CLIENT-CONTRACTOR BARGAINING ON NET PRESENT VALUE IN PROJECT SCHEDULING WITH LIMITED RESOURCES

Nursel Kavlak[1], **Gündüz Ulusoy**[1]
Funda Sivrikaya Şerifoğlu[2], **Ş. İlker Birbil**[1]

[1]Sabancı University, Orhanlı, Tuzla 34956
Istanbul, Turkey
(nursel@su.sabanciuniv.edu; gunduz@sabanciuniv.edu; sibirbil@sabanciuniv.edu)

[2]Düzce University, Konuralp Yerleşkesi 81620, Düzce, Turkey
(fundaserifoglu@duzce.edu.tr)

ABSTRACT

The client-contractor bargaining problem addressed here is in the context of a multi-mode resource constrained project scheduling problem with discounted cash flows, which is formulated as a progress payments model. In this model, the contractor receives payments from the client at predetermined regular time intervals. The last payment is paid at the first predetermined payment point right after project completion. The second payment model considered in this paper is the one with payments at activity completions. 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, which constitutes a hard constraint. The bargaining objective is to maximize the bargaining objective function comprised of the objectives of both the client and the contractor. The bargaining objective function is expected to reflect the two-party nature of the problem environment and seeks a compromise between the client and the contractor. The bargaining power concept is introduced into the problem by the bargaining power weights used in the bargaining objective function. Simulated annealing algorithm and genetic algorithm approaches are proposed as solution procedures. The proposed solution methods are tested with respect to solution quality and solution times. Sensitivity analyses are conducted among different parameters used in the model, namely the profit margin, the discount rate, and the bargaining power weights.

Published in the *Naval Research Logistics* 56(2) pp.93-112, 2009

1. INTRODUCTION

In this paper, 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 individual objectives of the client and the contractor are in conflict most of the times. Hence, the bargaining objective should consider the incentives of both parties.

In the literature, a number of exact and heuristic methods are proposed for solving the single objective resource constrained project scheduling problem with discounted cash flows (see, e.g., [14, 18]). Russell [28] introduces an initial version of the discounted cash flow problem in project scheduling with no resource constraints. Grinold [12] extends the model by Russell by introducing a project deadline. The NPV criterion and its impact on project scheduling are investigated by Bey *et al.* [4]. Baroum and Patterson [1] review 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 [9], Yang *et al.* [38], Içmeli and Erengüç [15], Baroum and Patterson [2], and Vanhoucke *et al.* [37]. A relatively recent review on project scheduling is provided by Kolisch and Padman [22].

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 leads to studies on a variety of heuristic procedures among others by Russell [29], Smith-Daniels and Aquilano [33], Padman and Smith-Daniels [26], Padman *et al.* [27], and Kimms [19]. Etgar *et al.* [11] present a simulated annealing (SA) solution approach to maximize the NPV of a project, where the net cash flow amounts are independent of the time of realization. Ulusoy *et al.* [36] solve the multi-mode resource constrained project scheduling problem with discounted cash flows using genetic algorithm (GA). They allow both positive and negative cash flows. In their paper they distinguish among four types of payment scheduling models:

- *Lump-sum payment.* The client pays the total payment to the contractor upon successful completion of the project.

- *Payments 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.
- *Progress payments.* 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.
- *Fixed number of payments at equal time intervals.* In this payment model, a fixed number of 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.

Mika *et al.* [25] consider 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 only. As solution methods, they employ SA and GA.

Kazaz and Sepil [17] present a mixed-integer programming formulation of the progress payment model with the objective of maximizing the NPV of the cash flows for the contractor. Sepil and Ortaç [30] test the performance of some heuristic procedures for resource-constrained projects with progress payments. They define cash inflows occurring periodically as progress payments, and cash outflows as costs incurred whenever an activity is completed.

Dayanand and Padman [5, 6] attack the problem of simultaneously determining the amount, location, and timing of the payments by the client to maximize the contractor's NPV. They deal with this problem further from the perspective of the client [7] and later investigate the problem in the context of client and contractor negotiation and stress the need for a joint view in their treatise of the payment-at-event-occurrences model [8]. Ulusoy and Cebelli [35] 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. A competitive perspective, on the other hand, is provided by Szmerekovsky [34]. He considers the case where both the client and the contractor are assumed to act in their own best interest rather than trying to compromise.

2. PROBLEM DEFINITION

The client-contractor bargaining problem addressed here is in the context of a multi-mode resource constrained project scheduling problem with discounted cash flows, which is formulated as a progress payments model. In this model, the contractor receives payments from the client at predetermined regular time intervals. The last payment is paid at the first predetermined payment point right after project completion. The second payment model considered in this paper is the one with payments at activity completions.

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 constitutes a hard constraint meaning that exceeding the deadline violates 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.

Contractor's cash outflows associated with an activity can occur anytime throughout its execution. However, it is assumed here that they are 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. The payments are specified as the sum of the costs incurred for all activities completed from the last payment point until the current payment point and multiplied with $(1 + \beta)$, where β is the profit margin. 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.

The bargaining objective function is expected to reflect the two-party nature of the problem environment. The bargaining objective function seeks a compromise between the client and the contractor. The bargaining power concept is introduced into the problem by the bargaining power weights used in the bargaining objective function. Further details of this function are given in Section 3.

The weight parameter as a notion of bargaining power is also used in the mathematical economics literature. For instance, Ervig and Haake [10] discuss the notion of weights for two agents that are involved in two different bargaining problems. Under certain assumptions about the class of bargaining problems and the individual utilities, they thoroughly analyze the equilibrium conditions. Again within the game theory field, Köbberling and Peters [23] investigate the effect of decision weights in the bargaining problems. These decision weights are incorporated through probabilistic weight functions that determine the risk attitude of a decision maker. By varying the engaged agents' attitudes towards risk, they discuss the different outcomes of the bargaining games.

Marmol et al. [24] analyze an equitable solution for the multi-criteria bargaining games. To deal with the multiple objectives, they minimize the maximum deviation from the best minimum payoff values, which can be attained when the problem is solved for each individual criterion. To this extent, the distance concept in their study bears some similar features to the one used in the present study. However, Marmol et al. [24] do not consider the concept of bargaining powers. In the present paper, the bargaining power weights reflecting the respective powers of the involved parties cause significant nonlinearities in the objective function. If applied in our work (after some modifications), the solution approach suggested by Marmol et al. [24] would result in a nonlinear constrained integer programming problem - an extremely difficult problem to solve. In this paper, the bargaining power weights are defined to introduce the impact of the difference in relative bargaining positions of the client and the contractor. Furthermore, they are meant to reflect perceived quantities and do not represent exact point values. The bargaining objective function introduced is formulated using a max min approach so as to improve the position of the worse-off party among the client and the contractor, which would correspond to an acceptable position for both parties for the given bargaining power weights. Köbberling and Peters [23] 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 [10] 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 the present study as well; that is, the objective function aims at 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 the players 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 the present study, the bargain between players doesn't constitute a favor exchange, but instead a pure trade-off among benefits is obtained. Mármol *et al.* [24] 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 used in this study, in a way that both identify the distance from the minimum level point for both players.

In practice, both the client and the contractor or an external consultant may use the overall approach presented here to reach a mutual agreement point through a series of trade-offs. It provides an environment for scenario analysis, which the interested parties may utilize.

