APPROACHES FOR MULTI-ATTRIBUTE AUCTIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÜLŞAH KARAKAYA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

JULY 2009

Approval of the thesis:

APPROACHES FOR MULTI-ATTRIBUTE AUCTIONS

submitted by **GÜLŞAH KARAKAYA** in partial fulfillment of the requirement for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Nur Evin Özdemirel ________________
Head of Department, **Industrial Engineering**

Prof. Dr. Murat Köksalan ________________
Supervisor, **Industrial Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Gülser Köksal ________________
Industrial Engineering Dept., METU

Prof. Dr. Murat Köksalan
Industrial Engineering Dept., METU ________________

Assist. Prof. Dr. Seçil Savaşaneril
Industrial Engineering Dept., METU ________________

Assoc. Prof. Dr. Yasemin Serin
Industrial Engineering Dept., METU ________________

Assist. Prof. Dr. Banu Yüksel Özkaya
Industrial Engineering Dept., HU ________________

Date: **23.07.2009**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Gülşah KARAKAYA

Signature :

ABSTRACT

APPROACHES FOR MULTI-ATTRIBUTE AUCTIONS

Karakaya, Gülşah
M.S., Department of Industrial Engineering
Supervisor: Prof. Dr. Murat Köksalan

July 2009, 89 pages

There is a growing interest in electronic auctions in the literature. Many researchers work on the single attribute version of the problem. Multi-attribute version of the problem is more realistic. However, this brings a substantial difficulty in solving the problem. In order to overcome the computational difficulties, we develop an Evolutionary Algorithm (EA) for the case of multi-attribute multi-item reverse auctions.

We generate the whole Pareto front using the EA. We also develop heuristic procedures to find several good initial solutions and insert those in the initial population of the EA. We test the EA on a number of randomly generated problems and compare the results with the true Pareto optimal front obtained by solving a series of integer programs.

We also develop an exact interactive approach that provides aid both to the buyer and the sellers for a multi-attribute single item multi round reverse auction. The buyer decides on the provisional winner at each round. Then the approach provides support in terms of all attributes to each seller to be competitive in the next round of the auction.

Keywords: online auctions, multi-attribute auctions, interactive approach, evolutionary algorithm

ÖZ

ÇOK AMAÇLI AÇIK ARTTIRMALAR İÇİN ÇÖZÜM YAKLAŞIMLARI

Karakaya, Gülşah
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Murat Köksalan

Temmuz 2009, 89 Sayfa

Literatürdeki elektronik açık arttırma çalışmaları giderek artmaktadır. Birçok araştırmacı bu problemi tek amaçlı olarak ele almıştır. Problemin çok amaçlı hali ise daha gerçekçidir. Buna karşın, çok amaçlılık problemin çözümüne büyük bir zorluk getirir. Hesaplama zorluklarının üstesinden gelmek için, çok amaçlı çok ürünlü açık arttırmalar için bir evrimsel algoritma geliştirdik.

Evrimsel algoritma kullanarak, bütün Pareto yüzeyini oluşturmaya çalıştık. Ayrıca, iyi başlangıç çözümleri ve bunları evrimsel algoritmanın başlangıç nufüsuna eklemek için sezgisel yöntemler de geliştirdik. Evrimsel algoritmayı rassal olarak oluşturulan problemler üzerinde denedik ve sonuçları tamsayı programlamalarla elde ettiğimiz gerçek Pareto yüzeylerle karşılaştırdık.

Ayrıca hem alıcıya hem de satıcıya çok amaçlı, tek ürünlü, çok turlu açık arttırmalarda yardım sağlayan bir etkileşimli yaklaşım geliştirdik. Alıcı her turda geçici bir kazanan seçmektedir. Bu yaklaşım, her satıcıya açık arttırmanın bir sonraki turunda rekabet edebilmesi için gerekli nitelikler konusunda destek sağlamaktadır.

Anahtar Kelimeler: elektronik açık arttırmalar, çok amaçlı açık arttırmalar, etkileşimli yaklaşım, evrimsel algoritma.

To My Families

&

Fatih

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Professor Murat Köksalan not only for his invaluable guidance and supervision but also his understanding and encouragement throughout this study. It has been a pleasure for me to work with him.

I would like to thank to my families for their endless and unconditional love and trust throughout my life. Moreover, I am grateful to my sister Ayşe who always motivated and supported me during this study.

I am thankful to my friends Diclehan Tezcaner, Burcu Özsayın, and Aslı Gül Buğdacı. I feel myself lucky to feel their existence whenever I need their ideas and support. Also I would thank to Aylin, Elif, Gülden, Eda and Çiğdem for their friendship.

I owe thanks to Professor Gülser Köksal, Associate Professor Yasemin Serin, Assistant Professor Banu Yüksel Özkaya and Assistant Professor Seçil Savaşaneril for their valuable comments and suggestions.

I also would like to thank to TÜBİTAK for providing a financial support during my M.S. study.

Lastly, I am grateful to my husband Ahmet Fatih Karakaya for being in my life and for his support in my graduate studies. Nothing would be possible or meaningful without the complimentary love and encouragement of him.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Reaching, sharing and exchanging information has become much easier with the help of information and communication technologies (ICT), and some special web based software. As Koppius et al. (2004) state, these new technologies led to some changes in business and made electronic markets very popular. Traditional auction process has also changed in parallel and integrated ICTs.

In today's world, auction processes are very common especially the electronic ones. With growing Internet technologies, online auctions have broadened by enabling people to sell and buy in a wide range of alternatives on the Internet.

Auctions are also used by companies and governments. Motorola is one of these companies that installed auction mechanisms in 2001. Metty et al. (2005) state that they enjoy 15% - 20% savings in indirect material cost and 25% - 50% savings in direct material cost by implementing an online negotiation program called MINT (Motorola Internet Negotiation Tool). A rather different example is the implementation of the government of Chile. They have been providing meals to public school children by an annual auction for many years (Catalán et al. 2009). Auctions are also commonly used in transportation problems (Sheffi, 2004; De Vries et al. 2003) and other large scale applications are reported in Hohner et al. (2003) and Sandolhm et al. (2006).

There is a growing amount of literature about online auctions (Bapna et al. 2008; Pinker et al. 2003). Many researchers have contributed on the single attribute version of the problem. In this thesis we deal with multi-attribute version of the problem which is more realistic. We first develop an exact interactive approach that provides aid both to the buyer and the sellers for a multi-attribute single item multi-round reverse auction. The buyer decides on the provisional winner at each iteration. Then the approach provides support in terms of all attributes to each seller to be competitive in the next round of the auction. We also develop an Evolutionary Algorithm (EA) for the case of multi-attribute multi-item reverse auctions to overcome the computational difficulties resulting from multi-attribute version of the problem. We try to generate the whole Pareto front using the EA. We also develop heuristic procedures to find several good initial solutions and insert those in the initial population of the EA.

The structure of the thesis is as follows: Some important definitions on multi-criteria decision making and background of the auction theory are given in Chapter 2. In Chapter 3, a literature review on multi-attribute auctions is presented. An exact interactive method for a multi-attribute single item multi-round reverse auction is developed in Chapter 4. In Chapter 5, an EA approach to multi-attribute multi-item reverse auctions is given. We test the EA on a number of randomly generated problems and compare the results with the true Pareto optimal front obtained by solving a series of integer programs. Lastly, future study issues and conclusive remarks are presented in Chapter 6.

CHAPTER 2

DEFINITIONS AND BACKGROUND

2.1 Definitions

In multi-objective optimization problems we try to optimize at least two, generally conflicting, objectives satisfying the given constraints. A multi-objective optimization problem can be formulated as follows:

“Maximize” $\{z_1(\mathbf{x}), z_2(\mathbf{x}),..., z_p(\mathbf{x}), \}$

s.to

$$\mathbf{x} \in \mathbf{X}$$

where,

$\mathbf{x}$: decision variable vector

$\mathbf{X}$: feasible decision space

$\mathbf{Z}$: feasible criterion/objective space

In the following definitions from 2.1 to 2.8 it is assumed that $\mathbf{z}(\mathbf{x}), \mathbf{z}(\mathbf{x}') \in \mathbf{Z}$ and $\mathbf{x}, \mathbf{x}' \in \mathbf{X}$.

Definition 2.1: $\mathbf{z}(\mathbf{x}')$ is said to *dominate* $\mathbf{z}(\mathbf{x})$ if and only if $z_j(\mathbf{x}') \geq z_j(\mathbf{x})$ for all j and $z_j(\mathbf{x}') > z_j(\mathbf{x})$ for at least one j.

Definition 2.2: $\mathbf{z}(\mathbf{x})$ is *nondominated* if and only if no $\mathbf{z}(\mathbf{x}')$ dominates it.

Definition 2.3: $\mathbf{z}(\mathbf{x})$ is *strictly dominated*, if and only if $z_j(\mathbf{x}') > z_j(\mathbf{x})$ for all j.

Definition 2.4: $\mathbf{z}(\mathbf{x})$ is said to be *weakly nondominated*, if and only if no $\mathbf{z}(\mathbf{x}')$ strictly dominates it.

Definition 2.5: If **z**(**x**) is nondominated then **x** is said to be *efficient*. Otherwise, **x** is said to be *inefficient*.

Definition 2.6: If **z**(**x**) is strictly dominated then **x** is said to be *strictly inefficient*. Otherwise, **x** is said to be *weakly efficient*.

Definition 2.7: If $\exists\, z_j \in$ **Z** for j= 1,2, . . .,k and $\exists\, w_j \ni \; w_j \geq 0$ j= 1,2, . . .,k satisfying $\sum_{j \neq i}^{k} w_j = 1$ and $\sum_{j \neq i}^{k} w_j z_j \geq z_i$, then z_i is said to be *convex dominated*.

Definition 2.8: If a nondominated solution, **z**(**x**), is convex dominated then **z**(**x**) is said to be *unsupported nondominated solution*. Otherwise **z**(**x**) is said to be *supported nondominated solution*.

Note that supported nondominated solutions can be found using weighted-sum objective functions. However, unsupported nondominated solutions cannot be found by such approaches. These solutions can be found by applying Tchebycheff method.

Definition 2.9: If **z**(**x**) consists of the worst objective function value for each objective among all nondominated solutions then **z**(**x**) is *nadir objective vector*.

In Figure 2.1 the classification of the solutions based on the domination rules where both objectives to be maximized are represented.

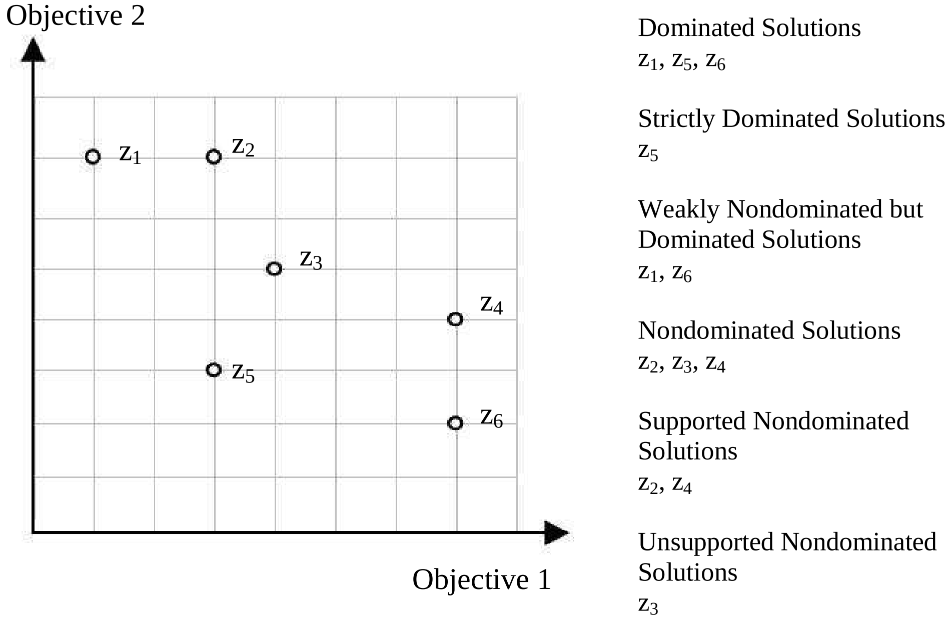


Figure 2.1 Classification of the solutions

2.2 Auctions

An auction is a way of buying and selling goods/services. The process is becoming very popular with the advances of the Internet technologies. Therefore, in the literature there are studies on online auctions, specifically English reverse auctions (further explained in Section 2.2.2).

The reasons why auctions become popular are listed in Talluri et al. (2007) as follows:
- improved information coordination with suppliers
- lower transaction times
- lower costs
- higher flexibility
- better supplier integration.

In addition to these, Rothkopf and Park (2001) state the prominent advantages as fairness and the appearance of fairness, i.e. all potential bidders have equal chance.

2.2.1 The Auction Process

Throughout the history, only valuable objects have been sold with the auction process. The auction process used to take place in a room or a square where an object was shown to the bidders by the auctioneer. People get together and make their bids to buy the offered good. However, in contrast to the traditional auctions, in today's world from cars to books, various types of things are being sold through auctions.

With the rapid development of technological infrastructure and invent of the Internet, the auction process has come to our houses. These technological developments eliminate the need for being in the auction place physically. There are some specialized websites that mediate buyer and seller to huge amount of goods being passed between people. Moreover, online auction sites allow people to buy (sell) inexpensive products/services even a pencil, in contrast to traditional auction houses.

McAfee and McMillan (1987) define an auction as "a market institution with explicit set of rules determining resource allocation and prices on the basis of bids from market participants". This is a general description for the auction and some classifications are done according to the transaction types. Auctions can be classified also based on the number of units to be sold, the number of attributes (price, quality, lead time, etc.) considered. The classification of the auctions with respect to the number of buyers and sellers is shown in Figure 2.2.

As can be seen from Figure 2.2, if there exist one buyer and one seller, the process is called negotiation, whereas if there are many buyers and many sellers it is called a double auction. eBay platform is an example for the double auction process. Reverse auction where there should be one buyer and many sellers is a common auction type in the literature. For instance a government (buyer) determines the supplier(s) for a bridge construction among many suppliers (sellers). The last type is forward auction where the seller is the auctioneer and the buyers are the bidders.

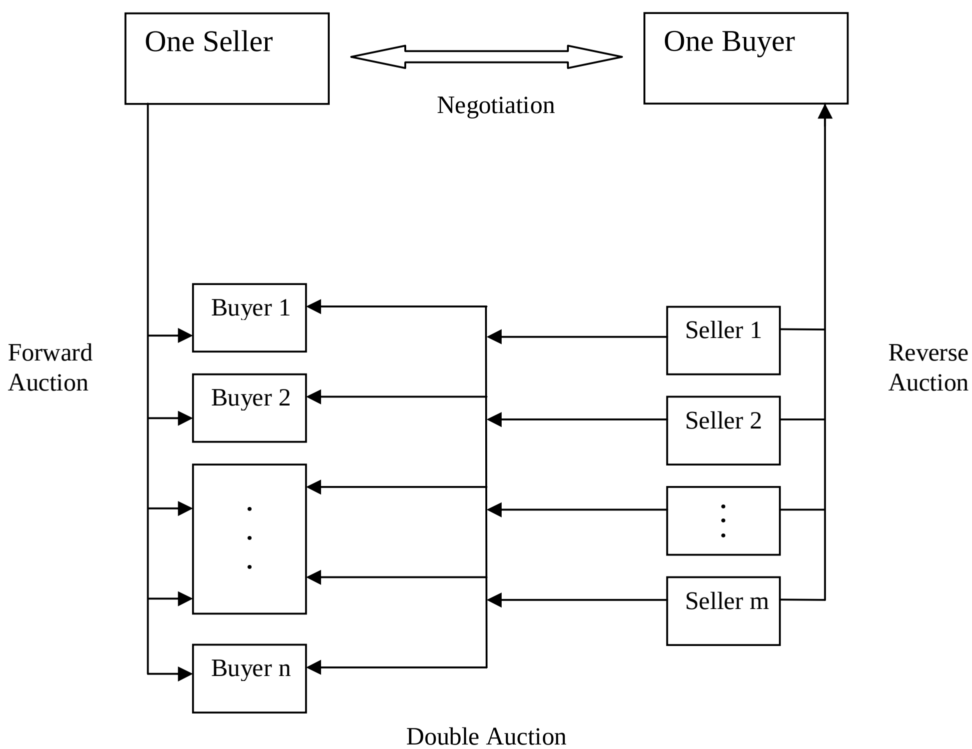


Figure 2.2 Auction types with respect to the number of sellers and buyers

2.2.2 Auction Mechanisms

Although Teich et al. (2004, 2006) group the auction mechanisms into three (English auction, Dutch auction, Vickrey's second price sealed-bid auction), in the literature auction mechanisms are generally given in four groups.

English Auction

It is the most known auction type. English auction is an ascending price auction, i.e. the auctioneer announces the minimum acceptable price and the price increases with the bids. The winning bidder is the one who bids the highest price. This type of auctions are generally open-cry auctions where all bidders know each other's bids.

To exemplify this type, in traditional antique sales the auction starts with a low price and with the bidders' raising hand the price increases. At the end, the antique is sold to the one who bids the most.

Dutch Auction

It is a descending price auction. In this type, the auctioneer announces a relatively high price and gradually decreases the price until a bidder accepts the current price. Generally this type of auctions are used for selling perishable goods. In the Netherlands, the flower sales have been done with this type of auction and its name comes from there.

