

$x_i - y_i$: the difference between the number of + and that of -. These cells have a link to a cell whose sign is opposite in another row

$|x_i - y_i|$: the absolute balance of this row

The absolute balance of the matrix: $ab_T = \sum_{i=1}^{m} | x_i - y_i |$

Consider column P_j

s_j : the number of +

n_j : the number of -

$k_j = \max(s_j, n_j)$

The maximum number of links in column P_j : $t_j = \min(s_j, n_j)$

The maximum links of the matrix: $Z_T = \sum_{j=1}^{n} t_j$

The average link of row C_i: $AL_i = \frac{x_i + y_i}{\sum_{i=1}^{m} x_i + y_i} * 2 * Z_T$

The difference between the end-point number and the average link in row C_i:

$$| x_i + y_i - AL_i |$$

The fairness of the matrix: $f_T = \sum_{i=1}^{n} | x_i + y_i - AL_i |$

Find a combination of links in the matrix such that there are Z_T links and the minimization of ab_T and f_T.

4.2 JOB-SHOP SCHEDULING

One case of the problem that is related to job-shop scheduling is depicted below.

	P_1	P_2	….	P_n
C_1	+	0	….	+
C_2	-	-	….	0
C_{m-1}	0	+	….	+
C_m	-	-	….	-

All rows are + or 0, otherwise are – or 0. + and – are matched in each column such that the matching number is maximum and the objective function $ab_T + f_T$ is minimized. This is a case of job-shop scheduling with row + representing one machine and row – being one job. Each job may have one or many tasks and each task has a start time. It is given that if a cell – is at column i, then its start time is t_i and some machine may process this task if the capacity of the machine is still available at this time. If this happens, there exists one cell + in column i matching with that cell -. How many tasks are satisfied in a period is depend on the capacity of machines at that time.

When the scheduling is performed, it has to comply with criteria below

- Jobs are fairly processed by machines.
- Machines should are fairly operated.

This job-shop scheduling problem turns out to allocate machines to perform a collection of jobs such that performance measure is minimised while satisfying the machine capacity constraints, processing time requirements. The shown matrix is only one case of our problem, so the writer conjectures that the problem may be NP-hard.

In the following parts, the search methods for the problem will be introduced and evaluated.

4.3 EXHAUSTIVE SEARCH (ES)

One combination is defined as a set in a column, whose cardinality is equal to a maximum set, which has opposite sign. For example, column i is as follows

+	1
-	2
+	3
0	4
+	5
-	6

Set (2, 6) has sign ‘-‘. Therefore, there are combinations whose sign is ‘+’ are (1, 3), (1, 5) and (3, 5). The purpose is to find one of the three combinations in order to make the objective function as small as possible. Note that the problem becomes complex when the dimension of the matrix is larger than 5x5 since matching by hand become complicated. In such situations so many combinations in each column need considering accompanied with combinations of other columns at the same time.

The purpose of ES is to consider the running time of matrices with relatively small size. Also, its results help to evaluate the optimal degree of following heuristic algorithms since the search method will absolutely give global minimum if it can scan all states. More importantly, from matrices causing difference between the result of ES and that of heuristics, the writer can devise various search strategies to improve optimal degree.

DFS is employed in ES. Note that the depth of the search tree is limited because when one node representing a combination is selected; the search will not visit other nodes that are at the same as the column of the newly selected node in later moves. However the method consumes a great amount of time as recursive and backtracking techniques are used to scan all nodes [20]. It is also unlikely to use branch and bound to decrease the number of visited nodes because the objective function is the sum of two absolute functions ($ab_T + f_T$).

- **Complexity**

Not take into account cells that are 0. Also, not consider columns with the same type, i.e. columns whose cells are – or 0, or + or 0.

Assume that there are Z_T links

With respect to column P_j, there are $\binom{k_j}{t_j}$ combinations to permute links.

Therefore, there are $U = \prod_{j=1}^{n} \binom{k_j}{t_j}$ cases that need considering to pick up which case minimizes ab_T and f_T. If the value of U is very large, finding the best solution by ES requires so much computation because the complexity of the method is exponential. To illustrate, a matrix n x n is considered.

Assume that all columns have $\binom{n-k}{k}$ combinations so the complexity is $\binom{n-k}{k}^n \approx \theta(n^{kn})$

To illustrate, the writer implemented experiments on Pentium IV 2.0GHZ computer to consider three criteria: Balance, Fairness, and Balance plus Fairness. In appendix A, it took about 47 minutes to run a matrix 9x11 with 1.6E+8 solutions, and approximately 12h: 30m for a matrix 10x11 with 2.268E+9. Obviously a polynomial-time algorithm solution is desirable for the combinatorial optimisation problem.

4.4 IMPROVED HILL-CLIMBING SEARCH (IHCS)

True hill-climbing search does not offer the possibility of backtracking, but merely selects the best child of any node and ignores the others and so can never return to an earlier node and re-continue. This means that it is easy to implement and consumes very little storage space (as it does not store alternative nodes) but may well get stuck [17].

The search strategy implemented by this new function is a variant on *hill-climbing* that backtracks. To deal with plateau, all best successors of the current state are kept to backtrack for finding a better solution.

The results of the method prove that it can find global optimal in most experiments so far and outperforms the hill-climbing (see appendix B) but it also spends more time reaching solutions and more memory storing equally-best successors.

This method is depicted below.

	P_1	P_2	….	P_n	x–y[(1)]	x+y[(2)]	Avg. Link	Balance +Fairness[(3)]
C_1	+	-	….	0			AL_1	
C_2	+	+	….	-			AL_2	
C_m	0	-	….	+			AL_m	

The original matrix has three types of cells. The first is cells that are absolutely chosen and the second is cells that are optionally chosen, i.e. concerned cells. When a second-type cell is chosen, it becomes selected; otherwise it is unselected. The last type is unconcerned cells, i.e. 0 or columns whose cells are – or 0, or + or 0. An example of these types of cells is shown as follows.

	P_1	P_2	P_3	P_4	x-y	x+y	Avg.Link	Fairness	Balance+ Fairness
C_1	-	=	0	0	-1	1	1.2	0.2	1.2
C_2	-	0	*	-	1	1	1.2	0.2	1.2
C_3	=	*	-	-	0	2	1.8	0.2	0.2
C_4	*	-	=	-	0	2	1.8	0.2	0.2
								OF=	2.8

Legend

Balance = |x - y|		+	unselected cell +
Fairness = |x + y - Avg.Link|		*	selected cell +
OF = ∑Balance+∑Fairness		-	unselected cell -
	unconcerned cell	=	selected cell -
	commited cell		

Initially, calculate initial values of column balance (x - y)[1], and the total endpoints (x + y)[2] based on committed cells.

Define the value of each unselected cell is

$$|x'_i - y'_i| + |x'_i + y'_i - AL_i| - |x_i - y_i| - |x_i + y_i - AL_i|$$

with $|x_i - y_i| + |x_i + y_i - AL_i|$: the current value of row i and column [3]

$|x'_i - y'_i| + |x'_i + y'_i - AL_i|$: the value of row i and column [3] if such the cell is selected.

Let a combination in a column be a largest set matching with an opposite sign set. The value of a combination is the total of the value of cells belonging to it.

1. Find a combination that has the smallest heuristic value. Select this set and disregard its column.
2. Update the value of combinations in the remaining columns if there is any change.
3. If there are many combinations that the same smallest value, select one out of such combinations and store remaining those.

Go on with unselected combinations till all columns have one selected combination. Compare the final value of the objective function with the previously stored value of this function if it is available to choose the smaller result. If there is still one stored combination in step 3, the search backtracks with this combination and repeat step 1 (see Figure 4.1).

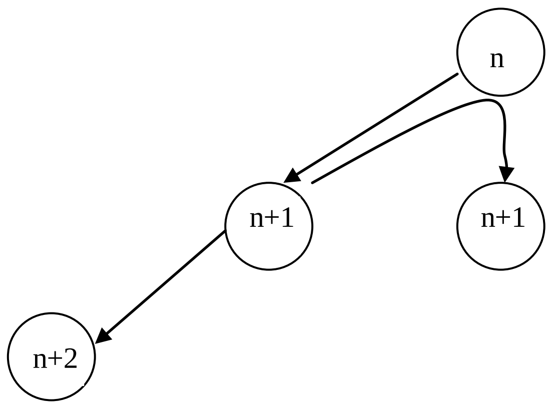


Figure 4.1: **Going back step n if there are two successors being the same good**

One drawback of the method is that when the plateau is too large and so many nodes are kept for backtracking, it shows inefficiency like ES. Furthermore, steps 1 and 2 require the comparisons and updates on combinations in remaining columns. This spends a great amount of time because when the dimension of the matrix increases, the number of combinations also grows exponentially (see Section 4.3). Hence, although the algorithm is polynomial-time, it cannot run on large matrices like 100x100. It is necessary to devise another method.

4.5 LOCAL REPAIR HEURISTICS

4.5.1 Description

In realistic problems a typical barter pool has hundreds of companies and thousands of products and services. In order to deal with such large matrices, e.g. 400x1000, it is necessary to find an algorithm satisfying the trade-off among runtime, memory capacity, and the optimal degree of solutions. The heuristic search algorithm consists of two phases:

Firstly, the initial state is established with committed cells. After that the selection of the next move is based on the value of cells instead of that of combinations. Also, if there are many equally best cells then one cell will be randomly selected in the next move. This saves memory and of course backtracking is not necessary here. A solution will come after a linear runtime because the number of cells increases proportionally with the size of matrix. Obviously the phase is HCS.

Secondly, pairs of cells are searched and exchanged to make the objective function decreased. One pair consists of two cells that are the same sign. One cell is selected and the other is unselected and they can be exchanged, i.e. they are in the same row or the same column. This phase will end when there are not such pairs left. Note that finding exchangeable cells can take many comparisons, and the number of these operations depends on each matrix. It is difficult to evaluate the complexity of this phase because a selected cell is substituted by an unselected cell, and later the former is likely to be exchanged again in order to reduce the value of the objective function. Breadth-first search (BFS) is used in the implementation of this phase. Experimental results show that the phase spends not much time and its runtime is almost linear (see evaluation of the method later).

The heuristic search algorithm is a kind of local repair heuristics with a variety of different search strategies. In this part, the writer only presents two search techniques:

HCS and BFS. In the next section, another search type will be added to improve optimal degree.

The algorithm outperforms IHCS because it is unnecessary to calculate all combinations of all columns. Also, it saves memory because it is unnecessary to store combinations for comparisons.

The method is presented in detail as follows.

	P_1	P_2	….	P_n	x–y (1)	x+y (2)	Avg. Link	Balance +Fairness (3)
C_1	+	-	….	0			AL_1	
C_2	+	+	….	-			AL_2	
C_m	0	-	….	+			AL_m	

Phase 1

- Calculate the initial state like IHCS (see Section 4.4).
- As for each column P_j, let s_j be the number of +, and n_j be the number of -. Let t_j be $\min(s_j, n_j)$. Assign the chance of the concerned cells to value t_j. For example, consider column 1

	P_1
C_1	+
C_2	-
C_3	+
C_4	0
C_5	+
C_6	-

  Cells (2, 1), (6, 1) are committed. At first, the chance of concerned cells (1, 1), (3, 1), and (5, 1) to create a direct link to one out of the two committed cells is 2.

- Chose a cell that has the smallest heuristic value, and mark it as a selected cell. If many cells are the same least weight, one of then will be randomly selected.
- Reduce the chance of the cells that have the same column as the selected cell by 1. If after the reduction, the chance of such cells is equal to 0, then all of them are not concerned for selection from now on (in this phase).

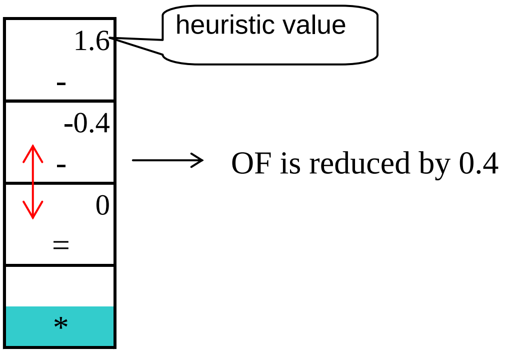
- Update cells of columns (1), (2), (3) and the value of cells that have the same row as the selected cell.
- Repeat this phase until there is not any concerned cell that needs selecting.

Phase 2

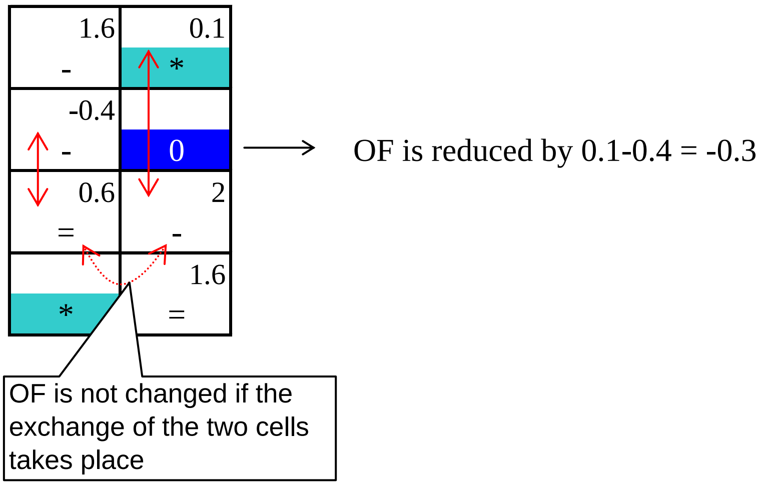
- Find pairs of exchangeable cells whose signs are the same. And, these exchanges at the same time make the value of the objective function decreased. Update value of concerned cells in the matrix after the exchanges happen.
- This phase ends when no combinations can be found.

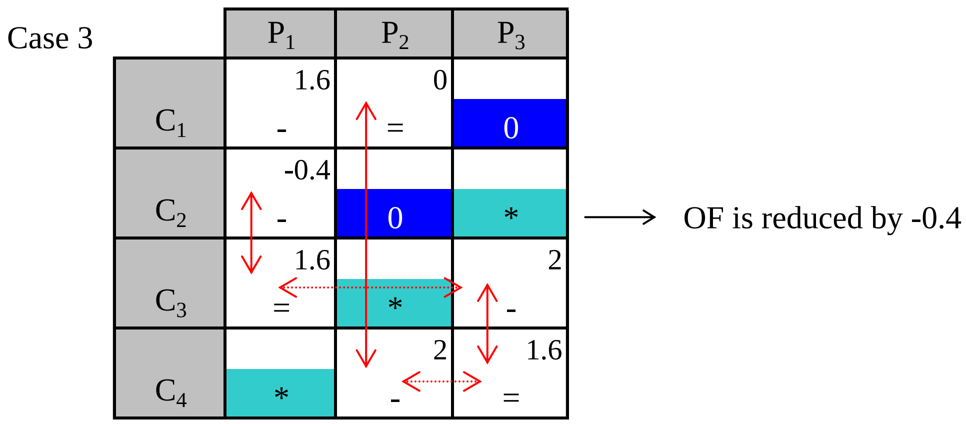
There are three basic kinds to exchange concerned cells in phase 2, which is illustrated by concrete examples as follows.

Case 1



Case 2





It is obvious that case 3 is a general case of case 1 and 2. Case 3 in the above example will be analysed to depict the method for searching pairs of cells that reduce the objective function (OF).

At first, since the heuristic value of unselected cell (2, 1) is -0.4, it is likely that OF is decreased if this cell is selected. For this reason, all selected cells in column 1 will be considered to check if one of them can be exchanged with (2, 1) and reduce OF. Only cell (3, 1) is already selected in column 1. However, since the sum of -0.4 and 1.6 is positive, the own exchange of (3, 1) and (2, 1) does not make OF decreased. It is necessary to find additional concerned cells to check a potential reduction of OF.

Unselect cell (3, 3) is checked for a next move. Notice that the role exchange, i.e. selected one becomes unselected one and vice versa, between (3, 1) and (3, 3) does not make OF decreased.

Next, selected cell (4, 3) is checked because it can be exchanged with (3, 3). However, till this time, the participation of (4, 3) in a search path still does not reduce OF since the sum of -0.4 and 1.6 is positive.