3. MATHEMATICAL FORMULATION

In this section, the formal mathematical programming model for the multi-mode resource constrained bargaining problem with progress payments is given. In the subsequent discussion, the following notation is employed:

J is the set of activities, where $i, j \in J$ denote the general activity indices, and $|J|$ denotes the last activity.

K is the set of resource types, where $k \in K$ denotes the resource index.

M_j is the set of modes for activity $j \in J$, where $m \in M_j$ denotes the mode index; the convention employed here is that as the mode number increases, the resource usage decreases and the activity duration increases.

t and q denote the time periods.

$i \prec j$: precedence relation; i.e., activity $i \in J$ precedes activity $j \in J$.

E_j : earliest finishing time of activity $j \in J$.

L_j : latest finishing time of activity $j \in J$.

d_{jm} : duration of activity $j \in J$ in mode $m \in M_j$.

D : predetermined deadline

r_{jkm} : consumption of resource $k \in K$ per unit time for activity $j \in J$ in mode $m \in M_j$.

R_k : availability limit of resource $k \in K$.

C_{max} : makespan.

C_{jm} : cost of activity $j \in J$ in mode $m \in M_j$ in real terms discounted to the starting time of the activity.

β : profit margin.

γ : discount rate per period.

α_t : continuous discount factor in period t, exp($-\gamma t$).

T_n : predetermined payment times, $n = 1, \ldots, N$ with $T_1 = 0$ and $T_N = D$.

P_{T_n} : client's payment at period T_n, $n = 1, \ldots, N$.

x_{jtm}: 1, if activity $j \in J$ is completed in period t with mode $m \in M_j$; 0, otherwise.

x: the decision vector with components x_{jtm}; i.e., the schedule.

F: feasible region of the overall problem.

The constraints are as follows:

$$\sum_{t=E_j}^{L_j} \sum_{m \in M_j} x_{jtm} = 1 \qquad j \in J \tag{1}$$

$$\sum_{t=E_j}^{L_j} \sum_{m \in M_j} t * x_{jtm} - \sum_{t=E_i}^{L_i} \sum_{m \in M_i} t * x_{itm} \geq d_{jm}, \qquad i \prec j;\ i,j \in J \tag{2}$$

$$\sum_{t=E_J}^{L_J} \sum_{m \in M_J} t * x_{Jtm} \leq D, \tag{3}$$

$$\sum_{j \in J} \sum_{m \in M_j} \sum_{q=t}^{t+d_{jm}-1} r_{jkm} * x_{jqm} \leq R_k, \qquad k \in K;\ t = 1, \ldots, C_{max}, \tag{4}$$

$$x_{jtm} \in \{0,1\}, \qquad j \in J;\ m \in M_j;\ t = 1, \ldots, C_{max}, \tag{5}$$

The constraint set (1) ensures that each activity is assigned. The constraint set (2) makes sure that all precedence relations are satisfied. The constraint (3) secures that the project is completed on or before the deadline. The constraint set (4) makes sure that for every single resource the required amount does not exceed the corresponding resource constraint at any given point in time throughout the project duration. The last set of constraints (5) shows that the variables are binary. Notice that a decision variable is feasible when it satisfies the above constraints. In other words, the feasible region of the problem is given by

$F = \{\mathbf{x} : \mathbf{x} \text{ satisfies constraints } (1) - (5)\}$.

Next, the bargaining objective of the mathematical model is presented. Let f_A and f_B denote the individual objective function value of the client (A) and the contractor (B), respectively. Then, we have

$$f_A(\mathbf{x}) = -\left(\sum_{n=1}^{N} \alpha_{T_n} * P_{T_n} \right), \tag{6}$$

$$f_B(\mathbf{x}) = \left(\sum_{n=1}^{N} \alpha_{T_n} * P_{T_n} \right) - \left(\sum_{j \in J} \sum_{t=E_j}^{L_j} \sum_{m \in M_j} C_{jm} * \alpha_{t-d_{jm}} * x_{jtm} \right), \tag{7}$$

where

$$P_{T_n} = (1+\beta) * \sum_{j \in J} \sum_{t=T_{n-1}+1}^{T_n} \sum_{m \in M_j} C_{jm} * x_{jtm} , \qquad n = 2,..., N. \tag{8}$$

Suppose that x^*_A and x^*_B are the optimal solutions for the client and the contractor over the feasible region, respectively. Formally,

$$x^*_A := \arg\max \{f_A(x) : x \in F \}, \tag{9}$$

$$x^*_B := \arg\max \{f_B(x) : x \in F \}, \tag{10}$$

Using now (6) and (7), the optimal objective function values for the client and the contractor are defined as $f''_A := f_A(x^*_A)$ and $f''_B := f_B(x^*_B)$, respectively. To introduce the objective of the bargaining problem, we define for the client the value $f'_A := f_A(x^*_B)$ and for the contractor the value $f'_B := f_B(x^*_A)$. These are the undesired values for the respective parties, and are defined as the objective function value for the player at the optimal solution of the other player.

After these definitions, the bargaining objective of our mathematical model becomes

$$\max \min \left\{ \left(\frac{f_A(x) - f'_A}{f''_A - f'_A} \right)^{w(A)}, \left(\frac{f_B(x) - f'_B}{f''_B - f'_B} \right)^{w(B)} \right\}, \tag{11}$$

where $w(A) \in (0,1)$ and $w(B) \in (0,1)$ with $w(A) + w(B) = 1$ denote the *bargaining power weights* for the client and the contractor, respectively. The terms in parentheses in the bargaining objective function are called the *bargaining values* for the client and the contractor, respectively. The bargaining value raised to power by the bargaining power weight results in the *weighted bargaining value*, where a higher value is more desirable for both parties. The fundamental metric proposed in the bargaining objective function formulation is the normalized distance of the party involved from the respective undesired solution. The bargaining objective function tries to maximize the minimum of the normalized distances of the *objective function values* of the client $f_A(x)$ and of the contractor $f_B(x)$ from f'_A and f'_B, respectively. In other words, the bargaining objective function is formulated so as to improve the position of the worse-off party among 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 weights are defined for both the client and the contractor. A large bargaining power weight implies a strong bargaining position. For each player, the optimal solution is the result of the respective single objective problem solved by a commercial solver. These values are employed in the bargaining objective function for normalization.

The overall mathematical model considered in this paper is given as follows:

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

s.t. $\quad f_A(\mathbf{x}) = -\left(\sum_{n=1}^{N} \alpha_{T_n} * P_{T_n} \right),$

$$f_B(\mathbf{x}) = \left(\sum_{n=1}^{N} \alpha_{T_n} * P_{T_n} \right) - \left(\sum_{j \in J} \sum_{t=E_j}^{L_j} \sum_{m \in M_j} C_{jm} * \alpha_{t-d_{jm}} * x_{jtm} \right),$$

$$P_{T_n} = (1+\beta) * \sum_{j \in J} \sum_{t=T_{n-1}+1}^{T_n} \sum_{m \in M_j} C_{jm} * x_{jtm}, \qquad n = 2, \ldots, N,$$

$$\mathbf{x} \in F.$$

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.