The First Price Sealed-bid Auction

In this type of auction, bidders submit their bids. Contrary to the English auction, bidders do not know each other's bids and also it is a single round auction. Bidders can submit their bids only once, it is not an iterative process. After the announcement of the winner, some negotiations may be done. The winning bidder is the one who bids the highest (lowest) price for the forward (reverse) process. The winner pays the highest (lowest) price.

The Vickrey's Second Price Sealed-bid Auction

It is similar to the First Price Sealed-bid Auction except that, in this type the winner – who bids the highest (lowest) price for the buying (selling) process - pays the second highest (lowest) price.

2.2.3 Auction Types

One variation of the auctions is based on the number of different items and the number of units for each item. Items can be single or multiple (different types) and units can be single or multiple. Different auction types are the combination of these two.

If there exists a single item to be auctioned, it is called a single-item single-unit auction. This is the simplest version of the auction types. If at least two identical units for a particular item are auctioned; it is called a single-item multi-unit auction.

Multi-item auctions are more complex than single-item auctions. In the single-item auction, the winning bidder gets the item and pays its price. However, in multi-item case it is not so trivial to determine the winning bidder(s). In such auctions, generally bidders offer a combination of items – bundle – they want to supply. For this bundle, they determine the attribute values and make their bids. Multi-item auctions are known as combinatorial auctions and it is one of the emerging research topics.

Another variation is based on the number of attributes in the auction process. If there is only one attribute, for instance price, is taken into consideration, this is called a single attribute auction whereas if there are more than one attribute (price, quality, lead time, warranty, etc.) it is called a multi-attribute auction

(Leskelä, 2007). Multi-attribute auctions are more realistic, but they bring computational complexity.

2.2.4 Valuation

Valuation is an important concept in the auction theory as mostly the bidding strategy depends on it. Since in some situations, the bidders do not know how much the object worth; they can only realize other bidders' valuation (McAfee and McMillan, 1987). They define two extreme models to describe the valuation types: independent private value model and common value model.

If each bidder knows the true value of the auctioned item to him/her certainly and his/her valuation is not affected by other bidders' bids it is called independent private value. McAfee and McMillan (1987) state, this valuation is generally valid for the auctions in which the item being sold is for the bidder's own use not for resale.

The common value model is explained by McAfee and McMillan (1987) as no one knows the true value of the bidding item. However, all bidders have the same valuation for the item. The bidders draw the item's value to them. They use the same tool to understand their valuation; for instance observing the market or using the same distribution.

2.2.5 Online Auctions

If the auction is held on the Internet, it is called an online auction. The bidder can see the price of the good/service and time left to the end of the auction. Generally there are three parties in the online auction processes namely website, seller, and buyer. Website holds the information about the buyer and the seller. It provides security for both parties. Also, the website offers a platform where buyers can reach many goods and sellers can reach many customers without a limitation of time and location. eBay, Gittigidiyor, uBid, etc. are some commonly known online auction websites.

Generally, online auction sites contain many categories. One can choose his/her desired category to find a good or simply can search with a keyword. After finding the desirable product, the website lists the description of the product, pictures of the product, shipping conditions, current bid, end time, number of bids, rating of the seller, etc.

CHAPTER 3

MULTI-ATTRIBUTE AUCTIONS

The multi-attribute auction process considers not only the price but also other attributes such as quality, lead time, etc. Typically, to evaluate bids, a function called value/scoring function is used. Commonly, to estimate the value function for multi-attribute auctions, linear weighted utility functions are used. Also, to determine the winner of the auction; Winner Determination Problems (WDP) are solved by applying this value/scoring function.

Multi-attribute auctions are used in many areas, one of which is e-commerce. Butler et al. (2008) explain the applications of multi-attribute preference models (MAPM) in e-commerce by giving examples of websites that use MAPM. They say that mentioned websites use different approaches to elicit the users' preferences.

In the literature, most of the studies in this field are on reverse auctions where the auctioneer is the buyer and the bidders are sellers. In the following, some approaches for multi-attribute reverse auctions are discussed.

3.1 Scoring Function

Bichler and Kalagnanam (2005) suggest a weighted-sum scoring function to evaluate bids. In their model they set lower and upper bounds for demand since the combinations hardly sum up to exact demand. Since they study multi sourcing, i.e. demand can be supplied by multiple suppliers, they add a limitation constraint on the number of winners. It is an important issue for two reasons:

Firstly, the buyer does not want to supply from too many sellers as it creates high overhead costs and complexity in managing. Also the buyer does not want to be supplied from very few sellers since there is a risk of not receiving the supplies on time with a limited number of sellers. Moreover, they add a homogeneity constraint to avoid different levels for an attribute among the winning bids.

Bichler and Kalagnanam (2005) use a weighted-sum scoring function. However, as Bellosta et al. (2004) state, although it is very common to use scoring functions based weighted-sum, it has some drawbacks such as the difficulty of determining weights. Also the solutions that can be found are limited with a weighted-sum scoring function.

Butler et al. (2006) differentiate attribute and objective and proposes an approach to define weights in multi attribute utility theory (MAUT) in two steps. In the first step, the decision maker (DM) specifies weights that show the impact of each attribute on each objective (predictive model). In the second step, he/she specifies weights for objectives according to their importance to his/her (preference model).

They exemplify the method and it is explained as follows: For a digital camera price, resolution and zoom are attributes whereas economy and functionality are the objectives of the user. Firstly, the relationship between attributes and objectives are defined (price and resolution are related with economy and resolution and zoom are related with functionality). Each attribute is normalized with respect to its range, for instance, if the range for price is between $200 and $500, the price value of $200 is scored as 1 and $500 is scored as 0. The values in the range are calculated by extrapolation. Then, according to the preferences of the user, weights of attributes for each objective are determined where weights can be positive or negative. Afterwards, weights (importance) of the objectives are defined. Finally, the weights determined in predictive and preference models are combined to determine the appropriate weights on the attribute to calculate the utilities of all alternatives.

3.2 Pricing Out

Teich et al. (2006) define the theoretical side of their method implemented on the Internet (NegotiAuction). Their approach is based on the 'pricing out' technique, i.e. they convert all attributes into monetary values. They try to define all attributes in terms of one attribute which is price. They use linear/integer programming to minimize the cost to buyer to determine the bid status. They define three bid status namely, active, inactive and semi-active. If a bid is "active", it will be among the winners. If it is "inactive", it will completely be outbid, whereas if it is "semi-active", it is partially active and it can be outbid next. They also develop the 'suggested price' decision support component for bidders. Suggested price refers to the best (highest) price that makes the bidder's bid active. They try to maximize the suggested price while keeping a decrement in the buyer's total cost with a maximization-type linear programming (LP).

Teich et al. (2006) use 'pricing out' approach for multi-attribute auctions which does not make it necessary to formulate the buyer's preference function. On the other hand, as stated in Talluri et al. (2007), it is not easy to convert all attributes into price. The problem is similar to determining the weights in the previous case.

Leskelä (2007) suggests a decision support tool for combinatorial auctions. In Leskelä et al. (2007), they formulate the problem for a single-attribute auction. However, they say that the formulation can be extended with the 'pricing out' approach for multi-attribute auction cases. In their approach, they provide bidders not only the 'suggested price' for a new bid, but also 'quantity decision support'. By quantity support mechanism (QSM), bidders can request suggestions for quantities to bid. They first solve the winner determination problem and use the dual prices of this problem as the cost estimates in quantity support problem.

In their study, inactive bids are also kept in the bid stream for the reason that an inactive bid can become active again, if an entering bid groups with it. Only dominated bids are deleted. A dominated bid is defined as a bid with the same quantities, but with a higher price than the same bidder's latter bid. They say that even if a bidder’s bid is higher than the suggested price, he/she may become among the winning bidders at the end. At first glance, that bid is inactive but with the entering bids it can become active unless it is dominated.

Leskelä et al. (2007) solve the quantity support problem for each inactive bidder separately whether they can become active by grouping the active bids. However, this approach cannot provide efficiency as it does not allow a new combination of inactive bids. Köksalan et al. (2008) develop a Group Support Mechanism (GSM). The main difference of GSM from QSM is that in QSM only one incoming bid can complement the active bids; whereas in the GSM inactive bids can make a combination with active bids or with inactive bids. Another difference is that in the GSM, sellers’ cost functions are not composed by just using the dual prices. Köksalan et al. (2008) define some ranges for fixed and variable costs by using industry estimates. These ranges are updated as information becomes available on inactive bids.

3.3 Interactive Approaches

Bellosta et al. (2004) claim that although it is very common to use scoring functions-based weighted-sum, it has some drawbacks. They say that weights are difficult to obtain and unsupported solutions cannot be obtained by using weighted-sum scoring functions. To overcome such limitations they suggest a multi-criteria model based on reference points for a single item English reverse auction.

In their mechanism, the buyer first defines an aspiration point. He/she also defines a reservation point consisting of the minimum acceptable levels of each attribute at the beginning of the auction. The bids are evaluated using scaled deviations from the aspiration levels. They give equal importance to all attributes. At each round of the process, ideal (consists of the maximum of each criterion among bids) and anti-ideal (consists of the minimum of each criterion among bids) points are determined by using the current bids for that round. By using these ideal and anti-ideal points, the deviations can be calculated. Tchebycheff method is applied; the maximum scaled deviation between all attributes is the deviation of that bid. The buyer chooses the bid with the smallest deviation as the best bid for that round. Also in this method, the buyer sets an increment amount for each round, therefore the reservation point is dynamic and at each round it improves. The buyer sends this new reservation point to the bidders except for the best bidder. Based on the given information, sellers update their bids. If a seller does not update his/her bid, i.e. he/she can not give a bid having lower deviation than that of the reservation point, he/she withdraws his/her bid. Otherwise, the process continues until one bidder, the winner remains.

Bellosta et al. (2004) state that their model can overcome some shortcomings of weighted-sum value functions. They give a numerical example to explain the process. However, they do not provide experimental results to compare the efficiency of the model.

Baykal (2007) studies combinatorial auctions. She develops a discount-based model for single attribute multi-unit auctions. She uses two models for multi-attribute multi-item single unit auctions. One is a linear model that can find the supported efficient solutions and the other is augmented weighted Tchebycheff method which can find both supported and unsupported efficient solutions. In the augmented weighted Tchebycheff method, she defines the ideal point by solving each objective independently. To avoid weakly nondominated but dominated solutions, she multiplies the differences of solutions and ideal points with a small positive constant (ε).

She solves these two models and compares the efficient solutions. Then she solves the linear model for different bundle-sized bids and she says that as the size of the bundles gets larger, the number of efficient solution decreases. Moreover she says that if the number of bundles increases, the solution space will get large and thus improved solutions can be found. However, solution time will also increase as the number of bundles increases.

She also uses an interactive method; namely Achievement Scalarizing Function (ASF). She applies a variation of Korhonen and Laakso's (1986) approach, to the multi-attribute multi-unit combinatorial auctions. By using this method, she tries to find the best combination of bids for single round.

Talluri et al. (2007) use data envelopment analysis (DEA) to propose a decision support system tool. They demonstrate the value function by using four attributes; namely price, quality, delivery, and quantity. They try to reflect the correlation between the attributes in their value function. They define weights for each attribute. It is very difficult for a buyer to define exact weights that reflect his/her preference information for attributes. Therefore, instead of using exact weights they define ranges. They divide the DEA model into two stages. In stage I, scores of each bid are evaluated. In stage II, the winning bids are determined.

Using DEA is a good way of demonstrating correlation between attributes. In the formula (for stage II), minimizing the number of bidders is the objective. However, this contradicts with the saying of Bichler and Kalagnanam (2005) about the number of sellers above. Although it is not mentioned in the paper, in open-cry auctions it would be realistic for each round guaranteeing an improvement in at least one attribute, i.e. decreasing the cost to buyer.

CHAPTER 4

AN ALGORITHM FOR MULTI-ATTRIBUTE SINGLE ITEM AUCTIONS

In this chapter we develop an interactive approach that provides aid both to the buyer (DM) and the sellers (bidders) in a multi-attribute single item multi-round reverse auction environment. After an overview of the approach and problem generation, we provide a numerical example in Section 4.4.

4.1 The Algorithm

We develop an approach that supports sellers to bid on a single item. The approach estimates the parameters of a preference function representing the buyer's preferences evaluated on multiple attributes and informs the sellers about the estimations to update their bids for the next round.

The formulation of our problem is as follows: There are I sellers and J attributes in the auction. Each seller gives one bid at each iteration and seller i's bid is represented as $s_i = (a_{i1}, a_{i2}, ..., a_{ij}, ..., a_{iJ})$ where a_{ij} stands for the level of attribute j offered by seller i. The preference function value of the buyer evaluated at seller i's bid is depicted as $u(s_i)$. We use a weighted L_α metric to represent the preferences of the bids to the buyer. This function serves to minimize/maximize the weighted difference of a point from the ideal point in terms of an L_α metric. We estimate the parameter values (α and w_j) based on the past preferences of the buyer to fit the following preference function as an estimate of the preference to bid s_i at any round.

$$u(s_i) = \left(\sum_j \left(w_j \left(a_{ij} - z_j^* \right) \right)^{\alpha} \right)^{1/\alpha}$$

where

w_j : weight of attribute j

z_j^* : ideal level of attribute j

α : parameter of the L_α metric

The contours of the estimated preference functions for different α and w_j combinations are illustrated in Figure 4.1.

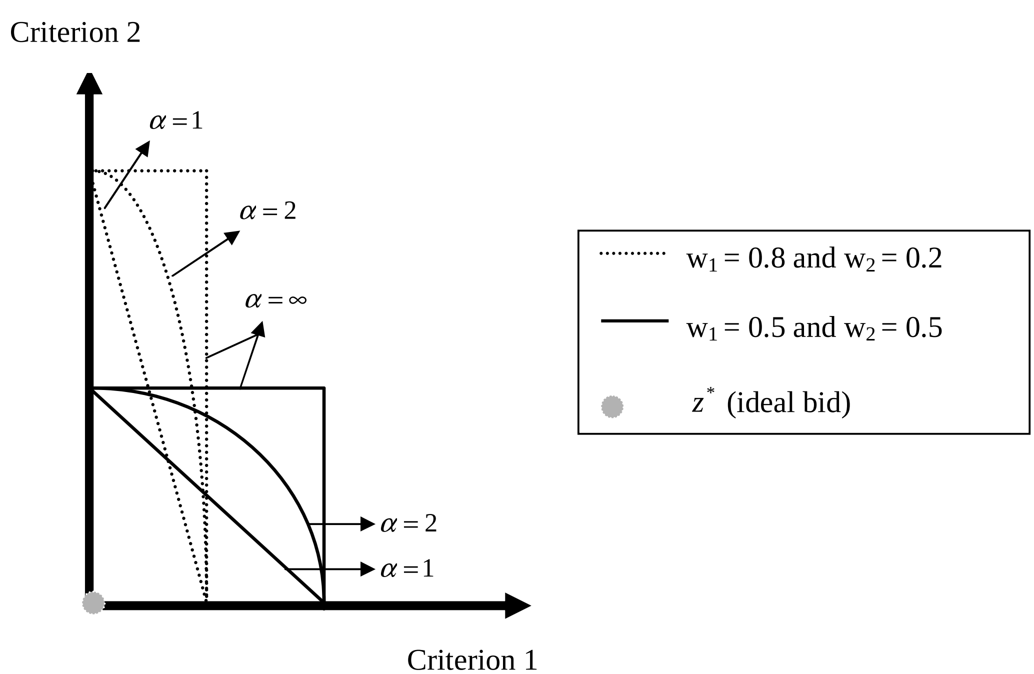


Figure 4.1 The contours for different α and w_j values

In our experiments, we deal with minimization type problems. Therefore, smaller $u(s_i)$ values are preferred by the buyer. If an attribute is of maximization type, we would simply replace $\left(a_{ij} - z_j^*\right)$ with $\left(z_j^* - a_{ij}\right)$ in the distance function. Here, without loss of generality, we assume all attributes are of minimization type. In the following, we explain the algorithm for minimization problems.

We use a constant threshold, "Δ" to represent a minimum preference difference by which the buyer can distinguish between bids. For instance if the buyer prefers A to B, then we require $u(B) \geq u(A)(1+ \Delta)$. Alternatively, we may choose a threshold, Δ', and require $u(B) \geq u(A) + \Delta'$. We will see later that the former leads to additional nonlinearity and using the latter is simpler as it is a linear relationship in Δ'.

In our approach, we estimate a preference function using the past preferences of the buyer. At each round, we expect the sellers to improve their bids by a predetermined "100θ" percent of the best bid of the current round. Therefore we improve the estimated preference function and inform the sellers about our estimations. The estimation of preference function and improvement procedures will be explained in detail in the following sections.

Let P and NP denote the sets of preferred and not preferred bids of the current round, respectively. Let X_h denote the set of constraints derived from the preferences of the buyer in round h where $X_0 = \phi$.

The algorithm can be summarized as follows:

Step 1. Sellers place initial bids. Set the round counter h = 0.

Step 2. Let P = NP = ϕ. Present the buyer all bids and ask him/her to choose the most preferred bid(s). Place the preferred bid(s) in set P and the remaining bids in set NP.

If at least one seller has bid profitably (i.e. improve the buyer's estimated value by 100θ %), go to *Step 3*. Otherwise go to *Step 5*.

Step 3. Update the preference constraint set;

$X_h = X_{h-1} \cup \{ u(s_m) \geq u(s_p).(1+\Delta) \quad \forall\, m \in NP \text{ and } p \in P \}$.

Fit a preference function that satisfies the constraint set X_h for the smallest positive integer α value. Let the estimated preference function value of the best bid of the current round be u^*.

Step 4. Move to a 100θ percent improved contour with a preference function value of $u^{(h)}$, i.e.,

$u^{(h)} = u^*(1-\theta)$.