Then unselected cell (4, 2) is checked. It can be exchanged with (4, 3) and this does not cause the change of OF.

At last, selected cell (1, 2) is paid attention. It can be exchanged with (4, 2) and makes OF decreased since the sum of -0.4+0 is negative.

In short, the role exchanges at the same time of a chain of concerned cells, (2, 1), (3, 1), (3, 3), (4, 3), (4, 2), and (1, 2), reduce OF by -0.4.

As for intuition searching a path like the above example is relatively simple. In practice when we face matrices of big dimension like 400x1000, the search becomes extremely complex and time-consuming. It requires an efficient data structure and a cute search algorithm to solve such matrices. The writer performed the work as follows.

Initially four arrays containing concerned cells are created. The first array consists of unselected cells +. The second comprises selected cells +. The third is unselected cells -. The last contains selected cells -. The four arrays are sorted by descending order of the heuristic value. The search of a chain of concerned cells is carried out on the first array and the second one, or the third array and the fourth one. For example, Figure 4.2 shows the example of Case 3.

Array of unselected cells -

Cell	(2,1)	(3,3)	(4,2)
Heuristic value	-0.4	2	2

Array of selected cells -

Cell	(1,2)	(3,1)	(4,3)
Heuristic value	0	1.6	1.6

Figure 4.2: **Search of a chain of exchangeable cells -**

4.5.2 Complexity of phase 1

The number of iteration to select concerned cells is Z_T. Therefore, the number of iterations is reduced from $\prod_{j=1}^{n}\binom{k_j}{t_j}$ to $\sum_{j=1}^{n} t_j$. Hence, its complexity is $O(\sum_{j=1}^{n} t_j)$

In the best case, each column has only one choice. This causes $t_j=1$, $\forall j=1...n$. Its complexity is $O(n)$

In the worst case, each column has $\left[\frac{m}{2}\right]-1$ choices if m is even and $\left[\frac{m}{2}\right]$ choices if m is odd. Its complexity is $O\left(\frac{m*n}{2}\right)$.

Therefore, in the average case, its complexity is $O\left(\frac{m*n}{4}\right)$

4.5.3 A complete illustration

Matrix 4 x 4

-	-	0	0
-	0	+	-
-	+	-	-
+	-	-	-

Phase 1

Establish the initial state with committed cells

				x-y	x+y	Avg.Link	Fairness	Balance& Fairness
0 -	0 -	0	0	0	0	1.2	1.2	1.2
-0.4 -	0	*	-	1	1	1.2	0.2	1.2
-1.6 -	*	-1.6 -	-	1	1	1.8	0.8	1.8
*	-1.6 -	-1.6 -	-	1	1	1.8	0.8	1.8
							OF=	6

Change the best unselect cell into selected one (HCS)

				x-y	x+y	Avg.Link	Fairness	Balance& Fairness
0 -	0 -	0	0	0	0	1.2	1.2	1.2
-0.4 -	0	*	-	1	1	1.2	0.2	1.2
-1.6 -	*	-1.6 -	-	1	1	1.8	0.8	1.8
*	-1.6 -	-1.6 -	-	1	1	1.8	0.8	1.8
							OF=	6

Randomly select
the best cell

				x-y	x+y	Avg.Link	Fairness	Balance& Fairness
0 -	0 -	0	0	0	0	1.2	1.2	1.2
-0.4 -	0	*	-	1	1	1.2	0.2	1.2
1.6 =	*	2 -	-	1	1	1.8	0.8	0.2
*	-1.6 -	-1.6 -	-	1	1	1.8	0.8	1.8
							OF=	4.4

Randomly select
the best cell

Select the best cell

				x-y	x+y	Avg.Link	Fairness	Balance& Fairness
0 -	0 -	0	0	0	0	1.2	1.2	1.2
-0.4 -	0	*	-	1	1	1.2	0.2	1.2
1.6 =	*	2 -	-	1	1	1.8	0.8	0.2
*	2 -	1.6 =	-	1	1	1.8	0.8	0.2
							OF=	2.8

				x-y	x+y	Avg.Link	Fairness	Balance& Fairness
1.6 -	0 =	0	0	0	0	1.2	1.2	1.2
-0.4 -	0	*	-	1	1	1.2	0.2	1.2
1.6 =	*	2 -	-	1	1	1.8	0.8	0.2
*	2 -	1.6 =	-	1	1	1.8	0.8	0.2
							OF=	2.8

Phase 2

Find couples of exchangeable cells (BFS)

				x-y	x+y	Avg.Link	Fairness	Balance& Fairness
1.6 -	0 =	0	0	-1	1	1.2	0.2	1.2
-0.4 -	0	*	-	1	1	1.2	0.2	1.2
1.6 =	*	2 -	-	0	2	1.8	0.2	0.2
*	2 -	1.6 =	-	0	2	1.8	0.2	0.2
							OF=	2.8

				x-y	x+y	Avg.Link	Fairness	Balance& Fairness
0 -	0 -	0	0	0	0	1.2	1.2	1.2
0.4 =	0	*	-	0	2	1.2	0.8	0.8
1.6 -	*	1.6 =	-	0	2	1.8	0.2	0.2
*	1.6 =	2 -	-	0	2	1.8	0.2	0.2
							OF=2.8 - 0.4 + 0 =	2.4

(equal to the result of exhaustive search, i.e. global minimum)

4.5.4 Evaluation

The algorithm is evaluated according to optimality of the solution and runtime. As for the first criterion, the writer created a testing set. The testing set consists of matrices of dimensions 4x15, 5x14, 15x3, 13x4, 8x8, and 9x9. These matrices are randomly generated with varying ratios of 0/+/-. For example, if 0% is 0, +% will be from 5 to 50 with an increment of 5. If 0% is 10 and +% is 20, then -% is 70. For simplicity, the writer considers $ab_T + f_T$ as the objective function. Table 4.1 shows a sample ratio table of a matrix kind.

Table 4.1**: A ratio table**

0% \ +%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0.15	0	0	0.2	0
10	0	0	0	0	0	0.133	0.047	0	0.463	
20	0	0	0	0	0	0.305	0.017	0.047		
30	0	0	0	0	0.185	0	0			
40	0	0	0	0	0.183	0				
50	0	0	0	0	0					
60	0	0	0.112	0.05						
70	0	0	0							
80	0	0								

The value of each cell is calculated from 10 matrices having ratios of 0, +, and –, which indicated by its position in this table. The values of their objective function, which are collected from the heuristic algorithm, are summed to give out a total. Similarly, the values of their objective function, which are collected from ES, are summed to give out a total. The difference of these two totals is divided by 10 to get an average difference for this cell.

Therefore, there are 440 matrices for each kind of the above matrices. The total number of matrices experimented is 2,640 matrices. The reason for choosing such matrices is that they can run with ES within an acceptable runtime.

In order to evaluate the hardness of the problem the ratio tables of such matrix kinds are built. It is recognised that there is no common pattern of these tables. This means that no fixed ratios of 0 and + make large average differences between the heuristic search algorithm and ES when all matrix kinds are concerned. Even if various testing sets of one matrix kind are tested, their ratio tables also are varied. The detail of this experiment is presented in Appendix C.

The experiment shows many interesting things. The total runtime to test 2,640 matrices with the heuristic search algorithm was very small; however, it took about one week to finish ES if the experiment is carried out on a Pentium IV 2.0GHZ computer. Particularly, matrices 9x9 whose 0% is 0 consumed a great amount of time.

Figure 4.3 shows an increase of over 30% of the global minimum when Phase 1 is compared with Phase 2. After Phase 2 the algorithm gets near optimal.

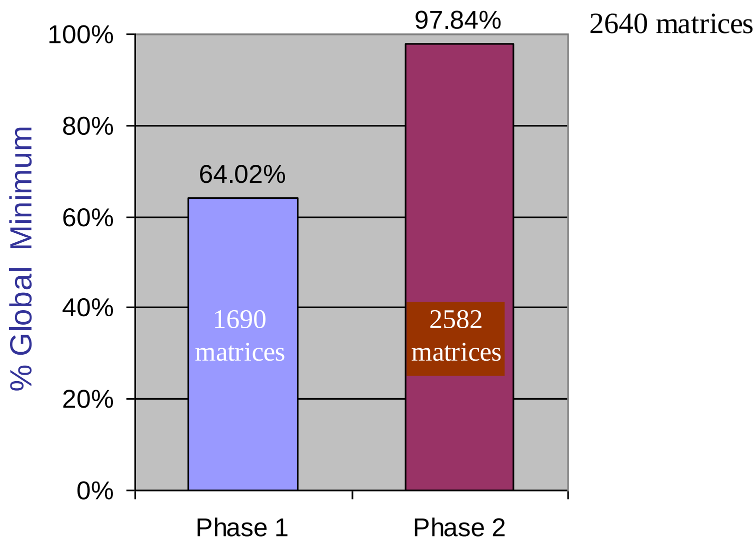


Figure 4.3: **Global minimum ratio of Phase 1 and Phase 2**

Figure 4.4 depicts the different ratios of local repair heuristics in the range that is defined from 0% (equivalent to the best values or global minima) to 100% (equivalent to the worst values). The best and worst values are derived from ES. As to each matrix kind, 440 matrices give the values of the heuristic search algorithm and ES. Therefore, each matrix has three values. The different ratio of this matrix is derived from the percentage of the value of the heuristic search algorithm in the range of the best value and the worst value. After that the 440 ratios are averaged. According to Figure 4.4, the largest different ratio is of 8x8 (0.241) and the smallest is of 15x3 (0%). Surprisingly the ratio of 9x9 is smaller than that of 8x8 although the dimension of the former is larger than that of the latter. The ratios prove that the heuristic search algorithm give results that are near the best.

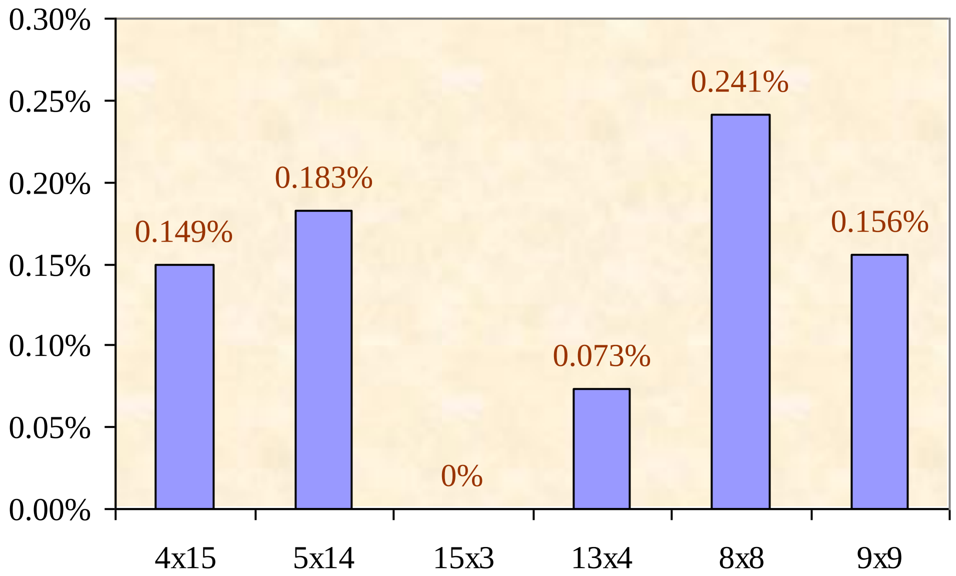


Figure 4.4: **Different ratios of the heuristic search algorithm in the range defined from 0% (the best values or global minima) to 100% (the worst values)**

As to the evaluation of the runtime of the algorithm, three matrices 400x1000 were randomly created with varying ratios of 0,+, and -. To deal with such big matrices, the writer build an C++ based application and using appropriate data structures like pointers, dynamic arrays. Complier Microsoft Visual C (MVC) allows the capacity of memory allocation very large. Unlike Turbo C 2.0 or Borland C 5.0, MVC not only takes physical memory but also the free space of hard disk for memory allocation. This partially eliminates the lack of memory.

The result of runtime is very good. This justifies that the algorithm is can be implemented to solve realistic problems. Table 4.2 shows the results of objective function and runtime of the three matrices. Notice that the results of runtime prove that the algorithm run linearly in practice.

Table 4.2**: The first experiment of three matrices 400x1000**

400 x 1000	Phase 1		Phase 2		Total time
	OF	Time	OF	Time	
0%=80 +%=7	4,130.13	11s	4,032.85	0s	11s
0%=60 +%=15	7,158.96	45s	6,235.83	5s	50s
0%=20 +%=35	32,407.55	2m:49s	9,247.06	1m	3m:49s

4.6 TECHNIQUE TO OVERCOME LOCAL MINIMUM

The above results show that the algorithm needs improving a little in optimal degree. It makes up 97.84% of global minimum of the testing set. After Phase 2 some cases get stuck in local minimum. To deal with the problem, some good search strategy needs devising.

Note that after Phase 2 each row has its own balance and fairness. It is likely that the largest balance and fairness can be reduced. However, the concerned cells of the corresponding row cannot be exchanged with cells of other rows to make the value of the objective function decreased more. It should be probed the reduction ability of the objective function by exchanging the role of one cell of this row with another of other row. After the exchange happens, if a cell of the former row can be exchanged with another cell of some row and this makes the value of objective function decreased more than its original value, then the process will be returned phase 2. Otherwise, the state must return to initial state, i.e. before the probe takes place. And, another cell of the row is tried on for the same purpose. If all concerned cells of the row cannot make its balance and fairness reduce, the process will go to the next-largest balance-fairness row and the probe for reduction continues.

In order to understand more clearly in abstraction, Figure 4.5 depicts moving up one level from local minimum in order to try to find a downward way.

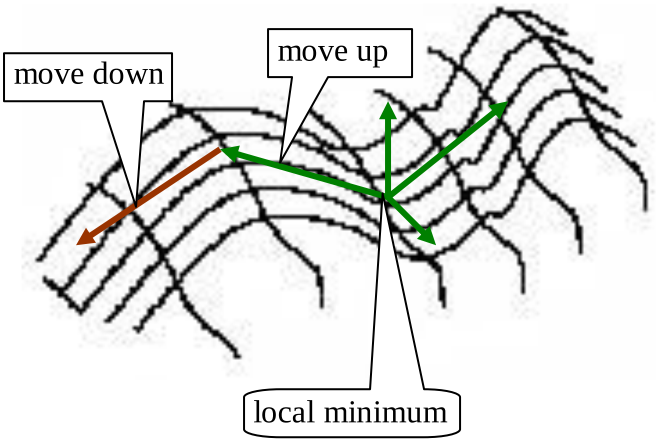


Figure 4.5: **Overcome local minimum**

The following example concretely illustrates the way to overcome local minimum.

Matrix 4x 4

-	+	-	-
-	0	-	+
-	+	-	+
-	-	+	+

After phase 2

				x-y	x+y	Avg.Link	Fairness	Balance+ Fainess
-	1.32 *	1.32 -	=	0	2	1.64	0.36	0.36
-	0	0.18 =	0.18 *	0	2	1.09	0.91	0.91
-	0 +	0 -	0 +	0	0	1.64	1.64	1.64
-	=	1.6 *	0.72 +	0	2	1.09	0.36	0.36
							OF=	3.27

The largest balance and fairness

				x-y	x+y	Avg.Link	Fairness	Balance+ Fainess
-	1.32 *	1.32 -	=	0	2	1.64	0.36	0.36
-	0	-0.18 -	0 *	1	1	1.09	0.91	1.09
-	0 +	0 =	-1.28 +	-1	1	1.64	1.64	1.64
-	=	1.6 *	0.72 +	0	2	1.09	0.36	0.36
							OF=3.27+0.18+0=	3.45

OF is increased

				x-y	x+y	Avg.Link	Fairness	Balance+ Fainess
-	1.32 *	1.32 -	=	0	2	1.64	0.36	0.36
-	0	0 -	0 +	0	0	1.09	1.09	1.09
-	0 +	1.28 =	1.28 *	0	2	1.64	0.36	0.36
-	=	1.6 *	0.72 +	0	2	1.09	0.36	0.36
						OF=3.45-1.28+0=		2.17

OF is reduced

The method called is Phase 3. It helps improve the ratio of global minimum by 2% in testing the 2640 matrices the same as the former experiment (see Figure 4.6).

