CLIENT-CONTRACTOR BARGAINING PROBLEM IN THE CONTEXT OF MULTI-MODE PROJECT SCHEDULING WITH LIMITED RESOURCES

by

NURSEL KAVLAK

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University
Fall 2005

CLIENT-CONTRACTOR BARGAINING PROBLEM IN THE CONTEXT OF A MULTI-MODE PROJECT SCHEDULING PROBLEM WITH LIMITED RESOURCES

APPROVED BY:

Prof. Dr. Gündüz Ulusoy …………………………….

(Dissertation Supervisor)

Assoc. Assistant Prof. Dr. Ş. İlker Birbil …………………………….

Assoc. Prof. Dr. Funda Sivrikaya Şerifoğlu …………………………….

DATE OF APPROVAL: …………………………….

ABSTRACT

This study focuses on the client-contractor bargaining problem in the context of multi-mode resource constrained project scheduling. The bargaining objective is to maximize the bargaining objective function comprised of the individual NPV maximizing objectives of both the client and the contractor.

Although the well-known multi- mode resource constrained project scheduling problem has been under investigation from various dimensions in the literature, this thesis proposes a two-player setting to this problem. The solution procedure takes the objectives of both players into account. One other proposal we have in this thesis is the bargaining weights concept we have used in the model, which is used to determine the bargaining power of each player through the negotiation process. The effect of bargaining weights assigned to each player on the solution has also been analyzed.

Different payment models have also been investigated in this thesis. We have used progress payments, payments at equal time intervals, and payments at activity completions in our tests.

Simulated Annealing Algorithm and Genetic Algorithm are proposed as solution procedures. Also the solution set of the problem is investigated by further analyzing the non-dominated solutions. We have conducted sensitivity analysis among different parameters we have used in the model. These parameters are profit margin, interest rate, and bargaining weights.

The bargaining objective function we have used has been an important part of the model itself. We have investigated different solution approaches by using two different bargaining objective function formulations in our tests.

ÖZET

Bu çalışma, çok modlu kaynak kısıtlı proje planlama çerçevesinde müşteri-müteahhit pazarlık problemine odaklanmaktadır. Pazarlığın amacı, hem müşteri hem de müteahhit'in bugünkü net değeri ençoklamayı hedefleyen amaç işlevlerinden oluşan pazarlık işlevini ençoklamaktır.

Bu problem yazında birçok açıdan incelenmiş olsa da, bu problem iyi tanınan modlu kaynak kısıtlı proje planlama problemine çok amaçlı bir bakış açısı önerir. Çözüm her iki oyuncunun da amacını dikkate alır. Bu tezdeki bir başka önerimiz de, modelde kullandığımız pazarlık ölçütü kavramıdır. Pazarlık ölçütleri, herbir oyuncunun pazarlık sürecindeki pazarlık gücünü belirlemek amacıyla kullanılırlar. Herbir oyuncuya atanan pazarlık ölçütlerinin çözüm üzerindeki etkisi de incelenmiştir.

Bu çalışmada, farklı ödeme modelleri de incelenmiştir. Testlerde kullandığımız ödeme modelleri şunlardır: ilerleme odaklı ödeme modeli, eşit zaman aralıklarında ödeme modeli, ve faaliyet bitimlerinde ödeme modeli.

Tavlama Benzetimi Algoritması, ve Genetik Algoritma çözüm yolları olarak önerilmiştir. Ayrıca, problemin çözüm kümesi baskın çözümler de tahlil edilerek incelenmiştir. Önerilen çözüm metodları kullanılarak birçok test uygulanmıştır. Modelde kullandığımız farklı değiştirgelerde hassasiyet taraması yapılmıştır. Bu değiştirgeler; kâr marjı, faiz oranı, ve pazarlık ölçütleridir.

Kullandığımız amaç işlevinin kendisi modelin önemli bir parçasıdır. Testlerimizde iki farklı amaç işlevi kullanarak, modelde farklı çözüm yaklaşımlarını inceledik.

ACKNOWLEDGEMENTS

I owe an enormous dept of gratitude to my thesis advisors, Gündüz Ulusoy, Ş. İlker Birbil, and Funda Sivrikaya Şerifoğlu, for all their advice and guidance. They have provided the motivation and stimulation that have prompted me to put all my effort to this work. I would not conclude this work without their support.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

1 PROBLEM FORMULATION

In this chapter we investigate the model by a literature survey followed by problem formulation and definition. Although, the literature provides a significant amount of study on relevant areas, the problem we investigate in this thesis differs by its objective definition. This chapter goes through the details of the introduced model.

1.1 Introduction

In this thesis, we consider the client-contractor bargaining problem in the context of multi-mode resource constrained project scheduling. The bargaining objective is to maximize the bargaining objective function comprised of the objectives of both the client and the contractor. The objective of the client is to minimize the net present value (NPV) of the payments to the contractor, whereas the objective of the contractor is to maximize the net return. The basic difficulty with this problem is that the individual objectives of both the client and the contractor are in conflict most of the times, and the bargaining objective should consider the incentives of both parties.

1.2 Literature Survey

In the literature, a number of exact and heuristic methods have been proposed for solving the single objective resource constrained project scheduling problem with discounted cash flows (see, e.g., Herroelen *et. al.* (1997), Kimms (2001a)). Russell (1970) introduced an initial version of the discounted cash flow problem in project scheduling with no resource constraints. Grinold (1972) extended the model of Russell by introducing a project deadline. The net present value criterion and its impact on project scheduling had been investigated by Bey *et al.* (1981). Baroum and Patterson (1996) had reviewed the development of cash flow weight procedures for the problem. Exact solution procedures for the resource constrained version of the problem are given among others by Doersch and Patterson (1977), Yang *et. al.* (1993), İçmeli and Erengüç (1996), Baroum and Patterson (1999), and Demeulemeester *et. al.* (2000). A relatively recent review on project scheduling in general is provided by Kolisch and Padman (2001).

However, exact methods become computationally impractical for problems of a realistic size, since the model grows too large quickly and hence, the solution procedures become intractable. This led to studies on a variety of heuristic procedures among others by Russell (1986), Smith-Daniels and Aquilano (1987), Padman and Smith-Daniels (1993), Padman *et.al.* (1997), and Kimms (2001b). Etgar *et. al.* (1996) present a simulated annealing algorithm solution approach in order to solve the problem of scheduling activities in a project to maximize its NPV for the case where the net cash flow magnitudes are independent of the time of realization. Ulusoy *et al.* (2001) investigated the multi-mode resource constrained project scheduling problem with discounted cash flows using genetic algorithm. They allowed both positive and negative cash flows. In their paper they distinguished among four types of payment scheduling models:

- *Lump-sum payment.* Here, the client pays the total payment to the contractor upon successful completion of the project.
- *Payment at event occurrences.* Payments are made at a set of event nodes. The problem is to determine the amount, location, and timing of these payments. Dayanand and Padman (1993, 1997) attacked the problem of simultaneously determining the amount, location, and timing of the payments by the client so as to maximize the contractor's NPV. They have dealt with this problem further from the perspective of the client (Dayanand and Padman, 1998). Later, Dayanand and Padman (1999) investigated the problem in the context of client and contractor negotiation stressing the need for a joint view. Ulusoy and Cebelli (2000) extended this payment model so as to include both the client and the contractor in a joint model. They introduce the concept of ideal solution, where the ideal solution for the contractor would be to receive the whole payment at the start of the project and for the client it would be a single payment at the completion of the project. They search for a solution where the client and the contractor deviate from their respective ideal solutions by an equal percentage. They call such a solution an equitable solution.
- *Progress payment.* The contractor receives payments at regular time intervals until the project is completed. The amount of payment is based on the amount of work accomplished since the last payment. Kazaz and Sepil (1996) presented a mixed-integer formulation of the progress payment with the objective of maximizing the NPV of the cash flows for the contractor. Sepil and Ortaç (1997) tested the performance of some heuristic procedures for resource-constrained projects with

progress payments. They defined cash inflows occurring periodically as progress payments, and cash outflows as costs incurred whenever an activity is completed.

- *Payment at equal time intervals.* In this payment model, payments are made at predetermined equal time intervals over the duration of the project, and the final payment is scheduled on project completion. The amounts of the payments are either predetermined and fixed or are based on the amount of work accomplished since the last payment. Note that the number of payments in this payment model is known and fixed in advance, whereas in progress payment model this is not the case.

Mika *et al.* (2004) considered the multi-mode resource constrained project scheduling problem with discounted cash flows in the context of the above payment scheduling models using positive cash flows. As solution methods they employed Simulated Annealing and Genetic Algorithm.

In the literature, there have been some examples of weight concept in bargaining problems, although none of them had the same bargaining weight definition we have used in this thesis. Köbberling and Peters (2002) have investigated the effect of decision weights in bargaining problems through the concept of probability weighting functions. In their approach, the solution to the bargaining problem depends exclusively on its image in utility space. Ervig and Haake (2005) view bargaining power as ordinary goods that can be traded in exchange economy involving two countries. The final solution they define satisfies two main properties. First, it should be Pareto optimal in the aggregate, i.e. there is no other package of subsidies and expenditures that makes both countries better off. This is the same property adopted in this study as well; that is our ultimate aim is finding the solution which ensures that there is no other point in the utility space that brings players to a better position at the aggregate level. The second property states that, if one compares the final solution with the scenario, in which both issues are treated separately, then neither of the players should be worse off in the final solution. So the favor exchange really should do a favor to both. In our study, the bargain between players doesn't constitute a favor exchange, but instead a pure trade-off among benefit obtained. Marmol *et al.* (2005) propose a solution concept for multi-criteria bargaining games, which is based on the distance to a utopian minimum level vector. The distance concept they introduce in their study is similar to the distance definition we have used in this study, in a way that both identify the distance from the minimum level point for both players.

1.3 Problem Definition

The problem under investigation here is formulated as a progress payments model. Contractor receives payments at predetermined regular time intervals. The last payment is also received at one of the predetermined payment periods. The project is represented on an Activity-on-Node (AON) project network. Activity durations are assumed to be deterministic. The project duration is bounded from above by a deadline imposed by the client. The deadline imposed constitutes a hard constraint meaning that exceeding the deadline will violate feasibility. Thus, there is no need to specify a penalty for exceeding the deadline. There is no explicitly stated bonus for the contractor to finish the project earlier than the deadline agreed upon.

Other than the progress payments model, we will also consider two other payment models; namely, payment at event occurrences and payment at equal time intervals. The reason for excluding lump-sum payment model is that such a model concentrates solely on finding the optimal schedule. Since the proposed bargaining model considers the amount and the time of cash flows along with the project schedule, starting with a predetermined total payment does not incorporate a bargaining objective.

Contractor's cash outflows associated with an activity can occur anywhere throughout the activity. However, it is assumed here that they will be discounted to the starting time of the activity. The cash inflows for the contractor, which represent the cash outflows for the client, occur at predetermined equal time intervals. In this context, the earned value for the contractor corresponds to the payments regarding the activities completed within that specific period of time. If the project is completed earlier than the deadline, then the last payment occurs at the deadline. The payments are specified as the sum of the costs incurred for all activities completed until that payment point and multiplied with $(1 + \beta)$, where β is the profit margin agreed upon by both parties. Note that activities in progress are not included in this sum. The problem is formulated under zero-lag finish-start precedence constraints and multi-mode renewable resource constraints.

Since we consider the client-contractor bargaining problem, the objective function should reflect the two-party nature of the problem environment. The objective function represents a compromise solution for the client and the contractor. The fundamental metric proposed is the distance of the party involved from the worst possible solution it can face. The bargaining value, f'_A is defined as the objective function value of the client at the solution point that maximizes the contractor's objective function. Similarly, the bargaining value, f'_B is the objective function value of the contractor at the solution point that maximizes the client's objective function. The bargaining objective function tries to maximize the minimum of distances respectively from f'_A and f'_B. In other words, the bargaining function is meant to improve the worse-off party between the client and the contractor. The relative bargaining positions of the client and the contractor differ in general. To introduce the impact of this difference in relative bargaining positions, bargaining power parameters are defined for both the client and the contractor. A large bargaining power parameter implies a strong bargaining position. The bargaining values f''_A and f''_B denote the individual optimal objective function values for the client and the contractor, respectively. For each player, optimal solution is the result of the single objective problem solved by the commercial solver. These values are involved in the bargaining objective function for normalization.

1.4 Mathematical Formulation

The mathematical formulation for the resource constrained bargaining problem with progress payments is presented below.

1.4.1 Notation

$f_A(\mathbf{x})$: objective function value of the client depending on the current schedule $\mathbf{x}$

$f_B(\mathbf{x})$: objective function value of the contractor depending on the current schedule $\mathbf{x}$

f'_A : the bargaining objective function value for the client at the optimal solution of the contractor

f'_B : the bargaining objective function value for the contractor at the optimal solution of the client

f''_A : optimal objective function value for the client

f''_B : optimal objective function value for the contractor

$w(A)$: bargaining power parameter for the client

$w(B)$: bargaining power parameter for the contractor

$w(A) \in (0,1)$, $w(B) \in (0,1)$ and $w(A)+ w(B)=1$.

J : set of activities, j ϵ **J**, j = 1,....,J

i<j : indicates precedence relation that activity i precedes activity j

M_j : number of modes for activity j

m : index for mode type, m=1, …, M_j

t : period index, t=1, 2, …, D.

$$x_{jtm} : \begin{cases} 1, \text{ if activity j is in mode m and completed in period t} \\ 0, \text{ otherwise} \end{cases}$$

d_{jm} : duration of activity j in mode m

β : profit margin

α : interest rate

c_t : continuous discount factor, exp(-αt)

C_{max} : makespan

N_{jm} : cost of activity j in mode m in real terms discounted to the starting time of activity j

D : predetermined deadline

T : set of predetermined payment times T_n, **T** = {T_0, T_1, ..., T_N}, where T_0 = 0 and T_N = D

P_{Tn} : client's payment at predetermined period T_n

k : index for resource type, k = 1, ..., K

R_k : availability limit of resource k

r_{jkm} : consumption of resource k per unit time for activity j in mode m

E_j : earliest finishing time of activity j

L_j : latest finishing time of activity j

$$\text{Max min} \left\{ \left(\frac{f_A(x) - f'_A}{f''_A - f'_A} \right)^{w(A)}, \left(\frac{f_B(x) - f'_B}{f''_B - f'_B} \right)^{w(B)} \right\} \tag{1.1}$$

Subject to:

$$f_A(X) = -(\sum_{t \in T} c_t * P_t) \tag{1.2}$$

$$f_B(X) = \sum_{t \in T} c_t * P_t - \sum_{i \in J} \sum_{t=E_j}^{L_j} \sum_{m=1}^{M_j} N_{jm} * c_{t-dur(j)} * x_{jtm} \tag{1.3}$$

$$P_{T_n} = (1+\beta) * \sum_{j \in J} \sum_{t=T_{n-1}+1}^{T_n} \sum_{m=1}^{M_j} N_{jm} * x_{jtm} \qquad T_n \in \mathbf{T}, n>0 \qquad (1.4)$$

$$\sum_{t=E_j}^{L_j} \sum_{m=1}^{M_j} x_{jtm} = 1 \qquad \text{for all } j \in \mathbf{J} \qquad (1.5)$$

$$\sum_{t=E_j}^{L_j} \sum_{m=1}^{M_j} t * x_{jtm} - \sum_{t=E_i}^{L_i} \sum_{m=1}^{M_j} t * x_{itm} \geq d_{jm} \qquad \text{for all } i<j, \text{ for all } i,j \in \mathbf{J} \qquad (1.6)$$

$$\sum_{t=E_J}^{L_J} \sum_{m=1}^{M_j} t * x_{Jtm} \leq D \qquad (1.7)$$

$$\sum_{j \in J} \sum_{q=t}^{t+d_j} \sum_{m=1}^{M_j} r_{jkm} * x_{jqm} \leq R_k \qquad k = 1, ..., K \text{ and } t = 1, ..., C_{max} \qquad (1.8)$$

$$f_A(x) \geq f'_A \qquad (1.9)$$

$$f_B(x) \geq f'_B \qquad (1.10)$$

$$x_{jtm} \in \{0,1\} \qquad \text{for all } j \in \mathbf{J}, m=1,\ldots M_j \text{ and } t = 1, ..., C_{max} \qquad (1.11)$$

Expressions (1.2) to (1.4) define respectively $f_A(\mathbf{x})$, $f_B(\mathbf{x})$, and P_{T_n} in terms of the decision variable and problem parameters. The constraint set (1.5) assures that each activity is assigned. The constraint set (1.6) makes sure that all precedence relations are satisfied. The constraint (1.7) secures that the project is completed on or before the deadline. The constraint set (1.8) makes sure that for every single resource the required amount does not exceed the corresponding resource constraint throughout the project duration. Finally, the constraints (1.9) and (1.10) ensure that the suboptimal solutions that yield negative numerators in the objective function are ignored. This guarantees that no adopted schedule can provide one of the players a solution worse than his/her worst case objective. The above mathematical programming formulation is a non-linear zero-one programming problem. Hence, one would expect that exact methods would fail even for moderate size problems.

The conventional solution procedures for the resource constrained project scheduling problem with discounted cash flows adopt the perspective of either the client or the contractor. Hence, these procedures developed for optimizing the benefit for one party only would not be expected to produce good solutions for the bargaining objective, which aims to merge the objectives of both the client and the contractor into a single function.

2 SOLUTION APPROACHES

In this chapter, first we introduce the solution methodologies we have used. Here we define the methodology and share the test results. Then we investigate the contractual preferences of the players in differing payment model settings.

2.1 Proposed Solution Methods

The resource constrained problem with NPV maximization bargaining objective does not depend on structured rules like completing the costly activities earlier, or delaying the project. Instead the rules change depending on the interest rate, the activity costs indices, and the precedence relations. In this sense a stepwise construction heuristic does not give good solutions without considering alternative solutions through search methods. We will employ two different meta-heuristics, namely Simulated Annealing (SA) and Genetic Algorithm (GA).

The test problems we have used are adopted from PSPLIB - A project scheduling problem library developed by Kolisch and Sprecher (1996). We have used their problem sets with activity numbers of 14, 20, and 32. All of the problem sets we have used has two nonrenewable resources and three modes for the activities. The first mode consists of the highest activity cost with the shortest duration, and the last mode consists of the lowest activity cost with the longest duration.

2.1.1 Simulated Annealing Algorithm 1 (SA1)

2.1.1.1 Solution Representation

Each solution is represented by two serial lists: activity starting time list, and mode list. The first list represents the finishing time of the activities in increasing order of their index numbers, and the second list represents the mode of the activity which identifies duration, and resource consumption figures for that particular activity. An example for this solution representation is as follows:

Finishing Time List:

1	3	6	9	8	4	13	10	13	19	25	30	21	30

Mode List:

1	2	2	3	1	2	3	1	2	3	1	2	3	1

Figure 2.1 Solution representation for SA 1 for a 14 activity problem

2.1.1.2 Neighborhood generation mechanism

We use two types of alternative generating methods. The first one is sliding an activity's starting time and the second one is swapping and sliding two activities' starting times.

a. **Sliding an activity:** Since we are looking for alternative solutions in the neighborhood of the current solution, sliding the considered activity's starting point should not affect the starting points of other activities. In this sense we consider the slack time window for each considered activity. For a particular activity, this time window starts with the latest of the finishing times of its predecessors, and ends with the earliest of the starting times of its successors (see Figure 2.(a), the shaded cells indicate the time periods that the activity is in progress). If the activity has no predecessors, time window starts at time 0, and if the activity has no successors, time window ends with the project deadline. Once these intervals are determined for each activity, we need to find out the specific time periods that the activity can start at. First we remove the considered activity from the schedule and add its resources to the available resource amounts at the time periods that the activity is in process (see Figure 2.(b)). Then through the determined time window with the revised resources we look for intervals at the length of the activity duration in which activity's resource requirement is satisfied in every single one of the consecutive time periods. Each of the starting times of these resource satisfied intervals, other than the original starting time of the activity, is considered as an alternative starting time and is included in the sliding neighborhood for the considered activity (see Figure 2.(c)). This procedure is repeated for each activity in the project and a complete set of sliding neighborhood is formed.