Note that the base model above considers the progress payment model at predetermined time intervals. The last payment period is scheduled at the deadline. The second payment model considered in this paper is the one with payments at activity completions. In this case, the term P_{Tn} in the above mathematical model should be replaced by

$$P_t = (1+\beta) * \sum_{j \in J} \sum_{m \in M_j} C_{jm} * x_{jtm}, \qquad t = 1, \ldots, C_{max}, \qquad (12)$$

and the first two equalities become

$$f_A(\mathbf{x}) = -\left(\sum_{t=1}^{C_{\max}} \alpha_t * P_t \right), \qquad (13)$$

$$f_B(\mathbf{x}) = \left(\sum_{t=1}^{C_{\max}} \alpha_t * P_t \right) - \left(\sum_{j \in J} \sum_{t=E_j}^{L_j} \sum_{m \in M_j} C_{jm} * \alpha_{t-d_{jm}} * x_{jtm} \right). \qquad (14)$$

4. SOLUTION APPROACHES

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 bargaining objective function. The optimization of such a bargaining objective function does not depend on structured rules like completing the costly activities earlier, or delaying the project. For instance, delaying a costly activity may be beneficial for the contractor but if the client's weighted bargaining value is less than the contractor's, the bargaining objective function value may be decreased; or delaying an activity may not have an effect at all. The max min objective function itself invalidates such structured rules. Rather than devising an algorithm based on heuristic decision rules, it has been decided to employ two different meta-heuristics, SA and GA, in order to exploit their well-established capability of searching through the solution space effectively. Since the same solution representation and generation are employed both in SA and GA, a description of these is provided first.

4.1 Solution Representation

A solution is represented by a combination of three serial lists: activity list, mode list, and idle time list. The structure of these lists is as follows:

1. Activity List:

   Activity list is a precedence-feasible permutation of activities. The dummy source and sink nodes are placed at the start and at the end of the list, respectively, by default. The list represents the priority ordering for the starting time of activities. That is, an activity appearing earlier in the list should start at the same time or at an earlier time than its immediate follower.

2. Mode List:

   The mode list shows the assigned modes for each activity in the activity list.

3. Idle Time List:

   The idle time value represents the exact idle time to be inserted before the start of the corresponding activity in the activity list.

An example for a 14-activity network is as given in Figure 1. In the example, the dummy activity 1 is assigned mode 1 and its start is delayed by 1 unit time, while activity 3 is assigned mode 2 and its starting time is delayed by 4 units of time.

Place Figure 1 about here

4.2 Feasible Solution Generation

Feasible solutions are generated as follows. A list of eligible activities is kept, which is initially composed of activities with no predecessors. One of the activities from this set is randomly chosen to be inserted into the next position on the activity list. Then the eligible set is updated by deleting the activity chosen and by inserting activities, all predecessors of which have already been inserted into the activity list.

For each feasible activity list, a set of random mode and idle time lists are generated among which a feasible combination is searched for. For each triple of activity list, mode list, and idle time list, the feasibility is checked according to both the resource limitations, and the deadline. Infeasible solutions are discarded. Problem sets used in tests are redefined by extending the deadlines for each problem set versus the original problem sets; details are given in Section 5 on computational results.

4.3. Simulated Annealing Method

4.3.1. Neighborhood Generation Mechanism

Neighboring solutions are generated by altering the three lists; i.e. the activity, mode and idle time lists. All feasible solutions generated by the mechanisms explained below are included in the neighborhood.

a. **Activity Replacement:** An activity is chosen randomly and is moved together with its corresponding mode and idle time assignments to each one of the feasible alternative locations one by one. 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, one mode at a time. All possible mode changes for all activities are tried.
c. **Idle Time Change**: By keeping the activity list and the mode list constant, idle time changes are applied, one at a time. All possible idle time changes for all activities are tried. For each of the activities all possible idle time changes are investigated by checking the feasibility of the solution on two dimensions, namely the deadline and the resource availability.

The solutions that yield negative numerators in the bargaining objective function are ignored since such solutions are clearly inferior. This guarantees that no adopted schedule can provide one of the players a value worse than f', which is the objective function value of that player at the optimal solution of the other player.

4.3.2. Cooling mechanism

The cooling scheme used here is a scheme effectively used by other researchers on complex problems ([3, 32]). The details are given below:

$$T_{init} = (f_{min} - f_{max}) / (\ln P_A^{init}) \quad (15)$$

$$c_r = (\ln P_A^{init} / \ln P_A^{f})^{1/(maxIter-1)} \quad (16)$$

$$T_{curr} = c_r * T_{curr} \quad (17)$$

where c_r is the cooling rate, T_{init} and T_{curr} are the initial and current temperatures, respectively; P_A^{init} and P_A^{f} are the initial and final acceptance probabilities, respectively; f_{min} and f_{max} are the minimum and maximum bargaining objective function values observed in an initial set of solutions, respectively; and *maxIter* is the maximum number of temperature reduction cycles.

4.3.3. Stopping Criterion

A fixed iteration count taken as the maximum number of temperature reduction cycles (*maxIter*) is adopted as the stopping criterion. The number of iterations carried for each problem depends on the number of activities; higher number of iterations is required in order to improve the objective function once the activity number of the problem increases. The (*maxIter*) value is determined based on several test runs on each type of problem – that is the minimum iteration count which delivers all observed improvement in the objective functions. Since the bargaining objective function is in *max min* format, the same bargaining objective function value may be observed at different solution points. Hence, other stopping criteria based on objective value improvements are not appropriate for the current problem. A record of the best solution ever encountered is kept throughout the run and is reported as the SA solution at termination.

4.3.4. The Algorithm

Step1: Find an initial solution and adopt it as the current solution *S*. Determine T_{init} and c_r. Take T_{curr} as T_{init}.

Step2: Generate all alternative solutions in the neighborhood of *S*.

Step3: Select randomly a solution *S'* from the neighborhood of *S*. If *S'* represents a better solution than *S* in terms of the bargaining objective, adopt *S'* as the current solution If *S'* represents a worse solution than *S*, adopt it as the current solution with probability of acceptance $P_A = e^{-(\Delta z / T_{curr})}$, where Δz denotes the absolute difference between the bargaining objective function values of *S* and *S'*. If *S'* is adopted as the current solution *S*, then go to Step 2. Repeat Step 3 until the number of neighborhood solutions tested reaches N_{test}.

Step 4: If the number of temperature reduction cycles reaches *maxIter*, then terminate. If not, then update T_{curr} and go to Step 2.

4.4. Genetic Algorithm

GA is applied by using the chromosome representation explained in Section 4.1 and exemplified in Figure 1. The initial population is generated employing the feasible solution generation mechanism explained in Section 4.2. The fitness value is set equal to the objective function value. Roulette wheel selection mechanism is employed, where the probability of selection is proportional to an individual's fitness [13]. Elitist selection is also applied such that the best few chromosomes are transferred directly to the new generation.

4.4.1. Crossover Operator

The most challenging problem when applying GA to multi-mode resource constrained scheduling problems is to reproduce feasible off-springs. *The Multi Component Uniform Order Based Crossover* (MCUOX) is employed, which preserves precedence feasibility when generating one offspring from parent chromosomes [31]. In this operator, one of the parents is selected randomly. Starting from the first activity, the next activity on that parent not assigned to the offspring yet is found. Then the mode assignment of that activity on each of the parents is determined and one is selected randomly. Finally, the idle time assignment of that activity on each of the parents is determined and one is selected randomly. This procedure is repeated until an offspring is generated fully.