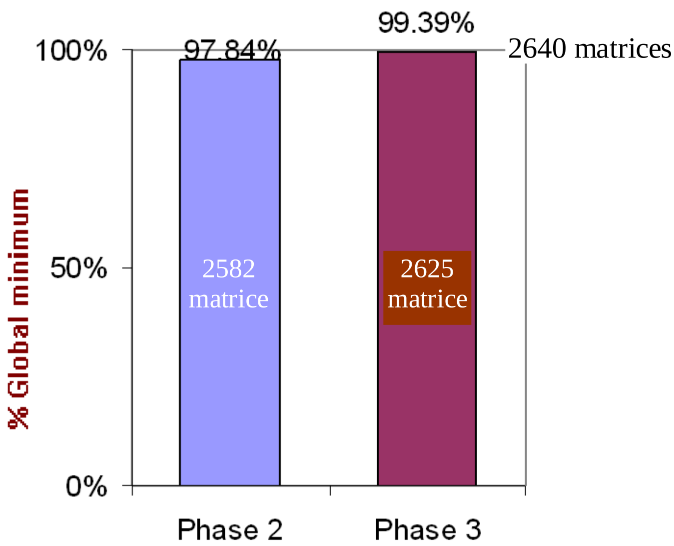


Figure 4.6: **Global minimum ratio of Phase 2 and Phase 3**

Figure 4.7 depicts average differences of the two experiments. Surprisingly, in the two experiments the average differences of 9x9 are smaller than that of 8x8. It is likely that the matrices 9x9, which were randomly created by accident, give results better than the matrices 8x8. Note that average differences in using Phase 3 are significantly reduced. The largest average difference is smaller than 0.013 and even as for matrices 15x3 this value is 0. The detail of this experiment is presented in Appendix C.

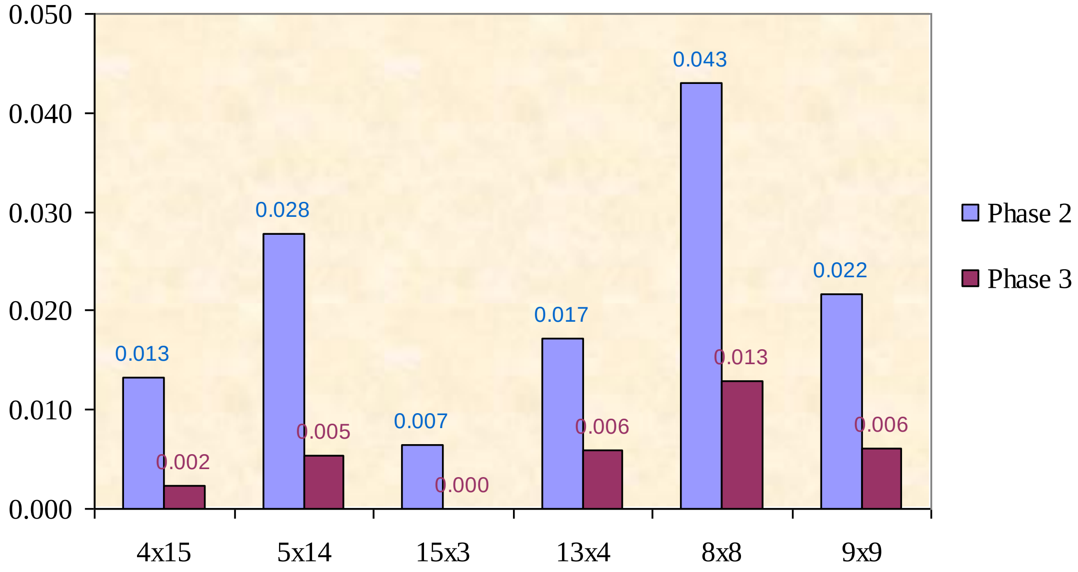


Figure 4.7: **Average difference of Phase 2 and Phase 3**

Note that the method is relatively similar to Simulated Annealing [12]; however, it does not use probability to perform moves. It also consumes a great amount of time. Particularly, it wastes time to perform phase 3 when global minimum is reached after Phase 2. Nevertheless, according to Section 2.3 no such general criterion exists in global optimisation for asserting if the global minimum has been reached. Hence, Phase 3 is still done.

Consider matrix m x n, there are about m x n trials to move up in the worst case. Such trials require many evaluations to check possible set-backs. If trial is unsuccessful, the previous state has to be recovered for next trials. For this reason, it is only practical for use only relatively small problems. For instance, three matrices 100x250 are tested and their results are presented in Table 4.3.

Table 4.3: **The second experiment of three matrices 400x1000**

100 x 250	Phase 1&2		Phase 3	
	Time	OF	Time	OF
0%=80 +%=7	0s	468.28	36s	468.28
0%=60 +%=15	0s	598.14	6m	597.61
0%=20 +%=35	0s	822.36	22m:18s	822.36

It is obvious that this method is inapplicable to solve realistic problems since in real situations, a barter trade exchange consists of hundreds of companies and thousands of products or services, the requirement matrix has to be enough big to contain all such parameters.

CHAPTER 5
PROTOTYPE IMPLEMENTATION

5.1 LINKING CONSUMERS TO SUPPLIERS

5.1.1 Statement of the problem

After the heuristic search algorithm creates the allocation matrix, concretely linking consumers to suppliers is not done yet. The purpose of this stage is to link buyers to sellers and minimize the number of links. This means that consumers will buy goods from the number of suppliers at least as possible. The following mathematical model represents this purpose.

Notice that after the requirement matrix is optimised, there is an allocation matrix. In the allocation matrix, each column has the number of demand equal to that of supply.

Consider column i.

Let x_1, x_2... x_u be the demand quantities of u consumers.

Let y_1, y_2... y_v be the supply quantities of v suppliers.

These values are integral and non-negative

There are the following constraints

$$\sum_{t=1}^{u} x_t = \sum_{k=1}^{v} y_k$$

$$\sum_{t=1}^{v} x_{tk} = x_k \quad \forall k=1,..,u$$

$$\sum_{t=1}^{u} x_{kt} = y_k \quad \forall k=1,..,v$$

Minimise the number of $x_{ij} > 0$, $\forall i=1..v, j=1..u$

(or maximise the number of $x_{ij} = 0$, $\forall i=1..v, j=1..u$)

There are many ways to solve the problem. The writer would like to introduce the simple and intuitive method below.

1. Create two arrays. One contains the demand quantities and another consists of supply ones.
2. Sort the demand array with descending order.
3. Sort the supply array with descending order.
4. Find all cases in which demand quantities are equal to supply ones. Link such pairs and eliminate they from the two arrays
5. Take the largest element of the demand array and that of supply array. There are two cases.
    - If the demand element is larger than supply one, then link the latter to the former and search one element out of the remaining elements of the supply array, whose value is equal to the remaining value of the demand element. If

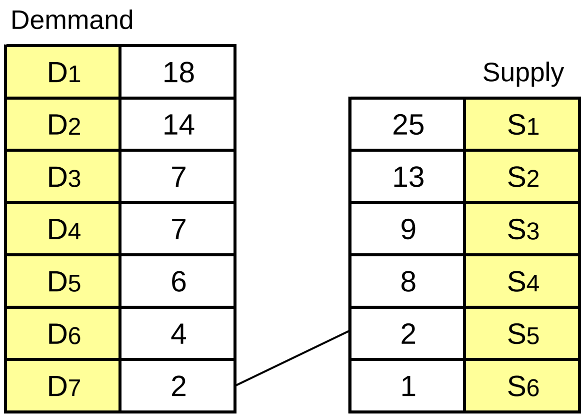
a supply element is found, the demand element is linked to the supply one. Otherwise, the next largest supply element(s) is/are linked to the demand until the demand quantity is equal to supply quantities. After that, all of the linked elements are deleted from the arrays.

- If the demand element is smaller than supply element, then link the latter to the former. The former is eliminated from its array. The remaining quantity of the latter will be supplied for the following links.

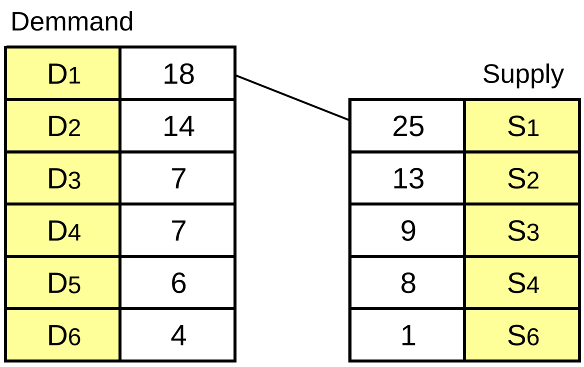
6. If the demand array still has element(s), go back to step 3.

Note that the method prefers large demand quantities to small demand ones because the number of links to supply elements of large demand elements is minimized before smaller demand ones are concerned.

5.1.2 Illustration



S5 supplies 2 units to D7



S1 supplies 18 units to D1

Demmand

D2	14
D3	7
D4	7
D5	6
D6	4

Supply

7	S1
13	S2
9	S3
8	S4
1	S6

sort

Demmand

D2	14
D3	7
D4	7
D5	6
D6	4

Supply

13	S2
9	S3
8	S4
7	S1
1	S6

S6 supplies 7 units to D3

Demmand

D2	14
D4	7
D5	6
D6	4

Supply

13	S2
9	S3
8	S4
1	S6

S2 supplies 13 units to D2

S6 supplies 1 units to D2

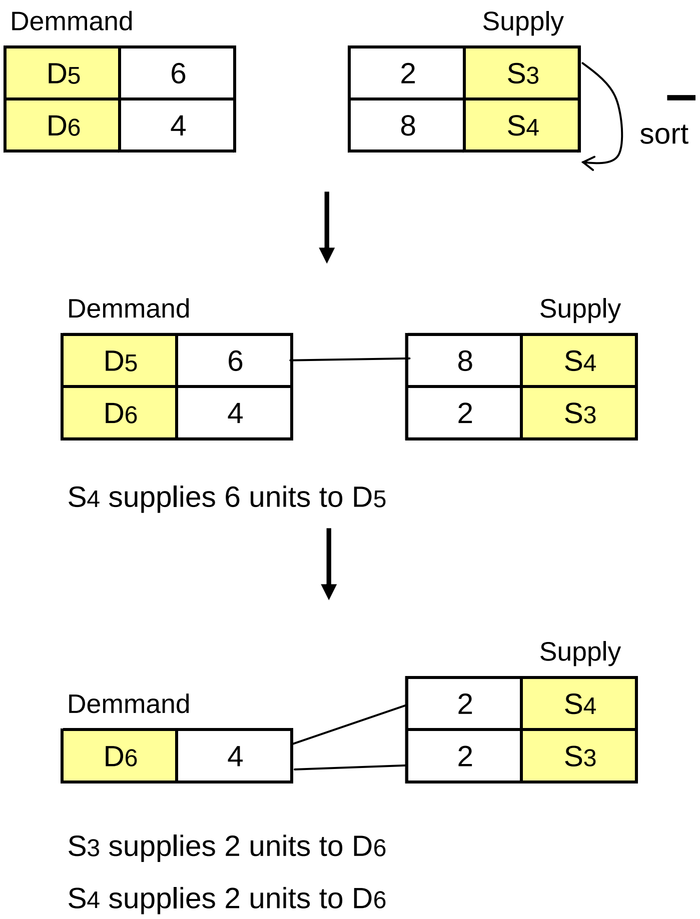
Demmand

D4	7
D5	6
D6	4

Supply

9	S3
8	S4

S3 supplies 7 units to D4



5.2 QUANTITY PROBLEM

In order to solve realistic problems that contain quantity of goods, the basic algorithm needs changing a little. The value of concerned cells is not only +, - but also many +'s' or –'s'. How many +'s' or –'s' is depended on the value. For example, if it is 3, then three signs of + will be created.

Below is the complete illustration of a quantity problem.

Customers' needs created by the prediction engine :

C_1	needs sell	needs buy
	4 P_1 4 P_4 4 P_5	5 P_2 3 P_6

C_2	needs sell	needs buy
	1 P_2 3 P_3 7 P_5	5 P_1 4 P_6

C_3	needs sell	needs buy
	6 P_1 1 P_5	1 P_2 2 P_4 5 P_6

C_4	needs sell	needs buy
	1 P_1 2 P_2 6 P_6	6 P_3 7 P_5

C_5	needs sell	needs buy
	1 P_2	7 P_4
	5 P_3	2 P_5
	4 P_6	

Requirement matrix 5 x 6 is derived from the predicted needs :

4	-5	0	4	4	-3
-5	1	3	0	7	-4
6	-1	0	-2	1	-5
1	2	-6	0	-7	6
0	1	5	-7	-2	4

Allocation matrix :

Phase 1

						x-y	x+y	Avg. Link	Fairness	Balance+ Fairness
4	-4	0	4	4	-3	5	19	15.83	3.17	8.17
-5	1	3	0	4	-3	0	16	15.83	0.17	0.17
1	0	0	-1	1	-4	-3	7	11.88	4.88	7.88
0	2	-6	0	-7	6	-5	21	17.42	3.58	8.58
0	1	3	-3	-2	4	3	13	15.04	2.04	5.04
									OF=	29.84

Phase 2

						x-y	x+y	Avg. Link	Fairness	Balance+ Fairness
1	-3	0	4	4	-3	3	15	15.83	0.83	3.83
-5	1	3	0	4	-3	0	16	15.83	0.17	0.17
4	-1	0	0	1	-4	0	10	11.88	1.88	1.88
0	2	-6	0	-7	6	-5	21	17.42	3.58	8.58
0	1	3	-4	-2	4	2	14	15.04	1.04	3.04
									OF=	17.50

Recommendations created by the linking method are

C_1	should sell	should buy
	1 P_1 for C_2	1 P_2 of C_2
	3 P_5 for C_4	2 P_2 of C_4
	4 P_4 for C_5	3 P_6 of C_4
	1 P_5 for C_5	

C_2	should sell	should buy
	1 P_2 for C_1	1 P_1 of C_1
	3 P_3 for C_4	4 P_1 of C_3
	4 P_5 for C_4	3 P_6 of C_4

C_3	should sell	should buy
	4 P_1 for C_2	1 P_2 of C_5
	1 P_5 for C_5	4 P_6 of C_5

C_4	should sell	should buy
	2 P_2 for C_1	3 P_5 of C_1
	3 P_6 for C_1	3 P_3 of C_2
	3 P_6 for C_2	4 P_5 of C_2
		3 P_3 of C_5

C_5	should sell	should buy
	1 P_2 for C_3	4 P_4 of C_1
	4 P_6 for C_3	1 P_5 of C_1
	3 P_3 for C_4	1 P_5 of C_3

5.3 EXTENSION OF THE PROBLEM

When applied in reality, the proposed algorithm needs have additional modifications:

1. Since the objective function is $ab_T + f_{T,}$ the priority of fairness and that of absolute balance is the same. However, the two priorities depend on the policy of each barter trade exchange. It may be that the barter trade exchange pays more attention to the fairness than the absolute balance or vice versa.

   For the general case, the objective function should be changed as follows:

   $$a*ab_T + b*f_T$$

   with a: priority of the absolute balance

   b: priority of the fairness

   a and b are parameters of the algorithm, and their value will be given in each concrete case.

2. Since each company has its own current balance, the current balance must be taken into account when the minimization of the absolute balance is concerned. Let cb_i be the current balance of company C_i. The company's balance plus fairness of this company should be changed as follows:

   $$|x_i - y_i + cb_i| + |x_i + y_i - AL_i|$$

3. For simplicity, we ignored the quantity of product in the requirement matrix. In reality, products or services of companies have their quantity in the requirement matrix. At the time, each cell of the matrix can contain many '+'s and many '-'s. Hence, it is necessary to have a procedure to create a suitable number of + and that of – for concerned cells with respect to each column. One reasonable proposal herein is the smallest allocation unit of each product should be the greatest common measure of the concerned cells of this column. One example of the proposal is as follows.

Consider column P_i

P_i
100
0
250
-200

The greatest common measure is 50. Therefore, the column is expressed below

P_i
+ +
0
+ + + + +
- - - -

It makes no sense if 100 is shown by 100 +, and -200 is 200 -. It wastes a great amount of time if the requirement matrix is optimised with such a representation.