Figure 2.2 An alternative sub-schedule generated by sliding

b. **Swapping and sliding two activities:** Resource constraint is a very restrictive factor for determining the sliding alternatives of a particular activity. So sometimes, interchanging the positions of two activities on the time schedule may create new alternatives to form complete intervals at the length of activity's duration that cannot be reached by individual sliding operations. In this instance, we determine the independent activity pairs in the project which have intersecting time windows (see Figure 3.(a)). These independent pairs should not have a precedence relation of any kind. At least one period of in process time of either activity need to intersect with the time window of the other activity in order to have a difference in the possible alternative intervals. Consider a particular pair. We first remove both of the activities from the schedule and add their resources to the available resource amounts at the time periods that these activities are in process (see Figure 3.(b)). Then with the revised resource amounts on the existing schedule, we repeat the sliding procedure for each activity individually through their own time windows (see Figure 3.(c)). Among the set of pairs of activities and their starting times, the solutions that we can find by only sliding should be disregarded. So, for each activity pair, at least one of the adopted starting times should be different from the sliding alternatives for that particular activity. For a particular pair, we figure out the possible alternatives in this way. This procedure is repeated for each independent pair in the project and a complete set of swapping and sliding neighborhood is formed.

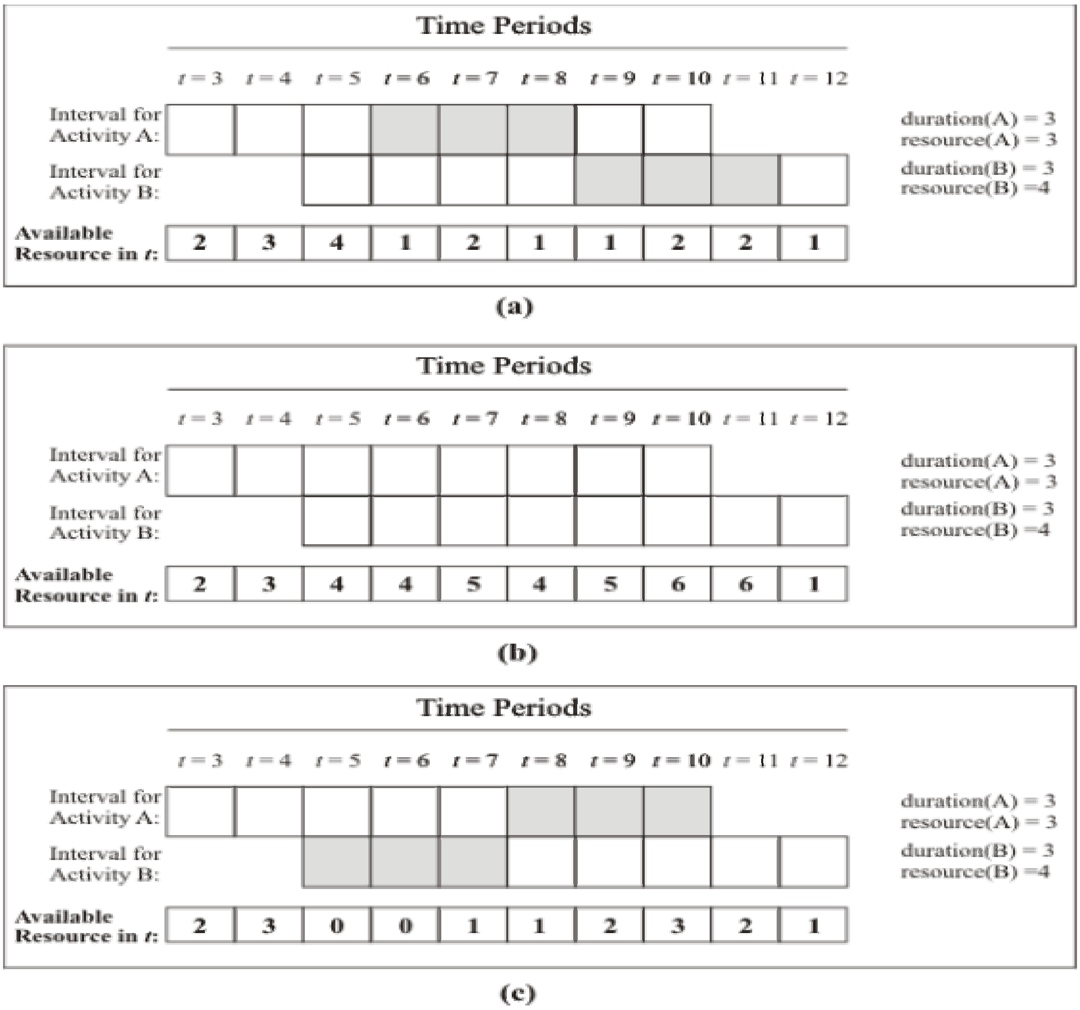

Figure 2.3 An alternative sub-schedule generated by swapping and sliding

2.1.1.3 Cooling mechanism

The cooling scheme we use was originally used by Baykasoğlu, Gindy and Cobb (2001), and adopted by Sivrikaya-Şerifoğlu and Tiryaki (2002). The details are given below:

$$T_{in}=(f_{min}-f_{max})/(\ln P_A^{init}) \quad (2.1)$$

$$T_{curr} = (\ln P_A^{init} / \ln P_A^{f})^{1/(maxIter-1)} * T_{curr} \quad (2.2)$$

where P_A^{init} is the initial acceptance probability and is set at 0.95, f_{min} is the minimum objective function value and is estimated by the minimum of generated neighborhood solutions at each iteration, f_{max} is maximum objective function value and is estimated by the maximum of generated neighborhood solutions at each iteration, P_A^f is the final acceptance probability set to be 0.001, and maxIter is the maximum number of temperature reduction cycles.

Throughout the SA1 search with this cooling scheme, modifying the plateau length L was also tested, but no significant improvement regarding the solution has been observed.

2.1.1.4 The Algorithm

STEP1: Find an initial solution. Determine the starting times for each activity. Since preemption is not allowed, the other information that we require in the following steps, like the activities in process at each time period, or the completed activities by a specified time period may easily be reached only by knowing the starting times.
STEP2: Determine the alternative solutions in the neighborhood.
STEP3: Among the complete set of all alternatives, we randomly pick one. If the selected alternative brings a better solution in terms of the bargaining objective, then it is adopted. If it is an inferior solution, i.e. its NPV is less than the existing NPV, then we determine whether to accept it or not according to the probability of acceptance P_A.

2.1.1.5 Stopping Criterion

A fixed iteration count (maxIter) is adopted as the stopping criterion. The number of carried iterations for each problem strictly depends on the number of activities. As the number of activities increase, iteration count increases. One other tested method is observing the improvement in the objective function. According to this method, process ends if there is not a significant improvement in the objective function. Test results show that this is not a convenient method for the tested objective function, and hence, has not been adopted as a stopping criterion for this study. Since objective function is in max min format, same objective function value may be observed at different solution points. For this reason, sometimes it is observed that a superior solution may be reached after several solutions with the same objective function value.

2.1.1.6 Tests

The tests conducted with SA1 gave satisfactory results (given in Table 2.1) when compared to the optimal solutions for small sized problems.

Table 2.1 Test results for SA1

	14 activity network	20 activity network	32 activity network
Progress payments at 10 time periods	92%	88%	82%
Progress payments at 5 time periods	95%	90%	84%
Payments at activity completion	97%	93%	87%

2.1.2 Simulated Annealing Algorithm 2 (SA2)

2.1.2.1 Solution Representation

Each solution is represented by three serial lists: activity list, mode list, and idle time list. The structure and feasibility status of these lists are explained as follows:

1. Activity List:
   Activity list is a string that includes all activities in a row. For example, for a network of 10 activities, an activity list may be as the following: 1-3-6-8-2-4-5-9-7-10. Since the source and the sink nodes are dummy activities, we place them at the start and the end as default. The list represents the starting priority ordering for the activities. That is, once an activity appears earlier in the list, it should start at the same time or at an earlier time than its immediate follower. For example, in our provided list, the starting time of activity 3 should be less than or equal to the starting time of activity 6. In other words, the list must be precedence feasible.

2. Mode List:
   The mode list shows the assigned modes for each activity in the list.

3. Idle Time List:

   The idle time value represents the exact idle time to be inserted before the corresponding activity starts.

Each feasible group of triple lists (namely activity list, mode list, and idle time list) is considered as a complete solution. An example for a 14 activity network is as given in Figure 2.4.

Activity List:

1	3	2	5	6	4	9	10	8	7	13	11	12	14

Mode List:

1	2	2	3	1	2	3	1	2	3	1	2	3	1

Idle Time List:

1	4	5	3	1	3	0	4	1	2	3	0	1	1

Figure 2.4 Solution representation for SA1 for a 14 activity problem

2.1.2.2 Initial Solution Generation Mechanism

We find the initial solution by generating a set of random feasible solutions, and then selecting the best point among them. Our initial solution set construction method for all lists is as follows:

a. **Activity List:**

   The method we use to generate an activity list is a common method where we keep a list of eligible activities, which at the beginning is composed of activities with no predecessors. Randomly we choose one of the activities from this set to insert into the next position on the chromosome. Then we update the set by deleting the activity chosen and by inserting activities, predecessors of which are all inserted into the chromosome.

b. **Mode List and Idle Time List:**

   For each feasible activity list we have, we generate mode lists and idle time lists randomly. For a network with 14 activities we generate 100 mode and idle time lists among which we again search for a feasible combination. For each group of activity list, mode list, and idle time list we have, we check the feasibility according to both the resource limitations, and the deadline. The schedule is, of course, constructed considering the precedence relations, an activity cannot be started until all its predecessors have been finished. The percentage of feasibility at this level is 11%.

2.1.2.3 Neighborhood Generation Mechanism

We use three types of alternative generating methods. The first one is replacement, the second one is mode change, and the third one is idle time change. Neighbors are created by one of these methods.

a. **Replacement:** In this method, for a selected activity from the activity list, an alternative location on the activity list is found. The activity together with its corresponding mode and idle time assignments is moved to each one of the feasible alternative locations one by one. If the new solution satisfies the deadline constraint, it is included into the neighborhood. This way, all feasible neighboring solutions are generated by applying the replacement operator.
b. **Mode change:** By keeping the activity list and the idle time list constant, mode changes are applied among the list, one mode at a time, and feasible solutions are included into the neighborhood. In this generation method, all possible solutions are searched in two dimensions: list length, and mode alternatives.
c. **Idle Time change**: By keeping the activity list and the mode list constant, idle time changes are applied among the list, one idle time at a time, and feasible solutions are included into the neighborhood. In this generation method, all possible solutions are searched in two dimensions: list length, and idle time alternatives.

2.1.2.4 Cooling mechanism

In SA2 studies we have used the same cooling scheme we have used in SA1. Throughout the SA2 search with this cooling scheme, modifying the plateau length was also tested, but again like SA1 no significant improvement regarding the solution has been observed.

2.1.2.5 The Algorithm

In SA2 we use exactly the same algorithm we have used in SA1.

2.1.2.6 Stopping Criterion

In SA2 we use exactly the same stopping criterion we have used in SA1.

2.1.2.7 Tests

The tests conducted with SA2 gave better results (given in Table 2.2) than SA1.

Table 2.2 Test results for SA2

	14 activity network	20 activity network	32 activity network
Progress payments at 10 time periods	94%	90%	84%
Progress payments at 5 time periods	96%	92%	86%
Payments at activity completion	99%	94%	88%

2.1.3 Genetic Algorithm

2.1.3.1 Solution Representation

Each solution is represented by three serial lists: activity list, mode list, and idle time list. The structure and feasibility status of these lists are explained as follows:

1. <u>Activity List:</u>
   A precedence-feasible permutation is kept as the activity list.

2. <u>Mode List:</u>
   The mode list shows the assigned modes for each box in the list.

3. <u>Idle Time List:</u>

The idle time value represents the exact time frame that an activity should spend when its turn on the list comes after all its predecessors are completed.

Each feasible group of triple lists (namely activity list, mode list, and idle time list) is considered as a complete chromosome. Figure 4 represents a chromosome example for a network of 14 activities.

2.1.3.2 Generating Feasible Chromosomes

The most important thing is to maintain the feasibility of the constructed solutions. In this sense we use a stepwise feasibility test through the algorithm. We first maintain the feasibility test through the activity list, and only feasible lists pass for the next level representations, i.e. mode list and the idle time list. The construction method for all list types is the same as we used in SA2.

2.1.3.3 Fitness Value

For each chromosome, the fitness value is determined by the original objective function value.

2.1.3.4 Selection

We use roulette wheel selection operator. With this approach the probability of selection is proportional to an individual's fitness. Goldberg (1989) explains that the analogy with a roulette wheel arises because one can imagine the whole population forming a roulette wheel with the size of any individual's slot proportional to its fitness. The wheel is then spun and the figurative "ball" thrown in. The probability of the ball coming to rest in any particular slot is proportional to the arc of the slot and thus to the fitness of the corresponding individual. The algorithm is summarized below:

STEP1. Sum the fitness of all the population members. Call this sum f_{sum}.

STEP2. Choose a random number, R_s, between 0 and f_{sum}.

STEP3. Add together the fitness of the population members (one at a time) stopping immediately when the sum is greater than R_s. The last individual added becomes the selected individual and a copy is passed to the next generation.

2.1.3.5 Crossover Operator

The most challenging problem we face when applying GA is to reproduce feasible off-springs. There are three strict constraints we need to deal with: precedence constraints, the deadline and resource constraints. When working on an activity list, taking these constraints into consideration has crucial importance for the next iterations. It is safer to continue with feasible solutions instead of dealing with infeasibility. In this sense, working with the following operator preserves feasibility in the activity list.

- Multi Component Uniform Order Based Crossover (MCUOX): By this recombination operator, which was proposed by Sivrikaya Şerifoğlu (1997), from a couple, we select one of the parents randomly. We find the activity on that parent not assigned to the child yet. Then we find the mode assignment of that activity on each of the parents, select one randomly. Finally we find the idle time assignment of that activity on each of the parents, select one randomly. We repeat it until the activity list of the child is completed. The offspring construction process continues until the number of feasible offspring reaches 1/3rd of the original population. 1/3rd is a design choice which enables highest improvement at each iteration when compared with other tested ratios: 1/6th, 1/4th, and 1/2.

2.1.3.6 Mutation Operators

- Replacement: Replacement operator is applied to randomly selected individuals until the number of feasible individuals reaches the 1/6th of the population. Here 1/6th is a design choice which enables highest improvement at each iteration when compared with other tested ratios: 1/4th, and 1/3rd. We select a parent chromosome randomly. Then we select an activity from the activity list of that chromosome. Next we change the position of that particular activity on the activity list. Activity's replacement window is determined according to the precedence relations. Once the activity is moved, whole list is adjusted accordingly. The corresponding mode list, and idle time list are also moved together with the activity list replacement. Then the feasibility of the whole new list is explored, and new solution is accepted as a child chromosome if it satisfies all feasibility constraints. The offspring construction process continues until the number of feasible offspring reaches 1/6th of the original population.

- Bit mutation: We define bit mutation in our three list model by changing mode or idle time for a specific activity on the list. Bit mutation is applied to the whole population in the pool other than the elitists, as the last step of new generation creation process. For each chromosome, other than the elitists, the bit mutation probability is 50%, that is we bit mutate either the mode or idle time assignment of a chromosome with probability of 0.5.

2.1.3.7 Population Management (Population Replacement Strategy)

Elitist strategy is used, such that two percent of the original population is carried in tact to the next generation in order to preserve the elites in the population. Then bit mutation is applied to the set of newly generated off-springs together with the selected chromosomes from the original population. Those chromosomes with a higher fitness value are more likely to be carried to the next generation. Mutation is applied to randomly selected chromosomes.

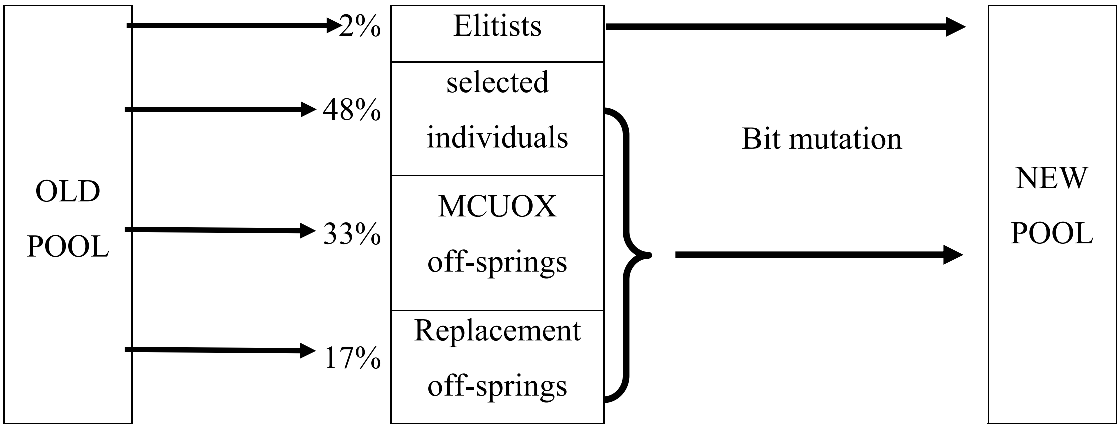

Figure 2.5 GA pool management scheme

2.1.3.8 Solution Generation

The whole cycle is repeated for 50-100 times. At each cycle, the chromosome with the highest fitness value ever is kept in memory. The final solution is the best fitness value reached ever after the last cycle.

2.1.3.9 Tests

The tests conducted with GA gave better results (given in Table 2.3) than both SA1 and SA2.

Table 2.3 Test results for GA

	14 activity network	20 activity network	32 activity network
Progress payments at 10 time periods	95%	91%	85%
Progress payments at 5 time periods	97%	93%	87%
Payments at activity completion	99%	95%	88%

2.1.3.10 Payment Model Studies with Genetic Algorithm

In order to increase the amount of testing carried over different payment models, we need to have the solutions for the single objective problems. But unfortunately the commercial optimizer we use (GAMS 20.0) couldn't find the solution for all the payment models we consider. In this sense, we used our activity list solution representation and GA in order to reach a solution for the single objective problems defined using different payment models. By this way we were able to find solutions for the bargaining objective of occupied with various other payment models. This enabled us being able to test payments at equal time intervals model as well as other models we have been testing and comparing with the optimal solutions. In this model although we didn't have the opportunity to compare with the optimal solution, we were able to distinguish the relation and differences with other payment models. We figured out that, once we have relaxed the timing of the last payment, and set it as the project finish time rather than the deadline, we had more chance for the project to be finished earlier. Main issue is that, once the contractor receives the last payment at the deadline no matter what, he may not have any motive to start the activities earlier.

2.1.4 Non-Dominated Solutions Set

Within the solution space we generate in both search methods we use, there exist some specific solutions which are dominant over others. Through the search methods we have incorporated, namely SA and GA, at each step we most likely eliminate inferior solutions. In this sense, identifying the non-dominant solutions prior to our search may provide ease in computation and even improvement in the generated solution space. Thus we have generated and analyzed non-dominated solution sets for specific problems. We have constructed tests for problem sets with 14 activities. The generation method we have used is summarized as given below:

2.1.4.1 Solution Representation

Each solution is represented by the exact same three serial lists we have used in GA: activity list, mode list, and idle time list.

2.1.4.2 Solution Space Generation

We use an adapted version of chromosome generation mechanics we have used in GA. We again use the exact same stepwise approach we have used in GA in order to generate feasible solutions. We first generate the Activity List, then generate the Mode List and Idle Time List.

2.1.4.3 Selection of Non-Dominated Solutions

Within the feasible solutions set we have, we identify the non-dominated solutions regarding the objective function values of the players. In this context, we could also be taking absolute objective function values of the players into consideration, but that way graphical scaling would be a problem since the absolute objective function values of the players are not normalized. We eliminate solutions, which have both players' objective functions inferior to another solution.