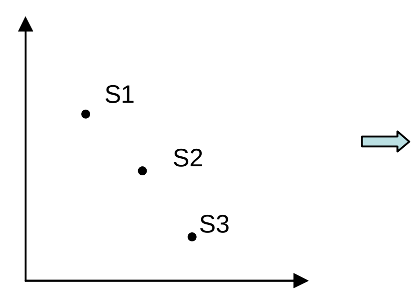
Recommend all sellers to move onto this contour by providing them with the current α, w_j and $u^{(h)}$ values together with the form of the preference function. Let sellers update their bids and set h = h + 1. Go to *Step 2.*

Step 5. Stop. s_p is (are) the winning seller(s) for $p \in P$. If there are more than one winning sellers, the buyer selects one of them.

In Step 4, sellers try to move onto the estimated contour. If they cannot achieve this by making profit, they move as close to the estimated contour as possible with zero profit. We assumed that sellers can gain even if they make no profit (bidding unprofitably) while we do not allow any loss for the sellers. Thus the algorithm continues even when some sellers give bids with zero profit.

In the following, the algorithm for two attribute case is illustrated:

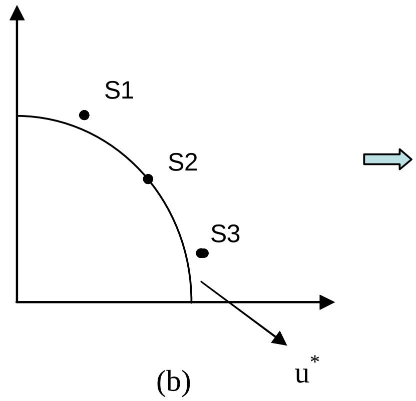
1



(a)

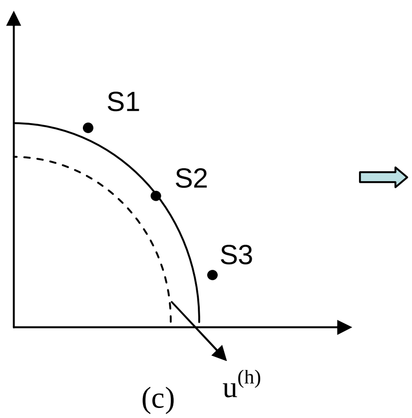
Sellers make their bids. Present the buyer all bids and he/she chooses S2.

2



Fit a preference function using the buyer's past preferences.

3



Improve the estimated preference function value and inform the sellers.

Figure 4.2 The visualization of the algorithm

Sellers update their bids and the auction continues if at least one seller gives profitable bids. Note that, in our study we consider nonnegative attribute values.

4.2 The Parameter Estimation Model

We estimate the parameters (weights) of the preference function by solving the following nonlinear (WALFA) problem for a given α value.

Parameters:

α : estimated parameter of the L_α metric

Δ: predetermined threshold level by which the buyer can distinguish between bids

w_l : lower bound for estimated weights of attributes

w_u : upper bound for estimated weights of attributes

a_{ij} : level of attribute j given by seller i

z_j^* : ideal level of attribute j

We use w_l and w_u restrictions in order to prevent using extreme weights. If one wishes to allow extreme weights, these restrictions can be relaxed.

Decision Variables:

ε : minimum difference between the preference function values of the preferred bids and the other bids

w_j: estimated weight of attribute j

Problem (WALFA)

$$\text{Max } \varepsilon \tag{1}$$

s.to

$$\sum_{j=1}^{J} w_j = 1 \tag{2}$$

$$w_l \le w_j \le w_u \qquad \forall j \tag{3}$$

$$u(s_i) = \left[\sum_{j=1}^{J} \left(w_j \left(a_{ij} - z_j^*\right)\right)^{\alpha}\right]^{1/\alpha} \qquad \forall i \tag{4}$$

$$u(s_m) \ge u(s_p)(1+\varepsilon) \quad \forall \quad s_p \text{ preferred to } s_m \text{ in all rounds so far} \tag{5}$$

$$\varepsilon \ge \Delta \tag{6}$$

For simplification we do not use any subscript to indicate rounds. To estimate the values of α and weights, we solve the nonlinear (WALFA) problem above. The objective (1) is to find the maximum ε value that satisfies the constraints. We use normalized weights (2) and to avoid extreme values of weights, we set upper and lower bounds (3). Bids are evaluated in terms of a weighted L_α preference function (4). The preferences of buyer are taken into consideration in (5). Also we keep the previous information. Since we take the threshold level as Δ, we add constraint (6).

At each round, for given α values we solve (WALFA) using GAMS 22.5 and the global optimization solver, BARON. We take the smallest positive integer α value that yields a feasible solution for the weight values satisfying the constraints. The reason for taking the smallest α value is that we want to fit a function that satisfies all constraints but has the least curvature. Therefore we start with a linear preference function case (α =1) and increase α by 1 if necessary.

We can show that there exist preference relations satisfied by α_1 but not α_2 for $\alpha_1 > \alpha_2$.

Example:

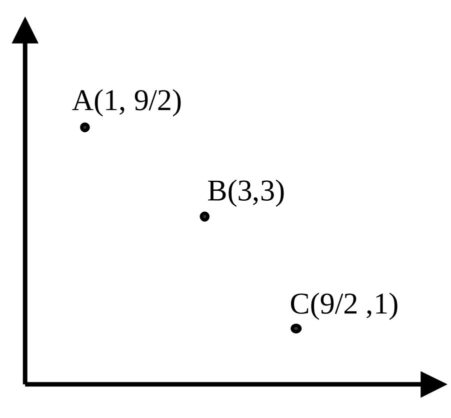


Figure 4.3 An example of bids

Let the buyer prefers B and the following relations exist

$$u(A) \geq u(B) \qquad (7)$$

$$u(C) \geq u(B) \qquad (8)$$

We can show that (7) and (8) can be satisfied for α = 2 but not α = 1.

For $\alpha = 2$

$$w^2 1^2 + (1-w)^2 (9/2)^2 \geq w^2 3^2 + (1-w)^2 3^2 \Rightarrow (45/4)(1-w)^2 \geq 8w^2$$

$$w^2 (9/2)^2 + (1-w)^2 1^2 \geq w^2 3^2 + (1-w)^2 3^2 \Rightarrow \underbrace{(45/4)w^2 \geq 8(1-w)^2}_{\text{for w = 0.5 it holds}}$$

For $\alpha = 1$

$$w\ 1 + (1-w)(9/2) \geq w\ 3 + (1-w)3 \Rightarrow 3/2\ (1-w) \geq 2w$$

$$w\ (9/2) + (1-w)1 \geq w\ 3 + (1-w)3 \Rightarrow \underbrace{3/2\ w \geq 2(1-w)}_{\text{No such weights exist!}}$$

In the following theorems and conjecture we assume that the underlying utility function of the buyer is in the following form:

$$u = \left(\sum_j \left(\lambda_j \left(a_j - z_j^* \right) \right)^t \right)^{1/t}$$

where

λ_j : weight of attribute j

a_j : level of attribute j

z_j^* : ideal level of attribute j

t : parameter of the underlying L_α metric

In our calculations, we pretend that we do not know the parameter values.

Theorem 1: For two attributes, J = 2, if preference constraints are satisfied for α_1 then they will also be satisfied by any α for $\alpha > \alpha_1$.

See Appendix A for the proof.

Conjecture: For any J > 2, if preference constraints are satisfied for α_1 then they will also be satisfied by any α for $\alpha > \alpha_1$.

Theorem 2: We satisfy the buyer's preferences with $\alpha \leq t$.

Proof: In our approach we start with $\alpha = 1$ and at each round if we can not satisfy the preference relations with the current α value, we increase it by 1. As we follow an ascending type procedure and for $\alpha = t$ the preferences are satisfied, the proof is obvious.

Theorem 3: If preference constraints are satisfied by α_h at round h, in the following rounds the estimated α value will be at least α_h ($\alpha_{h+1} \geq \alpha_h$).

Proof: At each round, we take the smallest α satisfying the preference constraints. If α_k does not satisfy the constraints at round k, as we keep the previous information, α_k can not satisfy the constraints of the following rounds. Therefore if the constraints are satisfied by α_h at round h, the smallest possible parameter value of the next round will be α_h.

We fit the preference function using α and corresponding weight values from (WALFA) problem. Let the estimated preference function value of the best bid of the current round be u^*. Then we construct a contour with a preference function value of $u^{(h)}$ where $u^{(h)} = u^*(1-\theta)$. We provide support to all sellers to move onto or closest to this estimated contour. We assume that at each round, sellers give their most profitable bids based on our estimations. Firstly, each seller tries to find profitable bids on the estimated contour. If composing bids on the estimated contour cause any loss, he/she moves to a point on his/her cost curve that is the closest point to the estimated contour.

4.3 Sellers' Model

The cost of a bid to seller i is a function, $f_i(s_i)$, of the attributes in the bid. We assign different cost functions to sellers and at each round, they update their bids according to their cost functions. Each seller first solves the profit model (P_SEL_i) to update his/her bid. By solving this he/she tries to compose a bid on the estimated contour with at least zero profit. If the objective function value of (P_SEL_i) is negative - indicating a loss, he/she then solves the zero profit model (Z_SEL_i) and updates his/her bid.

Parameters:

w_j : the weight of attribute j found from (WALFA)

α : the parameter of the L_α metric used in (WALFA)

z_j^* : ideal level of attribute j

$u^{(h)}$: preference function value of the estimated contour

f_i : the cost function of seller i

z_j^* values are typically the best attainable attribute values for each objective and can usually be extracted from the problem context.

Decision Variables:

a_{i1} : price seller i offers

a_{ij} : level of attribute j to be offered by seller i, $j \geq 2$

d_i : the difference corresponding to the preference function value of the bid of seller i, $u(s_i)$, and the contour, $u^{(h)}$ suggested to all sellers in round h.

Price is a typical attribute in auctions and we define it as attribute 1 for convenience of notation.

Problem (P_SEL_i)

$$\text{Max } z_i = a_{i1} - f_i(s_i) \qquad (9)$$

s.to

$$\left[\sum_{j=1}^{J} \left(w_j \left(a_{ij} - z_j^* \right) \right)^{\alpha} \right]^{1/\alpha} \leq u^{(h)} \qquad (10)$$

Problem (Z_SEL_i)

$$\text{Min } d_i \qquad (11)$$

s.to

$$\left[\sum_{j=1}^{J} \left(w_j \left(a_{ij} - z_j^* \right) \right)^{\alpha} \right]^{1/\alpha} - d_i \leq u^{(h)} \qquad (12)$$

$$a_{i1} - f_i(s_i) \geq 0 \qquad (13)$$

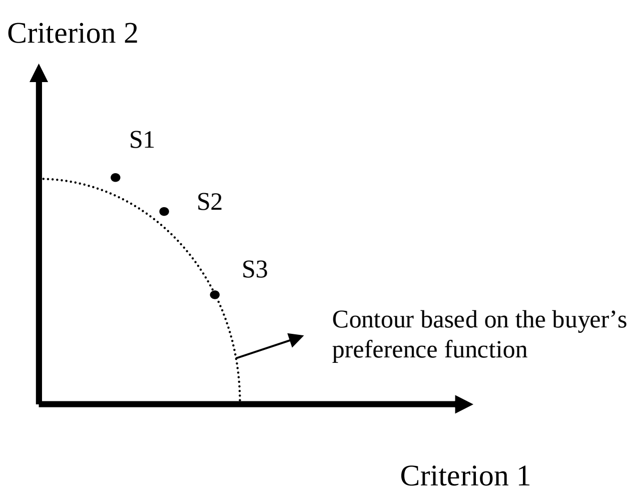
$$d_i \geq 0 \qquad (14)$$

In (P_SEL_i), the objective is to make a bid that maximizes the profit on the estimated contour while satisfying the constraints according to each seller's cost function. On the other hand, in (Z_SEL_i) – for the sellers who cannot bid profitably on the estimated contour – the objective is to move to a point that minimizes the difference between the estimated contour and the seller's cost function.

We generate the (P_SEL_i) and (Z_SEL_i) problems in such a way that sellers consider only their cost functions and our estimations while constructing their bids. However, sellers can have different constraints in real life. In this case, each seller could add his/her own constraint(s) and solves his/her (P_SEL_i) and if necessary (Z_SEL_i) problems.

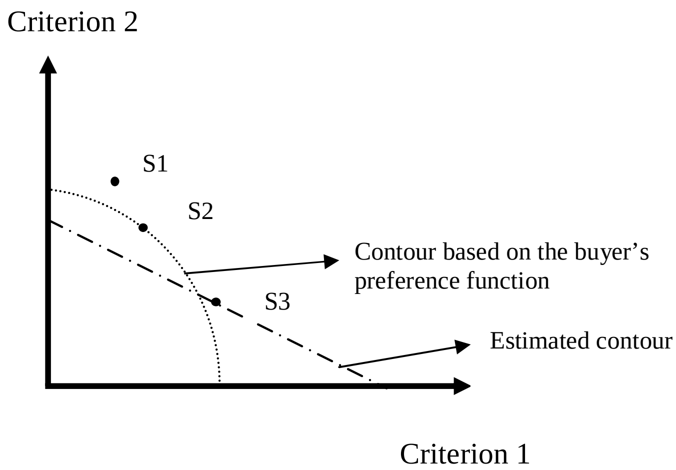
The ideal bid consists of the best attainable attribute values and generally it cannot be satisfied by any seller. At each round, we invite the sellers to a 100θ percent improved estimated contour. Although the estimated parameters of the preference function can change through auction, with 100θ percent improvement, we aim to converge to the ideal bid. As a seller cannot place the ideal bid, sellers have to give zero profit bids at some point and the auction will stop.

In the approach, we estimate the parameters of the preference function based on the preferences of the buyer and the shape of the estimated contour would change as the rounds progress. A seller who makes a bid with zero profit based on the estimations in round h can be the provisional of the next round. Therefore, sellers continue bidding even if they are not on the estimated contour. In the following we provide an example showing that a seller can be selected by the buyer even if he/she bids with zero profit based on our estimations.



According to his/her preference function the buyer selects S3

(a) Bids in Round h



Only seller 3 gives profitable bids on the estimated contour and seller 1 and 2 bid with zero profit. However, the buyer selects S2 at this round.

(b) Bids in Round h+1

Figure 4.4 The selection of a seller who makes bid with zero profit previously

4.4 Numerical Example

In this example, we consider two attributes – defect rate and price – and take the buyer's underlying preference function as follows:

$$u = \left(\left(0.6\left(a_p - z_p^*\right)\right)^4 + \left(0.4\left(a_d - z_d^*\right)\right)^4\right)^{1/4}$$

where,

a_p is the price value and a_d is defect rate value.

The parameter settings will be mentioned in Section 4.5.

It should be noted that although most of the numbers in the following tables have more than four digits after the decimal point, we round them up to four significant digits to simplify the tables.

We create the initial bids as mentioned above and represent them in Table 4.1:

Table 4.1 The Initial Bids

Round 0					
Seller	Price	Defect Rate	Pref.Fn_DM	Pref.Fn_Estimated	0_Profit Frequency
S1	12.5431	1.2	5.0173	8.0040	0
S2	10.2344	1.7	4.0946	6.5668	0
S3	10.0751	2.2	4.0323	6.4909	0
S4	8.6518	2.7	3.4689	5.6145	0
S5	7.7999	3.2	3.1418	5.0999	0
S6	8.1535	3.7	3.2955	5.3489	0
S7	**7.2629**	**4.2**	2.9832	4.8099	0

The provisional winner and his/her bid are written in bold in tables. For instance, in Round 0 according to his/her preference function, the buyer selects seller 7 (S7). Based on this information we estimate the parameters of the buyer's preference function. We start with $\alpha = 1$ and found the following values by solving the (WALFA) problem for this α value.

$w_d = 0.05$ and $w_p = 0.95$

$u^* = 4.8099$

$u^{(0)} = 4.8099(1\text{-}0.05) = 4.5694$

The estimated preference function values are given under "Pref.Fn_Estimated" column in Table 4.1, whereas "0_Profit Frequency" column represents how many times each seller has bid unprofitably up to that round. If all sellers have positive values in the last column at any round indicating all sellers bid unprofitably, the algorithm stops.

We inform sellers about the estimated α and weight values. Also we recommend each seller that his/her preference function value for his/her updated bid should not exceed 4.5694 which is the preference function value of the estimated contour after improvement.

Afterwards, each seller solves his/her own (P_SEL_i) problem and it turns out that all sellers except S1 and S2, can bid profitably. S1 and S2 solve (Z_SEL_i) and the resulting bids are provided in Table 4.2.

Table 4.2 Bids for Round 1

Round 1					
Seller	Price	Defect Rate	Pref.Fn_DM	Pref.Fn_Estimated	0_Profit Frequency
S1	7.9232	3.1208	3.1882	4.8479	1
S2	7.3235	3.6174	2.9720	4.6282	1
S3	**6.8897**	**4.1182**	2.8399	4.4977	0
S4	6.8493	4.6297	2.8727	4.5790	0
S5	6.8099	5.1285	2.9207	4.6582	0
S6	6.7711	5.6202	2.9846	4.7363	0
S7	6.7311	6.1266	3.0682	4.8167	0

In Round 1, the buyer selects S3 as the best among others and for $\alpha = 1$ we find the following weight values by solving (WALFA):
$w_d = 0.2009$ and $w_p = 0.7991$.

For the next round, the sellers are again provided with the information of the estimated α and weight values. Also they are given the preference function value of the estimated contour after improvement. The updated bids can be seen from Table 4.3.