4.4.2. Mutation Operators

- <u>Activity replacement:</u> A parent chromosome is selected randomly. Then an activity is selected randomly from the activity list of that chromosome. Next the position of that particular activity on the activity list is changed as follows. Activity's replacement window is determined according to the precedence relations - replacement window for an activity is basically the window within the activity list between the slot where the predecessors of that specific activity ends and the slot where the successors of that specific activity starts. The activity together with its mode and idle time assignments is moved to a location within the replacement window, and the whole list is adjusted accordingly. If the new solution satisfies all feasibility constraints, then it is accepted as an offspring.
- <u>Bit mutation:</u> Bit mutation is applied either to the mode or to the idle time assignment dimension following a four-step approach: First, for each chromosome, it's decided whether to bit mutate it or not. Once it is decided to apply bit mutation to a chromosome, the type of bit mutation, the activity to be mutated, and the direction of change are selected in sequence. If the mode bit mutation is going to be applied, the existing mode of the activity is replaced with another randomly chosen mode value. If

the idle time bit mutation is going to be applied, the existing idle time value is either decreased by 1 (if it is not 0), or increased by 1, where decision is randomly taken with 0.5 probability on each alternative of decreasing and increasing the current idle time.

4.4.3. Population Management

The GA pool management scheme is shown in Figure 2. New generations are formed by the offspring of the crossover and the activity replacement mutation operators, the best individuals selected according to the elitist strategy, and fitness-based selected individuals. Bit mutation is applied to the members of this new population except the elites.

Place Figure 2 about here

4.4.4. Termination

The whole cycle of generating a new population is repeated for 50-100 times, depending on the problem size. At each generation, the chromosome with the highest fitness value ever is kept in memory. The final solution is the solution with the best fitness value reached ever after the last cycle.

5. COMPUTATIONAL RESULTS

The test problems used are adopted from the project scheduling problem library PSPLIB ([20, 21]). The adoption has been made by eliminating the tardiness costs, relaxing the deadlines, and excluding the nonrenewable resources. For each problem size of 14, 20, and 32 activities, problem sets consisting of 30 problems are used adding up to a total of 90 test problems. Problems are selected from *Multi-Mode Resource Constrained Project Scheduling Problem (MRCPSP)* directory of PSPLIB. Within this directory following data sets are used for problems with 14, 20, and 32 activities respectively: *j12.mm*, *j18.mm*, and *j30.mm*. Within each of these sets there are around 550 problems, and among these 30 problems are selected randomly from each data set. These problem sets had been modified by disregarding non-renewable resources and extending the deadlines in order to increase feasibility. These modified data sets are provided in Kavlak [16]. All of the problem sets used involve two renewable 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. Initially, in order to investigate the behavior of the model and the performance of the solution procedures, in all tests the bargaining power weight of both the client and the contractor are set equal to 0.5.

For each problem set, initial solutions are generated as described in Section 4.2. First the activity lists are generated, then mode and idle time lists are generated randomly, and the resulting chromosome is subjected to feasibility checks. Table 1 illustrates the percentage of feasibility hits.

Place Table 1 about here

In SA, P_A^{init} is set at 0.95, and P_A^f is set to 0.001. A set of 100 solutions are generated using the feasible solution generation mechanism explained in Section 4.2 to determine the f_{min} and f_{max} values. Modifying the temperature plateau length, i.e. the number of neighbors tested at a given temperature level, has also been tested, but no significant improvement regarding the solution has been observed. Hence, N_{test} is set at 100. The fixed iteration count *maxIter* for each problem set is given in Table 2.

Place Table 2 about here

GA follows a two step pool management strategy as shown in Figure 2. When moving from the old population to the new one, elites constitute 2%, fitness-based selected individuals constitute 48%, off-springs generated by MCUOX operator constitute 33%, and off-springs generated by activity replacement operator constitute 17% of each new population.

The offspring generation process through MCUOX continues until the number of feasible offspring reaches 1/3rd of the original population. This proportion is a design choice, which enables highest improvement at each iteration when compared with other tested ratios of 1/6, 1/4, and 1/2.

Activity replacement mutation operator is applied to randomly selected individuals from the original population until the number of feasible individuals reaches 1/6th of the population. This proportion is again a design choice which enables highest improvement at each iteration when compared with other tested ratios of 1/4 and 1/3. These ratios were tested for 9 problem sets of which 3 have 14-activities, 3 have 20-activities, and the last 3 have 32-activities. Results showed that for all instances MCUOX offspring generation process continuing until the number of feasible offspring reaches 1/3rd of the original populations provided the highest improvement in the objective function. Similarly, continuing with the activity replacement mutation operator until the number of feasible offspring reaches 1/6th of the original populations provided for all instances the highest improvement in the objective function.

Bit mutation is applied with a four step approach, where selection probability is equal for the alternatives at each step. Figure 3 illustrates the bit mutation mechanism.

Place Figure 3 about here

Table 3 provides average percent deviations of the GA and SA solutions from the optimal solutions obtained by using the commercial solver GAMS 20.0. The *t*-test results stating whether GA delivering higher objective function values than SA is statistically significant or not are also reported in Table 3. With payments at 10 time periods and payments at 5 time periods, it is observed that GA delivering better results than SA is statistically significant, but it is not statistically significant for the instances with payments at activity completions. These results show that as the frequency of the payments increases and the number of activities decreases, better results are obtained and the likelihood of finding near optimal solutions increases, no matter which method is used. It is clear that the problem is getting more complex with increasing numbers of activities as can be seen from the increasing deviations for increasing numbers of activities in the rows of Table 3. As can be observed from the columns of Table 3, the deviations decrease with increasing frequency of payments, which indicates that the problem becomes less complex in that direction.

Place Table 3 about here

All tests are carried out on a Toshiba A10-S503 with Mobile Intel Celeron 2.20 GHz CPU, and 256 MB of RAM. Average CPU times for all test instances are given in Table 4. Here, for both SA and GA the number of objective function evaluations within the computational time is compatible. Still it is observed that the computation takes a little longer for GA when compared with SA at each instance. The reason for this may be due to the effort spent for the management of the objective function ranking list employed in GA to determine the elites.

Place Table 4 about here
----

6. SENSITIVITY ANALYSES

Sensitivity analyses are conducted for three important parameters used in the model: the profit margin, the payment discount rate, and the bargaining power weight. For profit margin and discount rate tests, exact optimal solutions obtained by using commercial solver GAMS.20.0 are used. For profit margin and discount rate tests, bargaining power weight of the players are kept constant and equivalent at 0.5. Hence, the effects of these two parameters are tested under equal conditions for both of the players. Bargaining power weight tests are held for varying discount rate γ with constant profit margin β at 0.1. In these tests, GA solutions are employed, since optimal solutions for these problems could not be delivered by the commercial solver due to the non-linear nature of the problem.

Since the client pays a profit margin 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, both payments at activity completions and payments at every 5 time periods payment models are tested. The client prefers less frequent payments, which leads to bulk payments, whereas the contractor prefers more frequent payments in order to be able to receive its return on investment as soon as possible. That is, while the client prefers payment for each specific activity deferred, the contractor prefers payments at activity completions. This brings a trade-off between the profit margin β and the frequency of the payments. Since increasing the profit margin or the payment frequency both brings an NPV change in the same direction, the tests show that, depending on both β and the discount rate γ, there is a trade-off between them at a constant NPV point. Although all these changes affect NPVs of the players, they don't have an affect on the bargaining power weights 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 one observes a significant effect on the NPVs, one doesn't see any bargaining power weight effect in the decision making process.

The sensitivity analyses are conducted on 30 test problems with 14 activities. As an example, the project network and the mode structure of the activities for one of the test problems are given in Figure 4 and Table 5, respectively. In all the test problems employed here, payments are due every 5 time periods. In this section, the different sensitivity analyses results are displayed as Box and Whisker plots, which give five number summaries: the smallest observation, lower quartile, median, upper quartile, and largest observation.