2.1.4.4 Neighborhood Search

Once we identify the initial non-dominated solutions set, we search the neighborhood in order to be able to extend the set we have. At this step we use two kinds of neighborhood generation operators:

1. <u>Replacement:</u> Replacement operator is applied to each individual in the set. We select a solution from the non-dominated set. Then we select an activity from the activity list of that solution. Next we change the position of that particular activity on the activity list. Activity's replacement window is determined according to the precedence relations. Once the activity is moved, whole list is adjusted accordingly. The solution construction process lasts when we incorporate each possible activity move for all solutions.

2. Swap: This operator is applied to mode list and idle-time list of the randomly selected individuals from the solution set. We select a solution randomly then select two cells again randomly from either the mode list or the idle-time list. We swap these two cells, and then check the feasibility of the generated solution.

2.1.4.5 Stopping Criterion

Once the neighborhood is generated, we again identify the non-dominated solutions within the whole solution set. We continue this generation cycle until no change occurs within the list for two cycles in a row. Here, we do not base the decision on the number of non-dominated solutions in the set, since an improvement not necessarily brings an increase in the number of non-dominated solutions. Improvements may occur as introducing new non-dominated solutions that may eliminate some already existing solutions.

Through the tests we have conducted, we observed that the set of non-dominated solutions appear as a polynomial curve on the border of the overall solution set. Figure 2.6 shows the non-dominant solutions set regarding objective function values of the contractor and the client. When we search for the best solution among non-dominated solutions set, we come up with a solution which is comparable with the solutions we have generated with metaheuristic search algorithms. This may lead us to a conclusion that if we choose our starting point from non-dominated solutions in other metaheuristic search algorithms we may improve the results we deliver. This dimension may be analyzed in further study.

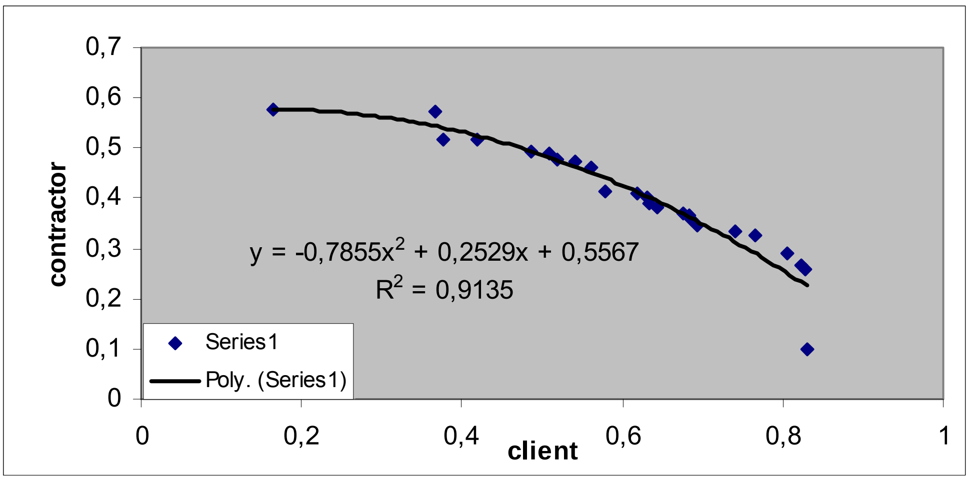

Figure 2.6 Non-dominated solutions curve

In these tests, once we fix a curve on the solution points we observe that the R^2 value of the curve is always higher than 0,9. This indicates a good fit of the curve. Hence, we may use this curve to identify acting-optimal solutions for the problem sets that we don't have any optimal solutions to compare the solutions we have generated with our metaheuristic algorithms. For example, in bargaining weight tests with higher activity numbers commercial solvers hadn't delivered optimal solutions, so that we may use this curve's equation to generate acting-optimal solutions.

2.1.2.7 Tests

The tests conducted with non-dominated solutions set gave comparable results (given in Table 2.4) with SA1 and SA2.

Table 2.4 Test results for non-dominated solutions set

	14 activity network	20 activity network	32 activity network
Progress payments at 10 time periods	94%	90%	84%
Progress payments at 5 time periods	96%	92%	86%
Payments at activity completion	98%	94%	87%

2.1.5 Test Results

Tables 2.6-2.8 summarize the test results for four different methods studied; namely, SA1, SA2, GA, and Non-dominated Solutions Set. In these tests we have w(A) = w(B) = 0,5. The legend for abbreviations used when reporting on the test results in Tables 2.6, 2.7, and 2.8 is provided in Table 2.5.

Table 2.5 Legend for abbreviations

n=14	refers to	problem network with 14 activities
n=20		problem network with 20 activities
n=32		problem network with 32 activities
SA_1		Simulated Annealing 1
SA_2		Simulated Annealing 2
GA		Genetic Algorithm
ND		Non-dominated Solution Set
pp10		progress payments at 10 time periods
pp5		progress payments at 5 time periods
ac		payments at activity completion

T-Tests we have conducted gave us the results summarized in Tables 2.6, 2.7, and 2.8. In this analysis, risk level indicates the probability of two sets belonging to equivalent populations. And likewise, significance level indicates the probability of belonging to equivalent populations and being incomparable.

Table 2.6 Comparison of different payment models

Among a problem set with 14 activities	SA_1 tests show that	pp5 delivers higher OFV than pp10 with a risk level of 0,1% ac delivers higher OFV than pp5 with a risk level of 0%
	SA_2 tests show that	pp5 delivers higher OFV than pp10 with a risk level of 0,1% ac delivers higher OFV than pp5 with a risk level of 0%
	GA tests show that	pp5 delivers higher OFV than pp10 with a risk level of 0,3% ac delivers higher OFV than pp5 with a risk level of 0,1%
	ND tests show that	pp5 delivers higher OFV than pp10 with a risk level of 8% ac delivers higher OFV than pp5 with a risk level of 10%
Among a problem set with 20 activities	SA_1 tests show that	pp5 delivers higher OFV than pp10 with a risk level of 0,2% ac delivers higher OFV than pp5 with a risk level of 0%
	SA_2 tests show that	pp5 delivers higher OFV than pp10 with a risk level of 0,1% ac delivers higher OFV than pp5 with a risk level of 0%
	GA tests show that	pp5 delivers higher OFV than pp10 with a risk level of 0,3% ac delivers higher OFV than pp5 with a risk level of 0%
	ND tests show that	pp5 delivers higher OFV than pp10 with a risk level of 8% ac delivers higher OFV than pp5 with a risk level of 10%
Among a problem set with 32 activities	SA_1 tests show that	pp5 delivers higher OFV than pp10 with a risk level of 0,1% ac delivers higher OFV than pp5 with a risk level of 0%
	SA_2 tests show that	pp5 delivers higher OFV than pp10 with a risk level of 0,1% ac delivers higher OFV than pp5 with a risk level of 0%
	GA tests show that	pp5 delivers higher OFV than pp10 with a risk level of 0,3% ac delivers higher OFV than pp5 with a risk level of 0%
	ND tests show that	pp5 delivers higher OFV than pp10 with a risk level of 8% ac delivers higher OFV than pp5 with a risk level of 10%

Table 2.7 Comparison of the solution methods

Among a problem set with 14 activities	SA_2 delivers higher OFV than SA_1 with a risk level of 0,2% GA delivers higher OFV than SA_2 with a risk level of 0,9% GA delivers higher OFV than ND with a risk level of 8% ND delivers higher OFV than SA_1 with a risk level of 0,9% OFVs of ND and SA_2 are incomparable with a significance level of 84%
Among a problem set with 20 activities	SA_2 delivers higher OFV than SA_1 with a risk level of 0,3% GA delivers higher OFV than SA_2 with a risk level of 0,9% GA delivers higher OFV than ND with a risk level of 8% ND delivers higher OFV than SA_1 with a risk level of 0,9% OFVs of ND and SA_2 are incomparable with a significance level of 84%
Among a problem set with 32 activities	SA_2 delivers higher OFV than SA_1 with a risk level of 0,2% GA delivers higher OFV than SA_2 with a risk level of 0,9% GA delivers higher OFV than ND with a risk level of 8% ND delivers higher OFV than SA_1 with a risk level of 0,9% OFVs of ND and SA_2 are incomparable with a significance level of 84%

Table 2.8 Comparison of the problem sets with different activity numbers

A network with 20 activities delivers higher OFV than a network with 14 activities with a risk level of 0%
A network with 32 activities delivers higher OFV than a network with 20 activities with a risk level of 0%

These results show that as the frequency of the payments increases and the number of activities decrease, the likelihood of finding near optimal solutions increase, no matter which method we use. Moreover, GA provided us with better results than SA 1 and SA 2 did. On the other hand, ND provided solutions which were compatible with SA algorithms. The success of ND stands from the strength of its initial population at each iteration. The major weakness of ND is that, the standard deviation of its results is significantly higher when compared with the standard deviation of the results of other solution methods.

2.2 The Contractual Preferences of the Players

We have enriched the experiments with two further payment structures, on top of the basic model which suggests a profit margin for the contractor on top of the total cost. In the model that we have used through the tests, we used three payment models:

- First one consists of a profit margin over the total cost paid by the client to the contractor at the predetermined payment points,
- Second one consists of a constant fee paid by the client to the contractor at the beginning of the project as an advance payment,
- Third one incorporates a benefit for the client as a function of early completion time.

Within these payment models, there are sub models we have tested, which are more likely to deal with the frequency and timing of the interim payments: payments at event occurrences, progress payments, and payments at equal time intervals. We conducted these tests by using commercial solver which delivers optimal solutions. The test results we have observed for each of these models may be summarized as given below:

1. Cost + Profit over cost: In this model the client pays a profit percentage (β) over the total cost at the predetermined payment points For example, if we consider $f_A(X)$ as the objective function of the client, and β as the profit margin of the contractor, the mathematical model would be shown as follows:

$$f_A(X) = -(\sum_{t \in \tau} c_t * P_t) \qquad (2.3)$$

$$P_{T_n} = (1+\beta) * \sum_{j \in J} \sum_{t=T_{n-1}+1}^{T_n} \sum_{m=1}^{M_j} N_{jm} * x_{jtm} \qquad T_n \in \mathbf{T},\ n>0 \qquad (2.4)$$

The number and amount of each payment (P_T) depends on the payment frequency, which is another parameter of the whole model. In this sense we have included sub-payment structures into the model:

a. Payments at event occurrences
b. Progress payments
c. Payments at equal time intervals

Each of these sub models has been included into our tests. Results show that the metaheuristic algorithms we propose deliver better results as we move to more frequent payments from bulk payments. This is due to the fact that as the frequency of the payments increase over the whole network, there is less room for activities to move in order to create more alternative solutions.

2. Cost + Constant Fee: In this model the client pays a constant fee for the services of the contractor at the end of the project in addition to the activity costs paid at each predetermined payment periods. This model puts pressure on the contractor to finish the project as soon as possible so as to receive the constant fee early.

$$f_A(X) = -(\sum_{t \in \tau} c_t * P_t) - F*c_D \tag{2.5}$$

$$f_B(X) = \sum_{t \in \tau} c_t * P_t - \sum_{i \in J} \sum_{t=E_j}^{L_j} \sum_{m=1}^{M_j} N_{jm} * c_{t-dur(j)} * x_{jtm} + F*c_D \tag{2.6}$$

$$P_{T_n} = \sum_{j \in J} \sum_{t=T_{n-1}+1}^{T_n} \sum_{m=1}^{M_j} N_{jm} * x_{jtm} \qquad T_n \in \boldsymbol{T},\ n>0 \tag{2.7}$$

Notation:

F: Constant fee

One important outcome of this model is that the individual objectives of both the client and the contractor over the adopted schedule are equal.

This proves that although the objective function we use tends to equate both objective functions by maximizing the minimum of them, the discrete nature of the model doesn't allow this happen most of the time. Recall that the objective function we used in our tests is:

$$Max\ min \left\{ \left(\frac{f_A(x) - f'_A}{f''_A - f'_A} \right)^{w(A)}, \left(\frac{f_B(x) - f'_B}{f'_B - f'_B} \right)^{w(B)} \right\} \tag{2.8}$$

In this function, we have two buckets, one for the client and one for the contractor. Since we are looking for maximizing the minimum of these two, and there is a trade-off between these two objectives, having these two equal at the optimum point is an expected result. However, the predetermined mode structures and all types of limitations prevent the model to equate two individual objectives. By giving an advance payment to the contractor, and leaving the amount of this advance payment to negotiations, we let the two objectives to have a continuous trade-off of benefits.

In Table 2.9, objective function split by each player is shown with the results of 6 tests for each payment type.

Table 2.9 Objective functions of the client and the contractor at two payment models

	1		2		3		4		5		6	
Bargaining Objectives	Contractor	Client	Contractor	Client	Contractor	Client	Contractor	Client	Contractor	Client	Contractor	Client
Cost+Constant fee	0,674	0,674	0,727	0,727	0,722	0,722	0,709	0,709	0,709	0,709	0,718	0,718
Cost+Profit over Cost	0,556	0,647	0,525	0,606	0,556	0,508	0,561	0,558	0,532	0,535	0,526	0,534

Table 2.9 clearly shows that the objective function tends to equate the individual objective functions of each player when there is room for it. On the other hand, when there is a continuous benefit subject to negotiation, balance between objective functions is maintained.

One important point concerning the constant fee model is that the bargaining powers of the players have significant effect on the decision process to determine the amount of constant fee within predetermined ranges. In the tests we conduct, we find out that the client favors the lower level of the constant fee, and the contractor favors the higher level of the constant fee, where their negotiation point differs within this range directed by their bargaining power. Since the constant fee is a marginal amount that effects both sides at the same marginal level, its absolute effect on both sides of the objective function differs depending on the f''_A, f''_B, f'_A, and f'_B values.

Results show that in the constant fee model, where the contractor doesn't get a profit margin over activity costs, both of the players favor longer and less costly schedules within the deadline. Thus, neither the client pays more, nor the contractor. Since we haven't defined any benefit for the client to be obtained when the project finishes earlier than the set deadline, the client itself favors cheaper modes of the activities. In this context the preferred schedules of the client and the contractor doesn't differ much. They both target the deadline at the lowest possible cost. This leads the negotiations to slide towards the fee rather than the schedule, since both have agreed on the scheduling rules.

Only if we include a profit margin for the contractor, which is higher than the interest rate (since client pays the cost at the beginning of the activities, and receives the payment earliest at the end of the activity), the contractor begins favoring more expensive modes for activities that results in much shorter schedules. This model, which includes both a constant fee, and a profit margin over the activity costs, lets negotiations include both the scheduling problem and distribution of the benefit which appears as the constant fee. Here, the bargaining power strongly affects both the schedule and the amount of constant fee, if it is subject to negotiation.

We have conducted several tests on this model to be able to identify the time-cost trade off relationship in the model:

a. In the first set of tests we have fixed the deadline at several levels and found the negotiated amount of the constant fee. The graph below shows the results for several of these tests. In these tests we have conducted, the lower limit for the constant fee was 0 and the higher limit was 300. As shown, once the deadline is relaxed over a negotiated schedule, the constant fee doesn't differ much over the negotiations. It only increases slightly. 20 is the minimum possible deadline on this problem set, hence the increased costs forces the constant fee down to zero in order to be able to balance the objective function ratios for both sides. Although the adopted network models and the mode selections doesn't differ much at higher deadline levels, a quantified benefit difference appears between the client and the contractor due to interest rate which affect both the client and the contractor, but at different levels. One other point is that as the deadline increases, the bargaining objective increases as well. This is due to increased flexibility on the schedule which lets the players to be able to choose cheaper activity modes.

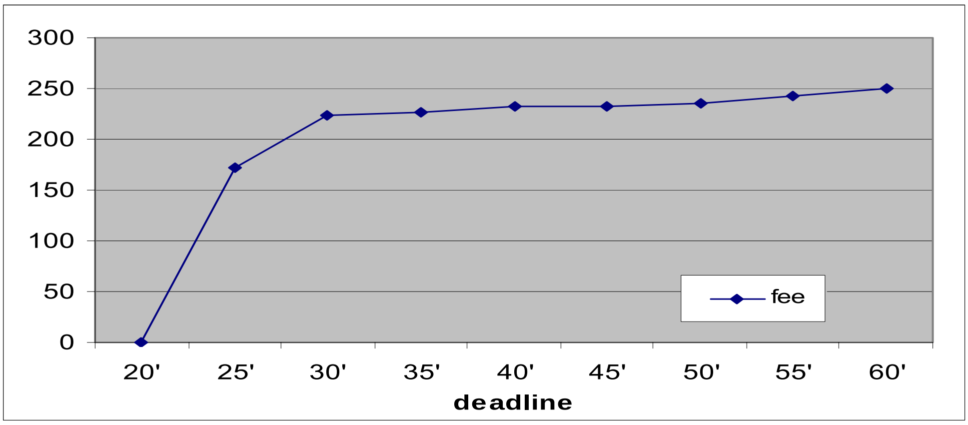

Figure 2.7 Constant deadline – negotiable fee

Figure 2.8 represents schedule examples for three different deadline settings.

D=40	activity list	1	4	3	2	5	6	10	8	7	13	11	9	12	14
	mode list	1	1	1	1	1	2	1	3	1	1	2	1	1	1
	idle time list	0	2	1	0	5	2	1	2	0	4	3	1	2	0

D=50	activity list	1	4	3	2	5	6	10	8	7	13	11	9	12	14
	mode list	1	1	1	1	1	2	1	3	1	1	2	1	1	1
	idle time list	0	12	1	0	5	2	1	2	0	4	3	1	2	0

D=60	activity list	1	4	3	2	5	6	10	8	7	13	11	9	12	14
	mode list	1	1	1	1	1	2	1	3	1	1	2	1	1	1
	idle time list	0	21	2	0	5	2	1	2	0	4	3	1	2	0

Figure 2.8 Schedule examples for different deadline settings

b. In the second set of tests we have fixed the constant fee and run the bargaining model to observe the trade-off on the deadline. Once we fixed the constant fee, through the negotiation process it became a given on the function, and appeared having no effect over the bargaining. Neither the schedule, nor the bargaining objective value changed as we changed the amount of the constant fee. The only thing it affected was the marginal increase in each player's NPV. And since this marginal change was the same in all the current schedule's objective function value, f''s and f's, no difference appeared in the final bargaining objective function value where only ratios are taken into account. The finish of the project has been postponed as much as possible, since by this way the NPV hurt of the activity costs are minimized. Once we include a profit margin for the contractor, schedule is shortened depending on the amount of β and α. If β provides a significant return on investment, the contractor tries to shorten the project time relying on its bargaining power.

c. In the third model, neither of the deadline and the constant fee were fixed; they were both negotiable, within certain ranges. This model also appeared to postpone the completion time of the project, as well as proposing activity modes which are much cheaper. By defining both the deadline and the constant fee as negotiable parameters, we kind of relaxed the problem and this brought better objective function values. In fact, the bargaining objective we get from this model is better than the previous ones. This is because we let more room for both the client and the contractor to look after their own benefits.