Table 4.3 Bids for Round 2

Round 2					
Seller	Price	Defect Rate	Pref.Fn_DM	Pref.Fn_Estimated	0_Profit Frequency
S1	8.1501	1.8513	3.2622	3.6852	2
S2	7.5452	2.3644	3.0253	3.4541	2
S3	6.9494	2.8533	2.7993	3.2394	1
S4	6.7549	3.3563	2.7422	3.2104	0
S5	6.5659	3.8574	2.7013	3.1941	0
S6	**6.3773**	**4.3576**	2.6798	3.1887	0
S7	6.1882	4.8590	2.6829	3.1941	0

In Round 2, the buyer selects S6. For $\alpha = 1$ we cannot find a feasible solution to (WALFA). We increase α by 1, set it 2 and find the following values:

$w_d = 0.3312$ and $w_p = 0.6688$.

The updated bids for Round 3 are given in Table 4.4.

Table 4.4 Bids for Round 3

Round 3					
Seller	Price	Defect Rate	Pref.Fn_DM	Pref.Fn_Estimated	0_Profit Frequency
S1	8.0032	2.4302	3.2080	3.3403	3
S2	7.4331	2.7688	2.9875	3.1205	3
S3	6.8663	3.1227	2.7754	2.9121	2
S4	6.3034	3.4888	2.5785	2.7203	1
S5	6.1455	3.9002	2.5523	2.7024	0
S6	**5.9883**	**4.3206**	2.5433	2.6987	0
S7	5.8076	4.7465	2.5475	2.7024	0

Again S6 is selected in Round 3. We cannot find a feasible solution for $\alpha = 2$. We increment α value by 1 and for $\alpha = 3$ we estimate the following weight values:

$w_d = 0.3783$ and $w_p = 0.6217$.

The updated bids for Round 4 are given in Table 4.5.

Table 4.5 Bids for Round 4

Round 4					
Seller	Price	Defect Rate	Pref.Fn_DM	Pref.Fn_Estimated	0_Profit Frequency
S1	7.9442	2.8844	3.1914	3.1917	4
S2	7.3763	3.1083	2.9735	2.9738	4
S3	6.8168	3.3525	2.7657	2.7660	3
S4	6.2679	3.6164	2.5739	2.5741	2
S5	5.7643	3.9017	2.4182	2.4184	0
S6	**5.6230**	**4.2628**	2.4156	2.4157	0
S7	5.4416	4.6318	2.4188	2.4188	0

S6 is selected in Round 4 again and the estimated parameter values are as follows:

$\alpha = 4$,

$w_d = 0.3999$ and $w_p = 0.6001$.

Sellers update their bids with the given information. In Round 5, only S6 and S7 give profitable bids. The updated bids for Round 5 are given in Table 4.6.

Table 4.6 Bids for Round 5

Round 5					
Seller	Price	Defect Rate	Pref.Fn_DM	Pref.Fn_Estimated	0_Profit Frequency
S1	7.9132	3.2559	3.1877	3.1880	5
S2	7.3432	3.3942	2.9703	2.9705	5
S3	6.7840	3.5537	2.7633	2.7635	4
S4	6.2399	3.7362	2.5726	2.5728	3
S5	5.7180	3.9426	2.4067	2.4069	1
S6	**5.2776**	**4.1875**	2.2948	2.2949	0
S7	**5.0870**	**4.5101**	2.2949	2.2949	0

In Round 5, the buyer is indifferent between S6 and S7 because the preference function values of the buyer for the two sellers are within the small threshold Δ value we use. We still write constraints indicating the preference of the bids of S6 and S7 over the remaining bids. These, however, do not provide us any new information, i.e. when we solve (WALFA), we found the same α and weight values as in Round 4. Still two sellers bid profitably, thus the algorithm continues. To support sellers for the next iteration, we improve the estimated contour and tell the sellers that their preference functions for their updated bids should not be on 2.1802 (2.2949 x 0.95) to be competitive. The estimated α and weight values of Round 4 are again used for Round 5.

The updated bids for Round 6 are given in Table 4.7.

Table 4.7 Bids for Round 6

Round 6					
Seller	Price	Defect Rate	Pref.Fn_DM	Pref.Fn_Estimated	0_Profit Frequency
S1	7.9132	3.2559	3.1877	3.1880	6
S2	7.3432	3.3942	2.9703	2.9705	6
S3	6.7840	3.5537	2.7633	2.7635	5
S4	6.2399	3.7362	2.5726	2.5728	4
S5	5.7180	3.9426	2.4067	2.4069	2
S6	5.2270	4.1764	2.2773	2.2774	1
S7	**4.7779**	**4.4410**	2.1970	2.1970	1

In this round, the buyer selects S7. When we check the 0_Profit_Frequency column, we see that all numbers are positive indicating that there is no seller bidding profitably. Therefore the algorithm stops and the winning seller is seller 7.

4.5 Experiments

We consider two cases in terms of the attribute numbers: two (defect rate-q, price-p) and three (defect rate-q, price-p, lead time-lt) attribute cases where all attributes to be minimized. In the two attribute case, each seller identifies a *defect rate* and a *price* value for the auctioned item. Here defect rate is used as an indicator of quality; smaller defect rate stands for higher quality. In the three attribute case, in addition to these attributes sellers also identify a *lead time* value about the delivery time of the item being auctioned as a new attribute. Smaller lead times causing higher costs to sellers, are preferred by the buyer. The relations between the attributes will be mentioned in the following sections.

For our example problems, we generate 7 sellers each having his/her own continuous cost function. The cost function of seller i for two attribute case is constructed in the following form:

$$\text{cost}_i = f_i(q_i) = \left(\frac{1}{(q_i - c)^2} + (6.5 - c) \right) 1.2$$

where,

q_i: defect rate value in seller i's bid

c: constant, c = 0.5(i-1)

i =1, 2,…, 7.

We consider a cost function which has a negative relation in defect rate, i.e. as defect rate decreases (or quality increases) cost increases. Then we assign such cost functions to sellers to generate nondominated bids. These cost functions provide each seller to dominate others at different defect rate ranges. For instance, seller 1 dominates the others for defect rates up to 1.54, whereas seller 2 dominates others for defect rates between 1.54 and 2.04. Each seller has his/her own dominating region with such cost functions. This mechanism is created in order to make the problem more interesting by facilitating the sellers to be competitive for different attribute combinations.

While assigning initial defect rates, we take these dominating regions into consideration to generate nondominated bids. We set the maximum defect rate as 10. Then we calculate initial cost values using sellers' cost functions mentioned above. After finding the cost values, we assign a profit rate to each seller and calculate price values for the initial round. We take the minimum profit rate as 20% and the maximum profit rate as 50%. We randomly generate numbers between 1.2 and 1.5 for each seller and multiply these numbers with sellers' cost values to assign initial prices. With this, we assign defect rates and price values to sellers to compose bids for the beginning round.

For the three attribute case the cost function is as follows:

$$\text{cost}_i = f_i(q_i, lt_i) = \left(\frac{1}{(q_i - c)^2} + (6.5 - c) \right) 1.2 + \left(\frac{15}{lt_i^{\,2}} \right)$$

where,

q_i: defect rate value in seller i's bid

lt_i: lead time value in seller i's bid

c: constant, c = 0.5(i-1)

i =1, 2,…, 7.

We construct a cost function that defect rate and lead time are related with cost separately. The relation of the defect rate is the same as that of in the two attribute case. Additionally we consider a negative relationship between cost and lead time, i.e. as lead time decreases cost increases. We take the minimum value of lead time as 3 and the maximum value of lead time as 10.

Cost functions can be constructed in different ways. Our aim was to generate cost functions in such a way that the sellers would be competitive for different combination of attribute values. Developing different, possibly more realistic, cost functions can be the subject of future research.

We use the same initial defect rates in the two attribute case. Then we randomly generate the lead time values and calculate the cost values according to the cost function constructed for three attribute case. Price calculation is the same for both cases. We use these initial bids in our experiments.

We take the maximum price value as 15 and scale it between 0 and 10 and take defect rate and lead time values as they are to bring all attributes to a similar scale. As we try to minimize all attributes, we take the ideal point such that all attributes have zero value.

In our example problems, we take the threshold, Δ, as 0.001. In Section 4.1 we discuss two different ways to use such a threshold level. We consider the former one and say that if the buyer prefers A to B, it requires $u(B) \geq u(A)(1+ \Delta)$. Although it causes nonlinearity in the model, we use it since our problems are small and do not cause significant computational complexity. However, for big problems taking a threshold, $\Delta`$, and require $u(B) \geq u(A) + \Delta`$ will be beneficial in terms of computational complexity. In some of our problems we apply the two methods by setting both Δ and $\Delta`$ to the same constant. The results are not significantly different.

The parameter θ is necessary for the improvement of the bids. If θ is too small, there will be lots of information about the true preference function; however the number of rounds will be too big. On the other hand, if θ is too large, the auction will end prematurely as it is unlikely for the sellers to make such improvements. In our calculations we set the improvement percentage, 100θ, be 5 where θ=0.05. We multiply the preference value of the estimated contour before improvement by 0.95 and find the preference value of the estimated contour that will be presented to the sellers. We set a lower bound of 0.05 and an upper bound of 0.95 for the weights in (WALFA) problem to avoid extreme values of weights in our experiments.

We use an underlying L_α metric representing the preference function of the buyer as follows:

$$u = \left(\sum_j \left(\lambda_j \left(a_j - z_j^* \right) \right)^t \right)^{1/t}$$

where

λ_j : weight of attribute j

a_j : level of attribute j

z_j^* : ideal level of attribute j

t: parameter of the underlying L_α metric

We test the algorithm on 17 problems for different combinations of t and λ_j values for both two and three attribute cases.

4.6 Results

To test the performance of the algorithm, we compare the results of the algorithm with the ones found using the exact parameters values. We want to see what would be the results if we know the preference function of the buyer explicitly and the sellers can bid with zero profit. Thus we solve the following (EX_PAR_i) problem with exact parameter values for each seller to find the best possible defect rate and price combination that the sellers can give with zero profit.

The parameters and the decision variables are the same as before and the model is as follows:

Problem (EX_PAR_i)

$$\text{Min } u(s_i) \tag{15}$$

s.to

$$u(s_i) = \left[\sum_{j=1}^{J}\left(\lambda_j\left(a_{ij} - z_j^*\right)\right)^t\right]^{1/t} \tag{16}$$

$$a_{i1} - f_i(s_i) \geq 0 \tag{17}$$

In (EX_PAR_i), the objective is to compose a bid that minimizes the distance from the ideal point in terms of L_α metric. Sellers try to find this combination without any loss.

When we look at the results of the example problem in Section 4.4, it can be said that our estimations are close to the true values and we guide the sellers well. The last rows of Table 4.8 and Table 4.9 in bold show that the winning sellers are the same (seller 7 for both) and the preference function values of the buyer are nearly the same for both cases. We say nearly as for simplicity we round up the values in the tables to four significant digits, although they look the same there are very small differences. However, these differences can be neglected due to rounding off. Also the preference function value of the buyer for all bids and the bid compositions are closely the same as we guide them with nearly exact parameter values.

Table 4.8 The Results of the Algorithm

Seller	Price	Defect Rate	Pref.Fn_DM
S1	7.9132	3.2559	3.1877
S2	7.3432	3.3942	2.9703
S3	6.7840	3.5537	2.7633
S4	6.2399	3.7362	2.5726
S5	5.7180	3.9426	2.4067
S6	5.2270	4.1764	2.2773
S7	**4.7779**	**4.4410**	**2.1970**

Table 4.9 The Results Found with Exact Values of Parameters

Seller	Price	Defect Rate	Pref.Fn_DM
S1	7.9132	3.2554	3.1877
S2	7.3433	3.3937	2.9703
S3	6.7841	3.5532	2.7633
S4	6.2400	3.7358	2.5726
S5	5.7181	3.9422	2.4067
S6	5.2272	4.1760	2.2773
S7	**4.7781**	**4.4407**	**2.1970**

In all problems for both two and three attribute cases, the winning seller(s) found with the algorithm and (EX_PAR_i) problem with exact parameter values are the same. We also compare the preference function values of the buyer for the winning sellers found with the algorithm and with (EX_PAR_i). To evaluate the performance of the algorithm for these values we use % deviations:

$$\% \text{ deviation} = \left(\frac{u(final\_bid) - u(optimal\_bid)}{u(optimal)}\right)100$$

In the formula $u(final\_bid)$ refers to the preference function value of the final bid of seller i found by the algorithm whereas $u(optimal\_bid)$ refers to the preference function value of the optimal solution found by (EX_PAR_i).

For each problem, we check the percent deviation of the buyer's preference function values of the winning sellers. We also calculate the percent deviations for each seller and report the averages over all sellers. As can be seen from Tables 4.10 and 4.11, the percent deviations are very small, i.e. for all problems the buyer's preference function found with the algorithm is close to that found by (EX_PAR_i) with the exact parameters. These imply that the estimation and guidance mechanisms of our approach worked well in all the test problems.

Table 4.10 % Deviations Between the Results of the Algorithm and Exact Solutions for Two Attribute Case

	α=1	**α=2**	**α=3**			**α=4**				
	w_p=0.8 w_d=0.2	**w_p=0.5 w_d=0.5**	**w_p=0.8 w_d=0.2**	**w_p=0.5 w_d=0.5**	**w_p=0.3 w_d=0.7**	**w_p=0.8 w_d=0.2**	**w_p=0.6 w_d=0.4**	**w_p=0.5 w_d=0.5**	**w_p=0.4 w_d=0.6**	**w_p=0.2 w_d=0.8**
Winning_seller(s)	0.0000	0.0013	6.4718	0.0000	0.3072	0.0003	0.0000	0.0209	0.0142	0.1308
Average	0.0000	0.0011	0.9537	0.0000	0.4753	0.0184	0.0000	0.1061	0.1665	2.4178

Table 4.11 % Deviations Between the Results of the Algorithm and Exact Solutions for Three Attribute Case

	α=1	**α=2**	**α=3**			**α=4**	
	w_p=0.3 w_d=0.4 w_{lt}=0.3	**w_p=0.2 w_d=0.1 w_{lt}=0.7**	**w_p=0.7 w_d=0.2 w_{lt}=0.1**	**w_p=0.2 w_d=0.1 w_{lt}=0.7**	**w_p=0.3 w_d=0.4 w_{lt}=0.3**	**w_p=0.7 w_d=0.2 w_{lt}=0.1**	**w_p=0.2 w_d=0.1 w_{lt}=0.7**
Winning_seller(s)	1.8714	0.0001	0.1456	0.2149	0.0191	0.0041	0.0083
Average	1.7383	0.0001	0.0256	0.2782	0.0183	0.0169	0.0026

For practical purposes we use constant Δ as 0.001 in our calculations. As we scale the attributes, the preference function values of the bids are not so different from each other. Therefore, we say that using constant threshold level is reasonable for our testing. However it is also possible to change the threshold depending on α value instead of taking it as constant. The reason is that for any given point, as α gets larger the preference value decreases for the same weights and the threshold level can change. Because of this, the threshold level of the buyer can be taken as $J^{1/\alpha}\Delta^*$ where J is the number of attributes showing the dimension and Δ^* is a predetermined value.

Another point is that, in our experiments, we deal with small number of sellers and at each iteration we present all bids to the buyer. However, for big sized auctions a representative group of bids can be selected using some filtering methods like in Steuer (1986) and the buyer can be asked to choose the best among them.

Lastly, ideal point can be dynamic. Although we use a static ideal point in our experiments, it can be updated at each round. After the buyer is presented the bids of the current round, point based on these bids he/she can update the ideal point for the next round.

CHAPTER 5

AN EVOLUTIONARY ALGORITHM APPROACH TO MULTI-ATTRIBUTE MULTI-ITEM AUCTIONS

The Evolutionary Algorithms (EAs) have been very popular during the last decade (http://neo.lcc.uma.es/opticomm/introea.html). The application of EAs in multi-objective optimization has also received growing interest from researchers (Fonseca and Fleming, 1995). EAs maintain many solutions in a single run; hence, they are particularly useful in multi-objective problems (Deb, 2001).

The advantages of EAs are described by Fonseca and Fleming (1995) as

- the ability to handle complex problems,
- involving features such as discontinuities, multimodality, disjoint feasible spaces and noisy function evaluations,
- reinforces the potential effectiveness of EAs in multi-objective search and optimization

In this chapter, we develop an EA for the case of multi-attribute multi-item reverse auctions. We try to generate the whole Pareto front using the EA. We also develop heuristic procedures to seed several initial solutions in the initial population. We test the EA on a number of randomly generated problems and compare the results with the true Pareto optimal front obtained by solving a series of integer programs.

5.1 EA Applications in Auctions

As discussed in Section 2.2.3, there are different types of auctions; one of which is the multi-attribute auction. Multi-attribute auction process considers not only the price but also other attributes.

Gaytán et al. (2005) apply an EA to solve multi-attribute multi-item reverse auctions. In their approach, sellers make their bids representing which items they want to supply without support for the first round. Then the buyer finds the efficient front by using the EA. The sellers give their bids for the second time without support. The buyer finds the new efficient front. Afterwards the buyer selects a criterion (i.e. price) and finds the point that has the maximum difference between the two Pareto fronts in terms of this criterion. If the buyer likes this new point, the auction ends and the sellers in the winning combination are informed. Otherwise the buyer expresses his/her preferences like: quality improvement while not increasing price too much, etc. to sellers. Sellers update their bids using this information and the auction continues.

5.2 Methodology

In this study, a multi-attribute multi-item reverse auction problem is solved by adopting an EA, namely the Non-Dominated Sorting Genetic Algorithm NSGA II (Deb et al. 2002).