4. Each product has its own price corresponding to a number of barter dollars. For example one unit of product rice corresponding to 10 barter dollars, and one unit in service laundry corresponding to 3 barter dollars. Furthermore, although in the same column, a product of different companies might have different price stipulated by each company. For these reasons, price need concerning in calculating absolute balance and fairness.
5. To create a link between a supply side and a demand one, we need consider trade terms of both sides. For instance, if business A needs a computer whose price is from 800 USD to 1,500 USD, then matching does not take place between A and manufacturer B whose computer price is over 1,500 USD.
6. Each business is granted an operating range and its barter activities in the exchange trade should be fluctuated in the range. Because of this, if the business supplies or demands one product, then additional value of barter dollars due to the transaction should not exceed the operating range. In the mentioned algorithm, when one cell is selected, it should be sure that the credit balance is still in the company's operating range.

Chapter 4 deals with a solution for a basic problem, which focuses on single units. This means barter members wish to trade in single units of goods. This chapter will develop the solution to solve realistic problems that involve quantities of goods, linking buyers to sellers, customers' credit balance, and so on.

5.4 PROCESS MODELLING

Figure 5.2 depicts the course of recommendation creation. The prediction database that was created by the prediction engine is the source of information for the recommendation scheduling engine. The engine will carry out all processes in the figure. If a process of the engine gets data or creates a new database, then it must perform such actions through ODBC.

Initially, the engine will input customers' predicted needs and then get price of products. It is likely that each customer has his own price for his products. Therefore, process 2.0 uses customer identification (ID) and product ID to get price from product database. Also, process 3.0 uses customer ID to get current balance from customer database. After that, process 4.0 collects the list of customers' predicted needs, price, and current balance to run optimisation work. The outcome of the process is optimised purchases. Process 5.0 then creates a recommendation database from the optimised purchases.

A following action which is not be mentioned herein is that after the recommendation database is created, the human broker can browse this database and make any necessary changes before the messages are sent to customers.

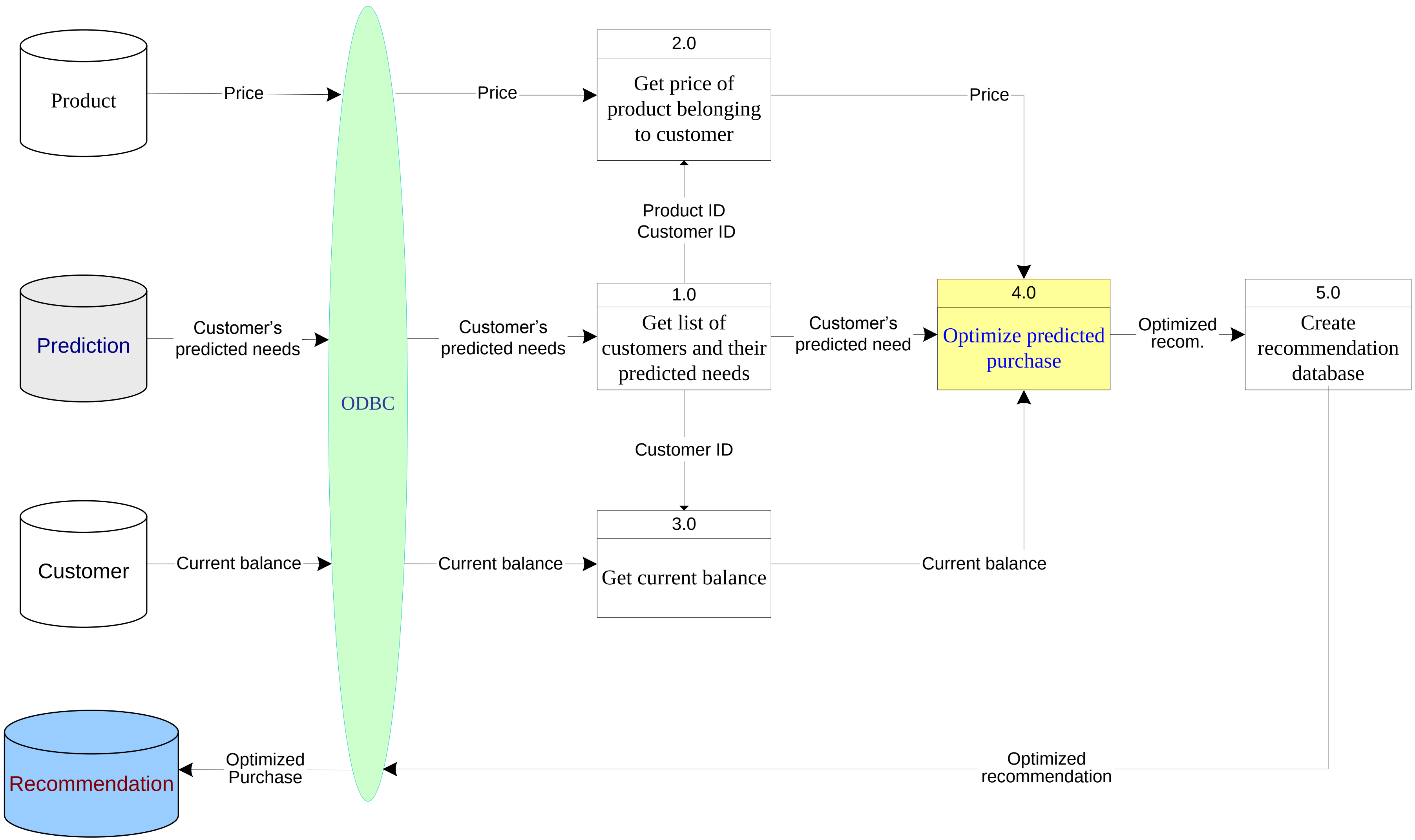


Figure 5.1: **Data flow diagram for creation of recommendations**

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

Barter business has increasingly developed in recent years under the form of barter trade exchanges. The trade exchanges disseminate best practices and new trading opportunities to enhance growth and profitability of barter members. The exchange provides a service - organizing the marketplace and matching buyers and sellers - for which it is compensated by the members. The trade exchanges earn income from commissions that are expressed as a percentage of the gross value of each transaction.

However, surprisingly, there is no complete software for automating the barter process, particular in matching buyers and sellers such that the trade exchanges can maximize trade volume over the long run and fairly allocate transactions to each member business. Most of transactions are still made through human broker so far. This irrelevance makes the barter trade exchanges cannot reach their maximum profits.

Therefore, this thesis focuses on finding a feasible solution of the matching problem and the implementation of the recommendation scheduling engine. Local repair heuristics with a variety of different search strategies is a suitable answer for this problem in realistic trade activities of today's barter business. The application to represent the recommendation scheduling engine should be coded with a low level programming language like C++ to solve realistic problems within an acceptable runtime.

If the engine is put into effect, it will increase benefits of both clients and barter trade exchanges. It helps customers predict needs of their own and where they can make transaction. Next, it is also a helpful assistant of human brokers. It helps them save time, energy in matching supply and demand. It also maximizes their revenue by actively encouraging trade activities in a fair behavior and hold of an account balance for all barter members.

6.2 FURTHER WORK

Although the proposed heuristic algorithm gains good optimal degree in the experiment of 2,640 matrices (about 97% of global minimum) and fast runtime, additional search methods needs seeking to improve optimal degree as well as keeping an acceptable runtime.

At the same time of the above improvements, the integration of other modules of the system like the e-catalog management, the prediction engine should take place soon to evaluate the running of the recommendation scheduling engine.

From feedback of users, human brokers and clients who receive recommendations, the optimization model will need readjusting for realistic trade activities of the barter trade exchange. For example, the priority of ab_T and f_T will be considering together to find suitable ratios (see Section 5.1) for each barter trade exchange.

REFERENCES

[1] Haddawy, P., (2002), “An Optimization Model of Brokering in Barter Trade Exchanges”

[2] Guttman, R. H., Moukas, A. G., and Maes, P., (June 1998), “Agent-Mediated Electronic Commerce: A Survey”, Knowledge Engineering Review, Vol. 13, No. 3

[3] Guttman, R. H., Moukas, A. G., and Maes, P., (April 1998), “Agent-Mediated Electronic Commerce: An MIT Media Laboratory Perspective”, Proceedings of the International Conference on Electronic Commerce (ICEC'98*),* Seoul, Korea

[4] Papadimitriou, C. H., and Steiglitz, K., (1998), “Combinatorial Optimization Algorithms and Complexity”, Prentice Hall Publishing, New Jersey

[5] Turban, E., and Aronson, J. E., (2001), “Decision Support Systems and Intelligent Systems”, Prentice Hall Publishing, New Jersey

[6] Turban, E., King, D., Lee, J., Warkentinn, M., and Chung, H. M., (2002), “Electronic Commerce 2002 A Managerial Perspective”, Prentice Hall Publishing, New Jersey

[7] Whisler, K., and Sullivan, J., (1996), “101 Ways to Grow Your Business with Barter – A Guide to Thriving in the 90s and Beyond”, WPR Publishing

[8] Profit from Waste International, (2001), Available at <URL: http://www.profit-from-waste.com>

[9] Ran, Y., Roos, N., and Herick, J. V. D., (2002), “Methods for Repair Based Scheduling”, Proceedings of the Twenty First workshop of the UK PLANNING and SCHEDULING Special Interest Group, PLANSIG 2002, pp. 79-86.

[10] Jonston, M. D., and Minton, S., (1994), “Analyzing a Heuristic Strategy for Constraint-Satisfaction and Scheduling”, Intelligent Scheduling, ed. M. Fox and M. Zweben, (San Francisco: Morgan-Kaufmann), ISBN 1-55860-260-7, pp. 257-289

[11] Resnick, P., Zeckhauser, R., and Avery, C., (October 1994), “Roles for Electronic Brokers”, Twenty-Second Annual Telecommunications Policy Research Conference

[12] Russell, S. J., and Norvig, P., (1995), “Artificial Intelligence A Modern Approach”, Prentice Hall Publishing, New Jersey

[13] Le Anh Vu, (2002), “An E-Catalog Management Framework for Barter Trade Exchanges”, AIT Thesis

[14] Kaewpitakkun, Y., (2002), “Recommendation Engine for automating broker-mediated process”, AIT Thesis

[15] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B., (1993), “Network Flows Theory, Algorithms, and Applications”, Prentice Hall Publishing, New Jersey

[16] Kang, N., and Han, S., (2002), “Agent-based E-marketplace System for more fair and efficient Transaction”, Decision Support Systems Vol. 34, No.2, pp.157-165

[17] Thorton, C., and Boulay, B., (1997), “Artificial Intelligence through Search”, Available at <URL: http://www.cogs.susx.ac.uk/local/books/ai-through-search/ch04/ch04-uh-4.html>

[18] Kerwin, T., (May 2002), “Artificial Intelligence - The Virtual Lecture Project”, Available at <URL: http://www-jcsu.jesus.cam.ac.uk/~tdk22/project/hill.html>

[19] Vardabash, K (Executive Director), IRTA Statistics Press Release 12-Mar-2002, International Reciprocal Trade Association Ed, Available at <URL: http://www.irta.com>

[20] Cormen, T. H., Leiserson, C. E., and Rivest, R. L., (1997), “Introduction to Algorithms”, Mc Graw Hill Publishing, Nineteenth Edition

[21] Törn, A., and Žilinskas, A., (1989), “Global Optimization”, Springer-Verlag, Nineteenth Edition, Berlin Heidelberg New York London Paris Tokyo

[22] Ling, C. X., and Li, C., (1998), “Data Mining for Direct Marketing: Problems and Solutions”, American Association for Artificial Intelligence (www.aaai.com)

[23] Albers, M., Jonker, C. M., Karami, M., and Treur, J., (1999), “Agent Models and Different User Ontologies for an Electronic Market Place”, Fourth International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM’99

[24] Satllings, W., (2001), “Operating Systems”, Prentice Hall Publishing, Fourth Edition, New Jersey, pp. 75

[25] Jonker, C. M., and Treur, J., (1998), “Compositional Design and Maintenance of Brokers Agents”, Proceedings of the 15th IFIP World Computer Congress, WCC'98, Conference on Information Technology and Knowledge Systems, IT&KNOWS'98, pp. 319-332

[26] Fulp, E. W., Ott, M., Reininger, D., and Reeves, D. S., (1998), “Paying for QoS: An Optimal Distributed Algorithm for Pricing Network Resources”, Sixth International Workshop on Quality of Service, pp. 75-84

[27] Goldman, C. V., Kraus, S., and Shehory, O., (2001), “Agent Strategies: for Sellers to Satisfy Purchase-Order, for Buyers to Select Sellers”, Proceedings of MAAMAW01

[28] Goldman, C. V., Kraus, S., and Shehory, O., (2001), “Equilibria Strategies for Selecting Sellers and Satisfying Buyers”, Proceedings of Fifth International Workshop on Cooperative Information Agents (CIA '01). Modena, Italy

[29] Bersetkas, D., and Gallager, R., (1992), “Data Network”, Second Edition, Prentice Hall, New Jersey

[30] Gerla, M., and Kleinrock, L., (April, 1980), “Flow Control: A Comparative Survey”, IEEE Transactions on Communications, Vol. 28, No. 4, pp. 1

[31] Materna, J., (2003), ‘Berkeley bartering firm exchanges cash woes for purchase points’, January 24, 2003, Available at

<URL:http://www.bizjournals.com/sanfrancisco/stories/2003/01/27/smallb1.html>

[32] Bailey, J. P., and Bakos, Y., (1997), “An Exploratory Study of the Emerging Role of Electronic Intermediaries”, International Journal of Electronic Commerce, Vol. 1, pp. 7-20

[33] Segev, A., and Beam, C., (1999), “Brokering Strategies in Electronic Commerce Markets”, Proceedings of the first ACM Conference on Electronic commerce, pp. 167-176

[34] Zanakis, S. H., and Evans, J. R., (1981), “Heuristic Optimization: why, when, and how to use it”, Interfaces 11, pp. 84-91

[35] Sutton, R., (May 1997), “Job-Shop Scheduling”, Available at <URL: http://www-anw.cs.umass. edu/~rich/book/11/node7.html>

APPENDIX A
SOME RESULTS OF ES

- Matrix 9 x 11

+	-	0	+	+	-	+	-	0	+	-
-	-	+	0	-	-	0	+	+	-	0
+	+	0	+	+	-	-	0	+	0	-
-	+	-	0	-	+	+	+	-	-	0
0	+	+	-	-	+	-	+	0	+	-
+	-	0	+	+	-	+	-	-	+	-
-	0	-	+	+	+	0	+	0	+	+
-	+	-	+	0	+	0	+	-	0	+
-	+	0	+	0	+	-	+	0	+	-

It takes 46m: 48s:406ms to exhaustively search all 1.62E+008 solutions

Balance |x-y|

											x-y	x+y	Al	| x-y|
*	=	0	+	*	=	*	=	0	+	-	0	6	6.31	0.00
-	=	*	0	=	=	0	*	*	=	0	-1	7	5.61	1.00
*	*	0	+	+	=	=	0	*	0	=	0	6	5.61	0.00
-	*	-	0	=	*	*	+	=	=	0	0	6	6.31	0.00
0	*	*	=	=	*	=	+	0	+	-	0	6	6.31	0.00
*	=	0	+	*	=	*	=	-	*	-	1	7	7.01	1.00
=	0	=	+	*	+	0	+	0	+	*	0	4	5.61	0.00
=	+	=	*	0	*	0	+	=	0	*	0	6	5.61	0.00
=	+	0	+	0	*	=	*	0	*	=	0	6	5.61	0.00

Absolute balance: 2.00
Fairness: 5.12

Fairness |x+y-Al|

											x-y	x+y	Al	|x+y-Al|
*	=	0	+	*	=	*	=	0	+	-	0	6	6.31	0.31
-	=	*	0	=	=	0	+	*	=	0	-2	6	5.61	0.39
*	*	0	+	+	=	=	0	*	0	=	0	6	5.61	0.39
-	*	-	0	=	*	*	+	=	=	0	0	6	6.31	0.31
0	*	*	=	=	*	=	+	0	+	-	0	6	6.31	0.31
*	=	0	*	*	=	*	=	-	+	-	1	7	7.01	0.01
=	0	=	+	*	+	0	+	0	*	*	1	5	5.61	0.61
=	+	=	+	0	*	0	*	=	0	*	0	6	5.61	0.39
=	+	0	+	0	*	=	*	0	*	=	0	6	5.61	0.39