An illustration of the bargaining objective's move over the finishing time is as follows for three of these models:

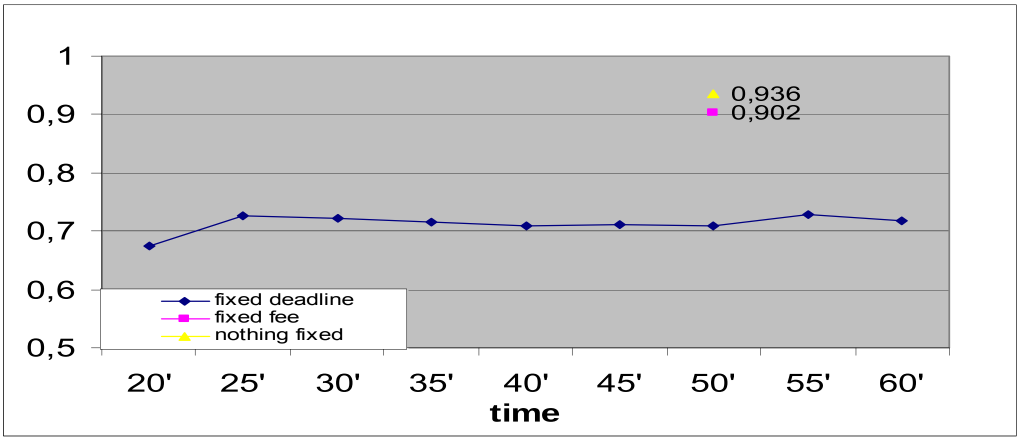

Figure 2.9 Bargaining objectives vs. finishing time

3. <u>Benefit for the client</u>: In this model, client receives a constant amount of benefit per each time period that the project is completed earlier than the deadline. In this model, contractor doesn't have any cost in benefit payments to the client. In a sense, this model proposes an additional benefit injected into the system regardless of any value trade off. In this model, we modify the base mathematical model by replacing client's objective function value that depends on the adopted schedule ($f_A(X)$) with the following function:

$$f_A(\mathbf{x}) = -(\sum_{t\in\tau} c_t * P_t) + (D- t^* \sum_{t=E_J}^{L_J}\sum_{m=1}^{M_j} t * x_{Jtm})*\sigma \qquad (2.10)$$

<u>Notation:</u>

σ: Client's benefit per time period

One of the most important outputs of this model is on adopted schedule. We have observed that as we increase the amount of benefit client receives the adopted schedule changes in a way that the project tends to finish earlier. This is due to the fact that, as the project is finished earlier the amount of extra value injected into the system increases, and both of the players gain benefit from this. We may observe in Table 2.10 the schedules for different benefit amounts per time period (σ).

Table 2.10 Schedules for different benefit amounts

	Activity	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
σ = 0	Finishing Time	0	14	15	24	20	20	20	24	25	25	30	30	30	30
	Mode	3	3	1	3	1	3	1	2	1	1	1	1	2	3
σ = 10	Finishing Time	0	10	5	20	15	15	20	24	24	20	30	30	30	30
	Mode	3	3	1	3	1	3	1	3	1	1	1	1	2	3
σ = 25	Finishing Time	0	10	5	10	15	14	19	20	20	24	24	24	24	24
	Mode	3	3	1	3	1	3	1	2	1	1	1	1	2	2
σ = 50	Finishing Time	0	5	10	19	15	15	19	20	20	20	24	24	24	24
	Mode	3	1	1	3	1	3	1	2	1	1	1	1	2	2
σ = 75	Finishing Time	0	5	9	19	15	14	19	20	20	20	23	24	24	24
	Mode	3	1	1	3	2	2	1	2	1	1	1	1	2	2
σ = 100	Finishing Time	0	3	6	14	11	10	15	16	16	16	20	20	20	20
	Mode	3	1	1	2	1	3	1	2	1	1	1	1	1	1
σ = 200	Finishing Time	0	3	6	5	11	10	15	16	16	16	20	20	20	20
	Mode	3	1	1	2	2	2	2	2	1	1	1	1	1	1
σ = 300	Finishing Time	0	3	6	5	11	10	15	16	16	16	20	20	20	20
	Mode	3	1	2	1	1	1	1	2	1	1	1	1	1	1
σ = 400	Finishing Time	0	3	6	15	11	10	15	16	16	16	19	20	20	20
	Mode	3	1	1	2	1	1	1	2	1	1	1	1	1	1
σ = 500	Finishing Time	0	3	6	15	11	10	15	16	16	16	19	20	20	20
	Mode	3	1	1	1	1	1	1	2	1	1	1	1	1	1

In the model we use, there are two parametric values for each player that defines the best and the worst possible solutions for the players. With benefit injection, the best possible absolute objective function value of the client has significant change although other parametric absolute objective function values are not affected. The amount of change is illustrated in Figure 2.10.

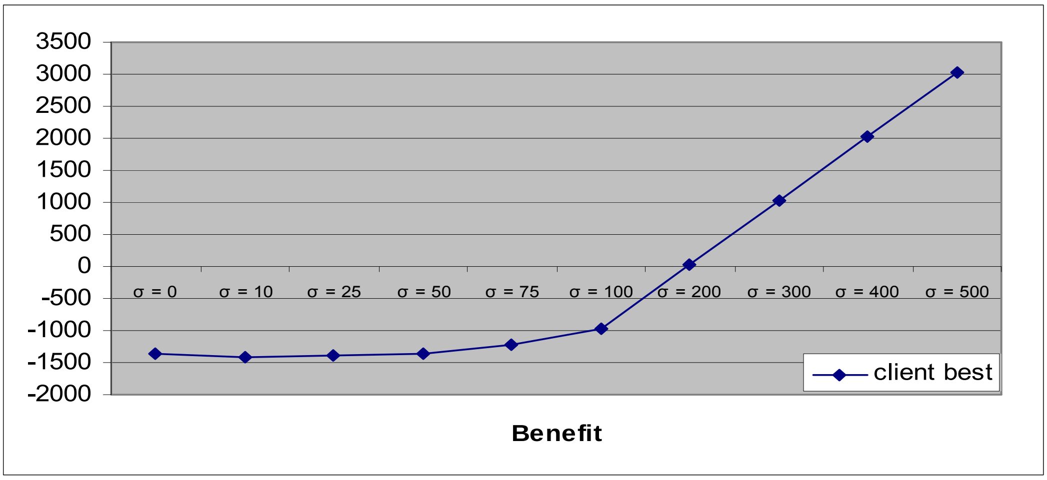

Figure 2.10 Best absolute OFV for the client

The key point we have concluded on these benefit tests is that both of the players gain quantitative value out of the benefit paid to the client. This leads increase in the final objective function as a result of with increased absolute objective values for both of the players. In Figures 2.11-2.15 we observe the amount of increase in objective function values of both players as the proposed benefit increases.

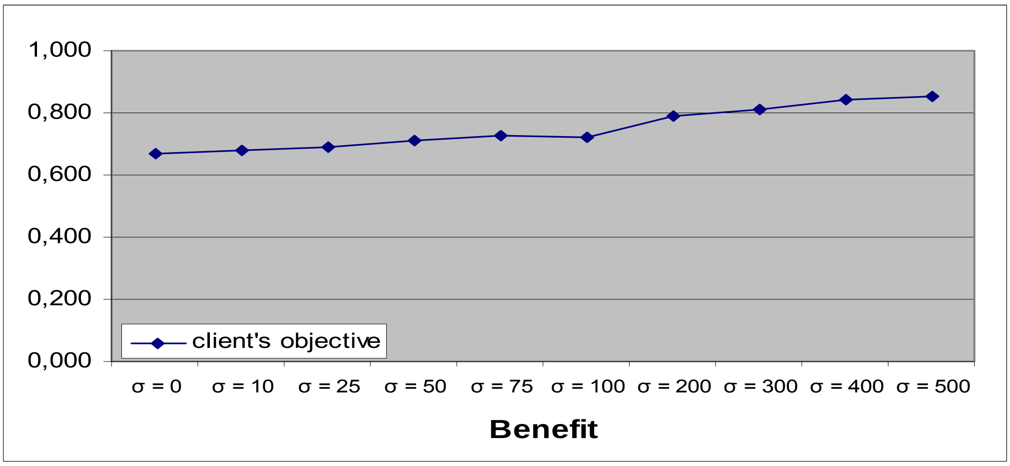

Figure 2.11 Client’s objective vs. client’s benefit

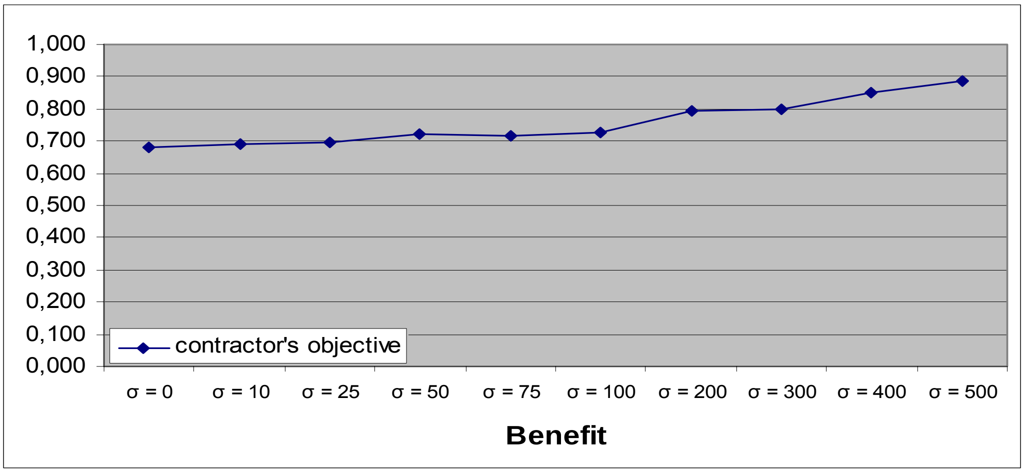

Figure 2.12 Contractor's objective vs. client's benefit

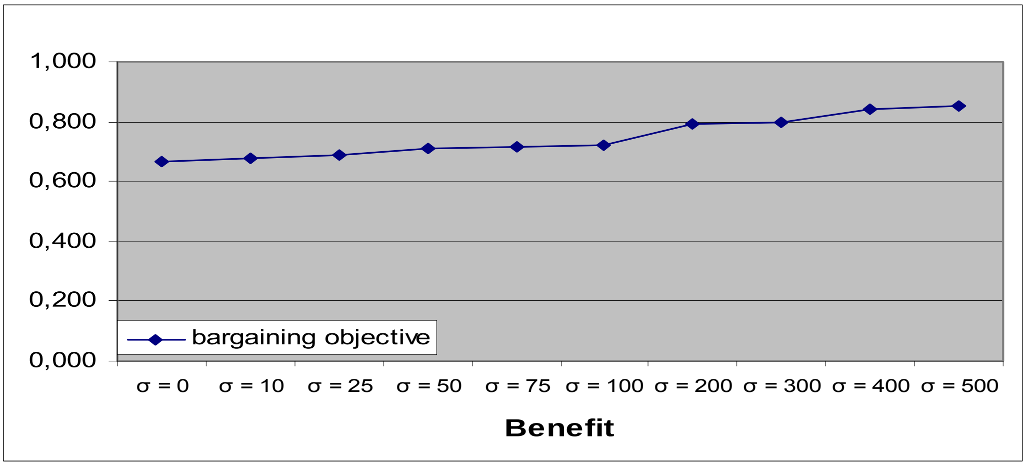

Figure 2.13 Bargaining objectives vs. client's benefit

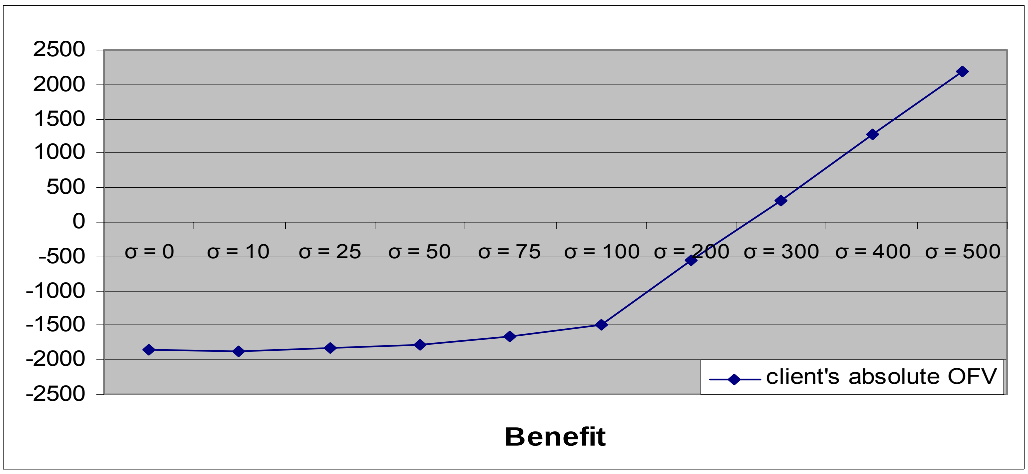

Figure 2.14 Client's absolute OFV vs. client's benefit

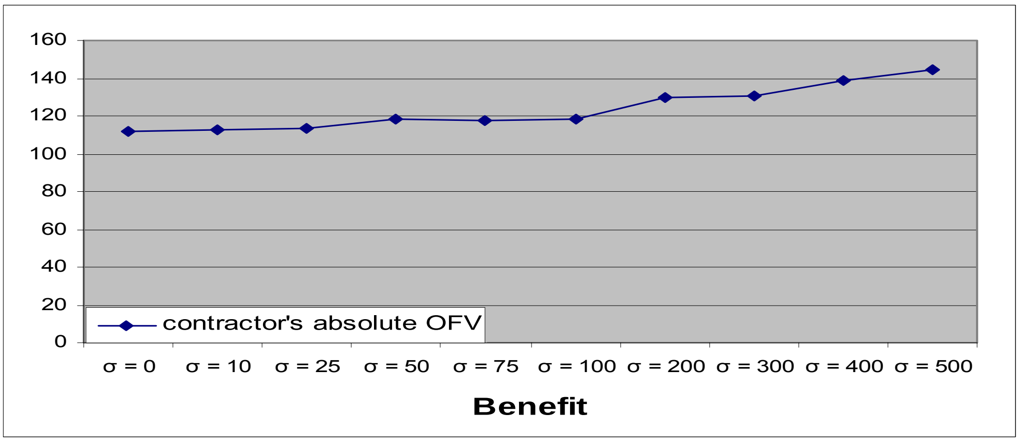

Figure 2.15 Contractor's absolute OFV vs. client's benefit

2.2.1 Structural Modifications That Would Be Proposed By the Client

In the model that we have used through the tests, the client pays either (cost + constant fee) or (cost + profit over cost). Since no specific benefit has been quantified for the client at the completion of the project, the client always has a negative NPV at the end. Therefore, objective of the client is to minimize the total cost and making sure that the project is completed by the deadline.

In the "cost + profit over cost" model, the client pays a profit percentage (β) over the total cost at the predetermined payment points. The number and amount of each payment (P_T) depends on the payment frequency, which is another parameter of the whole model. In this sense, the client prefers less frequent payments, which leads to bulk payments. By this way the payment for each specific activity may be deferred. In this model, the tests show that as the profit margin (β) is increased, the total amount client pays increases anyway. And since the model takes into account the ratio of difference between the adopted schedule and worst considered schedule, rather than the marginal difference, increase in β does not bring any additional bargaining power to the client. On the other hand, the NPV of the contractor is increased and the NPV of the client is decreased as if the client's bargaining power had been decreased. However, this realization in the NPVs does not affect the individual bargaining objectives of the players.

In the "cost + constant fee" model client pays a constant amount at the beginning of the project together with the activity costs paid at each predetermined payment periods. Since the ultimate objective of the client is to decrease its own costs, its preference is not having a constant fee at all, but once it is given, it doesn't affect client's decisions at all. That is a constant fee, free from its amount, doesn't change client's bargaining decision on the final schedule. The reason is that, once the fee is set as a cost for the client, the client adopts that cost both in its best and worst solutions. This lets all sets being even in terms of ratio.

In the model which incorporates an additional "benefit to the client", client always prefers increased benefits since he/she directly increases his/her objective function value as the benefit amount increases. One important point about this increase is that it is not linear on the benefit since client needs to share this value increase with the contractor up to an extent.

Among the sub payment models, client prefers the one which is more like a bulk payment. This tendency lets client promoting less payment frequencies. By this way, the client is able to postpone the payments for activities which have been already concluded.

One other preference of the client is to put activities on cheaper modes as much as possible. Once the deadline is met, client doesn't care much about the early finish of the project since it doesn't receive a benefit at the end of the project, which affects its NPV. This situation leads client to push the finish of the project towards the deadline. By this way, the client also postpones the payments. Once we include a quantified benefit for the client, for finishing the project earlier than the deadline, the client pushes the schedule by putting pressure on activities with smaller duration. This effort of the client totally depends on the amount of the benefit and its own bargaining power.

2.2.2 Structural Modifications That Would Be Proposed By the Contractor

There are two items in the contractor's balance sheet. One is the costs of the activities and the other one is the payments received from the client. The costs of the activities are paid at the beginning of the activities by default. On the other hand, the payments from the client may be received in several ways, either it comes as a "cost + constant fee" at the beginning of the project as an advance payment, or it comes as activity payments at predetermined payment points. If there is a profit margin defined for the contractor, it receives its payment at predetermined payment points as "cost + profit over cost".

In the "cost + profit over cost" model, contrary to the client, contractor prefers higher β values. By this way it can get more return on the costs it pays at the beginning of each activity. At payment periods, the contractor receives a payment which corresponds to the cost of the activities finished in that particular period plus a profit amount which is calculated over the total cost. In this sense, the contractor prefers more frequent payments, in order to be able to receive its return on investment as soon as possible. This leads the contractor to promote payments at activity completions. By this way, it may get its revenue without a long wait. This brings a trade-off between the profit margin (β) and the frequency of the payments. Tests show that, depending on both β and α, since increasing the profit margin or the payment frequency both bring an NPV change on the same direction, at a constant NPV point there is a trade-off between them. Although all these changes affect NPVs of the players, they don't have an affect on the bargaining powers of either side. The reason is that, once a specific parameter is defined in the model, it is pursued as given in all individual calculations (f''_A, f''_B, f'_A, and f'_B values), so even if we see the significant effect on the NPVs, we don't see any bargaining power effect in the decision making process.

In the "cost + constant fee" model, the contractor always prefers the maximum possible amount of the advance payment it receives from the client. When it is subject to negotiation over constant deadline, contractor shares the same preferences on the schedule with the client, which is deferring the project as soon as possible so that they both minimize their costs and pushing the negotiation over this constant fee. This situation brings that once a relaxed model at the cheapest activity modes is reached, the constant fee negotiation is directly linked with the α value and the deadline. And once the constant fee is fixed, the negotiated schedule is not affected by changes in the constant fee. The reason is that once it is defined as given, the individual calculations (f''_A, f''_B, f'_A, and f'_B values) are held accordingly, and this doesn't bring any difference in the bargaining objective function which fully relies on ratios. On the other hand, NPVs of both players are directly related with this constant fee.

In the model which incorporates an additional "benefit to the client", contractor also prefers increased benefit amounts as well as the client. This is due to the fact that as long as the additional value proposed in the whole system increases, bargaining dynamics enable contractor gain value from client's benefit increase.

If the contractor receives payments both as a constant fee and a predetermined amount of profit margin over total cost (β) there appears to be a clear negotiation over the constant fee and the schedule. That is, since there is a profit amount contractor may increase its NPV over the activity costs, the contractor prefers more costly activities. And at the same time the contractor always prefers a higher constant fee. However, the client's preferences are just the opposite. This brings a clear cut bargaining exchange over these two items. The tests show that once different profit margins are defined together with a constant fee, the negotiated project schedule is directly related to both of these parameters. Once the profit margin is decreased, contractor either pushes higher amounts of constant fee, or a much shorter schedule which consists of more expensive activities.

3 SENSITIVITY ANALYSES OF THE RESULTS

Three important parameters we have used in our base model are the profit margin, payment interest rate, and the bargaining power. We conduct several tests for several different models in order to be able to measure the absolute effect on the final result. The results we present here belong to a single model, which enables us to be able to compare the parameters with each other in different dimensions. For profit margin and interest rate tests, we have used exact optimals that we deliver by using commercial solver. For bargaining weight tests, we have used GA.

3.1 Sensitivity Analysis for Profit Margin (β)

Within the model profit margin (β) directly affects the amount of money the client pays to the contractor at each payment point.

$$P_{T_n} = (1+\beta) * \sum_{j \in J} \sum_{t=T_{n-1}+1}^{T_n} \sum_{m=1}^{M_j} N_{jm} * x_{jtm} \qquad T_n \in \mathbf{T},\ n>0 \qquad (3.1)$$

With the same schedule, increasing profit margin (β) would definitely increase the objective function of the contractor, and decrease the objective function of the client. And vice versa, decreasing the profit margin (β) would decrease the objective function of the contractor, and increase the objective function of the client within the same adopted schedule. But since schedule itself is also a variable in our model, we ran the model with different profit margin values in order to be able to observe the reaction of objective functions.