Place Figure 4 about here

Place Table 5 about here

6.1 Sensitivity Analysis for the Profit Margin

Within the model, the profit margin β directly affects the amount of payment made by the client to the contractor at each payment point (refer to Equation (8)).

When the schedule is kept constant, increasing β would clearly increase the objective function value of the contractor, and decrease the objective function value of the client and vice versa. Since the schedule itself is also a decision variable in the model, the model is run with different β values (0.010; 0.250; 0.500) at different discount rate γ levels (0.005; 0.010; 0.100). In Figures 5(a)–(c), examples can be found from a series of tests that display the best bargaining value (i.e., the minimum of the weighted bargaining values for the client and the contractor) for increasing profit margin β at different γ levels. In these figures, since the values for the client and the contractor are very close to each other, only the smaller of the weighted bargaining values for the client and the contractor are displayed.

Place Figure 5(a)–(c) about here

Here, it is observed that there is no correlation between β and the smaller weighted bargaining value. The main reason for this is the direct effect of β on payments at each payment time. Since variations of the payments in-between payment periods have significant effect on the adopted schedule, β exercises a major effect on the final schedule. Hence, the contractor can not always maintain his advantage on increased β. This result can be observed from Table 6, which shows the player whose weighted bargaining value corresponds to the bargaining objective function value at each β level. Since the model has a max min objective function, that particular player is the one who is less satisfied due to the inferior weighted bargaining value.

Place Table 6 about here

From Table 6, it is observed that there is no specific pattern. This results from the client's move for offsetting the contractor's profit margin advantage by causing major changes in the overall schedule. That is, β is not the single determinant, and other factors influence the outcome regarding the minimum of the weighted bargaining values.

Contrary to that, a direct relation is detected between the profit margin β and the objective function values of the client $f_A(x)$ and the contractor $f_B(x)$. Namely, as β increases, the objective function value of the client decreases and that of the contractor increases as expected. For the problem set, where γ is taken as 0.100, which is fairly high when compared to other tests, the initial values for both of the players at $\beta = 0.10$ are relatively close to each other, which indicates a trade-off between β and γ. The details of the progression of the objective function values for both players are shown in Table 7 displaying the average values over 30 test problems and for γ=0.010 in Figure 6.

Place Table 7 about here

Place Figure 6 about here

6.2 Sensitivity Analysis for the Discount Rate

In the bargaining model, discount rate γ is used in NPV calculations within the objective functions of the players as given in Equations (6) and (7). Discount rate γ may affect the schedule preferences of the players due to different payment amounts at each payment point, and due to the fact that the contractor pays activity costs in advance although s/he receives payments only at upcoming payment points or activity completion. This shows that the level of discount rate γ affects the contractor in two aspects, whereas it affects the client only in one. An increase in γ is a source of an increase in the objective function value for the client and just the opposite for the contractor. The details are given in Table 8 displaying the average values over 30 test problems and for β=0.25 in Figure 7.

Place Table 8 about here

Place Figure 7 about here

The schedule manipulations due to changes in the discount rate γ have more effect on the contractor's weighted bargaining value than they have on the client's. Test results are presented in Figures 8 (a)-(c), where only the smaller of the weighted bargaining values for the client and the contractor are displayed, since the values for the client and contractor are very close to each other. Here, it is clearly seen that increasing γ leads to increased weighted bargaining values. Consequently, the client and the contractor move to a position that is more desirable for them. The main reason for this is that as γ increases, the objective function value f_B for the contractor decreases due to decreased net realization. This leads the bargaining model to introduce schedule improvements to decrease the effects of γ from which the contractor benefits at the expense of the client's benefit. Although these changes still result in inferior f_B value for the contractor when compared with the results of cases with lower γ, the percentage increase in individual objective function value within the bargaining model itself is higher for both players, since the benefit of the client is coupled with schedule changes that the contractor benefits from.

Place Figure 8 (a)-(c) about here

6.3 Sensitivity Analysis for Bargaining Power Weights

Within the model the bargaining power weight for each player is applied in the final step of the bargaining objective function evaluation in order to set the exact realized value for the players. The bargaining objective function is presented below again for convenience.

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

Recall that in these tests, γ=0.05 and β=0.1, and the bargaining power weights of the client and the contractor add up to 1. The summary of the bargaining power weight tests conducted are given in Table 9 displaying the average values over 30 test problems.

Place Table 9 about here

The analysis of the bargaining values in Table 9 leads to the observation that, when the gap between the bargaining powers of the parties increases, the realized weighted bargaining values increase for both of the players. This is due to the fact that, if one of the players starts with a small bargaining power, his/her realized bargaining value increases considerably with increasing bargaining power, so that the other party tries to catch this increase by schedule changes. Objective function values for the players follow a similar trend as the weighted bargaining values do with respect to bargaining power weight.

Table 10 represents schedule examples at different bargaining power values. As the bargaining weight of the client increases, the contractor tries to compensate by changing the schedule. The contractor accomplishes this by adjusting the modes and the idle times. As can be observed in Table 10, on the average, the duration of the activities increase and the idle times decrease as the bargaining weight moves from the contractor to the client. This is a behaviour we expect to be valid in general. However, the changes in the duration and idle times for individual activities are not necessarily monotonic. For example, the mode for activity 13 is increased first, then decreased and later increased again. The delay time for activity 4, for example, is first decreased, then increased, and finally decreased again.

Place Table 10 about here

The average weighted bargaining values over 30 test problems for each player at different bargaining power weight couples are displayed for varying discount rates γ in Table 11 and in Figure 9 for γ=0.02. In Figure 9, only the smaller of the weighted bargaining values for the client and the contractor are displayed, since the values for the client and contractor are very close to each other. Here, one observes that, when the gap between bargaining power weights of the parties increases, the realized weighted bargaining values increase for both of the players. This is due to the fact that, if one of the players has a small bargaining power weight, his/her realized weighted bargaining value increases considerably, so that the other party tries to catch this increase by schedule changes. On the other hand, as discount rate γ increases, it is observed that both the client and the contractor become more content with increasing weighted bargaining values for all bargaining power weight couples. Although the change in γ doesn't change the pattern of increasing weighted bargaining values as the gap between bargaining power weights of the parties' increases, it is observed that the percentage change on the weighted bargaining values is decreased as γ increases. In other words, at high discount rate γ, there is less room for bargaining value improvements introduced by schedule changes when the gap between bargaining power weights of the parties is high.

Place Table 11 about here

Place Figure 9 about here

The progression of bargaining values of the players is illustrated in Figure 10, from which the direct relationship between the bargaining power weights and the bargaining values can be observed.

Place Figure 10 about here

Objective function values for the players follow a similar trend as the weighted bargaining values do with respect to bargaining power weight. In Figure 11, one clearly observes this trend.

Place Figure 11 about here

Increasing the profit margin, the discount rate, or the bargaining power weight, with everything else staying constant, leads to six key trends in the objective function values of the players, which are summarized in Table 12.