Absolute balance: 4.00
Fairness: 3.12

Balance & Fairness |x-y|+|x+y-Al|

											x-y	x+y	Al	|x-y|+|x+y-Al|
*	=	0	+	*	=	*	=	0	+	-	0	6	6.31	0.31
-	=	*	0	=	=	0	*	*	=	0	-1	7	5.61	2.39
*	*	0	+	+	=	=	0	*	0	=	0	6	5.61	0.39
-	*	-	0	=	*	*	+	=	=	0	0	6	6.31	0.31
0	*	*	=	=	*	=	+	0	+	-	0	6	6.31	0.31
*	=	0	+	*	=	*	=	-	*	-	1	7	7.01	1.01
=	0	=	+	*	+	0	+	0	+	*	0	4	5.61	1.61
=	+	=	*	0	*	0	+	=	0	*	0	6	5.61	0.39
=	+	0	+	0	*	=	*	0	*	=	0	6	5.61	0.39

Absolute balance: 2.00
Fairness: 5.12
Balance & Fairness: 7.12

- Matrix 10 x 11

+	+	-	0	0	+	+	-	-	+	+
-	-	-	0	+	+	-	+	+	+	0
-	-	-	+	0	+	0	+	+	0	+
+	+	+	0	+	-	+	+	+	0	+
0	0	+	+	+	-	+	-	0	+	-
+	+	-	-	-	0	+	+	+	+	-
0	-	+	+	+	0	+	+	+	+	+
-	-	0	+	+	+	0	+	+	+	-
-	+	0	+	-	+	+	+	+	0	0
-	+	+	+	+	+	-	+	+	0	+

It takes 12h: 28m: 55s:906ms to exhaustively search all 2.268E+009 solutions

Balance |x-y|

											x-y	x+y	Al	|x-y|
*	*	=	0	0	+	+	=	=	+	*	0	6	4.85	0.00
-	=	=	0	*	*	=	*	+	+	0	0	6	4.85	0.00
-	=	=	*	0	+	0	+	+	0	*	0	4	4.31	0.00
*	*	*	0	+	=	+	+	+	0	+	2	4	5.39	2.00
0	0	*	+	*	=	*	=	0	+	=	0	6	4.85	0.00
*	*	=	=	=	0	*	+	+	+	=	-1	7	4.31	1.00
0	=	*	+	+	0	+	+	+	+	+	0	2	5.39	0.00
=	=	0	+	+	*	0	*	+	+	=	-1	5	4.85	1.00
=	*	0	+	=	+	+	+	*	0	0	0	4	4.31	0.00
=	+	*	+	+	+	=	+	+	0	*	0	4	5.39	0.00

Absolute balance: 4.00
Fairness: 11.46

Fairness |x+y-Al|

											x-y	x+y	Al	|x+y-Al|
*	*	=	0	0	+	+	=	=	+	+	-1	5	4.85	0.15
-	=	=	0	*	*	=	+	+	+	0	-1	5	4.85	0.15
-	=	=	*	0	+	0	+	+	0	*	0	4	4.31	0.31
*	*	*	0	*	=	+	+	+	0	+	3	5	5.39	0.15
0	0	*	+	+	=	+	=	0	+	=	-2	4	4.85	0.31
*	*	=	=	=	0	+	+	+	+	=	-2	6	4.31	0.61
0	=	*	+	+	0	*	*	+	+	*	3	5	5.39	0.15
=	=	0	+	+	*	0	*	+	+	=	-1	5	4.85	0.15
=	*	0	+	=	+	*	+	+	0	0	0	4	4.31	0.31
=	+	*	+	+	+	=	+	*	0	*	1	5	5.39	0.39

Absolute balance: 14.00
Fairness: 2.67

Balance & Fairness |x-y|+|x+y-Al|

											x-y	x+y	Al	|x-y|+|x+y-Al|
*	*	=	0	0	+	+	=	=	+	*	0	6	4.85	1.15
-	=	=	0	*	*	=	*	+	+	0	0	6	4.85	1.15
-	=	=	*	0	+	0	+	+	0	*	0	4	4.31	0.31
*	*	*	0	+	=	+	+	+	0	+	2	4	5.39	2.85
0	0	*	+	*	=	*	=	0	+	=	0	6	4.85	1.69
*	*	=	=	=	0	*	+	+	+	=	-1	7	4.31	2.61
0	=	*	+	+	0	+	+	+	+	+	0	2	5.39	2.85
=	=	0	+	+	*	0	*	+	+	=	-1	5	4.85	1.15
=	*	0	+	=	+	+	+	*	0	0	0	4	4.31	0.31
=	+	*	+	+	+	=	+	+	0	*	0	4	5.39	1.39

Absolute balance: 4.00
Fairness: 11.46
Balance & Fairness: 15.46

APPENDIX B

COMPARISONS AMONG HCS, IHCS, AND ES

Table B.1: **The values of the objective function of some matrices are calculated by HCS, IHCS, and ES**

	Balance			Fairness			Balance plus Fairness		
	HCS	IHC	ES	HCS	IHC	ES	HCS	IHC	ES
6x6	6	6	6	2	2	2	8.5	8.5	8.5
5x6	2	2	2	0	0	0	4	4	4
9x11	8	2	2	3.71	3.12	3.12	13.69	7.12	7.12
10x11	8	4	4	4.38	2.67	2.67	15.46	15.46	15.46

Next, IHCS are compared with ES by the following method:

- Randomly generate matrices that have dimensions: 2x2, 3x3... 8x8 and 9x9.
- In each above kind, create 10 matrices corresponding to a ratio of 10%, 20%, 30% and 40% of + or -. The ratio of 0 is 10%. For example, if +% is 10%, then -% 80%. There are 8x10x 4 = 320 matrices in total.
- IHCS and ES are applied on such matrices.
- Depicts the experiment on 3-dimension charts. Axis x shows the dimension of matrix. Axis y shows the ratios of 10%, 20%, 30% and 40%. Axis z represents the difference between IHCS and ES.

Table B.2 shows the experiment in detail.

Table B.2: **The comparison between IHCS and ES**

		10%		20%		30%		40%	
		IHCS	ES	IHCS	ES	IHCS	ES	IHCS	ES
2x2	M1	0	0	2	2	2	2	2	2
	M2	0	0	4	4	0	0	0	0
	M3	0	0	2.67	2.67	0	0	0	0
	M4	0	0	2	2	0	0	2.67	2.67
	M5	0	0	0	0	0	0	0	0
	M6	2	2	0	0	2	2	0	0
	M7	0	0	0	0	0	0	0	0
	M8	0	0	0	0	0	0	0	0
	M9	0	0	0	0	2	2	0	0
	M10	0	0	0	0	0	0	2	2
3x3	M1	3.14	3.14	3.33	3.33	4	4	3.5	3.5
	M2	0	0	1	1	3.5	3.5	2.67	2.67
	M3	3.33	3.33	5	5	3.33	3.33	0	0
	M4	5	5	2.67	2.67	3	3	3.33	3.33
	M5	3.14	3.14	3	3	3.71	3.71	3	3
	M6	0	0	4	4	2.67	2.67	3.5	3.5
	M7	3.5	3.5	3	3	3.33	3.33	5.33	5.33
	M8	2.57	2.57	3	3	2	2	3	3
	M9	5.71	5.71	3.33	3.33	3	3	2	2
	M10	0	0	3	3	3.14	3.14	2.65	2.65
4x4	M1	5.54	5.54	6	6	1.85	1.85	6.86	6.86
	M2	3.85	3.85	6	6	4.53	4.53	.8	.8
	M3	4	4	3.87	3.87	3.43	3.43	4	4
	M4	0	0	4	4	4.53	4.53	5.2	5.2
	M5	0	0	3.71	3.71	3.85	3.85	2.4	2.4
	M6	4	4	0	0	4	4	6	6
	M7	5.14	5.14	6.67	6.67	5.73	4.8	4	4
	M8	0	0	6.4	6.4	3	3	7.43	7.43
	M9	4	4	3.85	3.85	4	4	8	8
	M10	0	0	0.92	0.92	3.38	3.38	4.31	4.31
5x5	M1	8.5	8.5	6.55	6.55	6	6	6.96	6.96
	M2	4.1	4.1	10.29	10.29	7	7	5.82	5.82
	M3	4.1	4.1	3.83	3.83	4	4	5.33	5.33
	M4	6	6	3.81	3.81	4.73	4.73	7.27	7.27
	M5	4.36	4.36	13.04	13.04	3	3	5.82	5.82
	M6	5.13	5.13	5.13	5.13	8.4	8.4	9.2	9.2
	M7	9.6	9.6	7.13	7.13	4.86	4.86	14	14
	M8	5.09	5.09	4.17	4.17	6	6	6.29	6.29
	M9	5.13	5.13	7.91	7.91	7.57	7.57	8.52	8.52
	M10	6.91	6.91	7.2	7.2	8	8	9.91	9.91
6x6	M1	10	10	8.06	8.06	11	11	6.94	6.94
	M2	5.47	5.47	5.33	5.33	9.6	8	11.33	11.33
	M3	5.77	5.77	2.58	2.58	5.5	5.5	6.7	6.7
	M4	5	5	4.88	4.88	7	7	3.61	3.61
	M5	5.89	5.89	5.45	5.45	4.18	4.18	8.41	8.41
	M6	8.23	8.23	11.48	11.48	5.43	5.43	4.24	4.24
	M7	0	0	5.81	5.81	6	6	11.76	11.76
	M8	6.25	6.25	5.03	5.03	5.82	5.82	8.79	8.79
	M9	6.24	6.24	6.4	6.4	7.74	7.74	10.67	10.67

		10%		20%		30%		40%	
		IHCS	ES	IHCS	ES	IHCS	ES	IHCS	ES
	M10	5.68	5.16	3.13	3.13	2.8	2.8	11.45	11.45
7x7	M1	3.2	3.2	8.93	8.93	13.14	13.14	6.13	4.98
	M2	7.47	7.47	5.12	5.12	9.14	9.14	7.3	7.3
	M3	6.7	6.7	6.7	6.7	4.95	4.95	9.45	9.45
	M4	7.24	7.24	9.71	9.71	12.8	12.8	16.77	16.77
	M5	7.62	7.62	9.12	9.12	6.19	6.19	10.88	10.51
	M6	6.22	6.22	5.04	5.04	8.1	8.1	11.21	11.21
	M7	5.52	5.52	2.86	2.86	8.4	8.4	7.91	7.91
	M8	8.5	8.5	7.14	7.14	9.33	9.33	7.43	7.43
	M9	4.26	4.26	6.42	6.42	8.61	8.61	8.37	8.37
	M10	5.18	5.18	8.45	8.45	2.78	2.78	5.77	5.77
8x8	M1	7.79	7.79	3.27	3.27	8.93	8.93	9.67	9.67
	M2	7.72	7.72	8.36	8.36	6.56	6.56	12.8	12.8
	M3	7.10	7.10	6.34	6.34	7.32	7.32	17.6	17.6
	M4	9.66	9.66	4.35	4.35	9.24	9.24	10	10
	M5	9.72	9.72	4.75	4.75	11.41	11.41	14	14
	M6	6.73	6.73	10.75	10.75	4.52	4.52	13.36	13.36
	M7	9.75	6.98	8.08	8.08	14.43	14.43	8.22	8.22
	M8	7	7	12.53	12.53	14	14	9.67	9.67
	M9	9.29	8.93	8.27	8.27	9.96	9.96	17.14	17.14
	M10	8.13	8.13	13.42	13.42	10.22	10.22	15.19	15.19
9x9	M1	14.14	14.14	9.41	9.41	10.48	10.48	7.75	7.75
	M2	7.35	7.35	11.30	10.75	8.94	8.94	6.73	6.73
	M3	5.68	5.68	8.62	8.62	10.34	10.34	13.62	13.62
	M4	12.17	12.17	10.63	10.63	16.11	16.11	19.67	19.67
	M5	8.49	8.49	15.32	15.32	11.86	11.86	11.18	9.64
	M6	8.9	8.9	16.23	16.23	6.29	4.86	8.86	8.86
	M7	6.67	6.67	9.38	9.38	4.57	4.57	7.75	5.75
	M8	8.37	8.37	7	7	6.39	6.39	15.78	15.78
	M9	4.54	4.54	9.6	9.6	9.29	9.29	11.29	11.29
	M10	8.97	8.97	10.83	10.83	17.64	17.64	11.03	11.03

Table B.3 depicts difference between the values of the objective function calculated by IHCS and ES. Figure B.1 and B.3 illustrate the difference table by 3-dimension graphs.

Table B.3: **Average difference between IHCS and ES**

	10%	20%	30%	40%
2x2	0	0	0	0
3x3	0	0	0	0
4x4	0	0	0.093	0
5x5	0	0	0	0
6x6	0	0.052	0.16	0.041
7x7	0	0	0	0.152
8x8	0.313	0	0	0
9x9	0	0.055	0.143	0.354