5.2.1 Problem Definition

The problem consists of one buyer, I sellers, J attributes and K items where there exists one unit for each item. All sellers can bid for all items in our problem. We try to find the best combination in terms of J attributes to supply all of these items. No extra constraint exists.

In our example problems we consider two objectives: *defect rate* and *price* where both are to be minimized. Each seller identifies a defect rate and a price value for each item he/she wants to supply separately. We used defect rate as an indicator of quality as in Chapter 4.

We consider two cases: *original case* and *discounted case* where there exists price discounts in the latter one.

Original Case:

In the original case, to calculate the defect rate of a combination, we sum up the offered defect rates in the combination. It is equivalent to taking the average. Alternatively, maximum of the offered defect rates in the combination can be taken as the defect rate of the combination. In some situations summing up the defect rates may not be reasonable. Weighted sum could be an option to calculate the defect rate of the combination. On the other hand, to calculate the total price of the combination we sum the offered prices in the combination.

We scale both objectives. We find the minimum possible and maximum possible values for combinations in terms of both objectives. Then we use these values to linearly scale the objectives between 0 and 1.

Discounted Case:

With the same bids, we also try to find the best combination if there exist price discounts. In this case, sellers have different threshold levels for applying discount. If the number of items supplied by a seller is greater than or equal to his/her threshold level, an amount of discount determined by that seller is applied. Discount amounts are different for different sellers. In our study we only consider one threshold level for each seller. Scaling is the same as the original case.

The mathematical formulations of the problems are provided below:

Mathematical Formulation of Original Case

Decision Variables

X_{ik} : $\begin{cases} 1 \text{ if item k is supplied by seller i} \\ 0 \text{ otherwise} \end{cases}$

Parameters

q_{ik} : defect rate given by seller i for item k

p_{ik} : price given by seller i for item k

where; i (sellers) : 1, 2, 3, . . . , I and k (items): 1, 2, 3, . . . , K

Problem(Original)

Minimize $z_1 = \sum_{i,k} X_{ik}q_{ik}$

Minimize $z_2 = \sum_{i,k} X_{ik}p_{ik}$

s.to

$\sum_{i} X_{ik} = 1$ $\quad\forall k$ (each item will be supplied by a seller)

$X_{ik} \geq 0$

Mathematical Formulation of Discounted Case

Decision Variables

X_{ik} : $\begin{cases} 1 \text{ if item k is supplied by seller i} \\ 0 \text{ otherwise} \end{cases}$

m(i): $\begin{cases} 1 \text{ if constraint (1) is active for seller i} \\ 0 \text{ if constraint (2) is active for seller i} \end{cases}$

t(i) : total number of items supplied by seller i in a combination

prc(i) : total price of items seller i is assigned to supply

Parameters

q_{ik} : defect rate given by seller i for item k

p_{ik} : price given by seller i for item k

th(i) : threshold level determined by seller i

d(i) : discount amount determined by seller i at threshold level th(i)

M: big number

where; i (sellers) : 1, 2, 3, . . . , I and k (items): 1, 2, 3, . . . , K

Problem(Discounted)

$$\text{Minimize } z_1 = \sum_{i,k} X_{ik}q_{ik}$$

$$\text{Minimize } z_2 = \sum_{i} prci(i)$$

s.to

$$\sum_{i} X_{ik} = 1 \qquad \forall k \text{ (each item will be supplied by a seller)}$$

$$t(i) = \sum_{k} X_{ik} \qquad \forall i \text{ (total number of items supplied by seller i)}$$

$$t(i) \geq th(i) - Mm(i) \qquad \forall i$$

$$prc(i) \geq \sum_{k} X_{ik}p_{ik} - M(1\text{-}m(i)) \qquad \forall i \qquad (1)$$

$$prc(i) \geq \sum_{k} X_{ik}p_{ik}(1\text{-}d(i)) \qquad \forall i \qquad (2)$$

X_{ik}, m(i) : binary

t(i), prc(i) ≥ 0

5.2.2 The EA

As mentioned above, NSGAII (Deb et al. 2002) is modified for our problem. We use real-valued representation. That is, each gene in a chromosome is represented by an integer reflecting the seller of that item.

3	6	8	. . .	. . .	4	4

Figure 5.1 An example of item – seller relation

In Figure 5.1, a chromosome showing which item is supplied by which seller is depicted. The chromosome cells refer to the items and the numbers in the cells refer to the sellers. The chromosome represents that item1 is supplied by seller 3 whereas item 2 is supplied by seller 6 and items K-1 and K are supplied by seller 4.

5.2.3 Overview of the Algorithm

Firstly, an initial parent population P_0 with population size of N is generated and sorted based on the nondomination sorting (Deb, 2001). That is, all nondominated solutions are in the first front. Once the first front is eliminated, all nondominated solutions of the resulting population are in the second front and so forth. A solution's nondomination level, rank, is used as its fitness function value. By using selection, recombination, and mutation operators, an offspring population, Q_0, with size N is created. For each generation, t, thereafter, parent population and offspring population are combined to form a global population, G_t. After sorting G_t, the best N members among the parents and the offspring are selected based on their nondomination level and crowding distance values. While selecting the N best solutions, their rank values are checked where smaller rank values are preferred. Firstly, the solutions with rank 1, which have the smallest rank, are selected. If the number of these solutions is less then the predetermined population size, N, we continue to fill the elite population with the solutions having rank 2. Whenever the total number of solutions exceed N, we eliminate those in a denser region based on a crowding distance measure at the current rank. We also eliminate all solutions in higher ranks (Deb, 2001). This maintains the diversity among the solutions. Afterwards, these best N members are referred as the new parent population members of the next generation, P_{t+1} where t is the current generation counter.

The outline of the algorithm is as follows:

- Generate the initial population P_0 by using the random number generator used in the NSGA II code, population size N.
- Evaluate objective values of the combinations and scale them
- Rank the solutions (nondomination sorting)
- Generate offspring population Q_0 of size N with Selection, Recombination and Mutation: Tournament Selection is used in the algorithm (Deb, 2001). In the tournament selection, two solutions from the parent population are selected. These solutions are compared and the better one is put into a mating pool. This procedure continues until the mating pool is filled with N solutions. Afterwards, uniform crossover is applied to the solutions in the mating pool (Deb, 2001). That is, two offspring are generated from two parents where each gene of one offspring is inherited with a probability p_c from either parent. The procedure is illustrated below:

Let P1 and P2 be two parent solutions in the mating pool and O1 and O2 be the offspring solutions generated from P1 and P2, where

P1

3	1	2	4

P2

4	1	2	4

For gene 1, a number is randomly generated, r. If $r < p_c$, than the first gene of O1 takes the value of 4 and the first gene of O2 takes the value of 3. Otherwise, the first gene of O1 takes the value of 3 and the first gene of O2 takes the value of 4. This procedure is repeated for each gene.

Lastly, mutation is applied to the offspring solutions created with crossover operation. The mutation probability is selected as p_m. Each gene can be mutated with a probability p_m and if it mutates, a randomly generated integer value between 1 and I is assigned to the mutated gene. With this, offspring population Q_0 is constructed.

For each generation perform the following operations:

- Combine parent and offspring populations and form a global population, G_t
- Rank the solutions in the global population and determine crowding distance between points having the same rank value
- Fill the elite population (this population is called elite as they have small ranks) with size N
- Set the elite population as P_{t+1}
- Create the next generation with genetic operators (selection, recombination and mutation).

5.3 Input Generation

We randomly generate defect rates and prices to construct our problems. We consider items from different categories and set different defect rate and price ranges for these categories. Some items are more valuable than others with small defect rates and high prices. We take 3 categories and corresponding attribute ranges are given in Table 5.1.

Table 5.1 Categories vs Attribute Ranges

Category	Defect Rate	Price
A	0.1 – 2.0	61 – 100
B	2.1 – 6.5	31 – 60
C	6.6 – 10.0	1 –30

We take the minimum defect rate as 0.1 and the maximum defect rate as 10.0 whereas we take the minimum price value as 1 and the maximum price value as 100. We randomly generate integer price values using a discrete uniform distribution.

Firstly, we randomly assign the categories to the items using uniform distribution. For each seller, we generate attribute values for each item using the defect rate and price value ranges of the corresponding category.

After generating the defect rate and price randomly, we use the defect rate as it is and to make it realistic (generally there is a tradeoff between defect rate and price; if one is small the other is large), we impose a negative correlation between these two attributes. We take the reciprocal of the defect rate and add it to the randomly generated price to find the final price given for the item that has been already assigned a defect rate and category.

For the discounted case, we also generate threshold levels (an integer between 2 and 7) for each seller randomly by using a discrete uniform distribution. Then, discount amounts between 3% (0.03) and 7% (0.07) are randomly generated with a discrete uniform distribution. Afterwards, these randomly generated discount amounts are added to the threshold levels to find the final discounts. The reason is that, we expect large discounts if the threshold level is high.

The true Pareto optimal frontiers for both the original and discounted case problems are obtained on GAMS 22.5 by using the ε-constraint method (Haimes et al. 1971). We use ε-constraint for defect rate and at each iteration, the ε value is increased by 0.1 over the ε value of the previous iteration since the minimum value of the defect rate among the items is 0.1.

5.4 Simulation Results

In our experiments we choose the selection strategy as tournament selection (Deb, 2001) and use uniform crossover with p_c =0.9 based on our preliminary experiments and bitwise mutation with p_m =0.01. We use the same seed for random number generator in each run. That is, we start each run with the same initial population and generate the same random numbers at each run.

We modify the code downloaded from the website of Kanpur Genetic Algorithms Laboratory (http://www.iitk.ac.in/kangal/codes.shtml) and we implement our algorithm in C programming language.

We test the algorithm for different problems. We construct three combinations of the number of items and sellers (IxK): 10x20, 30x30, and 30x100. For each combination we generate 5 problems and we experiment on these problems.

For each problem, the numbers of nondominated solutions for both original and discounted cases are given in Table 5.2.

We run different versions of the algorithm and test its performance against the true Pareto frontier. To test the performances, we use Hypervolume Indicator (Zitzler and Thiele, 1998) and the Inverted Generational Distance Metric (Van Veldhuizen and Lamont, 2000). The former measures the total objective space dominated by a population with respect to a given reference point. In our study we take the nadir point as the reference point as in Karahan (2007). On the other hand, the latter measures the algorithm's performance in terms of both convergence and diversity. In this metric, the Euclidean distance of each solution in the true nondominated front to the closest solution in the population generated by the algorithm is calculated. Then the average of these minimum distances is taken.

For the Hypervolume Indicator (HI) larger values are preferred, whereas for the Inverted Generational Distance Metric (IGDM) smaller values are desirable.

In the following, we give the results of the algorithm for a problem called as 30x100_pr4 in combination of 30 items and 100 sellers. We discuss the results for both original and discounted cases of this problem. All performance results for all problems are provided in Appendix C.

Table 5.2 Number of Nondominated Solutions on the True Pareto Front

Problem	Original Case	Discounted Case
10x20_pr1	39	69
10x20_pr2	31	40
10x20_pr3	30	22
10x20_pr4	37	49
10x20_pr5	23	34
30x30_pr1	61	174
30x30_pr2	54	159
30x30_pr3	88	191
30x30_pr4	76	209
30x30_pr5	70	172
30x100_pr1	85	209
30x100_pr2	57	129
30x100_pr3	96	161
30x100_pr4	74	168
30x100_pr5	79	184

5.4.1 Original Case

In the original case, we run 2 versions of the algorithm: *without seeding* and *seeding by sorting* as explained later. Our preliminary experiments show that when we run the algorithm for the original case without seeding, the solutions converge at the generation number of 7000 for 30x100 combination. We use different population sizes in our preliminary experiments and we get the best results for population size of 100. Thus in our experiments we set the population size to 100. In Figure 5.2, resulting solutions of running the algorithm for 7000 generations without seeding are represented.

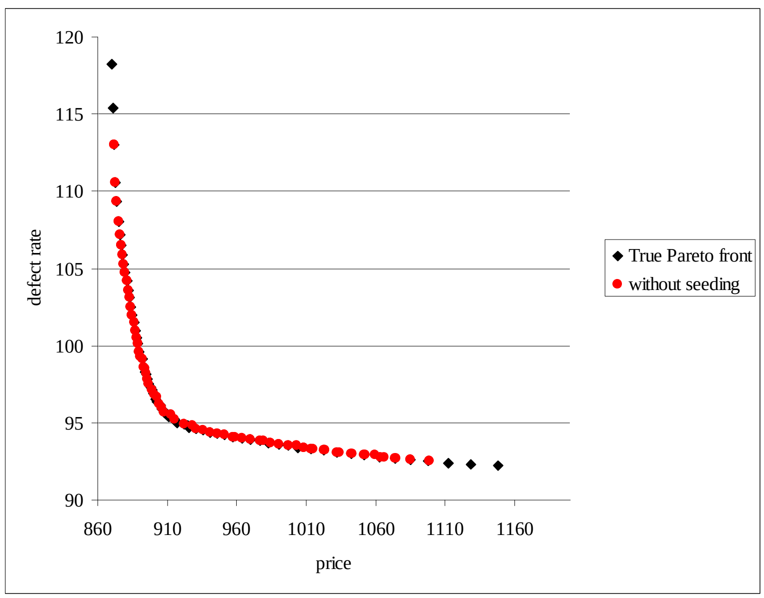


Figure 5.2 Solutions found by "without seeding" for Original Case

As can be seen from Figure 5.2, when we just run the modified NSGA II, convergence of the algorithm to the true front seems satisfactory; however we cannot capture some (extreme) regions. It is a deficiency of NSGA II.

We, then seed the initial population with the best solutions for each objective. We can solve the Problems (Original) in a solver to minimize one of the objectives with the addition of the other objective multiplied by a small constant to get a nondominated solution. For instance, to find the best combination in terms of the defect rate, the objective function will be in the following form.

Min $z_1 + z_2\Delta$

where

z_1: defect rate objective

z_2 : price objective

Δ : small positive constant

When we perform this for each objective we can find the best nondominated solutions for each objective. On the other hand, for the original case it is not difficult to obtain the best nondominated solutions for each objective without using a solver. We can find the best nondominated combination with a simple *sorting procedure*. Firstly, for each item, we choose the seller who offers the minimum price for that item. If there exists one seller, we assign this seller to that item. If there are more than one seller offering the minimum price for that item, then we compare the defect rate values offered by these sellers and choose the one giving the smallest defect rate value for that item. We continue this procedure for all items and find the best nondominated combination in terms of price objective. In the same manner, for each item we select the seller offering the minimum defect rate value and find the best nondominated combination for the defect rate objective. Then these two combinations are added to the initial population and the remaining (100-2=98) combinations are randomly generated. This is referred as "*seeding by sorting*" and the resulting solutions for 3000 generations can be seen in Figure 5.3.

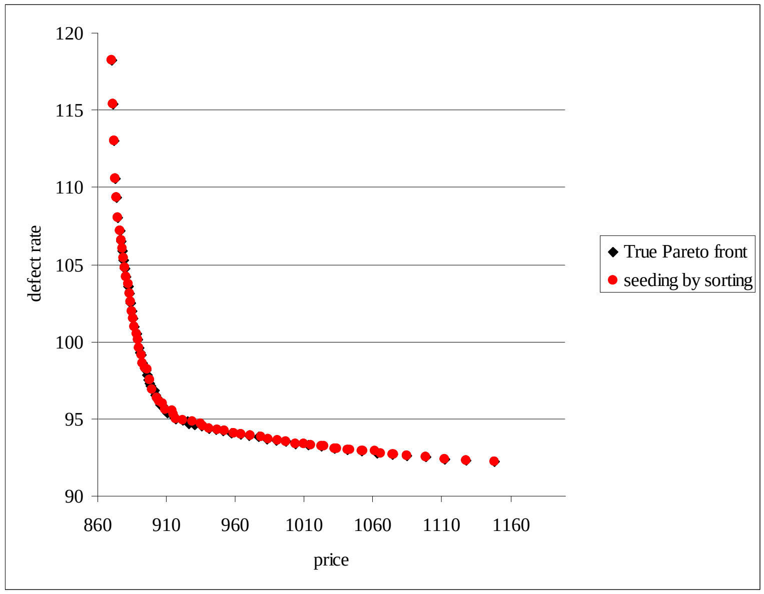


Figure 5.3 Solutions found by “seeding by sorting” for Original Case

When we look at the graph, we see that both the convergence and diversity are satisfactory. With these solutions, we can represent the true front well.

In the original case we have 74 nondominated solutions on the true Pareto front. We try to represent the Pareto front by at most 100 nondominated solutions. In Table 5.3, the performance measures in terms of HI* and IGDM, durations and the generation numbers for both without seeding and seeding by sorting versions are provided. We use HI* to represent the ratio of the hypervolumes of the algorithm and the true Pareto front. Larger HI* values are preferred.

HI* = Hypervolume of the algorithm / Hypervolume of the true Pareto front

Table 5.3 Performance Measures for Original Case 30x100_pr4

Version	**HI***	**IGDM**	**Duration (second)**	**Generation**
without seeding	0.9977	0.0102	64.45	7000
seeding by sorting	0.9985	0.0029	29.97	3000

The results show that for 30x100_pr4, seeding by sorting outperforms without seeding. To generalize, we look at the average HI* and IGDM values for different combinations in Table 5.4.

Table 5.4 Performance Measures for Original Case

Problem	Average HI*	Average IGDM
10x20_without seeding	0.9902	0.0107
10x20_seeding by sorting	0.9979	0.0034
30x30_without seeding	0.9921	0.0082
30x30_ seeding by sorting	0.9942	0.0057
30x100_without seeding	0.9976	0.0062
30x100_ seeding by sorting	0.9977	0.0050

The performance measures in Table 5.4 show that for the original case both versions seem good. However the seeding by sorting version of the algorithm with 3000 generations can be preferred, as it requires significantly fewer generations when compared with without seeding version (7000 generations) and it is a simple procedure to apply.