Place Table 12 about here
----

7. AN EXTENSION OF THE BARGAINING MODEL

The bargaining model formulation is enriched by introducing a benefit term into the objective function of the client as a function of completion earlier than the predetermined deadline. In this extended bargaining model, the client receives a constant amount of benefit per each time period that the project is completed earlier than the deadline. No cost, on the other hand, is imposed on the client through the contractor. In a sense, this model proposes an additional benefit generated in the system regardless of any value trade off. To obtain the extended bargaining model, the base mathematical model is modified by replacing the client's objective function value with the following function:

$$f_A(\boldsymbol{x}) = -\left(\sum_{n=1}^{N} \alpha_{T_n} * P_{T_n}\right) + \left(D - \sum_{t=E_{|J|}}^{L_{|J|}} \sum_{m=1}^{M_{|J|}} t * x_{|J|tm}\right) * \sigma \quad (18)$$

where σ is the client's benefit per time period resulting from early completion.

One of the more important impacts of this extension is on the adopted schedule. To demonstrate this, the example problem given earlier in Figure 4 and Table 5 is solved for different values of σ and the schedules obtained are displayed in Table 13. It is observed that as σ increases, the adopted schedule changes in a way that the project tends to finish earlier by choosing modes with higher cost and less duration. Similar analysis is carried for the remaining test problems. In all those tests, the schedule moves reported above are observed to be typical for all tested instances. The key conclusion reached as a result of the test problem solutions is that both players gain out of the benefit resulting from early completion. This leads to increases in the bargaining objective function value and the individual objective function values for both players due to the structure of the bargaining objective.

Place Table 13 about here
----

8. CONCLUSION

In this paper, 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 objective function comprised of the minimum of individual NPV maximizing objectives of both the client and the contractor.

In this context, we have proposed two main solution methods, namely SA and GA approaches. Based on comparisons with the optimal results obtained by using a commercial solver (GAMS 20.0), one can conclude that GA has provided better results than SA.

Sensitivity analyses results show that profit margin increase doesn't cause significant changes in the bargaining objective function value, but it has effect on objective function value of the players. That is, even if one of the parameters changes in favor of a player, the bargaining mechanism offsets this advantage with schedule changes. For example, even if increasing the profit margin seems to be increasing the objective function value of the contractor, when the bargain ends, the contractor does not necessarily end up with an advantageous position.

As expected, the objective function value of the contractor increases with increasing profit margin, whereas the objective function value of the client decreases. The tests we have conducted on the discount rate show that, although an increase in the bargaining objective function is observed with increasing discount rate, this is not a monotonic increase. However, variation in the discount rate has significant impact on the objective functions of the players. The objective function value of the contractor decreases with increasing profit margin, whereas the objective function value of the client increases. Weight tests implemented show that bargaining power weights have significant impact on the solution, not only on the weighted bargaining values of the players but also on the objective function values of the players.

During the problem solving stage, which appears as the bargaining stage, the variables interact with each other. For example, an increase in the profit margin has an impact in the same direction as an increase in the bargaining power weight of the contractor as would be expected or vice versa. There are various means of bargaining. The bargaining process is not only affected by the bargaining power weights. An increase in the profit rate can be tried to be counterbalanced by schedule adjustments. Likewise, if the discount rates are increasing due to market dynamics, the contractor may try to compensate for its loss by offering an increase on profit margin, which has a positive effect on the objective function value of the contractor.

An extension to the base model has also been investigated by introducing a benefit term into the objective function of the client as a function of completion earlier than the predetermined deadline. The key conclusion we have reached based on these benefit tests is that both players gain out of the benefit resulting from early completion. This leads to increases in the bargaining objective function value and the individual objective function values for both players due to the structure of the bargaining objective.

Although various analyses on the parameters have been conducted, still a series of combinations is open for research through the proposed model. The effect of bargaining power weights on different payment models may be an important source of analysis for further investigation of the problem. The influence of the bargaining power weights and the types of payment model on the model outcomes can be investigated. Other payment models not considered here may also be investigated using various solution procedures. A further study can be the investigation on a benefit sharing model between the client and the contractor for the increased benefit due to early completion. For example, a model, which will study the trade off between the client's benefit and the contractor's bonus for early completion, may produce further interesting results.

Acknowledgement: We thank the anonymous referees for their invaluable comments and suggestions, which have improved the paper considerably.

REFERENCES

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

[2] S.M. Baroum, and J.H. Patterson, "An exact solution procedure for maximizing the net present value of a project," Handbook on recent advances in project scheduling, J. Weglarz (Editor), Kluwer Academic, Dordrecht, 1999, pp.107-134.

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

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

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

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

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

[8] N. Dayanand, and R. Padman, "On payment schedules in contractor client negotiations in projects: An overview of the problem and research issues," Handbook on recent advances in project scheduling, J. Weglarz (Editor), Kluwer Academic, Dordrecht, 1999, pp. 477-508.

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

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

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

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

[13] D.E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison-Wesley Pub. Co., Boston, 1989.

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

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

[16] N. Kavlak, Client-Contractor Bargaining Problem In The Context Of Multi-Mode Project Scheduling With Limited Resources, Unpublished MSc. Thesis, Sabanci University, Istanbul, 2005.

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

[18] A. Kimms, Mathematical programming and financial objectives for scheduling projects, Kluwer Academic Publishers, Dordrecht, 2001.

[19] A. Kimms, 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 (2001), 221-236.

[20] R. Kolisch, A. Sprecher, and A. Drexl, Characterization and generation of a general class of resource–constrained project scheduling problems, Management Science 41 (1995), 1693–1703.

[21] R. Kolisch, and A. Sprecher, PSPLIB—A project scheduling problem library, European Journal of Operational Research 96 (1996), 205–216.

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

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

[24] A.M. Mármol, L. Monroy, and V. Rubiales, An equitable solution for multi-criteria bargaining games, European Journal of Operational Research 177 (2007), 1523-1534.

[25] M. Mika, G. Waligora, and J. Weglarz, 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 164 (2005), 639-668.

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

[27] R. Padman, D.E. Smith-Daniels, and V.L. Smith-Daniels, Heuristic scheduling of resource constrained projects with cash flows, Naval Research Logistics 44 (1997), 365-381.
[28] A.H. Russell, Cash flows in networks, Management Science, 16 (1970) 357-373.
[29] R.A. Russell, A comparison of heuristics for scheduling projects with cash flows and resource restrictions, Management Science 28 (1986), 1291-1300.
[30] C. Sepil, and N. Ortac, Performance of the heuristic procedures for constrained projects with progress payments, Journal of the Operational Research Society 48 (1997), 1123-1130.
[31] F. Sivrikaya-Şerifoğlu, A New Uniform Order-Based Crossover Operator for Genetic Algorithm Applications to Multi-component Combinatorial Optimization Problems. Department of Industrial Engineering, Boğaziçi University, PhD Dissertation, 1997.
[32] F. Sivrikaya-Şerifoğlu, and İ. Tiryaki, Multiprocessor task scheduling in multi-stage hybrid flowshops: A simulated annealing approach. Proceedings of the 2nd International Conference on Responsive Manufacturing, Gaziantep, Turkey, 2002, pp. 270-274.
[33] D.E. Smith-Daniels, and N.J. Aquilano, Using a late start resource constrained project schedule to improve project net present value, Decision Sciences 18 (1987), 617-630.
[34] J. G. Szmerekovsky, The impact of contractor behavior on the client's payment-scheduling problem, Management Science 51 (2005), 629-640.
[35] G. Ulusoy, and S.Cebelli, An equitable approach to the payment scheduling problem in project management, European Journal of Operational Research 127 (2000), 262-278.
[36] G. Ulusoy, F. Sivrikaya-Şerifoğlu, and Ş. Şahin, Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows, Annals of Operations Research 102 (2001), 237-261.
[37] M. Vanhoucke, E. Demeulemeester, and W. Herroelen, On maximizing the net present value of a project under renewable resource constraints, Management Science 47 (2001), 1113-1121.
[38] K.K. Yang, B.F. Talbot, and J.H. Patterson, Scheduling a project to maximize its net present value: an integer programming approach, European Journal of Operational Research 64 (1993), 188-198.