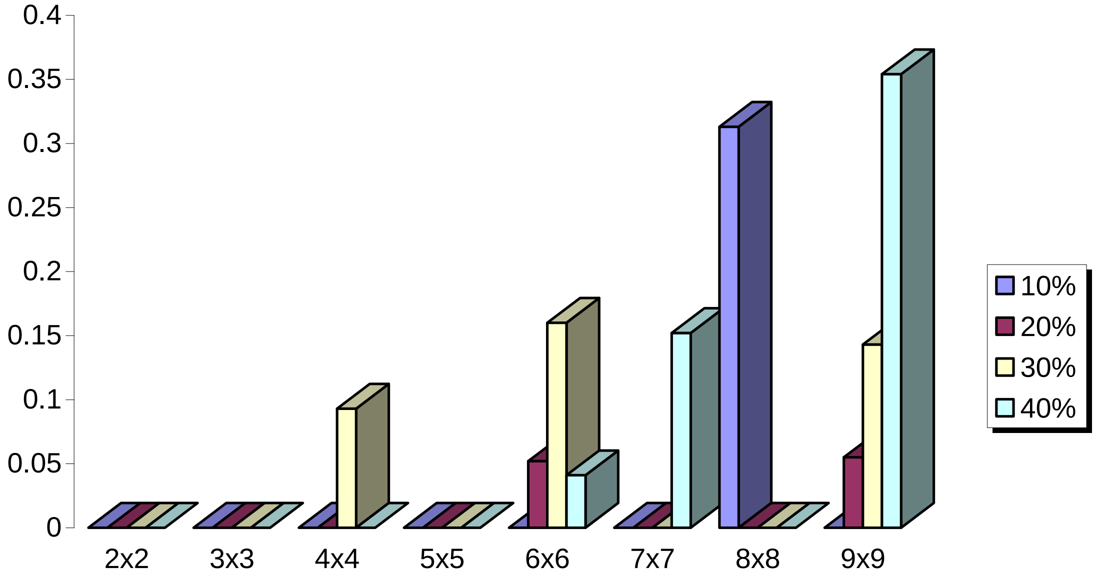


Figure B.1: **Demonstration of average difference by columns with depth**

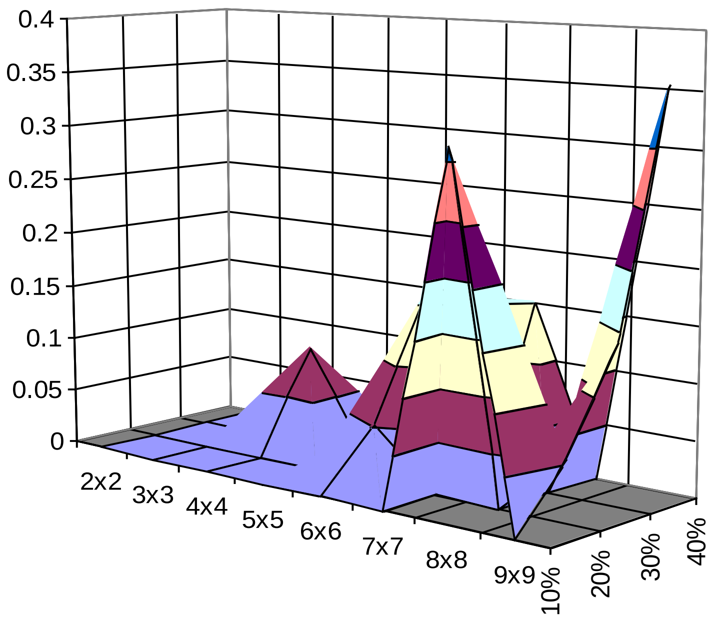


Figure B.2: **Demonstration of average difference by a surface chart**

APPENDIX C

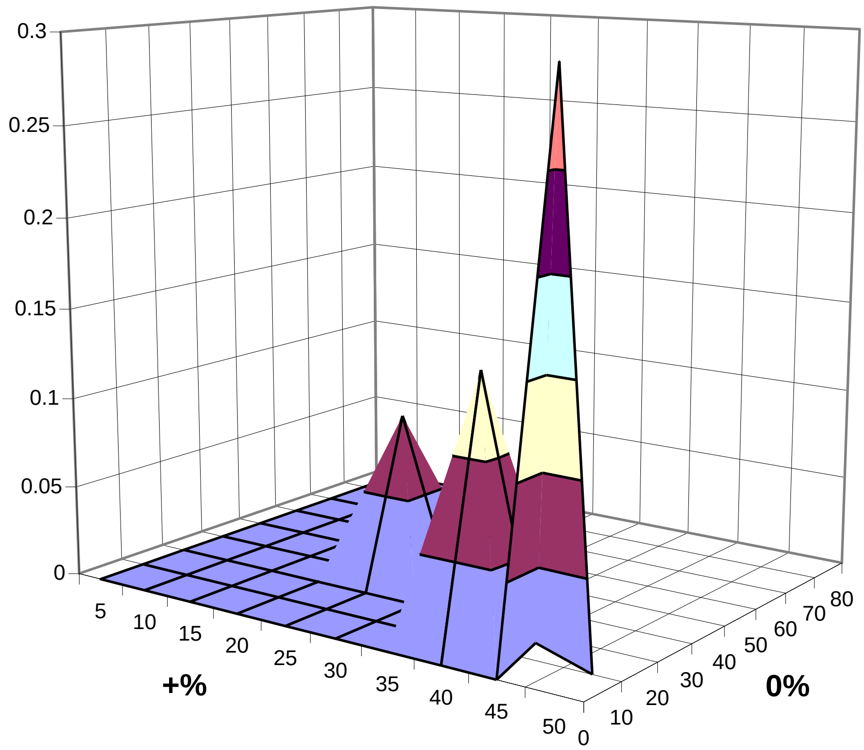
COMPARRISON BETWEEN PHASE 2 AND PHASE 3

440 Matrices 4x15

Results of phase 2

+% / 0%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0	0	0	0	0.3
10	0	0	0	0	0	0	0	0.143	0.009	
20	0	0	0	0	0	0	0.011	0		
30	0	0	0	0	0.092	0	0			
40	0	0	0	0	0.029	0				
50	0	0	0	0	0					
60	0	0	0	0						
70	0	0	0							
80	0	0								

7 out of 440 matrices cause differences. Ratio :1.59%

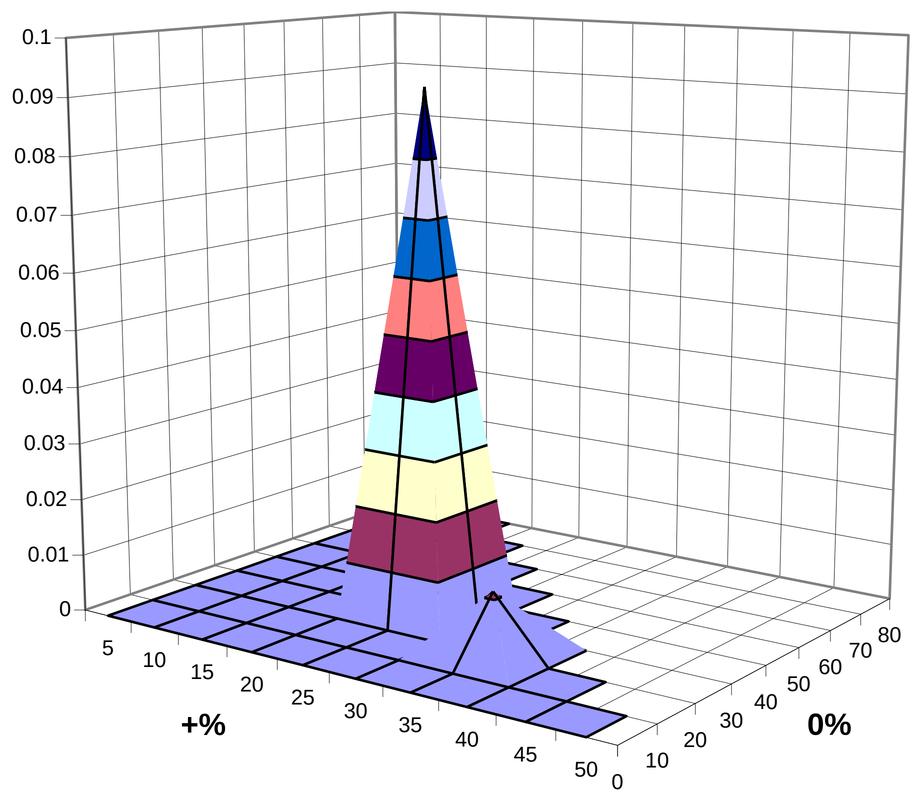


Average Difference between
Local Repair Heuristics and Exhaustive Search

Results of phase 3

0% \ +%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0.011	0		
30	0	0	0	0	0.092	0	0			
40	0	0	0	0	0	0				
50	0	0	0	0	0					
60	0	0	0	0						
70	0	0	0							
80	0	0								

2 out of 440 matrices cause differences. Ratio :0.46%



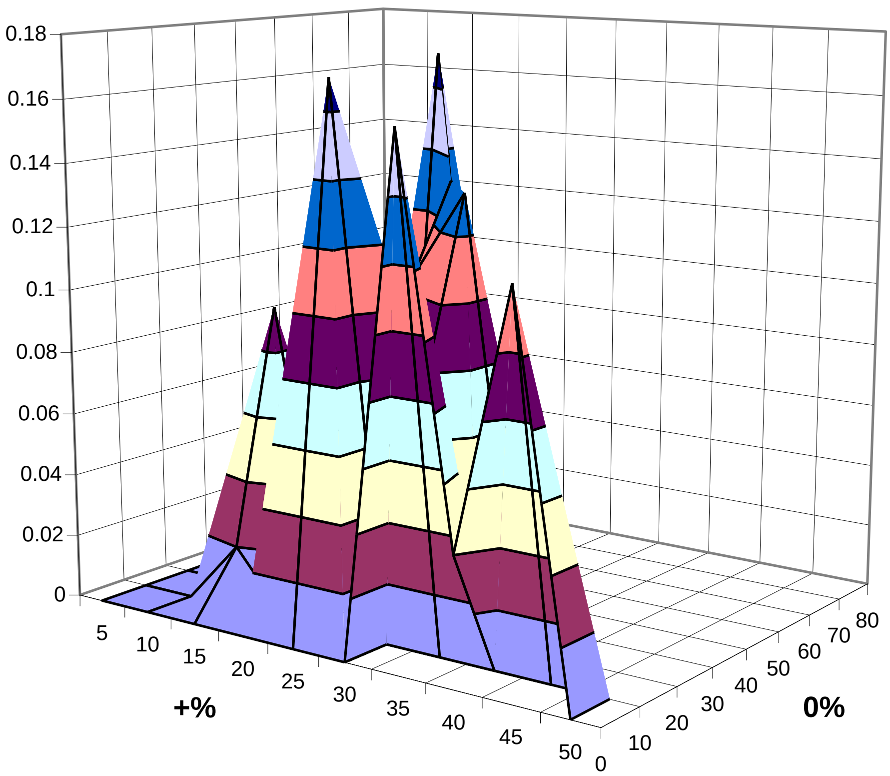
Average Difference between
Local Repair Heuristics and Exhaustive Search

440 Matrices 5x14

Results of phase 2

0% \ +%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0	0.16	0.04	0.12	0
10	0	0	0.02	0	0.17	0	0	0	0	
20	0	0	0.094	0	0	0.078	0	0		
30	0	0	0	0	0.105	0.133	0			
40	0	0	0	0	0.133	0				
50	0	0	0	0.171	0					
60	0	0	0	0						
70	0	0	0							
80	0	0								

12 out of 440 matrices cause differences. Ratio : 2.73%

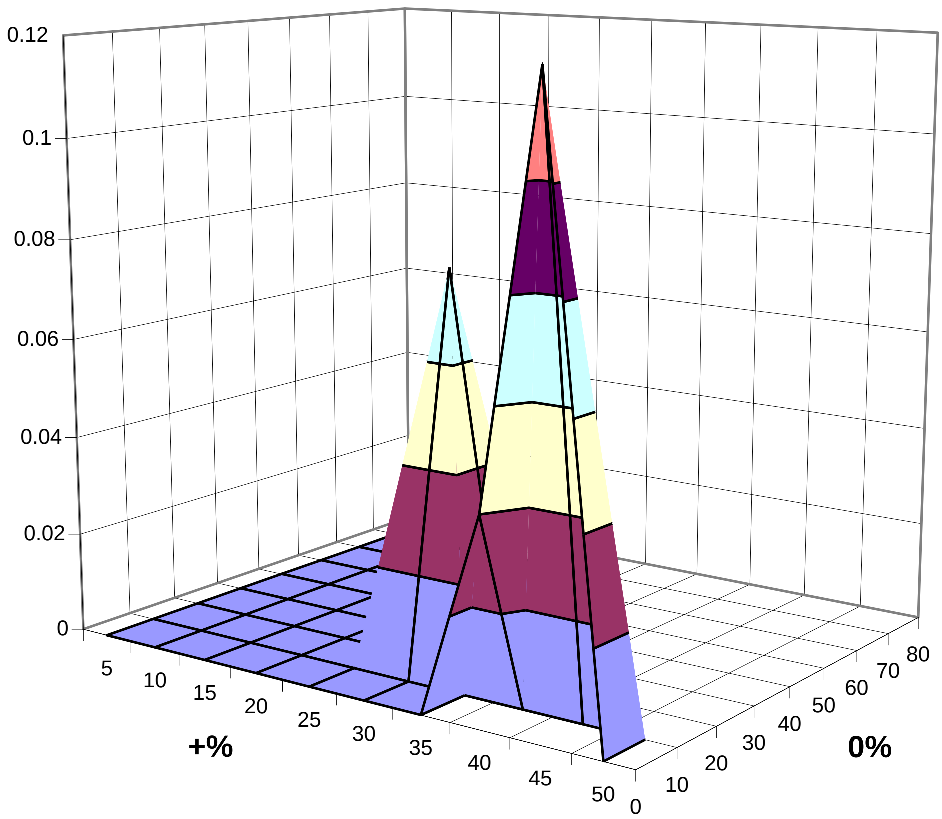


Average Difference between
Local Repair Heuristics and Exhaustive Search

Results of phase 3

0% \ +%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0	0	0.04	0.12	0
10	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0.078	0	0		
30	0	0	0	0	0	0	0			
40	0	0	0	0	0	0				
50	0	0	0	0	0					
60	0	0	0	0						
70	0	0	0							
80	0	0								

3 out of 440 matrices cause differences. Ratio :0.68%



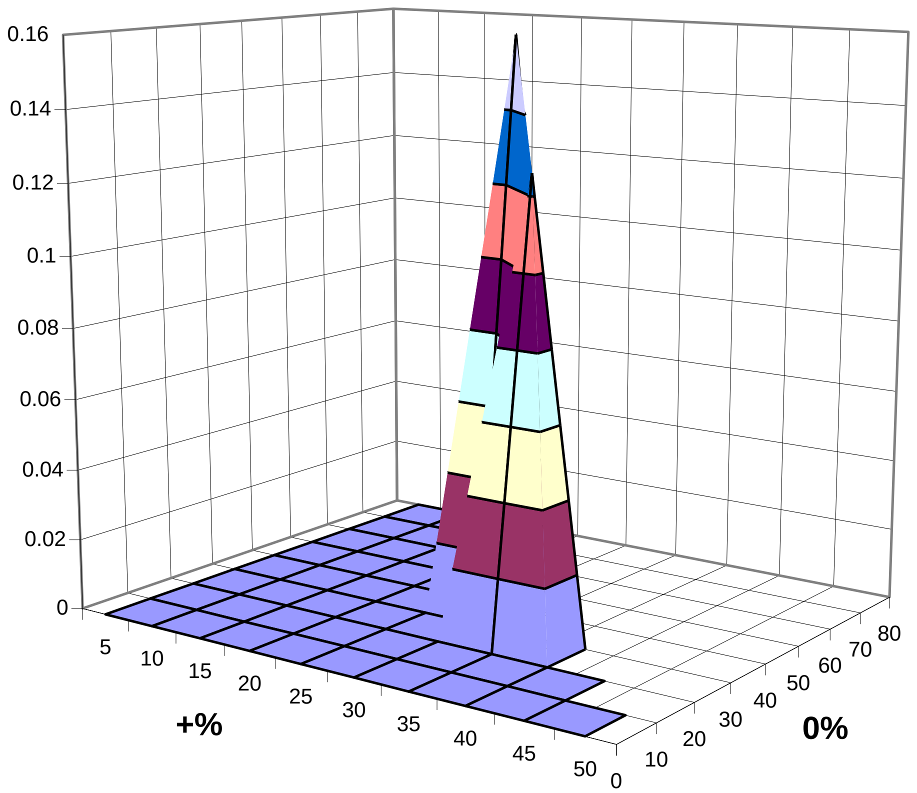
Average Difference between
Local Repair Heuristics and Exhaustive Search

440 Matrices 15x3

Results of phase 2

+% / 0%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0		
30	0	0	0	0	0	0	0.126			
40	0	0	0	0	0	0.16				
50	0	0	0	0	0					
60	0	0	0	0						
70	0	0	0							
80	0	0								

2 out of 440 matrices cause differences. Ratio : 0.46%

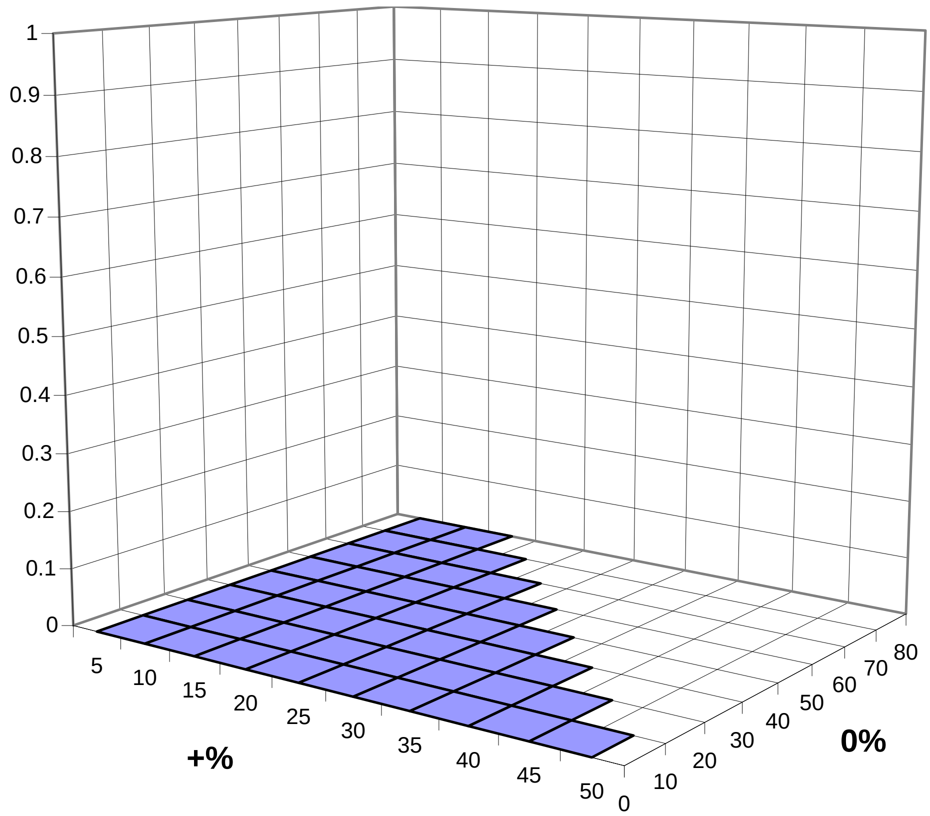


Average Difference between
Local Repair Heuristics and Exhaustive Search

Results of phase 3

0% \ +%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0		
30	0	0	0	0	0	0	0			
40	0	0	0	0	0	0				
50	0	0	0	0	0					
60	0	0	0	0						
70	0	0	0							
80	0	0								