5.4.2 Discounted Case

For the discounted case, we compare different versions of the algorithm running each for 10000 generations based on our preliminary experiments. We have 168 nondominated solutions on the true Pareto front and we try to approximate it with a population size of 200 according to our preliminary experiments. We observe that the price range on the true Pareto front is lower than that in the original case, as discount is applied on the price in the discounted case. We also observe that the defect rate range is wider in the discounted case. The reason is that in the discounted case we have additional constraints, i.e., we try to make groups of items to take advantage of the price discount. Therefore, the defect rate value of the combination gets larger.

In Figure 5.4, we present the solutions corresponding to the *without seeding* version of the algorithm.

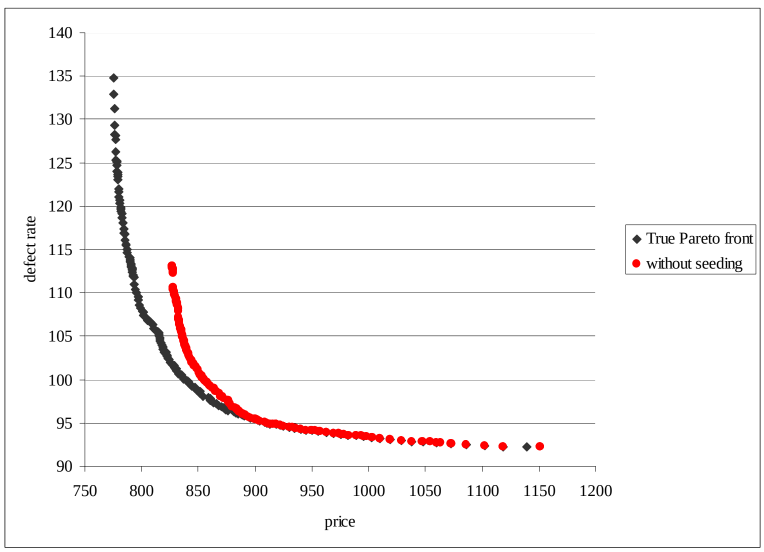


Figure 5.4 Solutions found by “without seeding” for Discounted Case

As can be seen in Figure 5.4, without seeding, we cannot capture some portions of the true front. Convergence does not seem bad, but it can be improved by increasing the number of generations. Afterwards, we seed the initial population with the best combinations for each objective. We first solve the

Problems (Discounted) in GAMS to find the best nondominated combinations for each objective. In fact, the sorting procedure gives the best nondominated solution for the defect rate objective. We use solver only to find the best nondominated combination for the price objective. We add these two combinations to the initial population and generate remaining solutions randomly and run the algorithm. This version is called the "*optimal seeding*". We give the nondominated solutions found with optimal seeding in Figure 5.5.

Convergence and diversity are good when we use optimal seeding and the true Pareto front can be represented well.

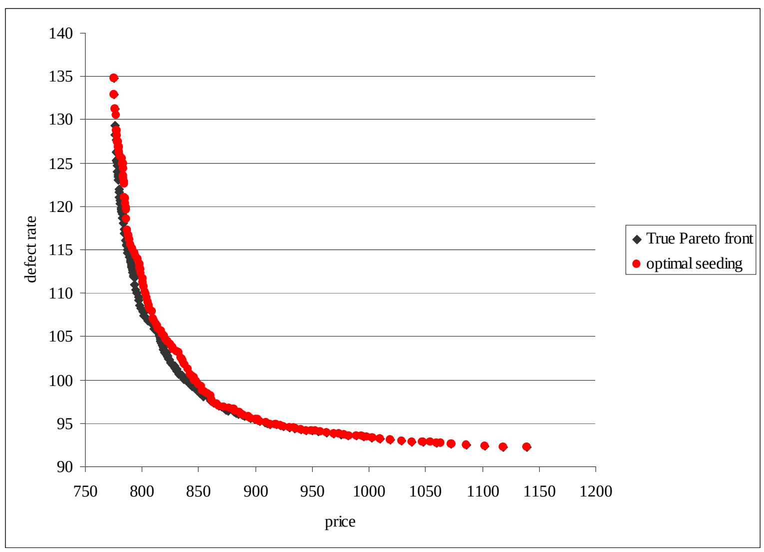


Figure 5.5 Solutions found by "optimal seeding" for Discounted Case

The initial population can be seeded by higher number of good combinations on the true Pareto front. However, these combinatorial problems are generally hard to solve. Typically good combinations for each objective are found and seed into the initial population. Therefore, we also consider a heuristic named as "*rank* *heuristic*" developed for the discounted case, for the cases where it is hard to find the optimal combination for price objective.

Overview of the Rank Heuristic

Firstly, we find the discounted prices offered by each seller for each item. Then we sort these discounted prices in ascending order for each item. At the top, the minimum discounted price for each item exists. Next, we select the seller who offers the maximum discounted price at the top and assign as many items as his/her threshold level to this seller. Let his/her threshold level be t1 and if t1 is greater than or equal to K where K is the total number of items to be supplied we assign all K items to him/her. Otherwise he/she will supply t1 items and we look for the seller having the second maximum discounted price at the top. If the seller is not assigned to supply any item, he/she is assigned to supply as many as his/her threshold level, t2, if the total items assigned (t1 + t2) is less that K. Else he/she is assigned to supply K - t1 items. If the seller offering the second maximum discounted price at the top is assigned to supply some amount, we skip and continue with the item offered by the next maximum discounted price. We progress to search until all K items are assigned to be supplied. Then we start from the item having the maximum discounted price at the top and search for the seller offering smaller price among the ones assigned to supply predetermined amount of items mentioned above. This process continues until all items are supplied.

To illustrate the procedure, say we have three items and four sellers. The discount amounts, threshold levels and the suggested prices for each seller are as follows:

Table 5.5 Original Prices

	Item 1	Item 2	Item 3
Seller 1	4	15	6
Seller 2	6	20	8
Seller 3	2	30	5
Seller 4	5	25	4

Table 5.6 Parameters Values

	Discount Amount	Threshold Level
Seller 1	10%	2
Seller 2	15%	1
Seller 3	10%	3
Seller 4	20%	2

Firstly, we find the discounted prices offered by each seller. The values are given in Table 5.7.

Table 5.7 Discounted Prices

	Item 1	Item 2	Item 3
Seller 1	3.6	13.5	5.4
Seller 2	5.1	17.0	6.8
Seller 3	1.8	27.0	4.5
Seller 4	4.0	20.0	3.2

When we sort the discounted prices in ascending order, the following table is constructed.

Table 5.8 Ordered Discounted Prices

Item 1	Item 2	Item 3
1.8 (S3)	13.5 (S1)	3.2 (S4)
3.6 (S1)	17.0 (S2)	4.5 (S3)
4.0 (S4)	20.0 (S4)	5.4 (S1)
5.1 (S2)	27.0 (S3)	6.8 (S2)

The minimum discounted prices offered for each item are in the first row of Table 5.8. In the cells of the table, discounted price and the seller by whom the price is offered exist. Then, we find the maximum discounted price among these minimum values. This value is 13.5 and offered by seller 1 for item 2. We assign item 2 to seller 1 and as his/her threshold level is 2 we say that one more item should be assigned to him/her. Total number of items assigned is 2. We continue since the total number of assigned items up to now is less than the total number of items to be assigned. Then we move to the next maximum which is 3.2 offered by seller 4. We assign item 3 to seller 4 and although the threshold level of seller 4 is 2, we assign him/her only one item as the total number of items assigned becomes 3. According to the heuristic, item 2 is assigned to seller 1, item 3 is assigned to seller 4 and remaining item, item 1, is assigned to seller 1 as we say that seller 1 will supply items as amount of his/her threshold level, 2.

With this approach, it is hoped to obtain a representative solution for the best combination in terms of price. Since the calculation for defect rate is the same for both original and discounted case, we can find the best combination for defect rate with sorting procedure mentioned above. These two combinations are seeded to the initial population and the result is as follows:

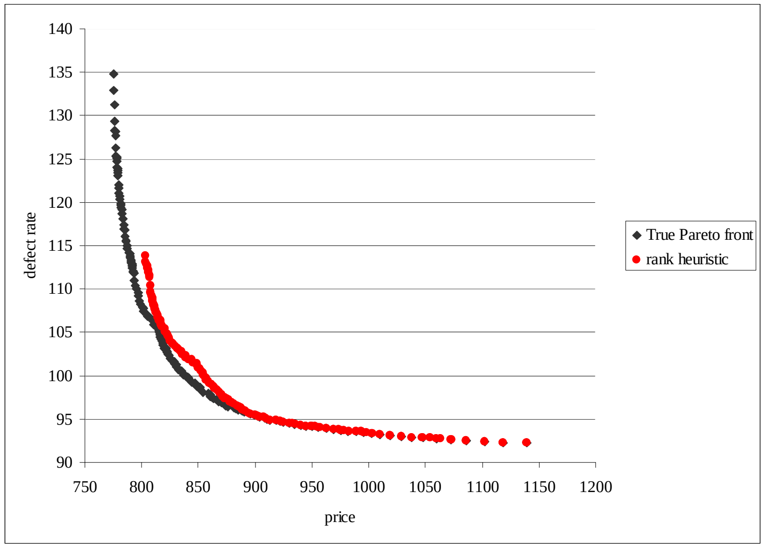


Figure 5.6 Solutions found by “rank heuristic” for Discounted Case

It can be seen in Figure 5.6 that the algorithm converges well to the true Pareto front however it cannot capture the whole front. In the following, the performance measures for 30x100_pr4 and for all problems are given in Tables 5.9 and 5.10, respectively.

Table 5.9 Performance Measures for Discounted Case 30x100_pr4

Version	**HI***	**IGDM**	**Duration (second)**	**Generation**
without seeding	0.8982	0.0876	163.77	10000
optimal seeding	0.9861	0.0110	213.04	10000
rank heuristic	0.9494	0.0611	178.57	10000

Table 5.10 Performance Measures for Discounted Case

Problem	Average HI*	Average IGDM
10x20_without seeding	0.9628	0.0393
10x20_optimal seeding	0.9967	0.0029
10x20_rank heuristic	0.9690	0.0307
30x30_without seeding	0.8071	0.1218
30x30_optimal seeding	0.9696	0.0205
30x30_rank heuristic	0.9374	0.0484
30x100_without seeding	0.8675	0.0824
30x100_optimal seeding	0.9770	0.0173
30x100_rank heuristic	0.8842	0.0801

The performance measures in Table 5.9 show that for our specific problem, 30x100_pr4, optimal seeding is the best. The worst one among three versions is without seeding with HI* value of nearly 90% and a small IGDM value. When we look at Table 5.10, the same analysis can be made: the best version is the optimal seeding and it has a hypervolume value nearly the same as the true Pareto front with very small IGDM values for all problems. On the other hand, the rank heuristic is a simple heuristic and it can be improved.

CHAPTER 6

CONCLUSIONS

In this study, we deal with multi-attribute reverse auction problems. We develop two approaches, one for multi-attribute single item auctions and the other for multi-attribute multi-item auctions. Former is an interactive approach where we support both parties (buyer and sellers), whereas the latter one is an EA approach.

In the interactive approach, we estimate the parameter values of the underlying preference function of the buyer using his/her past preferences. At each iteration we improve the estimated preference function values and inform the sellers about our estimations. Then the sellers update their bids and the auction continues if there exists at least one seller bidding profitably. We test the algorithm on a number of problems. The results show that in all test problems, the correct winning sellers (achieved with exact solutions) are found. Also the buyer's preference function is closely approximated. Therefore, it can be said that we guide the sellers well with asking reasonable number of questions.

We also develop an Evolutionary Algorithm (EA) for multi-attribute multi-item reverse auctions where we try to generate the whole Pareto front. In our study, we develop different heuristic procedures to find good initial solutions to seed the initial population. We modify the NSGA II (Deb et al. 2002) and apply different versions of the algorithm on randomly generated problems. We test the performance of the algorithm using both the Hypervolume Indicator and Inverted Generational Distance Metric. The results indicate that the algorithm can represent the true Pareto front well. As a future study, other multi-objective EAs developed for different problems can be adopted to solve our problem and comparisons can be made between the algorithms. Another point is that in our study, we consider single threshold level for each seller for the discounted case (Section 5.2.1). However, the algorithm can be tested for problems with different threshold levels for each seller.

We intend to develop an interactive EA for multi-attribute multi-item reverse auctions. For multi-round auctions using a preference based EA will be beneficial as we do not need to generate the whole Pareto front. In addition, as a future study, the interactive approach we develop for single item auctions can be improved for multi-item cases.

Our study shows that decision support tools have important potential benefits for all parties participating in auctions. We believe that this is an important area for future studies, especially for more complex auction environments. Also application of the approach in a web-based platform can be stated as another future work.

REFERENCES

Bapna, R., Wolfgang, J. and Shmueli, G., 2008, Price formation and its dynamics in online auctions, *Decision Support Systems*, 44 (3), 641-656.

Baykal, Ş., “Combinatorial Auction Problems,” Master’s Thesis, Department of Industrial Engineering, Middle East Technical University, 2007.

Bellosta, M.J., Brigui, I., Kornman, S. And Vanderpooten, D., 2004, A Multi-criteria Model for Electronic Auctions, *In Proceedings of ACM Symposium on Applied Computing*, 759-765, March 14-17, Nicosia, Cyprus.

Bichler, M., Kalagnanam, J., 2005, Configurable offers and winner determination in multi-attribute auctions, *European Journal of Operational Research*, 160(2), 380-394.

Butler, J.C., Dyer, J.S. and Jia, J., 2006, Using Attributes to Predict Objectives in Preference Models, *Decision Analysis*, 3(2), 100-116.

Butler, J.C., Dyer, J.S., Jia, J. and Tomak, K., 2008, Enabling e-transactions with multi-attribute preference models, *European Journal of Operational Research*, 186(2), 748-765.

Catalán, J., Epstein, R., Guajardo, M., Yung, D. and Martınez, C.,2009, Solving multiple scenarios in a combinatorial auction, *Computers & Operations Research*, 36(10), 2752-2758.

De Vries, S. and Vohra, R., 2003, Combinatorial Auctions : A Survey, *INFORMS Journal on Computing*, 15(3), 284-309.

Deb, K., 2001, Multi-Objective Optimization using Evolutionary Algorithms. Wiley, Chichester, UK.

Deb, K., Pratap, A., Agrawal, S., and Meyarivan, T., 2002, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.

Fonseca, C.M., Fleming, P.J., 1995, An Overview of Evolutionary Algorithms in Multiobjective Optimization, *Evolutionary Computation*, 3(1), 1–16.

Gaytán, J., Arroyo, P., Minott, S. and de Boer, L., 2005, An Interactive Method for Improving the Applicability of Electronic Reverse Auctions, *Proceedings of the 14th IPSERA Conference.*

Haimes, Y., Lasdon, L. and Wismer, D., 1971, On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization, *IEEE Transactions on Systems, Man, and Cybernetics 1*, 296–297.

Hohner, G., Rich, J., Ng, E., Reid, G., Davenport, A.J., Kalagnanam, J.R., Lee, H.S. and An, C., 2003, Combinatorial and Quantity-Discount Procurement Auctions Benefit Mars, Incorporated and Its Suppliers, *Interfaces*, 33(1), 23-35.

Karahan, İ., “Preference-Based Flexible Multiobjective Evolutionary Algorithms,” Master’s Thesis, Department of Industrial Engineering, Middle East Technical University, 2008.

Koppius, O.R., Van Heck, E. and Wolters, M.J.J., 2004, The importance of product representation online: empirical results and implications for electronic markets, *Decision Support Systems*, 38 (2), 161-169.

Korhonen, P.J. and Laakso, J., 1986, A visual interactive method for solving the multiple criteria problem, *European Journal of Operational Research*, 24(2), pp. 277-287.

Köksalan, M., Leskelä, R.L., Wallenius, H. and Wallenius, J., Improving Efficiency in Multiple-Unit Combinatorial Auctions: Bundling Bids from Multiple Bidders, *Decision Support Systems* (forthcoming).

Leskelä, R.L., "Decision Support for Multiple-Unit Combinatorial Auctions," Ph.D. Thesis, Department of Industrial Engineering and Management, Helsinki University of Technology, 2007.

Leskelä, R.L., Teich, J.E., Wallenius, H., Wallenius, J., 2007, Decision support for multi-unit combinatorial bundle auctions, *Decision Support Systems*, 43(2), 420-434.

McAfee, R. and McMillan, J., 1987, Auctions and Bidding, *Journal of Economic Literature*, 25(2), 699-738.

Metty, T., Harlan, R., Samelson, Q., Moore, T., Morris, T., Sorensen, R., Schneur, A., Raskina, O., Schneur, R., Kanner, J., Potts, K., Robbins J., 2005, Reinventing the Supplier Negotiation Process at Motorola, *Interfaces*, 35(1), 7-23.

Pinker, E., Seidman, Y. and Vakrat, Y., 2003. Managing online auctions: Current business and research issues, *Management Science*, 49(11), 1457-1484.

Rothkopf, M.H. and Park, S., 2001, An Elementary Introduction to Auctions, *Interfaces*, 31 (6), 83-97.

Sandholm, T., Levine, D., Concordia, M., Martyn, P., Hughes, R., Jacobs, J. and Begg, D., 2006, Changing the Game in Strategic Sourcing at Procter&Gamble: Expressive Competition Enabled by Optimization, *Interfaces*, 36(1), 55-68.