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
Duration	0	3	2	4	2	4	2	1	2	3	2	2	3	4
Starting Time	1	5	6	11	9	4	15	19	16	13	23	20	21	26
Finishing Time	1	8	8	15	11	8	17	20	18	16	25	22	24	30

Figure 1. Solution representation (chromosome) for a 14-activity problem

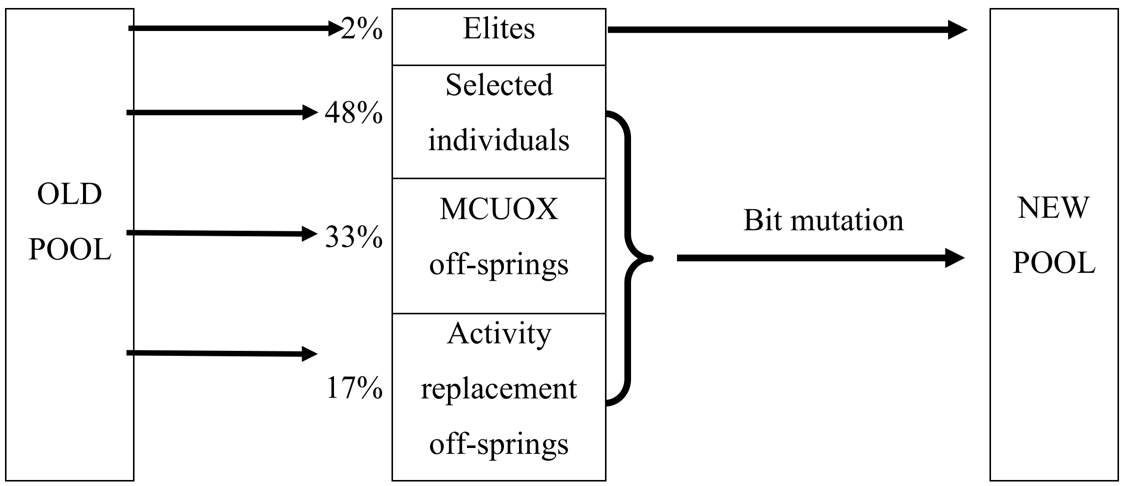

Figure 2. GA pool management scheme

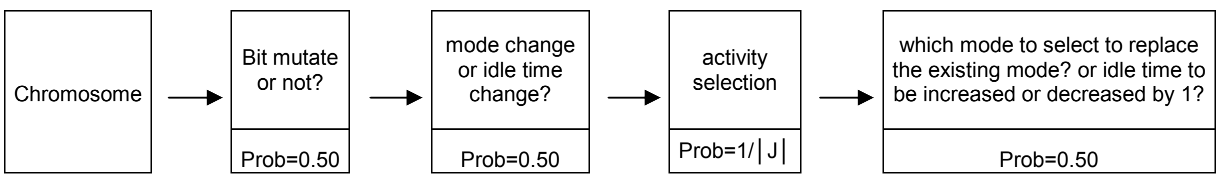

Figure 3. Bit mutation flow chart

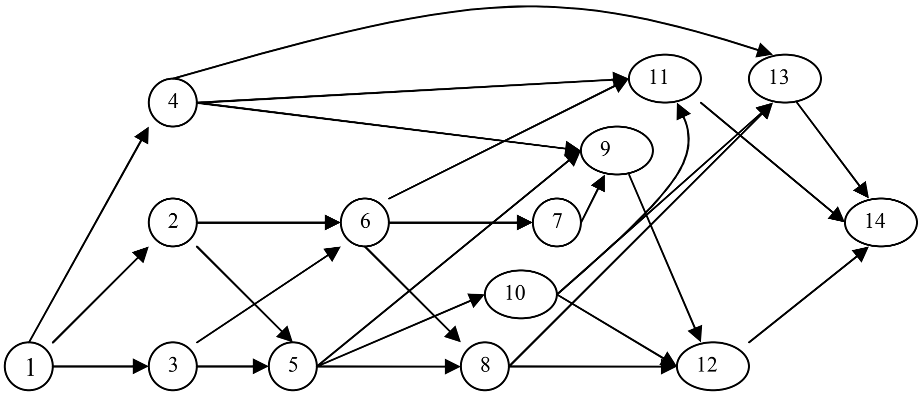

Figure 4. Project network for the example problem

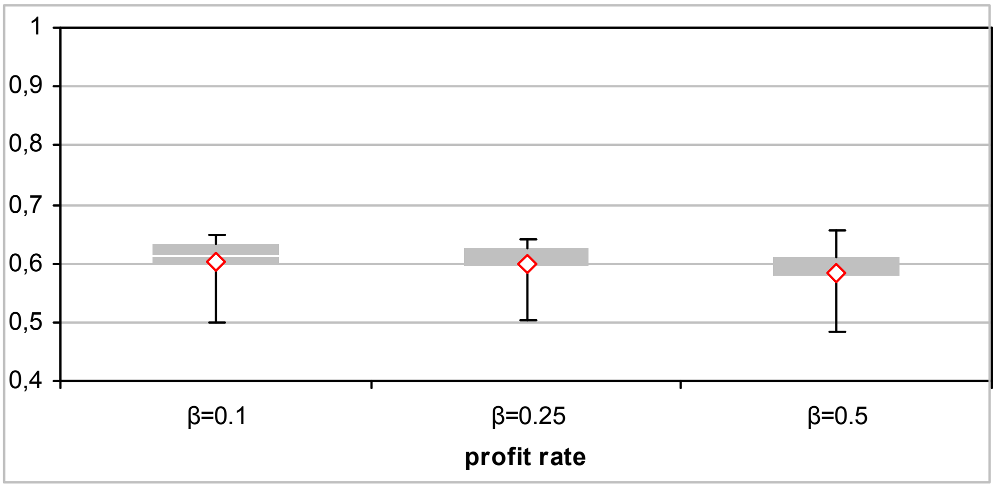

Figure 5 (a). Weighted bargaining values at different profit margin levels (γ =0.005)

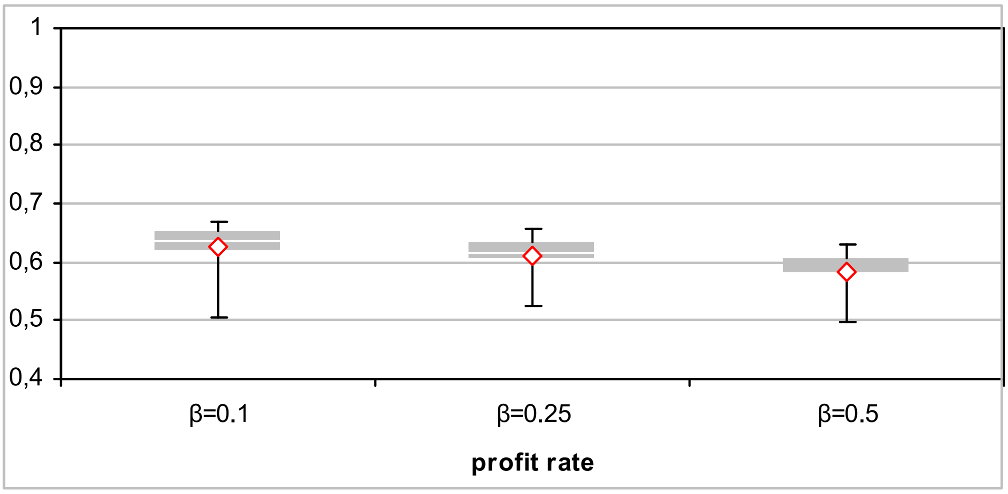

Figure 5 (b). Weighted bargaining values at different profit margin levels (γ =0.010)