In Figures 3.1 – 3.3, examples can be found from a series of tests that show final objective function results for increasing profit margin (β).

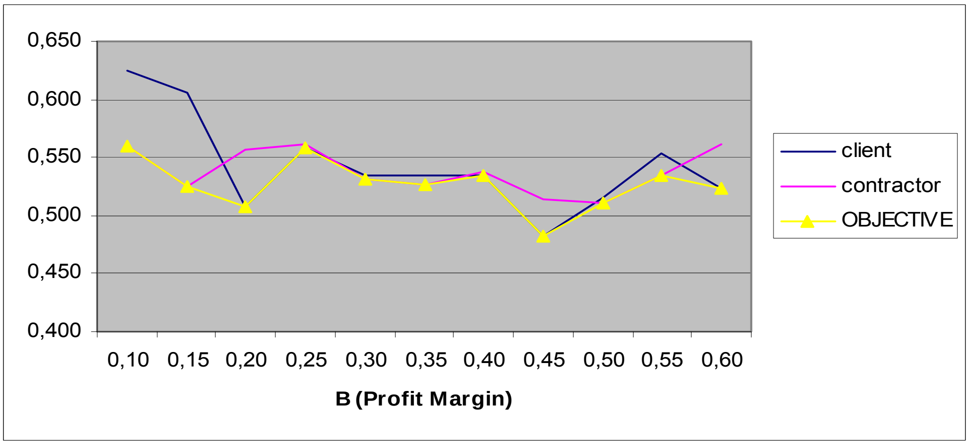

Figure 3.1 Bargaining objective at different profit margin levels (α=0,01)

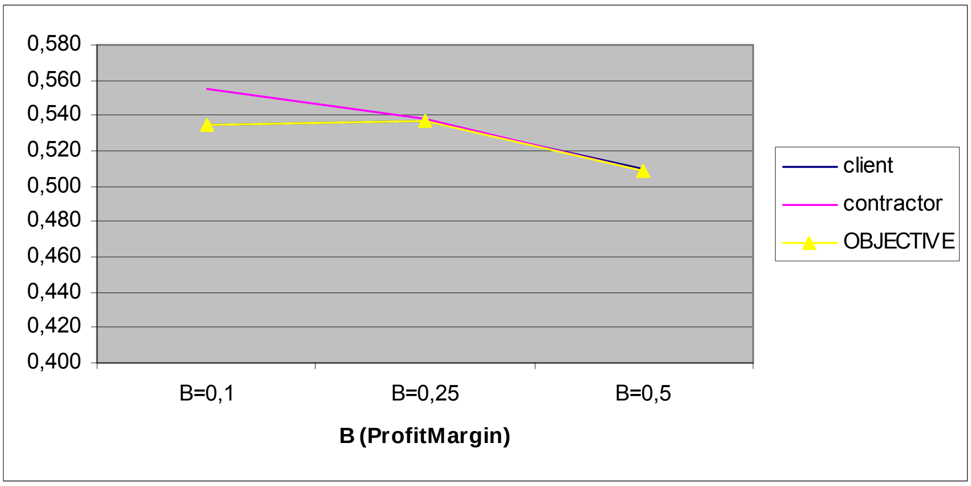

Figure 3.2 Bargaining objective at different profit margin levels (α=0,005)

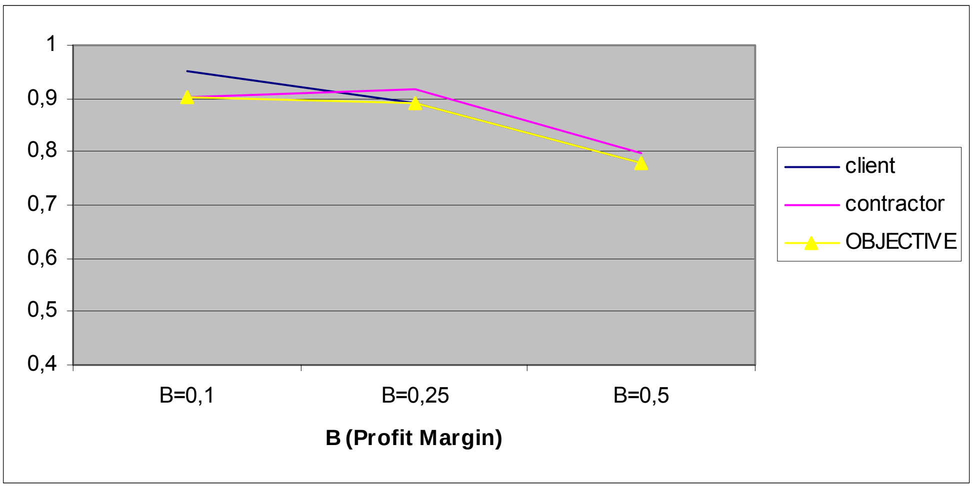

Figure 3.3 Bargaining objective at different profit margin levels (α=0,1)

Here we see that there is no correlation between the profit margin (β) and the final objective function value. The main reason for this is the direct effect of profit margin (β) on payments at each time module. Since variations of the payments in-between payment periods have significant effect on the adopted schedule, profit margin (β) has major effects on the final schedule.

Hence, contractor can not always keep his advantage on increased profit margin (β). We can observe this result from Table 3.1, which shows the owner of each final objective function. Since our model has a max min objective function, we can clearly say that the player who owns the final objective function is the one who had the inferior results in that specific game. Thus, the Table 3.1 shows the non-advantageous players for each case.

Table 3.1 Players who own the minimum objective function

(Interest Rate=0,01)	β =0,1	β =0,15	β =0,2	β =0,25	β =0,30	β =0,35	β =0,40
Minimum objective function belongs to:	Contractor	Contractor	Client	Client	Contractor	Contractor	Client

Here we see that, there is not a specific rule among players on determining the objective function. This solely results by client offsetting contractor's profit margin advantage by having major changes in the overall schedule.

Contrary to that, we see direct relation between profit margin and absolute objective function values of the players. Namely, as the profit margin increases, absolute objective function value of the client decreases and that of the contractor increases. For the problem set where interest rate is taken as 0,1, which is fairly high when compared to other tests, the beginning absolute values for both of the players (at Profit Margin=0,1) are close to each other, which indicates a trade-off between the profit rate and the interest rate. The details of the progression of absolute values for both players are shown in Figures 3.4-3.6 below:

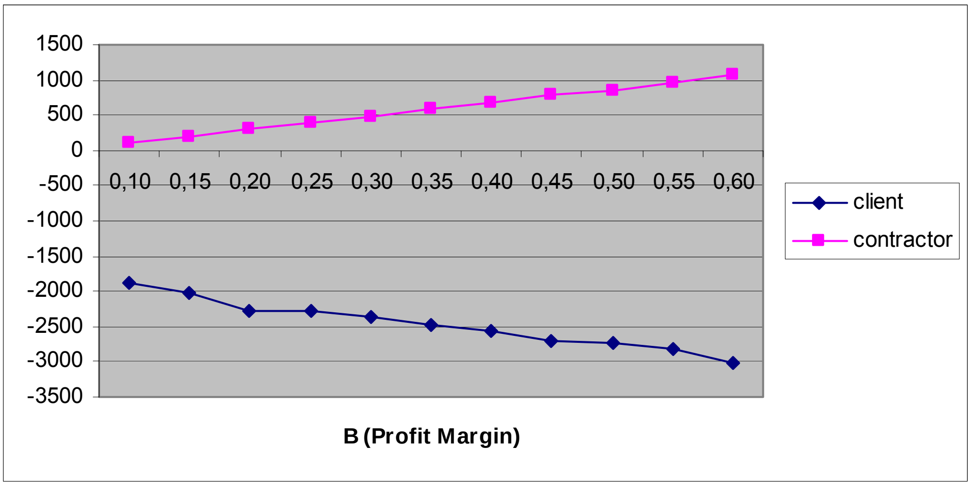

Figure 3.4 Absolute objective function values at different profit margin levels (α=0,01)

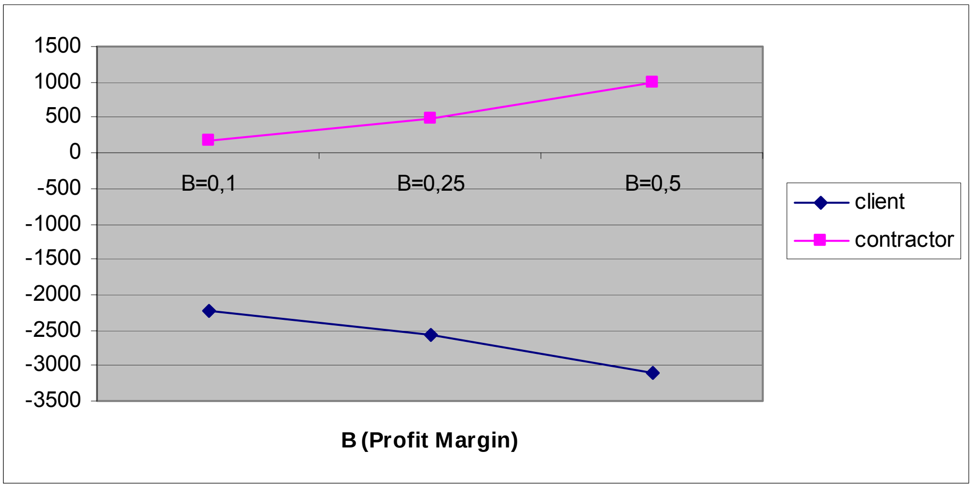

Figure 3.5 Absolute objective function values at different profit margin levels (α=0,005)

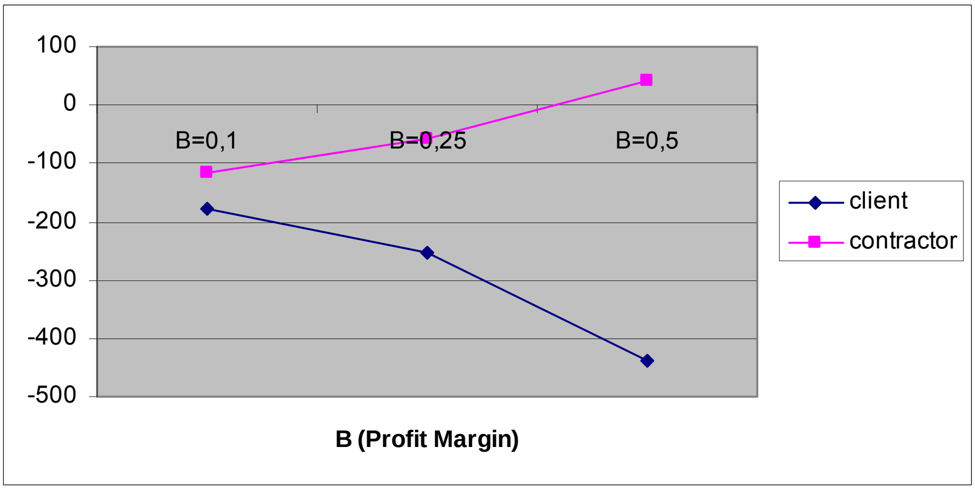

Figure 3.6 Absolute objective function values at different profit margin levels (α=0,1)

3.2 Sensitivity Analysis for Interest Rate (α)

Within the model Interest Rate (α) is used in NPV calculations within objective values of each player.

$$f_A(X) = -\left(\sum_{t \in T} c_t * P_t\right) \tag{3.1}$$

$$f_B(X) = \sum_{t \in T} c_t * P_t - \sum_{i \in J} \sum_{t=E_j}^{L_j} \sum_{m=1}^{M_j} N_{jm} * c_{t-dur(j)} * x_{jtm} \tag{3.2}$$

Interest Rate (α) may affect schedule preferences of the players due to different payment amounts at each payment point, and due to the fact that contractor pays activity costs in advance although he receives payments only at upcoming payment point. This shows that amount of Interest Rate (α) affects the contractor in two dimensions whereas it affects the client only in one. This brings the result that schedule manipulations due to Interest Rate (α) changes has more effect on the contractor's objective value than it has on the client's. Test results are presented in Figures 3.7-3.9.

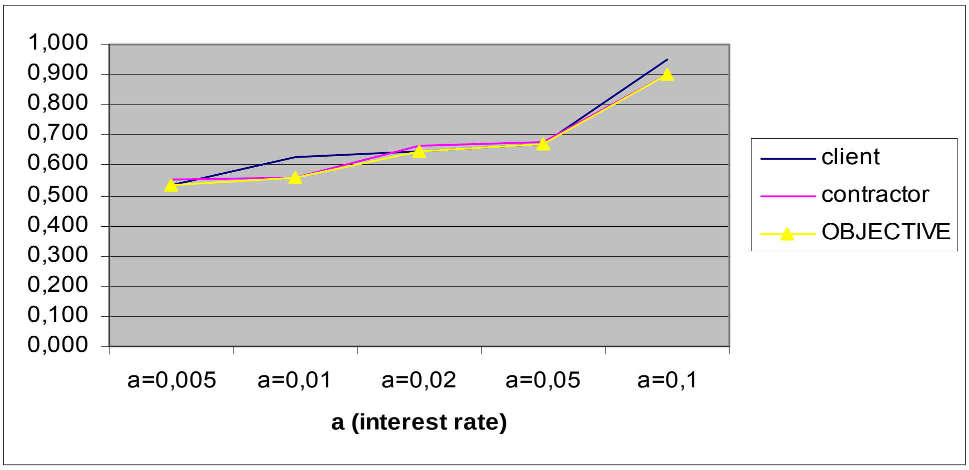

Figure 3.7 Bargaining objective at different interest rate levels (β=0,1)

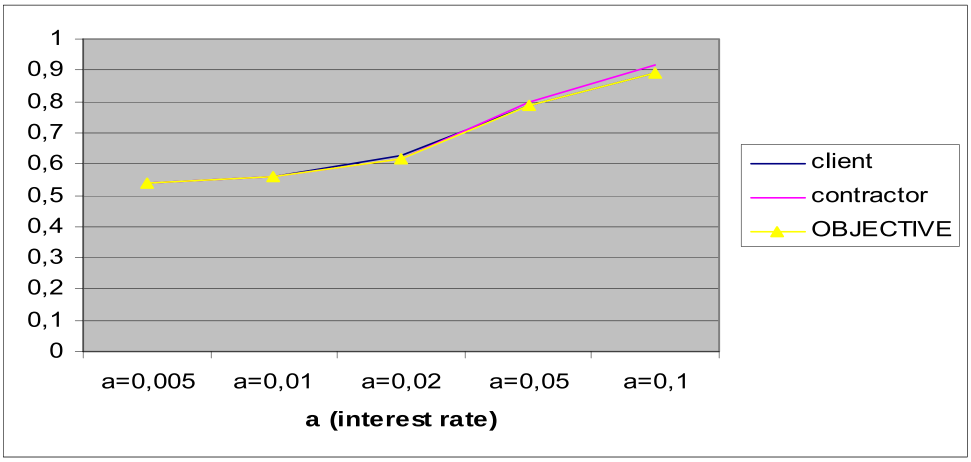

Figure 3.8 Bargaining objective at different interest rate levels (β=0,25)

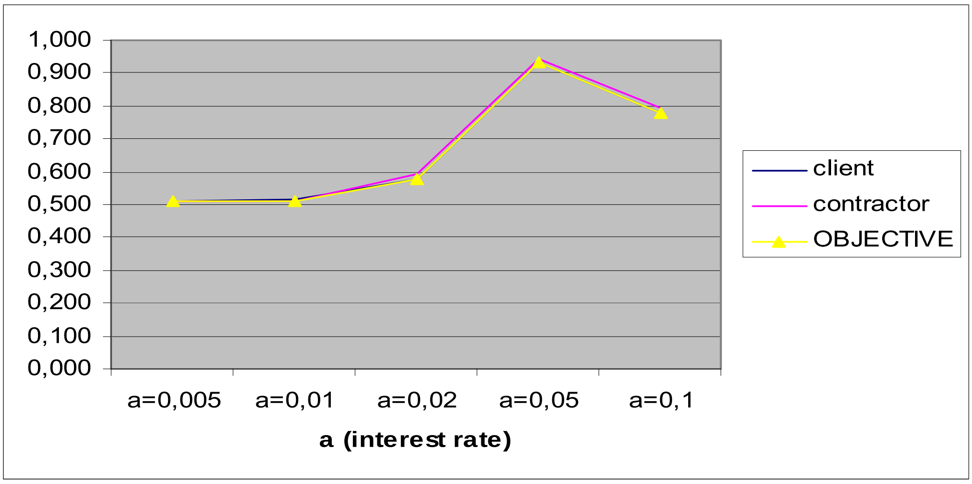

Figure 3.9 Bargaining objective at different interest rate levels (β=0,5)

Here, it is clearly seen that increasing interest rate had led to improved objective functions in most of the cases. Only in the last case where profit margin is set at high, we couldn't observe a monotonic increase in the objective function since the client does not have much room to improve since the highest possible absolute objective function value for him/her is equal to zero. The main reason for improved objective function in most of the cases is that as the interest rate increased in the model, individual objective function values for each player had been decreased due to decreased net realization. This led the model to introduce schedule improvements that both players benefit at the same time, to decrease the effects of the interest rate. Although these changes still brought inferior objective function values when compared with the results of models with higher interest rates, the percentage increase within the model itself had been higher for both players since they were looking for the same direction.

Increasing interest rate is a source of absolute objective function value increases for the client and just the opposite for the contractor. The details are given in Figures 3.10-3.12.