Sheffi, Y., 2004, Combinatorial Auctions in the Procurement of Transportation Services, *Interfaces*, 34(4), 245-252.

Steuer, R.E., 1986, Multiple Criteria Optimization: Theory, Computation, and Application, John Wiley, New York.

Talluri, S., Narasimhan, R. and Viswanathan, S., 2007, Information Technologies for Procurement Decisions: A decision Support System for Multi-attribute e-reverse Auctions, *International Journal of Product Research*, 45 (11), 2615-2628.

Teich, J.E., Wallenius, H., Wallenius, J., Koppius, O.R., 2004, Emerging multiple issue e-auctions, *European Journal of Operational Research*, 159, 1–16.

Teich, J.E., Wallenius, H., Wallenius, J., Zaitsev, A., 2006, A Multi-Attribute e-Auction Mechanism for Procurement: Theoretical Foundations, *European Journal of Operational Research*, 175, 90–100.

Van Veldhuizen, D. A. and Lamont, G. B., 2000, On measuring multiobjective evolutionary algorithm performance, *Proceedings of the 2000 Congress on Evolutionary Computation, IEEE*, 1, 204–211.

Zitzler, E. and Thiele, L., 1998,. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study, *Parallel Problem Solving from Nature, PPSN V*, 292–301.

APPENDIX A

PROOF OF THEOREM 1

Theorem 1: For two attributes, J=2, if the preference constraints are satisfied for α_1 then they will also be satisfied by any α for $\alpha > \alpha_1$ where the preference function is in the following form.

$$u(s_i) = \left(\sum_j \left(w_j * \left(a_{ij} - z_j^*\right)\right)^{\alpha} \right)^{1/\alpha}$$

where

$u(s_i)$: preference function value of the bid offered by seller i

w_j : weight of attribute j

z_j^* : ideal level of attribute j

α : parameter of the L_α metric

j=1, 2.

Proof:

Let $A(a_1, a_2)$, $B(b_1, b_2)$ and $C(c_1, c_2)$ be three bids and let u be a preference function where smaller values imply higher preference levels. Let

$$u(A) \geq u(B) \text{ and } u(C) \geq u(B) \qquad (*)$$

We consider the following figure, to visualize the possible regions where bid B may be located.

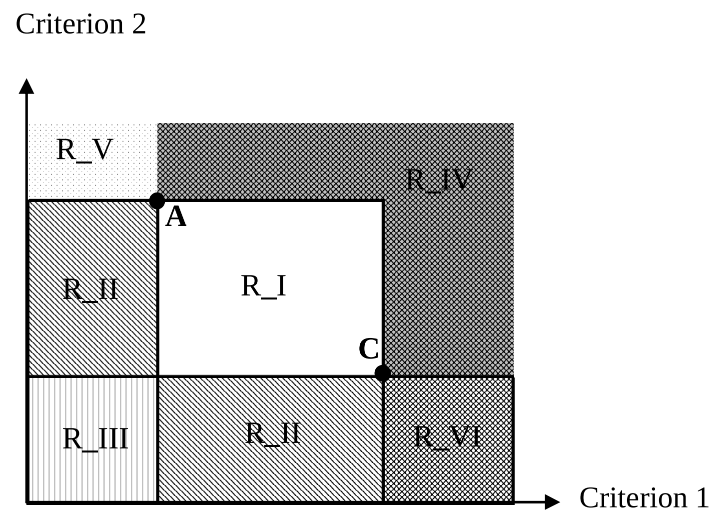


Figure A.1 The possible regions for bid

Note that if we can show that the solutions in regions R_I, R_V, R_VI satisfy (*), then the proof of the theorem is complete, because

- we can find solutions in R_II that dominates the solutions in R_I and if we can show that the solutions in region R_I satisfy (*), the proof of the theorem for the solutions in region R_II is trivial.

- the solutions in region R_III dominates A and C, any monotone preference function satisfies (*).

- the solutions in region R_IV are dominated, B cannot be in this region for such a preference relation.

Therefore, we prove the theorem by showing (*) holds for B in region R_I, R_V and R_VI.

For B in region R_I,

$c_1 \geq b_1 \geq a_1$ (1)

$a_2 \geq b_2 \geq c_2$ (2)

All ≥ 0

For $a = 1$

$$\left.\begin{aligned} w\ a_1 + (1-w)a_2 \ \geq\ w\ b_1 + (1-w)b_2 \\ w\ c_1 + (1-w)c_2 \ \geq\ w\ b_1 + (1-w)b_2 \end{aligned}\right\}$$

From preference relation

$$\Rightarrow \qquad \frac{b_1 - a_1}{a_2 - b_2} \leq \frac{1-w}{w} \leq \frac{c_1 - b_1}{b_2 - c_2}$$

Let $X = \dfrac{b_1 - a_1}{a_2 - b_2}$ and $Y = \dfrac{c_1 - b_1}{b_2 - c_2}$, where $X, Y \geq 0$. Then

$$\frac{1}{1+Y} \leq w \leq \frac{1}{1+X} \qquad \Rightarrow \qquad X \leq Y$$

$$\frac{b_1 - a_1}{a_2 - b_2} \leq \frac{c_1 - b_1}{b_2 - c_2} \qquad \Rightarrow \qquad \boxed{(b_1 - a_1)\ (b_2 - c_2) \leq (c_1 - b_1)\ (a_2 - b_2)} \qquad (3)$$

For $a = 2$

$$\left.\begin{aligned} w^2 a_1{}^2 + (1-w)^2 a_2{}^2 \ \geq\ w^2 b_1{}^2 + (1-w)^2 b_2{}^2 \\ w^2 c_1{}^2 + (1-w)^2 c_2{}^2 \ \geq\ w^2 b_1{}^2 + (1-w)^2 b_2{}^2 \end{aligned}\right\}$$

From preference relation

$$\Rightarrow \qquad \frac{b_1{}^2 - a_1{}^2}{a_2{}^2 - b_2{}^2} \leq \left(\frac{1-w}{w}\right)^2 \leq \frac{c_1{}^2 - b_1{}^2}{b_2{}^2 - c_2{}^2}$$

We check whether $\dfrac{b_1^2 - a_1^2}{a_2^2 - b_2^2} \le \dfrac{c_1^2 - b_1^2}{b_2^2 - c_2^2}$ holds.

$$(b_1^2 - a_1^2)\,(b_2^2 - c_2^2) \le (c_1^2 - b_1^2)\,(a_2^2 - b_2^2)$$

$$(b_1 - a_1)\,(b_1 + a_1)\,(b_2 - c_2)\,(b_2 + c_2) \le (c_1 - b_1)\,(c_1 + b_1)\,(a_2 - b_2)\,(a_2 + b_2)$$

From (3), we only need to check whether the following inequality holds.

$(b_1 + a_1)\,(b_2 + c_2) \le (c_1 + b_1)\,(a_2 + b_2)$ from (1) and (2), it holds.

For $a = n$

$$\left.\begin{aligned} w^n a_1^n + (1-w)^n a_2^n &\ge w^n b_1^n + (1-w)^n b_2^n \\ w^n c_1^n + (1-w)^n c_2^n &\ge w^n b_1^n + (1-w)^n b_2^n \end{aligned}\right\}$$

From preference relation

$$\Rightarrow \qquad \frac{b_1^n - a_1^n}{a_2^n - b_2^n} \le \left(\frac{1-w}{w}\right)^n \le \frac{c_1^n - b_1^n}{b_2^n - c_2^n}$$

We check whether $\dfrac{b_1^{n+1} - a_1^{n+1}}{a_2^{n+1} - b_2^{n+1}} \le \dfrac{c_1^{n+1} - b_1^{n+1}}{b_2^{n+1} - c_2^{n+1}}$ holds, if $\dfrac{b_1^n - a_1^n}{a_2^n - b_2^n} \le \dfrac{c_1^n - b_1^n}{b_2^n - c_2^n}$.

$$\frac{b_1^n - a_1^n}{a_2^n - b_2^n} \le \frac{c_1^n - b_1^n}{b_2^n - c_2^n} \quad \Rightarrow \quad (b_1^n - a_1^n)\,(b_2^n - c_2^n) \le (c_1^n - b_1^n)\,(a_2^n - b_2^n)$$

$$\Rightarrow (b_1 - a_1)\,(b_1^{n-1} + b_1^{n-2} a_1 + \cdots + a_1^{n-1})\,(b_2 - c_2)\,(b_2^{n-1} + b_2^{n-2} c_2 + \cdots + c_2^{n-1}) \le$$
$$(c_1 - b_1)\,(c_1^{n-1} + c_1^{n-2} b_1 + \cdots + b_1^{n-1})\,(a_2 - b_2)\,(a_2^{n-1} + a_2^{n-2} b_2 + \cdots + b_2^{n-1})$$

Multiply both sides with $b_1 b_2$.

$$(b_1-a_1)\,(b_1{}^{n}+b_1{}^{n-1}a_1+\cdots+b_1a_1{}^{n-1})\,(b_2-c_2)\,(b_2{}^{n}+b_2{}^{n-1}c_2+\cdots+b_2c_2{}^{n-1})\ \le$$
$$(c_1-b_1)\,(c_1{}^{n-1}b_1+c_1{}^{n-2}b_1{}^{2}+\cdots+b_1{}^{n})\,(a_2-b_2)\,(a_2{}^{n-1}b_2+a_2{}^{n-2}b_2{}^{2}+\cdots+b_2{}^{n})$$

(4)

Write the following inequality.

$$(b_1-a_1)\,(b_2-c_2)\,\Big((b_1{}^{n}+\cdots+b_1a_1{}^{n-1})c_2{}^{n}+(b_2{}^{n}+\cdots+b_2c_2{}^{n-1})a_1{}^{n}+c_2{}^{n}a_1{}^{n}\Big)\le$$
$$(c_1-b_1)\,(a_2-b_2)\,\Big((c_1{}^{n-1}b_1+\cdots+b_1{}^{n})a_2{}^{n}+(a_2{}^{n-1}b_2+\cdots+b_2)c_1{}^{n}+a_2{}^{n}c_1{}^{n}\Big)$$

(5)

Add (4) and (5);

$$(b_1{}^{n+1}-a_1{}^{n+1})\,(b_2{}^{n+1}-c_2{}^{n+1})\ \le\ (c_1{}^{n+1}-b_1{}^{n+1})\,(a_2{}^{n+1}-b_2{}^{n+1})\,.$$

For B in region R_V,

$c_1\ge a_1\ge b_1$ (6)

$b_2\ge a_2\ge c_2$ (7)

All ≥ 0

For $a=1$

$$\left.\begin{aligned} w\ a_1+(1-w)a_2\ &\ge\ w\ b_1+(1-w)b_2 \\ w\ c_1+(1-w)c_2\ &\ge\ w\ b_1+(1-w)b_2 \end{aligned}\right\}\quad \text{From preference relation}$$

$$\Rightarrow \qquad \frac{1-w}{w} \le \frac{a_1-b_1}{b_2-a_2} \quad \text{and} \quad \frac{1-w}{w} \le \frac{c_1-b_1}{b_2-c_2}$$

$$w \ge Max\left\{\frac{b_2-a_2}{(b_2-a_2)+(a_1-b_1)}, \frac{b_2-c_2}{(b_2-c_2)+(c_1-b_1)}\right\}$$ there exist such weights

For $a=n$

$$\left.\begin{array}{l} w^n a_1{}^n + (1-w)^n a_2{}^n \ge w^n b_1{}^n + (1-w)^n b_2{}^n \\ w^n c_1{}^n + (1-w)^n c_2{}^n \ge w^n b_1{}^n + (1-w)^n b_2{}^n \end{array}\right\}$$ From preference relation

$$\Rightarrow \qquad \frac{1-w}{w} \le \frac{a_1{}^n-b_1{}^n}{b_2{}^n-a_2{}^n} \quad \text{and} \quad \frac{1-w}{w} \le \frac{c_1{}^n-b_1{}^n}{b_2{}^n-c_2{}^n}$$

$$w \ge Max\left\{\frac{b_2{}^n-a_2{}^n}{\left(b_2{}^n-a_2{}^n\right)+\left(a_1{}^n-b_1{}^n\right)}, \frac{b_2{}^n-c_2{}^n}{\left(b_2{}^n-c_2{}^n\right)+\left(c_1{}^n-b_1{}^n\right)}\right\}$$

there exist such weights.

For B in region R_VI,

$b_1 \ge c_1 \ge a_1$ (8)

$a_2 \ge c_2 \ge b_2$ (9)

All ≥ 0

For $a = 1$

$$\left.\begin{array}{l} w\ a_1 + (1-w)a_2 \ge w\ b_1 + (1-w)b_2 \\ w\ c_1 + (1-w)c_2 \ge w\ b_1 + (1-w)b_2 \end{array}\right\}$$ From preference relation

$$\Rightarrow \qquad \frac{1-w}{w} \geq \frac{b_1 - a_1}{a_2 - b_2} \quad \text{and} \quad \frac{1-w}{w} \geq \frac{b_1 - c_1}{c_2 - b_2}$$

$$w \geq Min \left\{ \frac{a_2 - b_2}{(a_2 - b_2) + (b_1 - a_1)}, \frac{c_2 - b_2}{(c_2 - b_2) + (b_1 - c_1)} \right\} \qquad \text{there exist such weights}$$

For $a = n$

$$\left. \begin{aligned} w^n a_1{}^n + (1-w)^n a_2{}^n &\geq w^n b_1{}^n + (1-w)^n b_2{}^n \\ w^n c_1{}^n + (1-w)^n c_2{}^n &\geq w^n b_1{}^n + (1-w)^n b_2{}^n \end{aligned} \right\} \qquad \text{From preference relation}$$

$$\Rightarrow \qquad \frac{1-w}{w} \geq \frac{b_1{}^n - a_1{}^n}{a_2{}^n - b_2} \quad \text{and} \quad \frac{1-w}{w} \geq \frac{b_1{}^n - c_1{}^n}{c_2{}^n - b_2{}^n}$$

$$w \geq Min \left\{ \frac{a_2{}^n - b_2{}^n}{(a_2{}^n - b_2{}^n) + (b_1{}^n - a_1{}^n)}, \frac{c_2{}^n - b_2{}^n}{(c_2{}^n - b_2{}^n) + (b_1{}^n - c_1{}^n)} \right\}$$

there exist such weights

#.

APPENDIX B

RESULTS OF THE INTERACTIVE APPROACH

2_attributes & alfa=1

w(price)=0.8, w(defect)=0.2

The algorithm

Bidder	price	defect	DM_u[1]
1	8.1483	1.8562	4.7170
2	7.5490	2.3543	4.4970
3	6.9483	2.8561	4.2770
4	6.3443	3.3669	4.0570
5	5.7490	3.8543	3.8370
6	5.1445	4.3663	3.6170
7	**4.5452**	**4.8645**	**3.3970**

EX_PAR with exact parameters

Bidder	price	defect	DM_u[1]
1	8.1482	1.8563	4.7170
2	7.5490	2.3544	4.4970
3	6.9483	2.8562	4.2770
4	6.3435	3.3689	4.0570
5	5.7490	3.8543	3.8370
6	5.1444	4.3664	3.6170
7	**4.5451**	**4.8646**	**3.3970**

2_attributes & alfa=2

w(price)=0.5, w(defect)=0.5

The algorithm

Bidder	price	defect	DM_u[1]
1	8.2109	1.7088	2.8672
2	7.6873	2.0693	2.7634
3	7.1717	2.4487	2.6859
4	6.6635	2.8448	2.6376
5	**6.1635**	**3.2537**	**2.6206**
6	5.6744	3.6715	2.6358
7	5.1974	4.0969	2.6828

EX_PAR with exact parameters

Bidder	price	defect	DM_u[1]
1	8.2056	1.7200	2.8672
2	7.6805	2.0803	2.7634
3	7.1636	2.4592	2.6858
4	6.6538	2.8547	2.6375
5	**6.1522**	**3.2631**	**2.6206**
6	5.6615	3.6802	2.6358
7	5.1827	4.1050	2.6828

[1] DM_u : preference function value of the DM

2_attributes & alfa=3

w(price)=0.8, w(defect)=0.2

The algorithm

Bidder	price	defect	DM_u[1]
1	7.8673	4.2224	4.2073
2	7.2747	4.5083	3.8960
3	6.6825	4.8123	3.5873
4	6.0915	5.1204	3.2824
5	5.5017	5.4343	2.9831
6	4.9137	5.7492	2.6924
7	**4.3281**	**6.0604**	**2.4147**

EX_PAR with exact parameters

Bidder	price	defect	DM_u[1]
1	7.8501	4.8953	4.2045
2	7.2582	5.0391	3.8937
3	6.6679	5.2044	3.5857
4	6.0801	5.3710	3.2815
5	5.4956	5.5431	2.9829
6	4.9157	5.7206	2.6924
7	**4.3420**	**5.9074**	**2.2680**

w(price)=0.5, w(defect)=0.5

The algorithm

Bidder	price	defect	DM_u[1]
1	8.0588	2.1533	2.7427
2	7.5297	2.4079	2.5990
3	7.0195	2.6912	2.4797
4	6.5338	2.9994	2.3930
5	**6.0765**	**3.3319**	**2.3473**
6	**5.6519**	**3.6869**	**2.3484**
7	5.2659	4.0610	2.3975