0 out of 440 matrices cause differences. Ratio : 0%



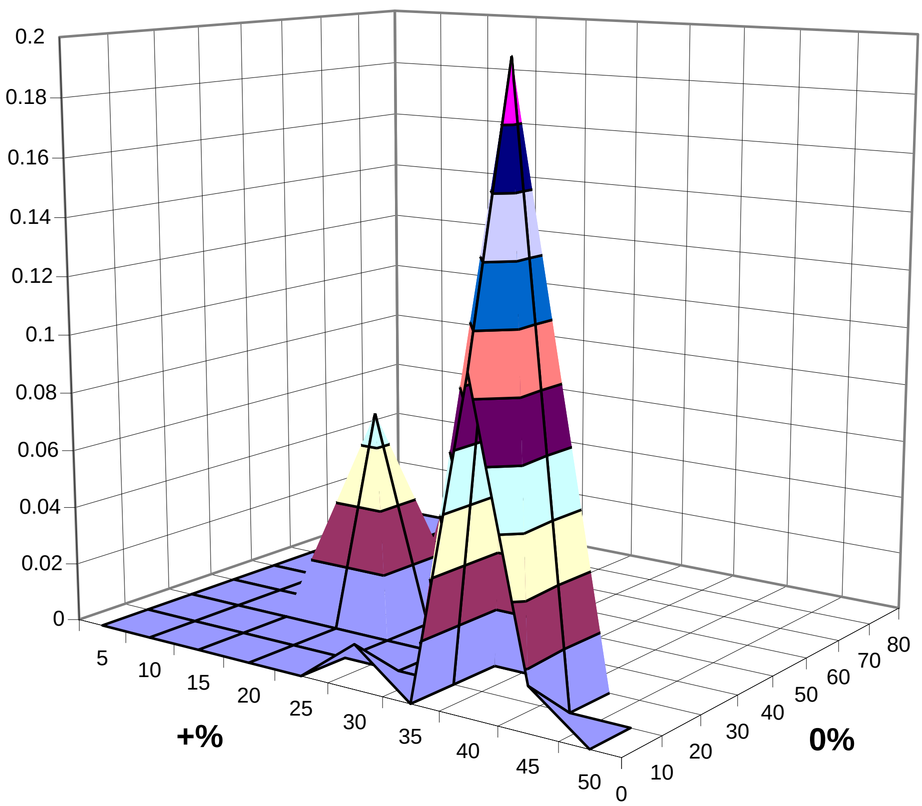
Average Difference between
Local Repair Heuristics and Exhaustive Search

440 Matrices 13x4

Results of phase 2

0% \ +%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0.015	0	0.108	0.015	0
10	0	0	0	0	0	0	0	0.2	0	
20	0	0	0	0	0	0	0	0		
30	0	0	0	0.071	0	0	0.08			
40	0	0	0	0	0	0.16				
50	0	0	0	0	0.107					
60	0	0	0	0						
70	0	0	0							
80	0	0								

7 out of 440 matrices cause differences. Ratio : 1.59%

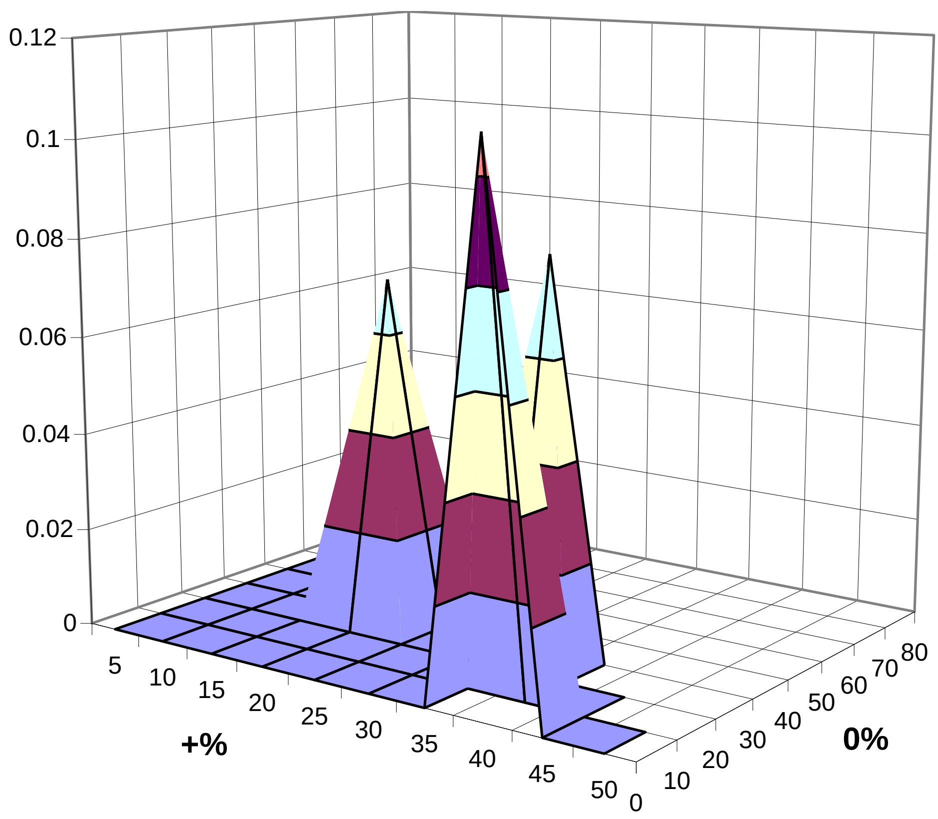


Average Difference between
Local Repair Heuristics and Exhaustive Search

Results of phase 3

0% \ +%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0	0	0.108	0	0
10	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0		
30	0	0	0	0.071	0	0	0.08			
40	0	0	0	0	0	0				
50	0	0	0	0	0					
60	0	0	0	0						
70	0	0	0							
80	0	0								

3 out of 440 matrices cause differences. Ratio :0.68%



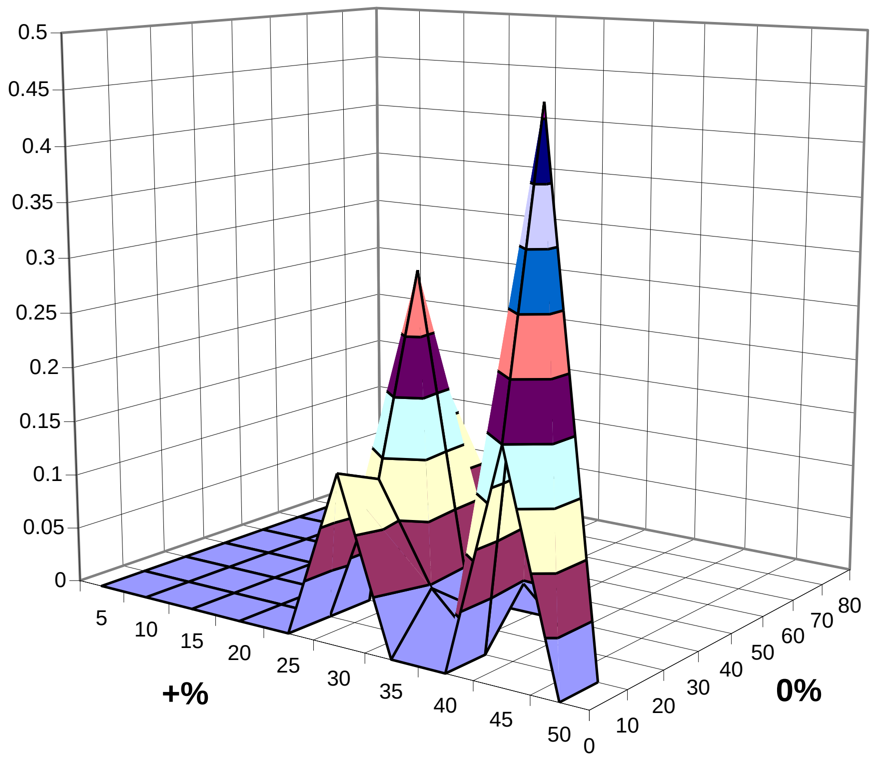
Average Difference between
Local Repair Heuristics and Exhaustive Search

440 Matrices 8x8

Results of phase 2

0% \ +%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0.15	0	0	0.2	0
10	0	0	0	0	0	0.133	0.047	0	0.463	
20	0	0	0	0	0	0.305	0.017	0.047		
30	0	0	0	0	0.185	0	0			
40	0	0	0	0	0.183	0				
50	0	0	0	0	0					
60	0	0	0.112	0.05						
70	0	0	0							
80	0	0								

18 out of 440 matrices cause differences. Ratio : 4.09%

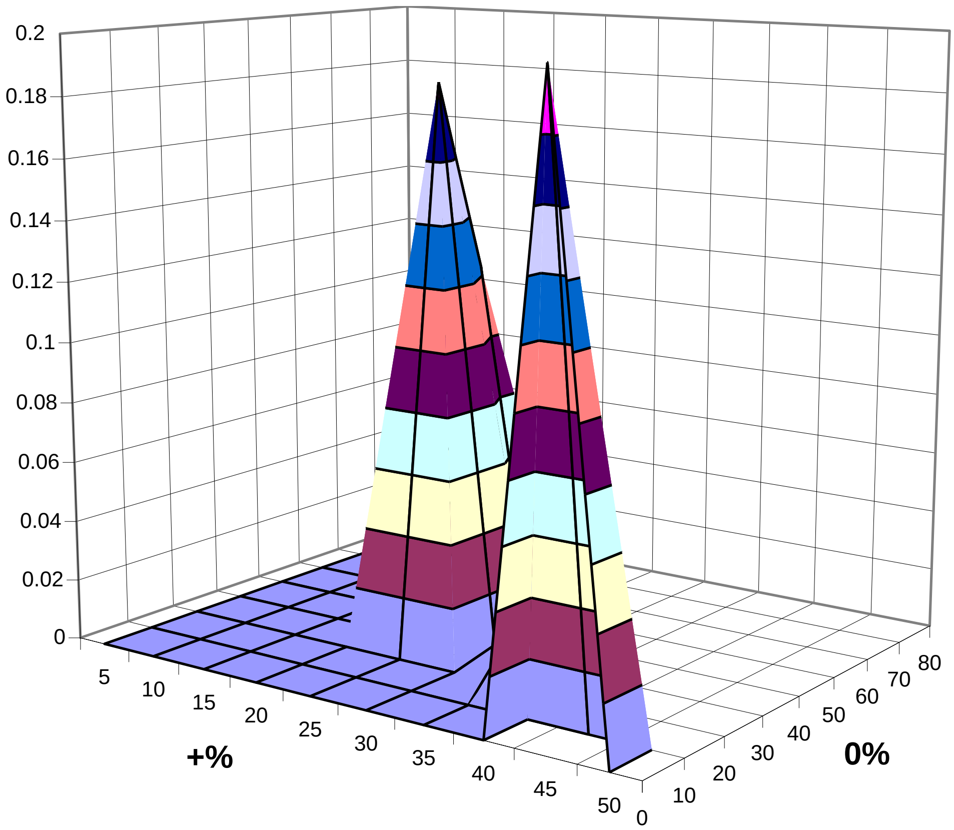


Average Difference between
Local Repair Heuristics and Exhaustive Search

Results of phase 3

0% \ +%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0	0	0	0.2	0
10	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0.017	0.044		
30	0	0	0	0	0.185	0	0			
40	0	0	0	0	0.123	0				
50	0	0	0	0	0					
60	0	0	0	0						
70	0	0	0							
80	0	0								

5 out of 440 matrices cause differences. Ratio : 1.14%



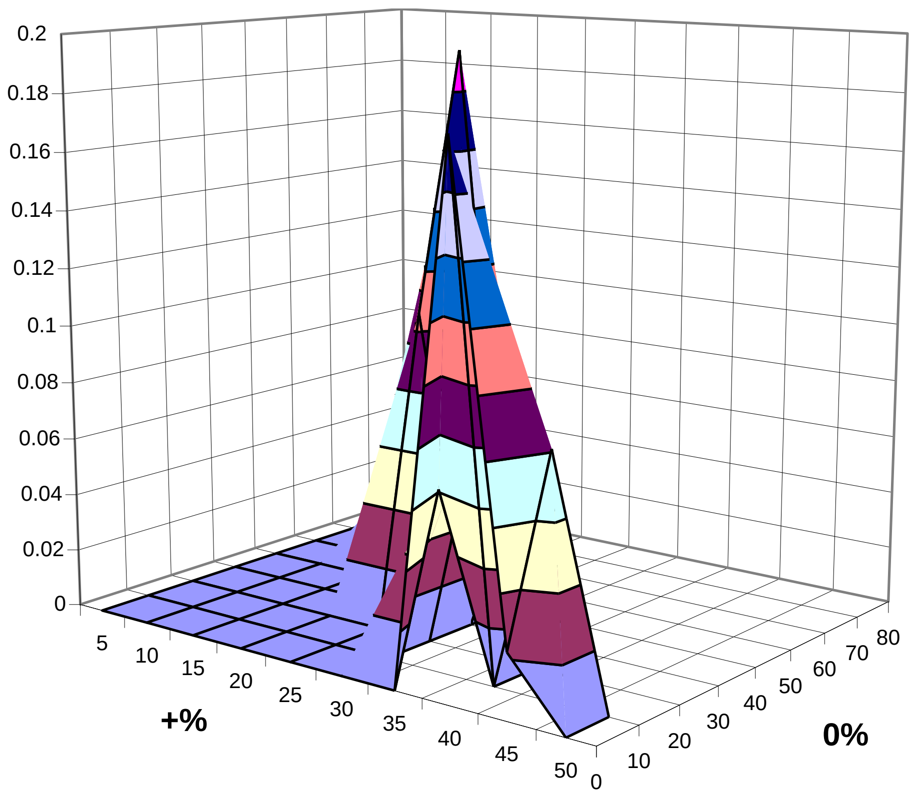
Average Difference between
Local Repair Heuristics and Exhaustive Search

440 Matrices 9x9

Results of phase 2

0% \ +%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0	0	0.178	0.022	0
10	0	0	0	0	0	0	0.061	0	0.081	
20	0	0	0	0	0	0	0.086	0		
30	0	0	0	0	0.106	0	0.057			
40	0	0	0	0	0.194	0				
50	0	0	0.034	0	0					
60	0	0	0.135	0						
70	0	0	0							
80	0	0								