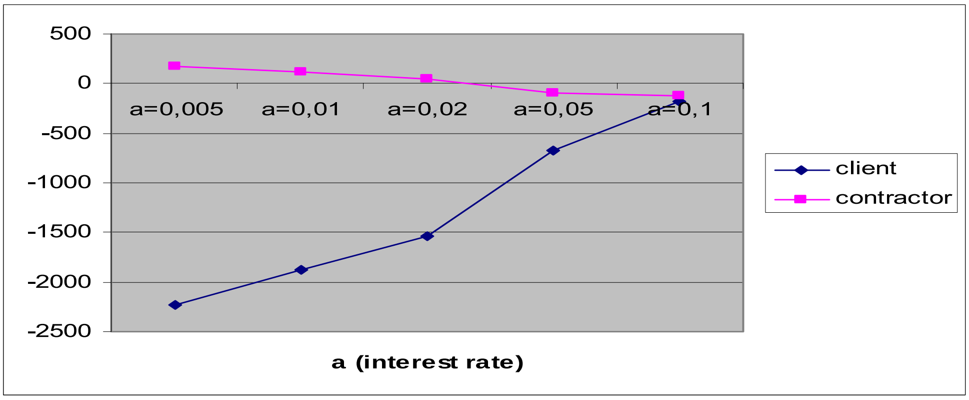

Figure 3.10 Absolute objective function values at different interest rate levels (β=0,1)

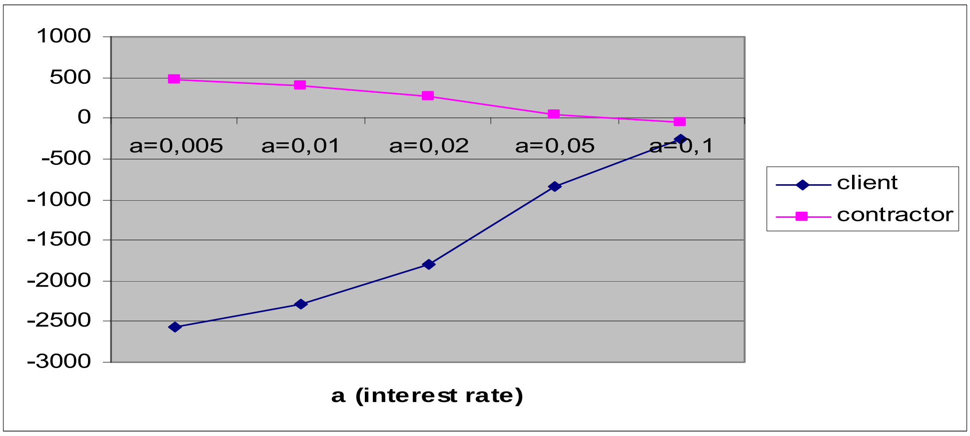

Figure 3.11 Absolute objective function values at different interest rate levels (β=0,25)

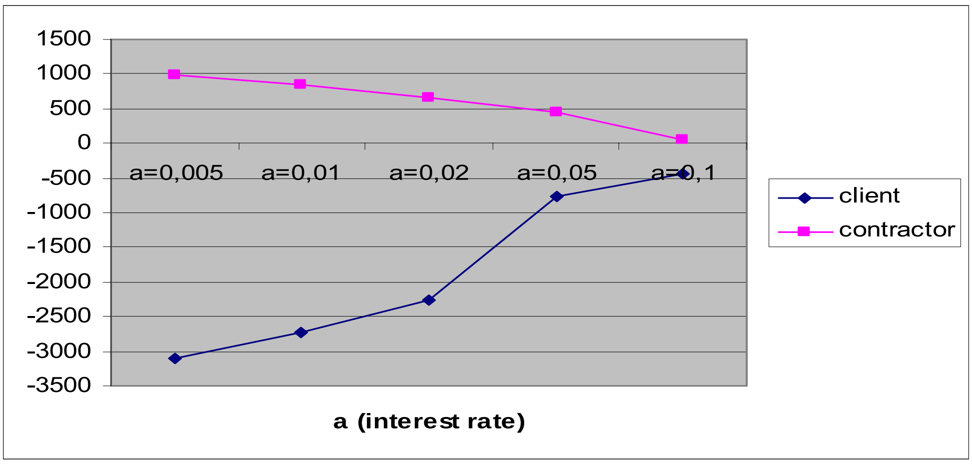

Figure 3.12 Absolute objective function values at different interest rate levels (β=0,5)

3.3 Sensitivity Analysis for Bargaining Power (w(A), w(B))

Within the model Bargaining Power for each player is used in the final step of objective function evaluation in order to set the exact realized value for the players:

$$Max\ min \left\{ \left(\frac{f_A(x) - f'_A}{f''_A - f'_A} \right)^{w(A)}, \left(\frac{f_B(x) - f'_B}{f''_B - f'_B} \right)^{w(B)} \right\} \qquad (3.3)$$

In our tests we have used bargaining powers that add up to 1, by this way we manage trade-off between power changes. As the bargaining power of a player increases, the actual objective value, which is the outcome of the adopted schedule, also increases. The summary of the weight tests we have conducted are given in Table 3.2:

Table 3.2 Summary of the weight tests

			Value	Objective FV	Realized Objective Value for Parties	Absolute OFV
w1	client	w(A)	0,1	0,186	0,845	-2478
	contractor	w(B)	0,9	0,830	0,846	144
w2	client	w(A)	0,2	0,306	0,789	-2314
	contractor	w(B)	0,8	0,761	0,803	136
w3	client	w(A)	0,3	0,410	0,765	-2173
	contractor	w(B)	0,7	0,677	0,761	126
w4	client	w(A)	0,4	0,442	0,721	-2129
	contractor	w(B)	0,6	0,581	0,722	115
w5	client	w(A)	0,5	0,515	0,718	-2030
	contractor	w(B)	0,5	0,526	0,725	109
w6	client	w(A)	0,6	0,578	0,720	-1943
	contractor	w(B)	0,4	0,456	0,731	101
w7	client	w(A)	0,7	0,672	0,757	-1815
	contractor	w(B)	0,3	0,391	0,755	93
w8	client	w(A)	0,8	0,708	0,759	-1766
	contractor	w(B)	0,2	0,238	0,751	76
w9	client	w(A)	0,9	0,810	0,827	-1628
	contractor	w(B)	0,1	0,100	0,794	60

Here, for the first five weight couples, client has smaller actual objective than the contractor, and for the last four weight couples situation is vice versa. This is solely led by the bargaining powers players have. Here the mechanism is that, for the player who has smaller bargaining power, the realized objective increases as we are taking power of a number which is <1 and >0, so that the other player who has higher bargaining power increases its objective by modifying the adopted schedule. Table 3.3 represents schedule examples at different weight points.

Table 3.3 Schedules at different weight points

w1	activity list	1	2	3	6	4	7	5	9	8	10	13	11	12	14
	mode list	1	1	1	1	1	1	1	1	1	2	1	2	1	1
	idle time list	0	2	0	4	3	2	0	0	3	1	2	0	2	0
w2	activity list	1	2	3	6	4	5	7	9	8	10	13	11	12	14
	mode list	1	1	1	1	1	1	1	1	1	2	2	2	1	1
	idle time list	0	2	0	4	3	2	0	0	3	1	0	0	2	0
w3	activity list	1	2	3	6	4	5	7	8	9	10	13	11	12	14
	mode list	1	1	1	1	1	2	1	1	1	2	2	2	1	1
	idle time list	0	2	0	4	3	1	0	0	3	1	0	0	2	0
w4	activity list	1	3	2	5	6	7	4	8	9	10	13	11	12	14
	mode list	1	1	1	2	1	2	3	1	1	1	2	1	1	1
	idle time list	0	0	2	1	4	0	0	0	3	2	0	0	2	0
w5	activity list	1	2	3	6	5	7	4	8	10	9	11	12	13	14
	mode list	1	1	1	1	2	1	3	1	1	2	2	1	2	1
	idle time list	0	2	0	4	1	0	0	0	2	1	0	2	0	0
w6	activity list	1	3	2	4	6	5	7	9	8	10	11	12	13	14
	mode list	1	1	1	3	3	1	1	2	2	1	1	1	1	1
	idle time list	0	0	2	0	1	2	0	1	0	1	2	1	1	0
w7	activity list	1	4	2	3	6	5	7	9	8	10	11	13	12	14
	mode list	1	1	1	3	1	3	2	1	1	3	1	1	1	1
	idle time list	0	3	1	0	4	0	0	3	1	0	1	1	0	0
w8	activity list	1	2	3	6	5	4	10	7	8	9	11	13	12	14
	mode list	1	1	2	3	1	3	1	2	3	2	2	2	3	1
	idle time list	0	1	1	1	2	0	1	0	0	1	0	0	0	0
w9	activity list	1	3	2	6	5	4	10	7	11	9	8	13	12	14
	mode list	1	2	2	3	1	3	1	2	3	2	3	2	3	1
	idle time list	0	0	0	1	2	0	1	0	0	1	0	0	0	0

In Figure 3.13 we observe the net realized bargaining objectives for each player at different weight couples. Here we see that, once the gap between bargaining powers of the parties increase, the realized objective values increase for both of the players. This due to the fact that, if one of the players has a small bargaining power, his/her realized objectives increases a lot, that the other party tries to catch this increase by schedule changes.

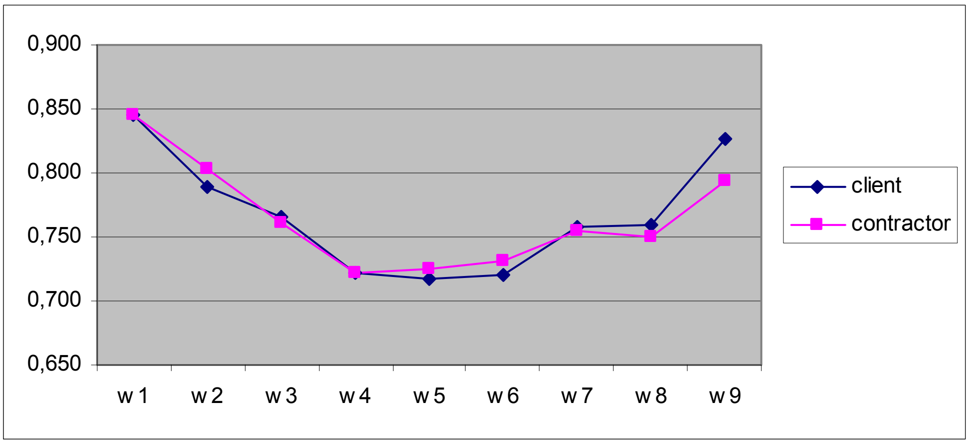

Figure 3.13 Bargaining objectives vs. bargaining power

When we look at the progression of actual objectives of the players in Figure 3.14, we observe the direct relationship between the bargaining power and the actual objective.

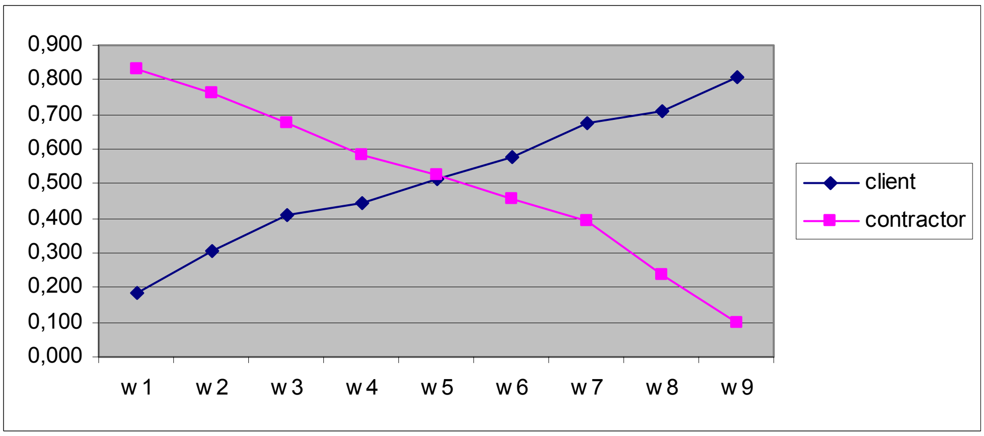

Figure 3.14 Actual objectives of the players vs. bargaining power

Absolute objective function value for the players follow a similar trend as the actual objective does with respect to bargaining power. In Figure 3.15, we clearly observe this trend, but since the range is defined with the absolute gap between the worst and best solutions of the players, in this figure it seems like that the client has a sharper trend than the contractor has. But this difference is totally driven by the gap between the worst and best case solutions, which are 1363 for the client and 114 for the client in this case.

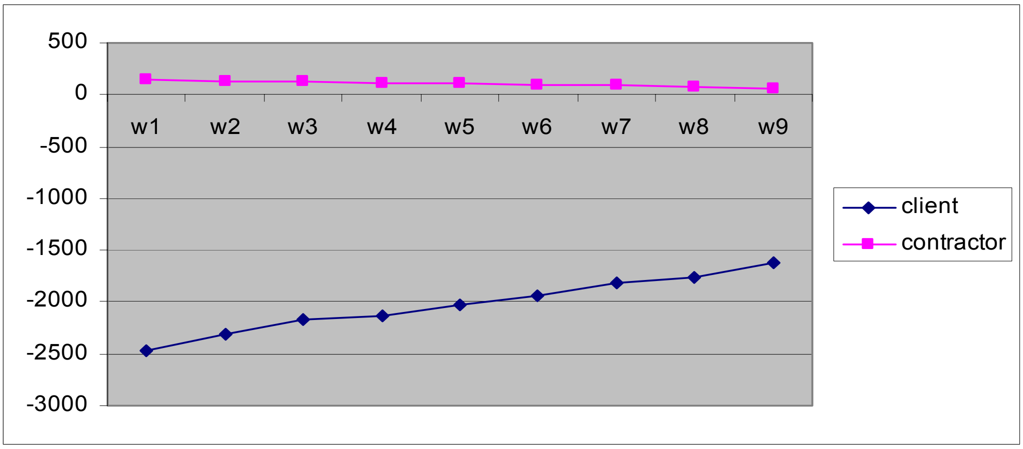

Figure 3.15 Absolute objective function values of the players vs. bargaining power

In terms of absolute objective function values of the players, we have observed six key conclusions (ceterus paribus):

- Absolute OFV of the client decreases as the profit margin increases
- Absolute OFV of the contractor increases as the profit margin increases
- Absolute OFV of the client increases as the Interest Rate increases
- Absolute OFV of the contractor decreases as the Interest Rate increases
- Absolute OFV of the client increases as his/her Bargaining Power increases
- Absolute OFV of the contractor increases as his/her Bargaining Power increases

This brings us to the conclusion that, during problem solving stage, which appears as the bargaining stage, the variables may substitute each other. For example, an agreement on increasing the profit margin may correspond to increasing the bargaining power of the contractor, or vice versa.

4 EXTENSION OF THE MODEL

In addition to the max min objective function, we also accomplished some tests with a second objective function formulation. We used GA in these tests.

4.1 Maximizing the Minimum Model

In this model, the objective function we use is:

$$\text{Max min} \left\{ \left(\frac{f_A(x) - f'_A}{f''_A - f'_A} \right)^{w(A)}, \left(\frac{f_B(x) - f'_B}{f'_B - f'_B} \right)^{w(B)} \right\} \tag{4.1}$$

Our ultimate aim is for each player, to maximize the distance from their worst solution point. In this first objective function equation; we get to achieve this by maximizing the minimum of these distance rates. For each player, we take the distance rates into consideration, not the absolute distance, since we need to find out the improvement rate, not the absolute gap between the current point and the worst point.

4.2 Maximizing the Multiplication Model

In this model, the objective function we use is:

$$\text{Max} \quad \left(\frac{f_A(x) - f'_A}{f''_A - f'_A} \right)^{w(A)} * \left(\frac{f_B(x) - f'_B}{f''_B - f'_B} \right)^{w(B)} \tag{4.2}$$

For this objective function, test results show that, the objective function values for each player is close to each other when we are working with low interest rates. This is inline with the fact that in low interest rate environments schedule changes in our model correspond to benefit exchange among players. Here we can refer to the common square rule, which tells that if we are trying to maximize the multiplication of two numbers those add up to a constant value, we better choose these two equal to each other. In this sense, this objective function does exactly the same: In order to be able to maximize the multiplication of distance rates, it’s wise to choose the distance rates close to each other. On the other hand, if the interest rate is high, schedule changes not necessarily lead to benefit exchange among players.

We observe the progression in two dimensions: one is by taking the interest rate as constant and gradually increasing the profit margin, and the other one is by taking the profit margin as constant, and gradually increasing the interest rate.

For each player and for each objective function we test progression of individual distance rate, which is a component of each objective function equation. In Figures 4.1 and 4.2, you may observe the individual objective function values for each player obtained at constant interest rate.

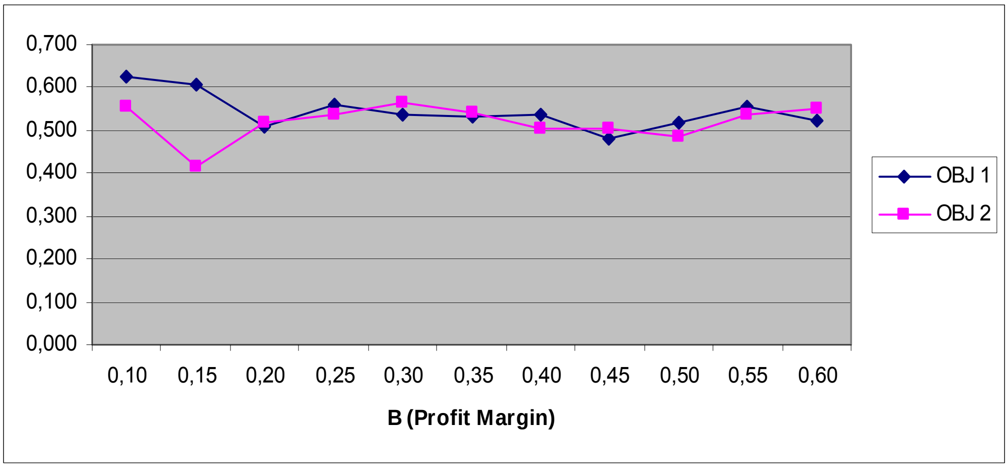

Figure 4.1 Client's OFV at different profit margin levels (α=0,01)

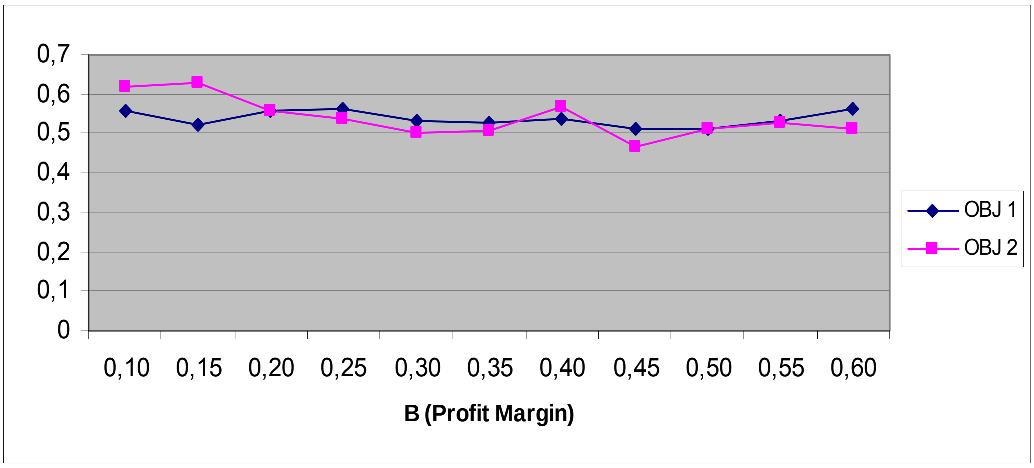

Figure 4.2 Contractor's OFV at different profit margin levels (α=0,01)

For the first objective function equation, although the client seems to start at a higher level than the contractor at high profit margin levels, actually for this first equation both of the players end up with similar objective function values. In the second objective function equation; we observe a similar reaction, only this time contractor seems to have inferior objective functions when compared to the client at lower profit margin levels. As the profit margin level increases, this gap erodes, and both contractor and client adopt similar objective function values. The main reason for this is that at lower profit margin levels, schedule changes have greater effect on distance rate for both the contractor and the client. Hence, same schedule changes may have greater effects on distance rates on lower profit margin levels. One other point we observe is that for both the client and the contractor, the distance rates which we quantify as objective function values stabilize in higher profit margin levels in both first and second objective functions equations.

The absolute value progression with respect to increasing profit margin for both of the players at various interest rate amounts is shown in Figures 4.3 – 4.10. Here we clearly observe that for both of the objective functions, the absolute value for the client decreases and the absolute value for the contractor increases as the profit margin increases. This comes as the trivial result of the fact that as the profit margin increases the net cost of the client increases, and the net profit of the contractor increases, free from the interest rate.