EX_PAR with exact parameters

Bidder	price	defect	DM_u[1]
1	8.0587	2.1534	2.7427
2	7.5296	2.4080	2.5990
3	7.0195	2.6913	2.4797
4	6.5337	2.9995	2.3930
5	**6.0764**	**3.3320**	**2.3473**
6	**5.6517**	**3.6870**	**2.3484**
7	5.2658	4.0611	2.3975

w(price)=0.3, w(defect)=0.7

The algorithm

Bidder	price	defect	DM_u[1]
1	**8.6252**	**1.2059**	**1.7899**
2	8.2227	1.5832	1.7976
3	7.8286	1.9883	1.8695
4	7.4448	2.4113	2.0091
5	7.0707	2.8475	2.2068
6	6.7067	3.2933	2.4477
7	6.3549	3.7462	2.7183

EX_PAR with exact parameters

Bidder	price	defect	DM_u[1]
1	**8.4870**	**1.3217**	**1.7845**
2	8.1532	1.6220	1.7967
3	7.8872	1.9655	1.8691
4	7.6848	2.3439	2.0040
5	7.5381	2.7492	2.1935
6	7.4369	3.1746	2.4252
7	7.3722	3.6150	2.6875

[1] DM_u : preference function value of the DM

2_attributes & alfa=4

w(price)=0.8, w(defect)=0.2

The algorithm

Bidder	price	defect	DM_u[1]
1	7.8390	5.5448	4.1860
2	7.2450	5.6654	3.8711
3	6.6520	5.8022	3.5579
4	6.0607	5.9465	3.2471
5	5.4717	6.0899	2.9401
6	4.8861	6.2323	2.6394
7	**4.3051**	**6.3777**	**2.3489**

EX_PAR with exact parameters

Bidder	price	defect	DM_u[1]
1	7.8309	6.2304	4.1847
2	7.2364	6.2434	3.8699
3	6.6434	6.2592	3.5569
4	6.0526	6.2773	3.2463
5	5.4652	6.2896	2.9397
6	4.8822	6.3202	2.6393
7	**4.3068**	**6.3513**	**2.3489**

w(price)=0.6, w(defect)=0.4

The algorithm

Bidder	price	defect	DM_u[1]
1	7.9132	3.2559	3.1877
2	7.3432	3.3942	2.9703
3	6.7840	3.5537	2.7633
4	6.2399	3.7362	2.5726
5	5.7180	3.9426	2.4067
6	5.2270	4.1764	2.2773
7	**4.7779**	**4.4410**	**2.1970**

EX_PAR with exact parameters

Bidder	price	defect	DM_u[1]
1	7.9132	3.2554	3.1877
2	7.3433	3.3937	2.9703
3	6.7841	3.5532	2.7633
4	6.2400	3.7358	2.5726
5	5.7181	3.9422	2.4067
6	5.2272	4.1760	2.2773
7	**4.7781**	**4.4407**	**2.1970**

w(price)=0.5, w(defect)=0.5

The algorithm

Bidder	price	defect	DM_u[1]
1	8.0469	2.2045	2.7012
2	7.5134	2.4568	2.5399
3	6.9977	2.7371	2.3988
4	6.5048	3.0417	2.2890
5	6.0391	3.3703	2.2245
6	**5.6054**	**3.7206**	**2.2172**
7	5.2081	4.0910	2.2708

EX_PAR with exact parameters

Bidder	price	defect	DM_u[1]
1	7.9925	2.4967	2.6957
2	7.4539	2.6736	2.5351
3	6.9399	2.8790	2.3955
4	6.4595	3.1161	2.2876
5	6.0240	3.3867	2.2243
6	**5.6445**	**3.6920**	**2.2167**
7	5.3292	4.0309	2.2680

[1] DM_u : preference function value of the DM

w(price)=0.4, w(defect)=0.6

The algorithm

Bidder	price	defect	DM_u[1]
1	8.1813	1.7738	2.2119
2	7.7011	2.0475	2.1164
3	7.2541	2.3545	2.0593
4	**6.8449**	**2.6918**	**2.0570**
5	6.4772	3.0555	2.1199
6	6.1522	3.4420	2.2457
7	5.8701	3.8477	2.4220

EX_PAR with exact parameters

Bidder	price	defect	DM_u[1]
1	8.1251	1.9211	2.2088
2	7.6514	2.1304	2.1148
3	7.2305	2.3796	2.0591
4	**6.8770**	**2.6697**	**2.0567**
5	6.6014	2.9994	2.1172
6	6.4058	3.3645	2.2388
7	6.2832	3.7590	2.4104

w(price)=0.2, w(defect)=0.8

The algorithm

Bidder	price	defect	DM_u[1]
1	**9.0007**	**0.9997**	**1.2553**
2	8.7247	1.3871	1.3527
3	8.4552	1.8042	1.5622
4	8.1904	2.2402	1.8509
5	7.9282	2.6889	2.1818
6	7.6701	3.1466	2.5343
7	7.4170	3.6107	2.8985

EX_PAR with exact parameters

Bidder	price	defect	DM_u[1]
1	**8.8982**	**1.0453**	**1.2537**
2	8.9509	1.3279	1.3480
3	9.2267	1.6759	1.5329
4	9.6627	2.0724	1.7919
5	10.0000	2.5108	2.0997
6	10.0000	2.9804	2.4406
7	10.0000	3.4549	2.8006

3_attributes & alfa=1

w(price)=0.3, w(defect)=0.4, w(lead time)=0.3

The algorithm

Bidder	price	defect	lead time	DM_u[1]
1	**6.7074**	**1.0700**	**3.5163**	**3.4951**
2	6.3079	1.5682	3.5205	3.5758
3	5.9079	2.0715	3.5112	3.6543
4	5.5156	2.5661	3.5101	3.7341
5	5.1118	3.0683	3.5119	3.8144
6	4.7142	3.5669	3.5105	3.8942
7	4.3137	4.0674	3.5104	3.9742

EX_PAR with exact parameters

Bidder	price	defect	lead time	DM_u[1]
1	**10.5295**	**1.0626**	**3.0000**	**3.4309**
2	9.9293	1.5626	3.0000	3.5109
3	9.3297	2.0625	3.0000	3.5909
4	8.7298	2.5624	3.0000	3.6709
5	8.1307	3.0620	3.0000	3.7509
6	7.5314	3.5616	3.0000	3.8309
7	6.9288	4.0629	3.0000	3.9109

[1] DM_u : preference function value of the DM

3_attributes & alfa=2

w(price)=0.2, w(defect)=0.1, w(lead time)=0.7

The algorithm

Bidder	price	defect	lead time	DM_u[1]
1	6.4351	2.5403	3.0000	2.4761
2	6.0510	2.8913	3.0000	2.4409
3	5.6691	3.2503	3.0000	2.4086
4	5.2889	3.6211	3.0000	2.3791
5	4.9108	4.0015	3.0000	2.3526
6	4.5350	4.3901	3.0000	2.3292
7	**4.1624**	**4.7842**	**3.0000**	**2.3091**

EX_PAR with exact parameters

Bidder	price	defect	lead time	DM_u[1]
1	9.6545	2.5274	3.0000	2.4761
2	9.0786	2.8791	3.0000	2.4409
3	8.5061	3.2385	3.0000	2.4086
4	7.9363	3.6096	3.0000	2.3791
5	7.3695	3.9907	3.0000	2.3526
6	6.8064	4.3795	3.0000	2.3292
7	**6.2479**	**4.7741**	**3.0000**	**2.3091**

3_attributes & alfa=3

w(price)=0.7, w(defect)=0.2, w(lead time)=0.1

The algorithm

Bidder	price	defect	lead time	DM_u[1]
1	5.3388	4.5400	10.0000	3.7784
2	4.9451	4.7094	10.0000	3.5119
3	4.5530	4.8846	10.0000	3.2493
4	4.1628	5.0694	10.0000	2.9920
5	3.7751	5.2627	10.0000	2.7422
6	3.3911	5.4628	10.0000	2.5029
7	**3.0123**	**5.6693**	**10.0000**	**2.2787**

EX_PAR with exact parameters

Bidder	price	defect	lead time	DM_u[1]
1	8.0079	4.5499	10.0000	3.7784
2	7.4174	4.7184	10.0000	3.5119
3	6.8290	4.8976	10.0000	3.2493
4	6.2435	5.0829	10.0000	2.9920
5	5.6641	5.2752	9.9253	2.7422
6	5.0988	5.4806	9.5709	2.5022
7	**4.5465**	**5.6932**	**9.1020**	**2.2754**

w(price)=0.2, w(defect)=0.1, w(lead time)=0.7

The algorithm

Bidder	price	defect	lead time	DM_u[1]
1	6.5154	1.9790	2.2560	2.2560
2	6.1472	2.3406	2.2329	2.2329
3	5.7836	2.7134	2.2124	2.2124
4	5.4228	3.1021	2.1944	2.1944
5	5.0648	3.5040	2.1788	2.1788
6	4.7106	3.9151	2.1657	2.1657
7	**4.3612**	**4.3332**	**2.1549**	**2.1549**

EX_PAR with exact parameters

Bidder	price	defect	lead time	DM_u[1]
1	9.5650	3.4941	3.0000	2.2492
2	8.9827	3.7157	3.0000	2.2260
3	8.4043	3.9523	3.0000	2.2056
4	7.8309	4.2031	3.0000	2.1879
5	7.2638	4.4670	3.0000	2.1728
6	6.7045	4.7462	3.0000	2.1603
7	**6.1552**	**5.0392**	**3.0000**	**2.1502**

[1] DM_u : preference function value of the DM

w(price)=0.3, w(defect)=0.4, w(lead time)=0.3

The algorithm

Bidder	price	defect	lead time	DM_u[1]
1	6.1577	1.8931	3.6898	2.0079
2	5.8528	2.1715	3.6121	1.9435
3	5.5621	2.4781	3.5444	1.8966
4	**5.2937**	**2.8117**	**3.4736**	**1.8719**
5	**5.0426**	**3.1745**	**3.4047**	**1.8734**
6	4.8134	3.5549	3.3436	1.9034
7	4.6049	3.9554	3.2817	1.9619

EX_PAR with exact parameters

Bidder	price	defect	lead time	DM_u[1]
1	9.1882	1.9123	3.7616	2.0075
2	8.7276	2.1849	3.6844	1.9431
3	8.2908	2.4918	3.6090	1.8963
4	**7.8790**	**2.8283**	**3.5372**	**1.8716**
5	**7.5007**	**3.1852**	**3.4690**	**1.8731**
6	7.1533	3.5652	3.4024	1.9030
7	6.8398	3.9625	3.3401	1.9616

3_attributes & alfa=4

w(price)=0.7, w(defect)=0.2, w(lead time)=0.1

The algorithm

Bidder	price	defect	lead time	DM_u[1]
1	5.3301	5.1552	10.0000	3.7412
2	4.9346	5.3061	10.0000	3.4679
3	4.5402	5.4605	10.0000	3.1968
4	4.1472	5.6163	10.0000	2.9292
5	3.7560	5.7804	10.0000	2.6667
6	3.3672	5.9506	10.0000	2.4124
7	**2.9821**	**6.1212**	**10.0000**	**2.1715**

EX_PAR with exact parameters

Bidder	price	defect	lead time	DM_u[1]
1	7.9862	5.7549	10.0000	3.7401
2	7.3929	5.7884	10.0000	3.4669
3	6.8014	5.8309	10.0000	3.1960
4	6.2126	5.8782	10.0000	2.9286
5	5.6275	5.9348	10.0000	2.6664
6	5.0484	5.9921	10.0000	2.4124
7	**4.4782**	**6.0589**	**10.0000**	**2.1714**

w(price)=0.2, w(defect)=0.1, w(lead time)=0.7

The algorithm

Bidder	price	defect	lead time	DM_u[1]
1	6.3547	4.2809	3.0000	2.1680
2	5.9615	4.4822	3.0000	2.1536
3	5.5697	4.6956	3.0000	2.1416
4	5.1797	4.9151	3.0000	2.1319
5	4.7921	5.1429	3.0000	2.1242
6	4.4073	5.3833	3.0000	2.1183
7	**4.0265**	**5.6330**	**3.0000**	**2.1139**

EX_PAR with exact parameters

Bidder	price	defect	lead time	DM_u[1]
1	9.5308	4.3265	3.0000	2.1680
2	8.9436	4.4482	3.0000	2.1536
3	8.3601	4.5844	3.0000	2.1416
4	7.7816	4.7318	3.0000	2.1319
5	7.2099	4.8949	3.0000	2.1242
6	6.6477	5.0746	3.0000	2.1182
7	**6.0989**	**5.2732**	**3.0000**	**2.1138**

[1] DM_u : preference function value of the DM

APPENDIX C

PERFORMANCE MEASURES OF THE EVOLUTIONARY ALGORITHM FOR ALL PROBLEMS

Table C.1 Performance Measures for Original Case Problem 10x20

Problem 10x20	HI*	IGDM
pr1_without seeding	0.9902	0.0110
pr1_seeding by sorting	0.9993	0.0015
pr2_without seeding	0.9971	0.0076
pr2_ seeding by sorting	0.9981	0.0026
pr3_without seeding	0.9943	0.0170
pr3_ seeding by sorting	0.9941	0.0082
pr4_without seeding	1.0000	0.0000
pr4_ seeding by sorting	0.9981	0.0040
pr5_without seeding	0.9693	0.0178
pr5_ seeding by sorting	1.0000	0.0000
Average_without seeding	**0.9902**	**0.0107**
Average_ seeding by sorting	**0.9979**	**0.0034**

Table C.2 Performance Measures for Original Case Problem 30x30

Problem 30x30	HI*	IGDM
pr1_without seeding	0.9979	0.0051
pr1_ seeding by sorting	0.9992	0.0012
pr2_without seeding	0.9961	0.0029
pr2_ seeding by sorting	0.9991	0.0011
pr3_without seeding	0.9804	0.0162
pr3_ seeding by sorting	0.9863	0.0128
pr4_without seeding	0.9972	0.0054
pr4_ seeding by sorting	0.9977	0.0029
pr5_without seeding	0.9889	0.0117
pr5_ seeding by sorting	0.9889	0.0104
Average_without seeding	**0.9921**	**0.0082**
Average_ seeding by sorting	**0.9942**	**0.0057**

Table C.3 Performance Measures for Original Case Problem 30x100

Problem 30x100	HI*	IGDM
pr1_without seeding	0.9958	0.0067
pr1_ seeding by sorting	0.9969	0.0059
pr2_without seeding	0.9996	0.0031
pr2_ seeding by sorting	0.9978	0.0055
pr3_without seeding	0.9973	0.0073
pr3_ seeding by sorting	0.9971	0.0052
pr4_without seeding	0.9977	0.0102
pr4_ seeding by sorting	0.9985	0.0029
pr5_without seeding	0.9978	0.0039
pr5_ seeding by sorting	0.9983	0.0053
Average_without seeding	**0.9976**	**0.0062**
Average_ seeding by sorting	**0.9977**	**0.0050**

Table C.4 Performance Measures for Discounted Case Problem 10x20

Problem 10x20	HI*	IGDM
pr1_without seeding	0.9866	0.0092
pr1_optimal seeding	0.9989	0.0014
pr1_rank heuristic	0.9990	0.0013
pr2_without seeding	0.9823	0.0664
pr2_optimal seeding	0.9995	0.0010
pr2_rank heuristic	0.9823	0.0664
pr3_without seeding	0.9833	0.0205
pr3_optimal seeding	0.9870	0.0085
pr3_rank heuristic	0.9827	0.0205
pr4_without seeding	0.9960	0.0157
pr4_optimal seeding	0.9993	0.0014
pr4_rank heuristic	0.9691	0.0151
pr5_without seeding	0.8656	0.0845
pr5_optimal seeding	0.9990	0.0020
pr5_rank heuristic	0.9121	0.0500
Average_without seeding	**0.9628**	**0.0393**
Average_optimal seeding	**0.9967**	**0.0029**
Average_rank heuristic	**0.9690**	**0.0307**

Table C.5 Performance Measures for Discounted Case Problem 30x30

Problem 30x30	HI*	IGDM
pr1_without seeding	0.7385	0.1636
pr1_optimal seeding	0.9874	0.0093
pr1_rank heuristic	0.9529	0.0391
pr2_without seeding	0.8175	0.1194
pr2_optimal seeding	0.9461	0.0370
pr2_rank heuristic	0.9528	0.0376
pr3_without seeding	0.8577	0.0846
pr3_optimal seeding	0.9805	0.0124
pr3_rank heuristic	0.9700	0.0351
pr4_without seeding	0.8341	0.1030
pr4_optimal seeding	0.9547	0.0317
pr4_rank heuristic	0.9003	0.0740
pr5_without seeding	0.7874	0.1385
pr5_optimal seeding	0.9794	0.0121
pr5_rank heuristic	0.9112	0.0564
Average_without seeding	**0.8071**	**0.1218**
Average_optimal seeding	**0.9696**	**0.0205**
Average_rank heuristic	**0.9374**	**0.0484**

Table C.6 Performance Measures for Discounted Case Problem 30x100

Problem 30x100	HI*	IGDM
pr1_without seeding	0.8692	0.0711
pr1_optimal seeding	0.9720	0.0232
pr1_rank heuristic	0.8777	0.0681
pr2_without seeding	0.7789	0.1444
pr2_optimal seeding	0.9780	0.0156
pr2_rank heuristic	0.7666	0.1645
pr3_without seeding	0.9021	0.0588
pr3_optimal seeding	0.9732	0.0185
pr3_rank heuristic	0.9541	0.0425
pr4_without seeding	0.8982	0.0876
pr4_optimal seeding	0.9861	0.0110
pr4_rank heuristic	0.9494	0.0611
pr5_without seeding	0.8891	0.0500
pr5_optimal seeding	0.9759	0.0182
pr5_rank heuristic	0.8729	0.0646
Average_without seeding	**0.8675**	**0.0824**
Average_optimal seeding	**0.9770**	**0.0173**
Average_rank heuristic	**0.8842**	**0.0801**