12 out of 440 matrices cause differences. Ratio : 2.72%

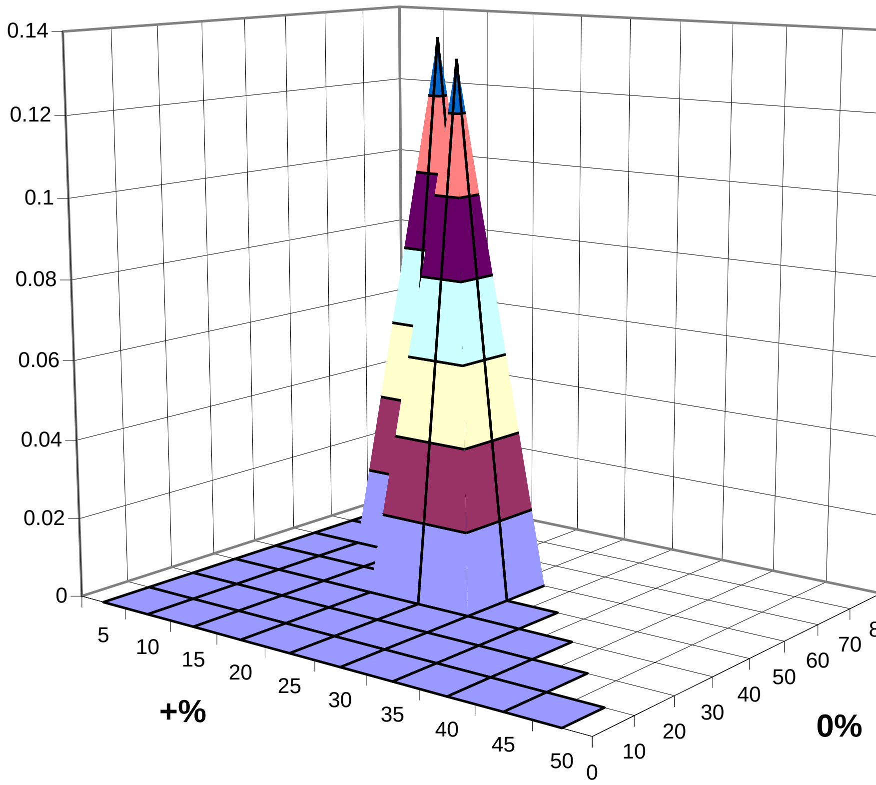


Average Difference between
Local Repair Heuristics and Exhaustive Search

Results of phase 3

0% \ +%	5	10	15	20	25	30	35	40	45	50
0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0		
30	0	0	0	0	0	0	0			
40	0	0	0	0	0.133	0				
50	0	0	0	0	0					
60	0	0	0.135	0						
70	0	0	0							
80	0	0								

2 out of 440 matrices cause differences. Ratio : 0.46%



Average Difference between
Local Repair Heuristics and Exhaustive Search

Matrix 30 x 30

-	-	-	+	+	+	+	+	+	0	0	-	+	+	-	-	-	+	+	-	-	+	+	+	-	-	-	-	+	+
+	+	+	+	-	-	-	0	-	+	+	+	-	-	-	-	-	-	+	+	+	+	0	0	-	-	+	+	+	-
+	+	+	+	-	-	-	-	-	-	-	-	-	-	+	-	+	-	+	0	0	-	-	-	-	-	-	-	-	-
-	-	-	-	-	+	+	+	-	-	-	0	0	0	0	+	+	+	-	-	+	+	+	+	+	+	+	+	+	+
-	-	-	+	+	+	-	-	-	+	+	+	0	0	0	-	-	-	+	+	+	-	-	-	-	-	+	+	+	+
0	0	0	-	-	-	+	+	+	-	-	-	+	+	-	+	-	+	-	+	+	+	-	-	-	+	+	-	-	-
-	-	-	-	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	+	+	+	+	+	+
-	-	-	-	-	-	+	+	+	+	+	-	-	-	+	+	+	-	-	-	0	0	0	0	-	-	-	+	+	+
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	0	0	0	0	0	0	0	-	-	-
0	0	0	0	0	0	-	-	-	-	-	+	+	+	+	-	+	-	+	-	+	-	+	-	+	- +	-	+	-	
-	-	-	-	-	-	-	+	+	+	+	+	-	+	-	+	0	-	+	0	-	+	0	-	+	0	+	-	+	0
-	+	-	+	0	0	-	-	-	+	+	-	+	-	+	0	-	+	-	-	-	-	+	+	+	-	-	-	+	+
+	+	+	-	-	-	+	+	-	-	+	+	+	-	-	-	+	-	+	+	0	0	-	-	+	+	-	-	-	-
+	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	+	+
-	-	-	-	+	-	+	-	+	-	+	-	+	-	-	+	+	-	-	-	+	+	+	-	-	-	-	+	+	+
-	+	+	+	-	-	-	+	+	+	0	0	0	-	+	-	+	+	+	-	-	-	0	0	+	-	+	-	-	-
-	-	-	-	-	-	+	+	+	+	0	-	0	-	0	-	0	+	+	-	-	+	-	0	+	+	+	+	+	+
+	+	+	+	+	0	0	-	-	-	-	-	-	-	-	+	+	+	+	+	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	+	-	-	+	+	-	-	0	0	-	+	-	0	-	+	-	-	-	+	-	-	+	+	+	-
+	+	-	-	+	+	-	-	+	+	-	-	+	+	-	-	+	+	-	-	0	0	-	+	-	+	0	0	-	+
-	-	-	-	-	-	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-	+	+	+	-	-	-	+	+
-	-	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-	-	-	+	+	+	+	+	+	-	+	-	+
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-
+	+	+	+	+	-	0	0	0	0	0	-	-	-	-	+	+	+	+	+	+	-	-	-	-	-	-	+	-	-
-	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	+
-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	0	0	0	0	0	0	-	-	-	-	+
+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	0	0	0	0	-	+	+	+	+	-	-	0
-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+

Result of phase 2

																														x-y	Balance	x+y	Avg. Link	Fairness	Balance& Fairness
=	=	=	*	*	*	*	*	*	0	0	=	*	*	=	=	=	*	+	=	=	*	*	*	=	=	=	-	+	+	0	0	24	24.47	0.47	0.47
*	*	*	*	=	=	=	0	=	+	*	*	=	=	=	=	=	=	*	*	*	*	0	0	=	-	*	*	+	=	0	0	24	23.6	0.4	0.4
*	*	*	*	=	=	=	=	=	=	=	=	=	-	*	=	*	=	*	0	0	=	=	-	=	-	=	-	=	=	-10	10	24	24.47	0.47	10.47
=	=	=	=	=	*	*	*	=	=	=	0	0	0	0	*	+	*	=	=	*	*	*	*	*	*	+	*	+	+	2	2	22	22.72	0.72	2.72
=	=	=	*	*	*	=	=	=	+	*	*	0	0	0	=	=	=	*	*	*	=	=	-	=	-	*	*	*	*	0	0	24	23.6	0.4	0.4
0	0	0	=	=	=	*	*	*	=	=	=	+	*	=	*	=	*	=	*	*	*	-	-	=	*	*	=	=	=	-2	2	24	23.6	0.4	2.4
-	=	=	=	*	*	*	*	*	*	*	*	*	*	*	=	=	=	=	-	=	-	=	=	=	-	=	=	=	=	-4	4	26	26.22	0.22	4.22
*	*	*	*	*	*	*	+	+	+	+	*	*	*	*	*	+	=	=	=	=	=	=	=	=	*	*	*	*	*	11	11	25	26.22	1.22	12.22
=	=	=	=	=	=	*	*	*	*	*	=	=	=	*	*	*	-	=	-	0	0	0	0	=	-	=	*	*	*	-1	1	23	22.72	0.28	1.28
*	*	*	*	*	*	*	*	*	+	+	*	+	*	*	*	*	*	*	*	0	0	0	0	0	0	0	=	=	=	14	14	20	20.1	0.1	14.1
0	0	0	0	0	0	=	=	=	=	=	*	*	*	*	-	+	=	*	-	*	=	*	=	*	=	*	=	*	=	-1	1	21	20.97	0.03	1.03
-	=	=	=	=	=	=	*	*	+	*	*	=	*	=	*	0	-	*	0	=	*	0	=	*	0	*	=	*	0	0	0	22	21.85	0.15	0.15
-	*	=	*	0	0	=	=	=	*	*	-	*	=	*	0	=	*	=	-	=	=	*	*	*	=	=	=	*	*	0	0	24	23.6	0.4	0.4
*	*	*	=	=	=	*	*	=	=	+	*	*	=	=	-	*	-	*	*	0	0	=	=	*	*	=	=	=	=	-1	1	25	24.47	0.53	1.53
*	*	*	=	=	=	=	=	=	=	=	=	=	=	=	-	=	=	=	-	=	=	-	=	0	0	0	0	*	*	-13	13	23	22.72	0.28	13.28
=	=	=	=	*	-	*	=	*	=	*	=	*	=	=	*	*	-	=	-	*	*	*	=	=	-	=	*	*	*	0	0	26	26.22	0.22	0.22
=	*	*	*	=	=	=	*	*	+	0	0	0	=	*	=	*	*	*	=	-	=	0	0	*	=	*	-	=	=	0	0	22	21.85	0.15	0.15
=	=	=	=	=	=	+	*	+	*	0	-	0	=	0	=	0	*	*	=	=	*	=	0	*	*	*	*	*	*	0	0	22	21.85	0.15	0.15
*	*	*	*	*	0	0	=	=	=	=	=	=	=	-	*	*	*	*	*	=	-	=	=	=	-	=	-	=	=	-4	4	24	24.47	0.47	4.47
-	=	=	=	=	*	=	=	*	*	=	-	0	0	=	*	=	0	=	*	=	=	-	*	=	=	*	*	*	=	-6	6	24	23.6	0.4	6.4
*	*	=	=	*	*	=	=	+	*	=	=	*	*	=	=	*	*	=	=	0	0	-	*	=	*	0	0	=	*	0	0	24	22.72	1.28	1.28
=	=	=	=	=	=	*	+	+	*	*	*	*	*	*	*	*	=	=	=	=	=	*	*	*	=	=	=	*	*	0	0	28	26.22	1.78	1.78
=	=	*	*	*	*	+	+	+	*	=	=	=	=	-	=	=	=	=	=	*	*	*	*	*	*	=	*	=	*	0	0	26	26.22	0.22	0.22
*	*	*	*	*	*	+	+	*	*	*	*	*	*	*	*	+	*	*	*	*	*	*	*	*	*	*	*	+	*	26	26	26	26.22	0.22	26.22
-	=	=	=	=	=	=	=	=	=	=	-	=	-	=	=	*	=	*	=	*	-	*	=	*	=	*	=	*	=	-12	12	26	26.22	0.22	12.22
*	*	*	+	*	=	0	0	0	0	0	=	=	-	=	*	*	*	*	*	*	=	=	-	=	=	=	*	=	=	0	0	22	21.85	0.15	0.15
=	=	*	=	*	=	*	=	*	=	*	=	*	=	*	=	*	-	*	=	*	-	*	=	+	=	*	=	*	+	0	0	26	26.22	0.22	0.22
=	=	*	*	*	*	*	+	+	*	*	*	+	*	*	=	=	=	=	0	0	0	0	0	0	=	=	=	=	+	0	0	20	20.97	0.97	0.97
*	*	*	*	*	*	+	*	+	+	+	=	=	=	=	=	=	=	0	0	0	0	=	*	*	*	*	=	=	0	1	1	21	21.85	0.85	1.85
-	*	=	*	=	*	=	+	=	+	=	*	=	*	-	*	=	*	=	*	=	*	=	*	=	*	=	*	=	*	0	0	26	26.22	0.22	0.22
																															108			13.59	121.59

Result of phase 3

																														x-y	Balance	x+y	Avg. Link	Fairness	Balance& Fairness
=	=	=	*	*	*	*	*	*	0	0	=	*	*	=	=	=	*	+	=	=	*	*	*	=	=	=	=	*	+	0	0	26	24.47	1.53	1.53
*	*	*	*	=	=	=	0	=	+	*	*	=	=	=	=	=	=	*	*	*	*	0	0	=	-	*	*	+	=	0	0	24	23.6	0.4	0.4
*	*	*	*	=	=	=	=	=	=	=	=	=	-	*	=	*	=	*	0	0	=	=	-	=	-	=	-	=	=	-10	10	24	24.47	0.47	10.47
=	=	=	=	=	*	*	*	=	=	=	0	0	0	0	*	+	*	=	=	*	*	*	*	*	*	+	*	+	+	2	2	22	22.72	0.72	2.72
=	=	=	*	*	*	=	=	=	+	*	*	0	0	0	=	=	=	*	*	*	=	=	-	=	-	*	*	*	*	0	0	24	23.6	0.4	0.4
0	0	0	=	=	=	*	*	*	=	=	=	*	*	-	*	=	*	=	*	*	*	-	-	=	*	*	=	=	=	0	0	24	23.6	0.4	0.4
-	=	=	=	*	*	*	*	*	*	*	*	*	*	*	=	=	=	=	-	=	-	=	=	=	-	=	=	=	=	-4	4	26	26.22	0.22	4.22
*	*	*	*	*	*	*	+	+	+	+	*	*	*	*	*	+	=	=	=	=	=	=	=	*	*	*	*	*	*	11	11	25	26.22	1.22	12.22
=	=	=	=	=	=	*	*	*	*	*	=	=	=	*	*	*	-	=	-	0	0	0	0	=	-	=	*	*	*	-1	1	23	22.72	0.28	1.28
*	*	*	*	*	*	*	*	*	+	+	*	+	*	*	*	*	*	*	*	0	0	0	0	0	0	0	=	=	=	14	14	20	20.1	0.1	14.1
0	0	0	0	0	0	=	=	=	=	=	*	*	*	*	-	+	=	*	-	*	=	*	=	*	=	*	=	*	=	-1	1	21	20.97	0.03	1.03
-	=	=	=	=	=	=	*	*	+	*	*	=	*	=	*	0	-	*	0	=	*	0	=	*	0	*	=	*	0	0	0	22	21.85	0.15	0.15
-	*	=	*	0	0	=	=	=	*	*	-	*	=	*	0	=	*	=	-	=	=	*	*	*	=	=	=	*	*	0	0	24	23.6	0.4	0.4
*	*	*	=	=	=	*	*	=	=	+	*	*	=	=	-	*	-	*	*	0	0	=	=	*	*	=	=	=	=	-1	1	25	24.47	0.53	1.53
*	*	*	=	=	=	=	=	=	=	=	=	=	=	=	-	=	=	=	-	=	=	-	=	0	0	0	0	*	*	-13	13	23	22.72	0.28	13.28
=	=	=	=	*	-	*	=	*	=	*	=	*	=	=	*	*	-	=	-	*	*	*	=	=	-	=	*	*	*	0	0	26	26.22	0.22	0.22
=	*	*	*	=	=	=	*	*	+	0	0	0	=	*	=	*	*	*	=	-	=	0	0	*	=	*	-	=	=	0	0	22	21.85	0.15	0.15
=	=	=	=	=	=	+	*	+	*	0	-	0	=	0	=	0	*	*	=	=	*	=	0	*	*	*	*	*	*	0	0	22	21.85	0.15	0.15
*	*	*	*	*	0	0	=	=	=	=	=	=	=	=	*	*	*	*	*	=	-	=	=	=	-	=	-	=	=	-5	5	25	24.47	0.53	5.53
-	=	=	=	=	*	=	=	*	*	=	-	0	0	=	*	=	0	=	*	=	=	-	*	=	=	*	*	*	=	-6	6	24	23.6	0.4	6.4
*	*	=	=	*	*	=	=	+	*	=	=	+	*	=	=	*	*	=	=	0	0	-	*	=	*	0	0	=	*	-1	1	23	22.72	0.28	1.28
=	=	=	=	=	=	*	+	+	*	*	*	*	*	*	*	*	=	=	=	=	=	*	*	*	=	=	-	+	*	0	0	26	26.22	0.22	0.22
=	=	*	*	*	*	+	+	+	*	=	=	=	=	-	=	=	=	=	=	*	*	*	*	*	*	=	*	=	*	0	0	26	26.22	0.22	0.22
*	*	*	*	*	*	+	+	*	*	*	*	*	*	*	*	+	*	*	*	*	*	*	*	*	*	*	*	+	*	26	26	26	26.22	0.22	26.22
-	=	=	=	=	=	=	=	=	=	=	-	=	-	=	=	*	=	*	=	*	-	*	=	*	=	*	=	*	=	-12	12	26	26.22	0.22	12.22
*	*	*	+	*	=	0	0	0	0	0	=	=	-	=	*	*	*	*	*	*	=	=	-	=	=	=	*	=	=	0	0	22	21.85	0.15	0.15
=	=	*	=	*	=	*	=	*	=	*	=	*	=	*	=	*	-	*	=	*	-	*	=	+	=	*	=	*	+	0	0	26	26.22	0.22	0.22
=	=	*	*	*	*	*	+	+	*	*	*	+	*	*	=	=	=	=	0	0	0	0	0	0	=	=	=	=	+	0	0	20	20.97	0.97	0.97
*	*	*	*	*	*	+	*	+	+	+	=	=	=	=	=	=	=	0	0	0	0	=	*	*	*	*	=	=	0	1	1	21	21.85	0.85	1.85
-	*	=	*	=	*	=	+	=	+	=	*	=	*	-	*	=	*	=	*	=	*	=	*	=	*	=	*	=	*	0	0	26	26.22	0.22	0.22
																															108			12.15	120.15