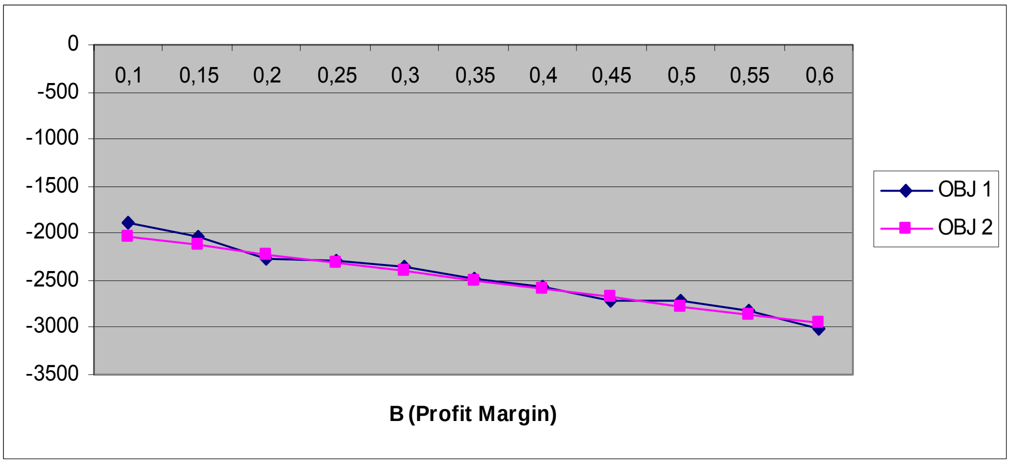

Figure 4.3 Client's absolute OFV at different profit margin levels (α=0,01)

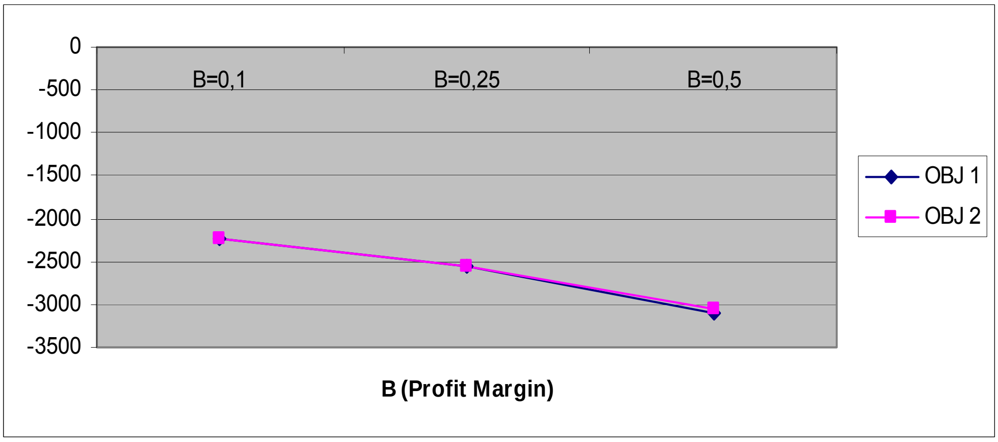

Figure 4.4 Client's absolute OFV at different profit margin levels (α=0,005)

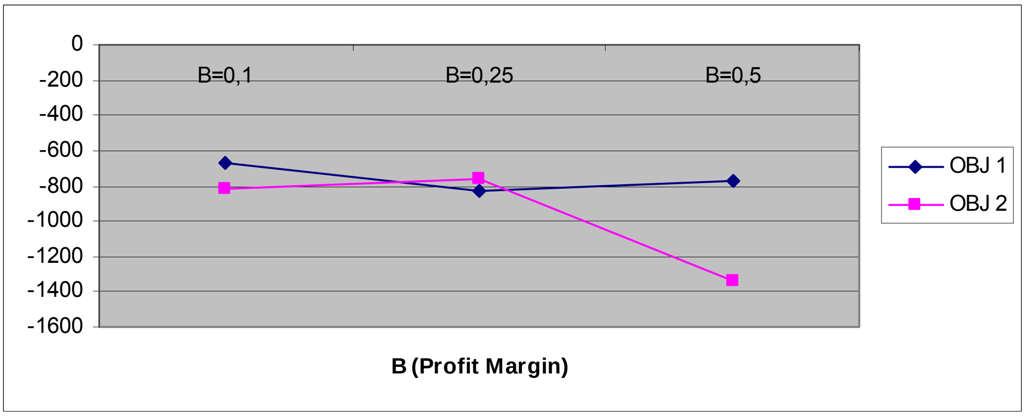

Figure 4.5 Client's absolute OFV at different profit margin levels (α=0,05)

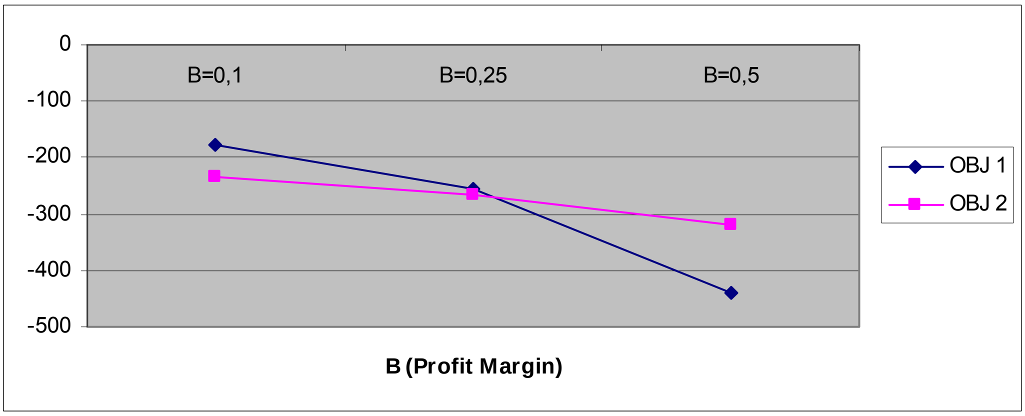

Figure 4.6 Client's absolute OFV at different profit margin levels (α=0,1)

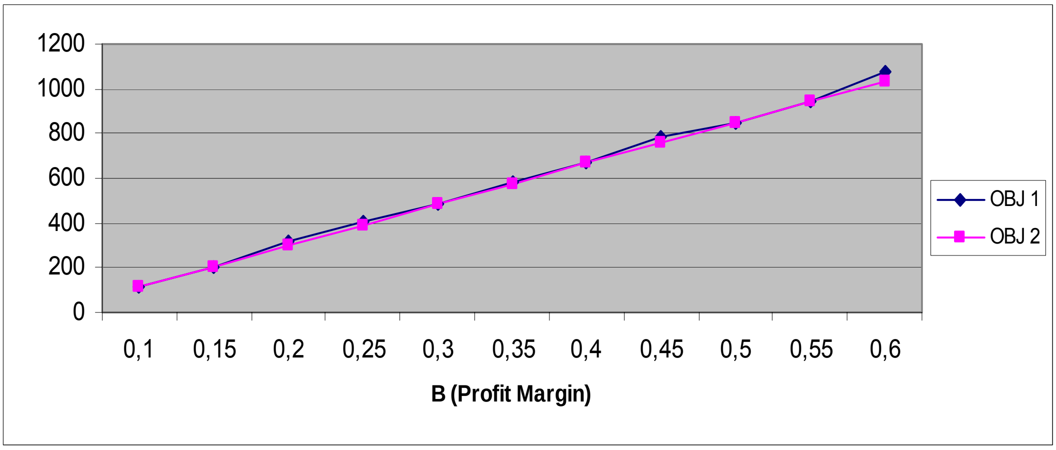

Figure 4.7 Contractor's absolute OFV at different profit margin levels (α=0,01)

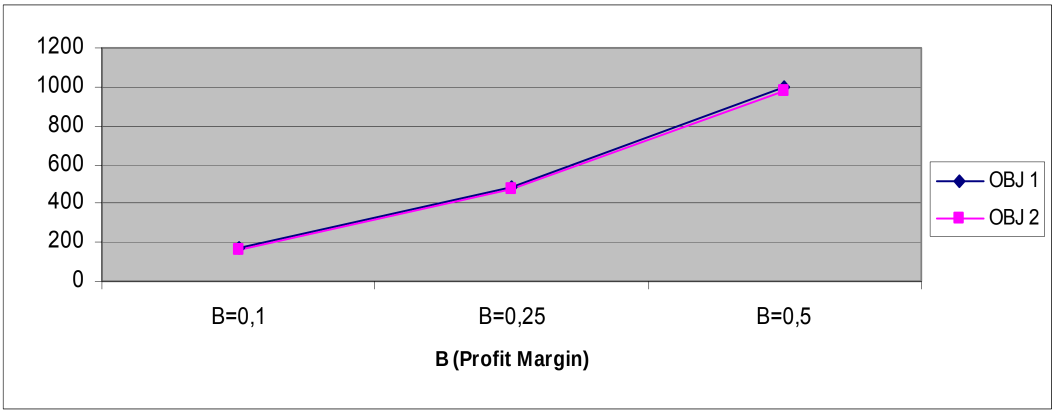

Figure 4.8 Contractor's absolute OFV at different profit margin levels (α=0,005)

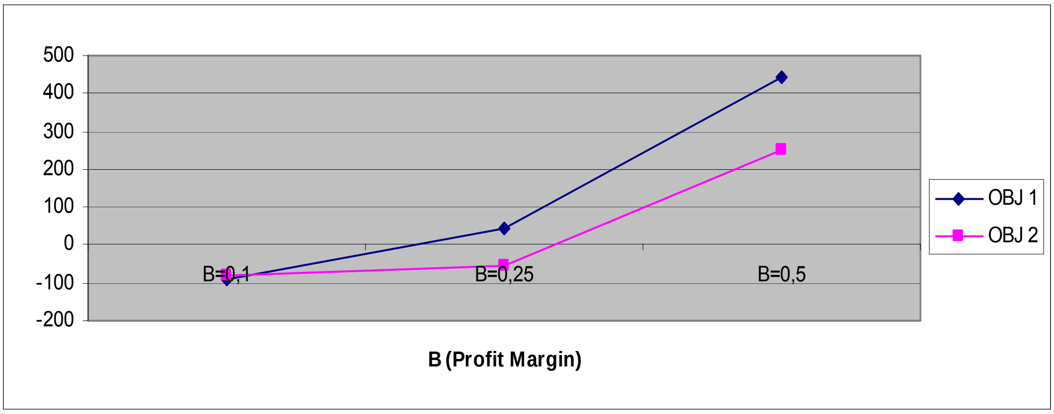

Figure 4.9 Contractor's absolute OFV at different profit margin levels (α=0,05)

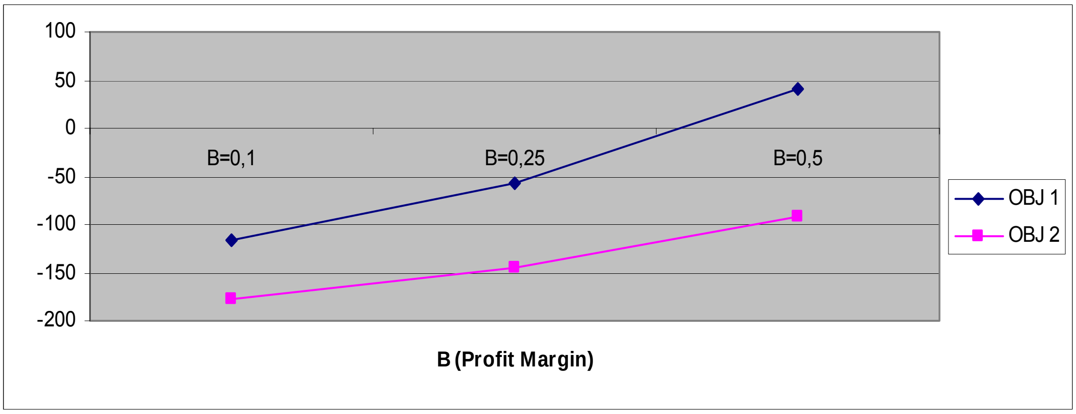

Figure 4.10 Contractor's absolute OFV at different profit margin levels (α=0,1)

The second set of graphs we examine consists of the progression of the individual objective function values for both the client and the contractor under constant profit margin and varying interest rate:

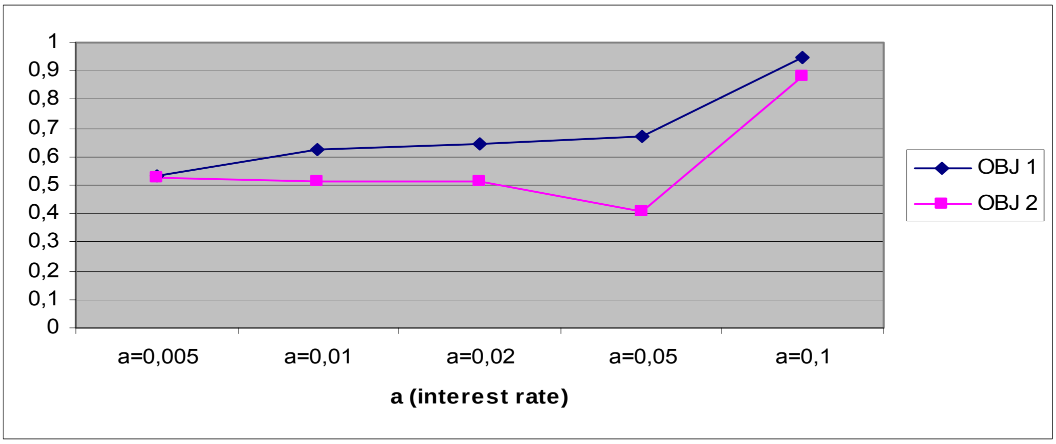

Figure 4.11 Client's OFV at different interest rate levels (β=0,1)

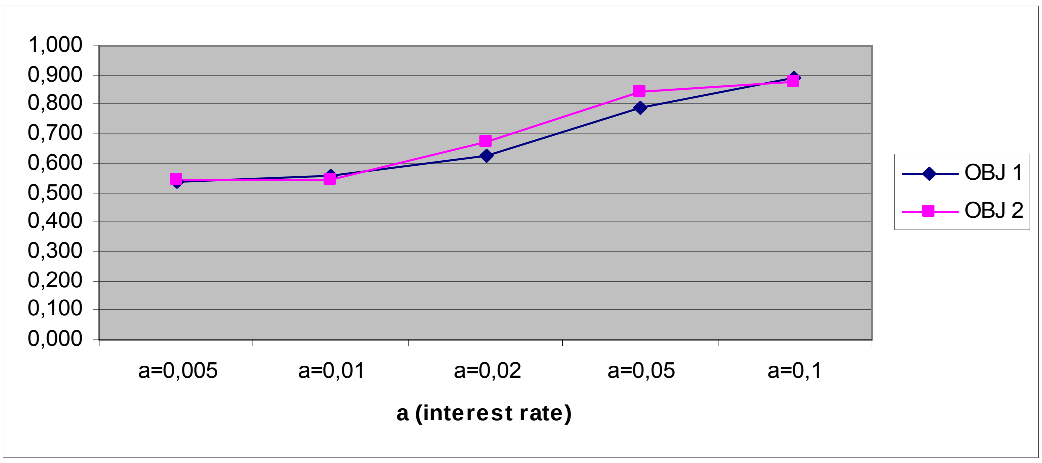

Figure 4.12 Client's OFV at different interest rate levels (β=0,25)

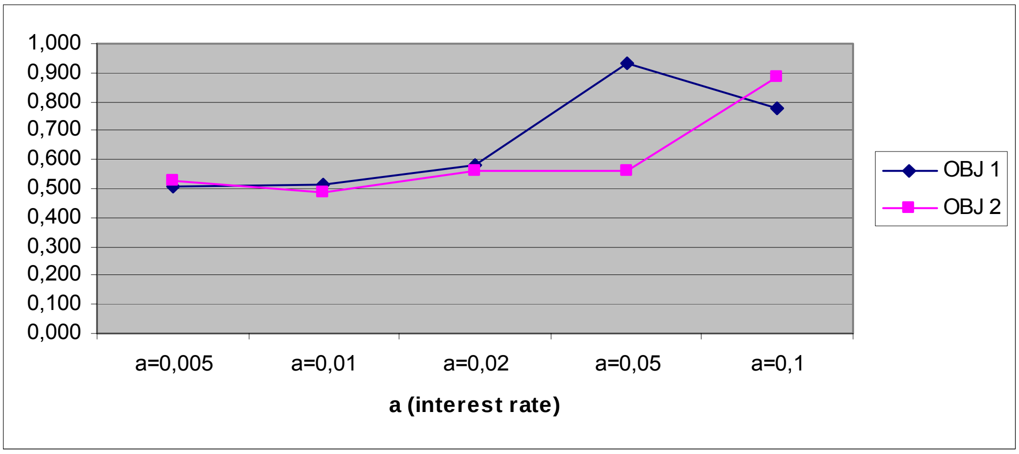

Figure 4.13 Client's OFV at different interest rate levels (β=0,5)

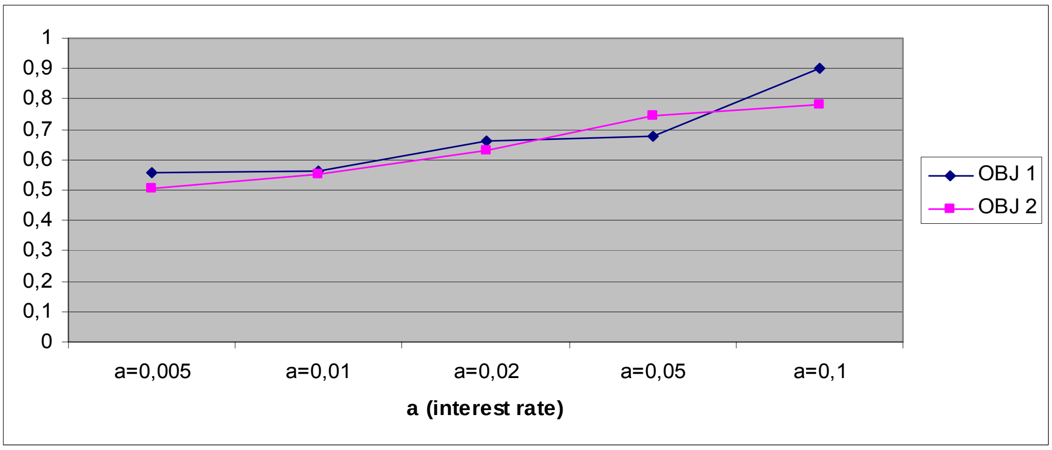

Figure 4.14 Contractor's OFV at different interest rate levels (β=0,1)

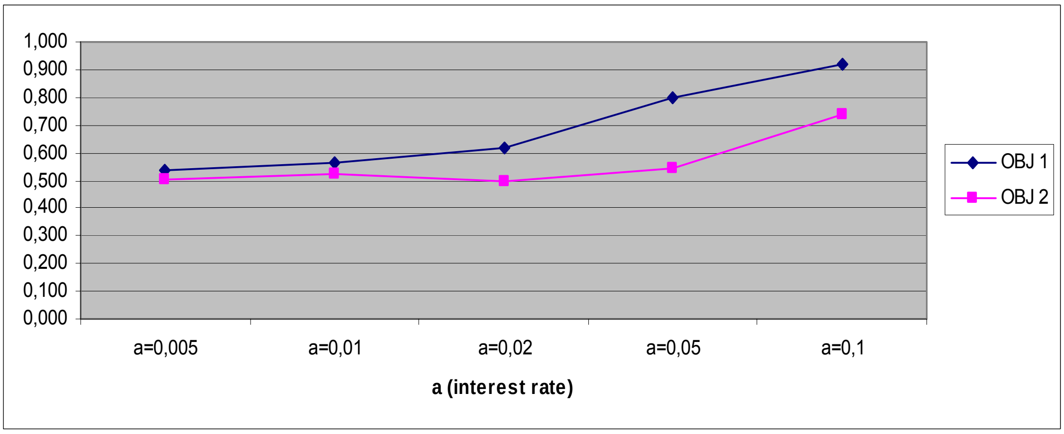

Figure 4.15 Contractor's OFV at different interest rate levels (β=0,25)

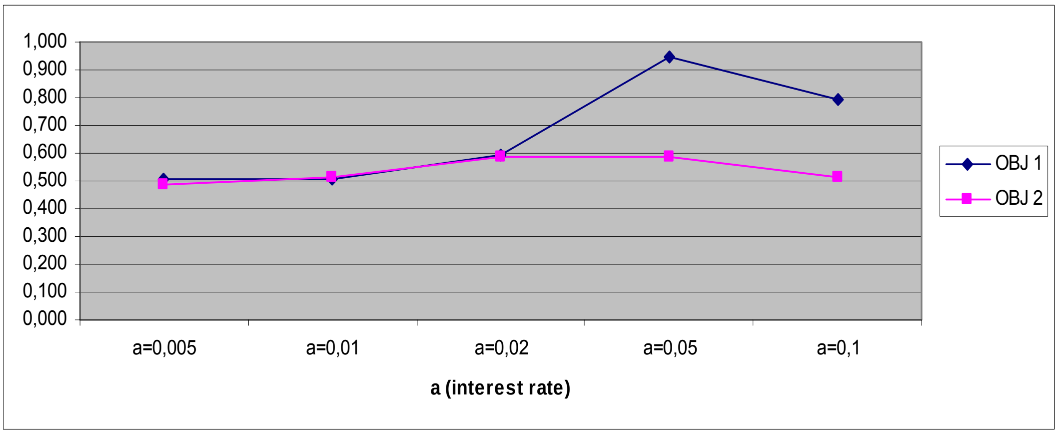

Figure 4.16 Contractor's OFV at different interest rate levels (β=0,5)

In Figures 4.11-4.16 we observe that for the first objective function equation, both the client and the contractor increase their objective function values as the interest rate increases. The reason for this progression is that since the interest rate increases, schedule changes has more dramatic effects on objective improvements, hence the distance rate improves for each player. On the other hand, in the second objective function we observe that as the interest rate increases, the objective function value for the contractor and the objective function value for the client doesn't necessarily follow a trend. This is totally driven by the equation dynamics of that specific equation. That is, since the objective function aims maximizing the multiplication of both objective functions, it does not create fair player values. This may lead to an optimal solution which builds a gap between player's realized benefits.

When we look at the progression of absolute objective function values of the players at different interest rate amounts in Figures 4.17 – 4.22, we clearly see that, for both objective functions, as the interest rate increases, the absolute value for the client increases and the absolute value for the contractor decreases. This results from the fact that increasing interest rate decreases the NPV of the contractor's profit as well as the NPV of the client's cost.

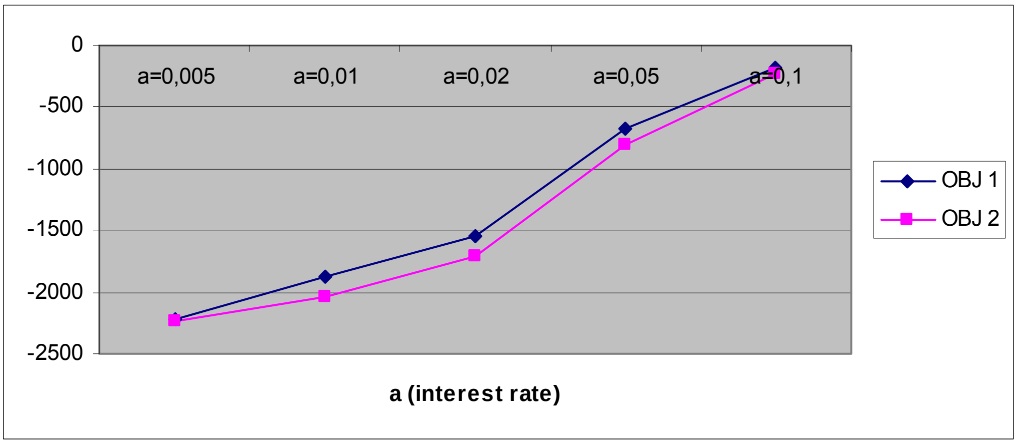

Figure 4.17 Client's absolute OFV at different interest rate levels (β=0,1)

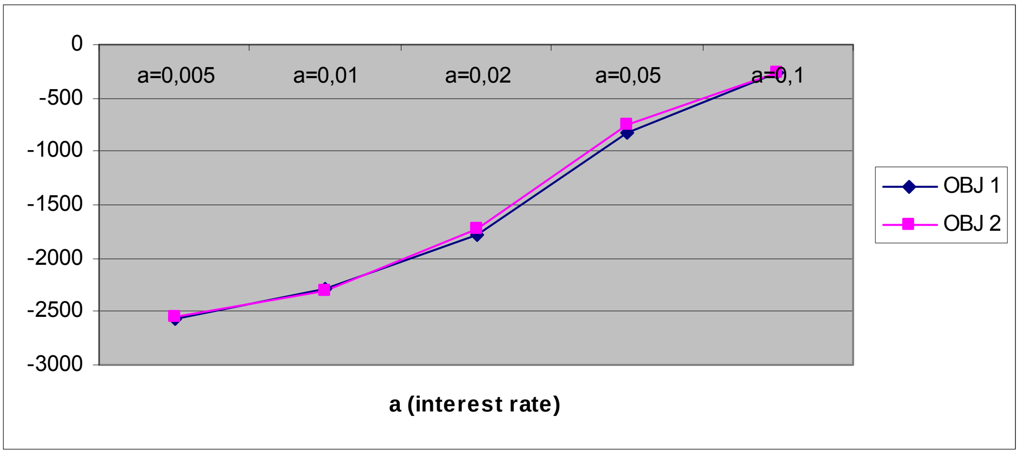

Figure 4.18 Client's absolute OFV at different interest rate levels (β=0,25)

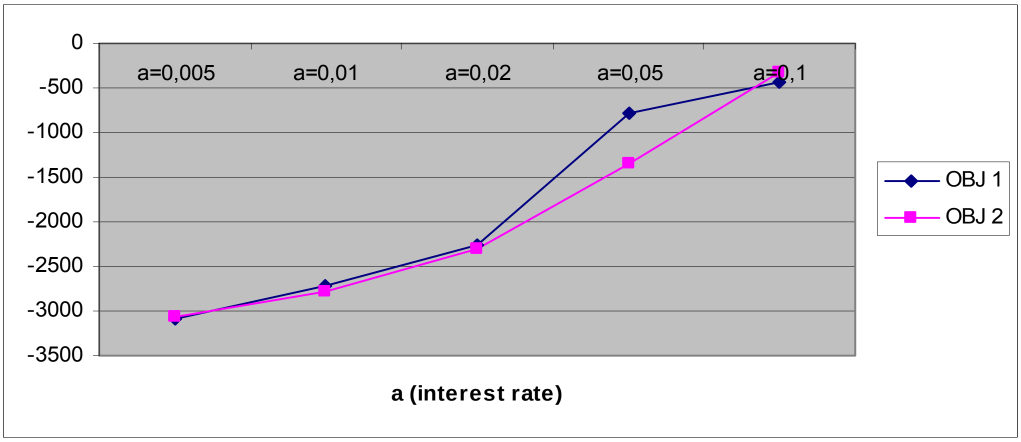

Figure 4.19 Client's absolute OFV at different interest rate levels (β=0,5)

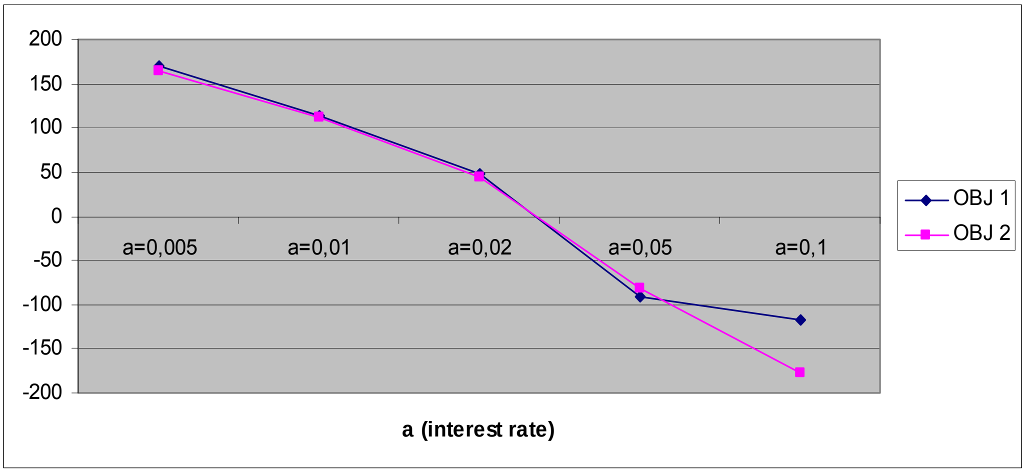

Figure 4.20 Contractor’s absolute OFV at different interest rate levels (β=0,1)

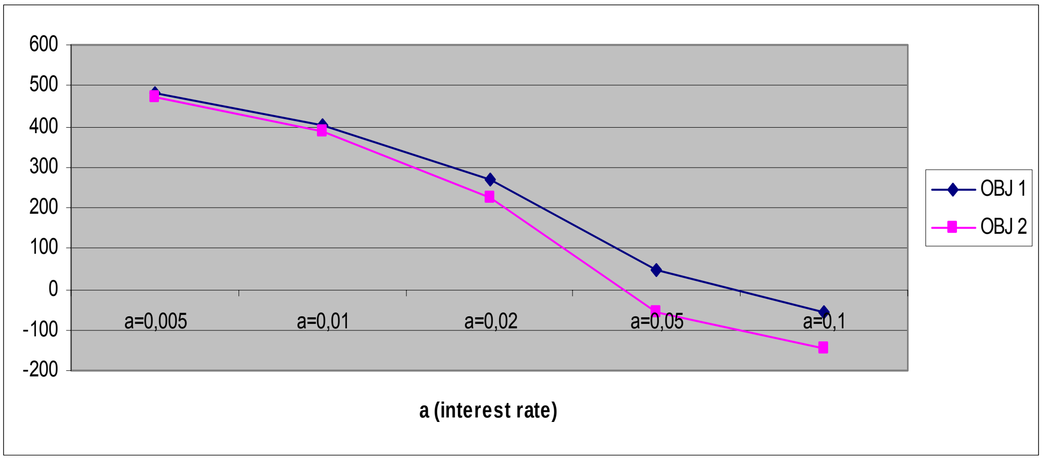

Figure 4.21 Contractor’s absolute OFV at different interest rate levels (β=0,25)

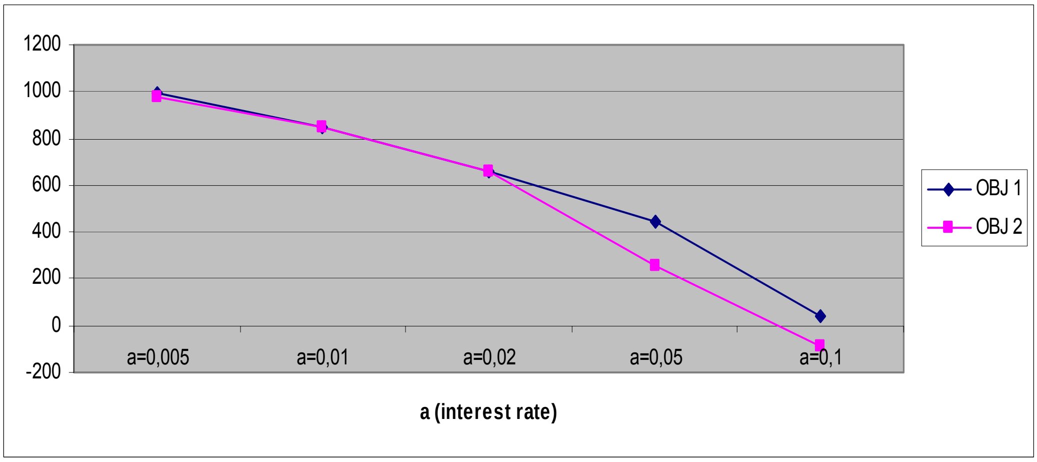

Figure 4.22 Contractor’s absolute OFV at different interest rate levels (β=0,5)

5 CONCLUSION

In this thesis, we have investigated the client-contractor bargaining problem in the context of multi-mode resource constrained project scheduling. The bargaining objective is to maximize the bargaining function comprised of the individual NPV maximizing objectives of both the client and the contractor.

In this context, we have proposed two main solution methods, namely Simulated Annealing Algorithm, and Genetic Algorithm. We have defined two different application methods for the Simulated Annealing, and on top of these we have introduced a solution search method by using non-dominated solutions set among the feasible set. Among these four solution methods, Genetic Algorithm had provided the best results when compared with the optimal results obtained by using a commercial solver (GAMS 20.0).

One important point in the solution procedure is that the representation of the model in the solution algorithm has significant effect on the solution. The activity list solution representation provides improved results in Simulated Annealing when compared with the activity finishing time list.

We have tested two main payment models to set the payment structure of the client to the contractor, cost + profit over cost, and cost + constant fee. Among these models, cost + constant fee model provided continuity in the problem and enabled both of the parties to end up with the same objective function value. In both of the models we have observed that although variations in the contractor benefits have effects on the absolute objective function values of both players, it does not have dramatic effects on the normalized objective functions and the schedule, since same changes are effective on reference values we use in the normalization.

When we introduce a benefit for the client for each time period the project is completed before deadline, we observe a significant increase on the absolute objective function values of both players. Bargaining objective improvements are also observed with increased benefit amounts. This indicates that this introduced extra benefit is negotiated within the system and hence both parties take share from it.

Sensitivity analyses results show that profit margin increase doesn't cause significant changes in the bargaining objective function, but it has huge effect on absolute objective functions of the players. As expected absolute objective function value of the contractor is increased with increased profit margin, and the absolute objective function value of the client is decreased. The tests we have conducted on interest rate showed that, although an increase in the bargaining objective function is observed with increasing interest rate, this is not a monotonic increase. However, interest rate variance has significant impact on the absolute objective functions of the players; the absolute objective function value of the contractor is decreased with increased profit margin, and the absolute objective function value of the client is increased. Weight tests we have implemented showed that bargaining weights have significant impact on the solution, not only on the absolute values of the players but also on the objective functions of the players.

We have also tested a second objective function formulation besides our basic formulation, which is a max min function. The second formulation introduced is a maximization of a multiplication function. The test results showed that although the results were compatible for both of the objective functions, the first objective function provided better results at extremes.

5.1 Further Research

Although various analyses with the parameters have been conducted with the solution parameters, still a series of combinations is open for research through the proposed model. The effect of bargaining weights on different payment models may be an important source of analyses through further investigation of the problem. Different payment models may also be investigated more thoroughly with different solution procedures. On top of these combining the non-dominated solutions set procedure with metaheuristic algorithms may provide a source of further research. Also, the non-dominated solutions curve may be used to identify acting-optimal solutions for the problem sets that we don't have any optimal solutions to compare the solutions we have generated with our metaheuristic algorithms. For example, in bargaining weight tests with higher activity numbers commercial solver we have used (GAMS 20.0) hadn't delivered optimal solutions, so that we may use this curve's equation to generate acting-optimal solutions. By this way, a benchmark may be set by using the curve fit to the non-dominated solutions set.

REFERENCES

[1] Baroum, S.M. and J.H. Patterson (1996). The development of cash flow weight procedures for maximizing the net present value of a project. *Journal of Operations Management*, 14:209-227.

[2] Baroum, S.M. and J.H. Patterson (1999). An exact solution procedure for maximizing the net present value of a project. In: *Handbook on Recent Advances in Project Scheduling*, ed. J. Weglarz, Kluwer Academic, Dordrecht, 107-134.

[3] Baykasoglu, A., Gindy N.N.Z. and R.C. Cobb (2001). Capability based formulation and solution of multiple objective cell formation problems using simulated annealing. *Integrated Manufacturing Systems*, 12: 258-274.

[4] Bey, R.B., Doersh, R.H. and J.H. Patterson (1981). The net present value criterion: its impact on project scheduling. *Project Management Quarterly*, 12: 35-45.

[5] Dayanand, N. and R. Padman (1993). Payments in Projects: A contractor's model. *The Heinz School*, Carnegie Mellon University, Pittsburgh, PA, Working Paper, 93-71.

[6] Dayanand, N. and R. Padman (1997). On modeling payments in projects. *Journal of the Operational Research Society*, 48, 906-918.

[7] Dayanand, N. and R. Padman (1998). Project contracts and payment schedules: The client's problem. *The Heinz School*, Carnegie Mellon University, Pittsburgh, PA, Working Paper, 95-23.

[8] Dayanand, N. and R. Padman (1999). On payment schedules in contractor client negotiations in projects: An overview of the problem and research issues. In: *Handbook on Recent Advances in Project Scheduling*, ed. J. Weglarz, Kluwer Academic, Dordrecht, 477-508.

[9] Demeulemeester, E., Vanhoucke, M. and W. Herroelen (2000). On maximizing the net present value of a project under renewable resource constraints. *Operations Management Group*, *Department of Applied Economics*, Katholieke Universiteit Leuven, Working Paper.

[10] Doersch, R.H. and J.H. Patterson (1977). Scheduling a project to maximize its present value: a zero one programming approach. *Management Science*, 23: 882-889.

[11] Ervig, U. and Haake, C.J. (2005). Trading bargaining weights. *Journal of Mathematical Economics*, 41: 983-993.

[12] Etgar, R., Shtub, A. and L.J. Leblanc (1996). Scheduling projects to maximize net present value – the case of time-dependent, contingent cash flows. *European Journal of Operational Research,* 96: 90-96.

[13] Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Pub. Co., Boston.

[14] Grinold, R.C. (1972). The payment scheduling problem. *Naval Research Logistics Quarterly*, 9: 123-136.

[15] Herroelen, W.S., Dommelen, P. and E.L. Demeulemeester (1997). Project network models with discounted cash flows: A guided tour through recent developments. *European Journal of Operational Research*, 100: 97-121.

[16] İçmeli, O. and S.S. Erengüç (1996). A branch-and-bound procedure for the resource-constrained project scheduling with discounted cash flows. *Management Science*, 42: 1395-1408.

[17] Kazaz, B. and C. Sepil (1996). Project scheduling with discounted cash flows and progress payments. *Journal of the Operational Research Society*, 47: 1262-1272.

[18] Kimms, A. (2001a). *Mathematical Programming and Financial Objectives for Scheduling Projects*, Kluwer Academic Publishers, Dordrecht.

[19] Kimms, A. (2001b). Maximizing the net present value of a project under resource constraints using a Lagrangian relaxation based heuristic with tight upper bounds. *Annals of Operations Research*, 102: 221-236.

[20] Köbberling, V. and Peters, H. (2003). The effect of decision weights in bargaining problem. *Journal of Economic Theory*, 110: 154-175.

[21] Kolisch, R and Sprecher (1996). PSPLIB - A project scheduling problem library. *Instituten für Betriebswirtschaftslehre der Universitat Kiel*, Kiel University, Kiel, Germany, Unpublished manuscript.

[22] Kolisch, R. and R. Padman (2001) An integrated survey of deterministic project scheduling. *OMEGA*, 29: 249-272.

[23] Mármol, A.M., Monroy, L. and V. Rubiales (2005). An equitable solution for multicriteria bargaining games. *European Journal of Operational Research*, in press.

[24] Mika, M. , Waligora, G. and J. Weglarz (2004). Simulated annealing and tabu search for multi-mode resource constrained project scheduling with positive cash flows and different payment models. *European Journal of Operational Research*, in press.

[25] Padman, R. and D.E. Smith-Daniels (1993). Early-tardy cost trade-offs in resource constrained projects with cash flows: an optimization guided heuristic approach. *European - Journal of Operations Research*, 64: 295-311.

[26] Padman, R., Smith-Daniels, D.E. and V.L. Smith-Daniels (1997). Heuristic scheduling of resource constrained projects with cash flows. *Naval Research Logistics*, 44: 365-381.

[27] Russell, A.H. (1970). Cash flows in networks. *Management Science*, 16: 357-373.

[28] Russell, R.A. (1986). A comparison of heuristics for scheduling projects with cash flows and resource restrictions. *Management Science*, 28: 1291-1300.

[29] Sepil, C. and N. Ortac (1997). Performance of the heuristic procedures for constrained projects with progress payments. *Journal of the Operational Research Society*, 48: 1123-1130.

[30] Sivrikaya-Şerifoğlu, F. (1997). A New Uniform Order-Based Crossover Operator for Genetic Algorithm Applications to Multi-component Combinatorial Optimization Problems. *Department of Industrial Engineering,* Boğaziçi University, PhD Thesis.

[31] Sivrikaya-Şerifoğlu, F. and İ. Tiryaki (2002). Multiprocessor task scheduling in multi-stage hybrid flowshops: A simulated annealing approach. *Proceedings of the 2nd International Conference on Responsive Manufacturing,* Gaziantep, Türkiye, 270-274.

[32] Smith-Daniels, D.E. and N.J. Aquilano (1987). Using a late start resource constrained project schedule to improve project net present value. *Decision Sciences*, 18: 617-630.

[33] Ulusoy, G. and S. Cebelli (2000). An equitable approach to the payment scheduling problem in project management. *European Journal of Operational Research*, 127: 262-278.

[34] Ulusoy, G., Sivrikaya-Şerifoğlu, F. and Ş. Şahin (2001). Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows. *Annals of Operations Research*, 102: 237-261.

[35] Yang, K.K., Talbot, B.F. and J.H. Patterson (1993). Scheduling a project to maximize its net present value: an integer programming approach. *European Journal of Operational Research*, 64: 188-198.