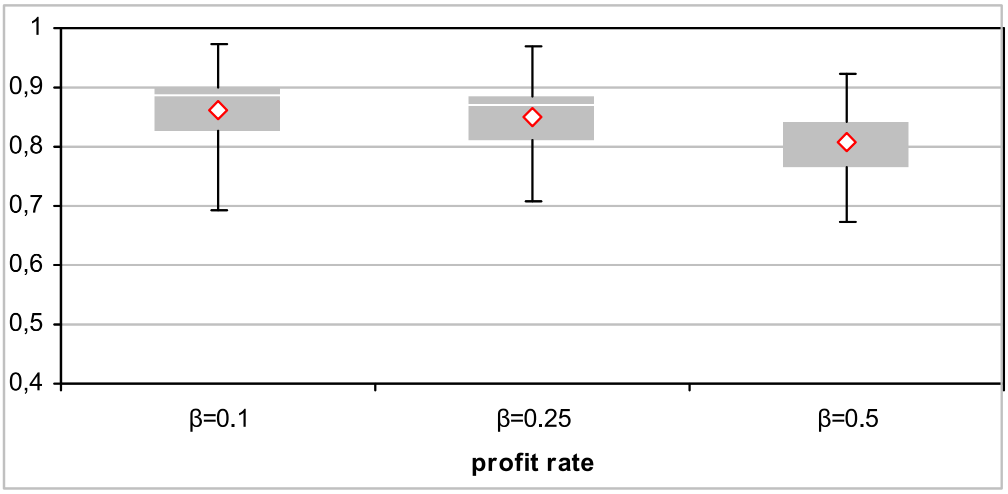

Figure 5 (c). Weighted bargaining values at different profit margin levels (γ =0.100)

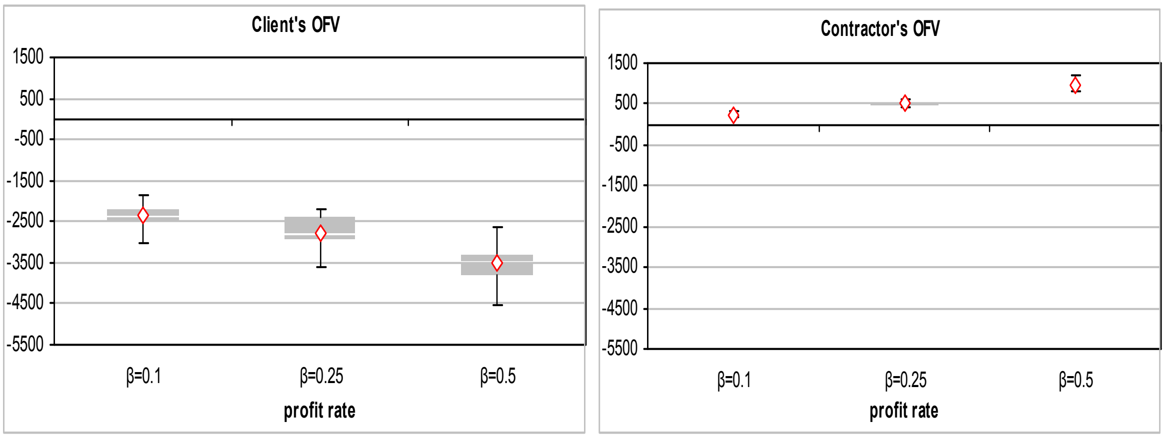

Figure 6. Objective function values at different profit margin levels (γ=0.010)

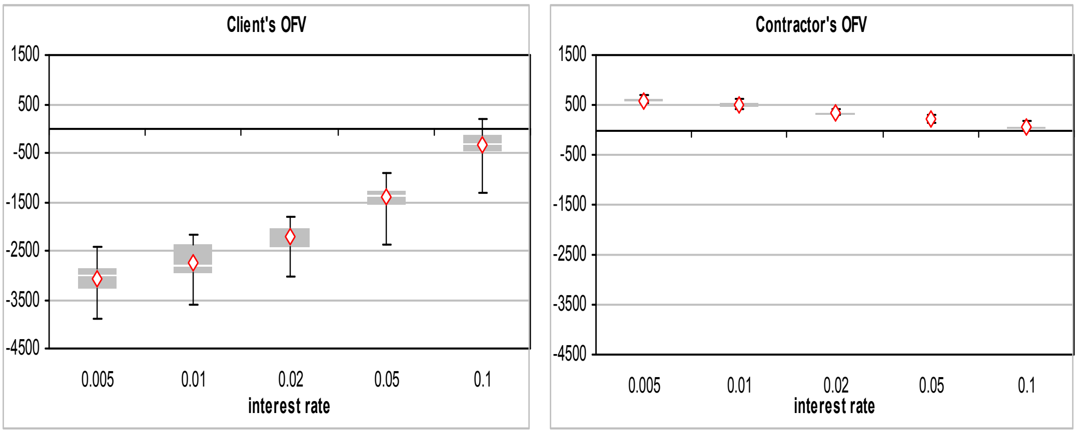

Figure 7. Objective function values at different discount rate levels (β=0.25)

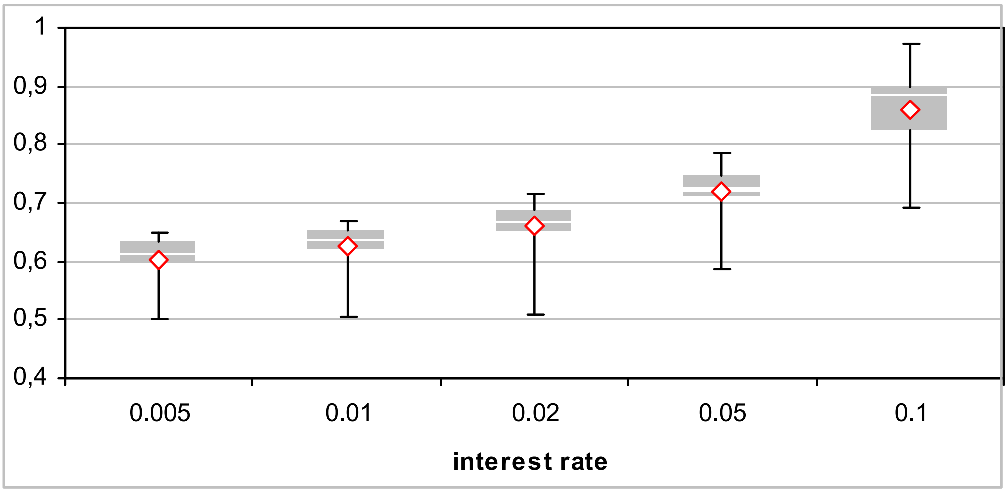

Figure 8 (a). Weighted bargaining values at different discount rate levels (β=0.10)

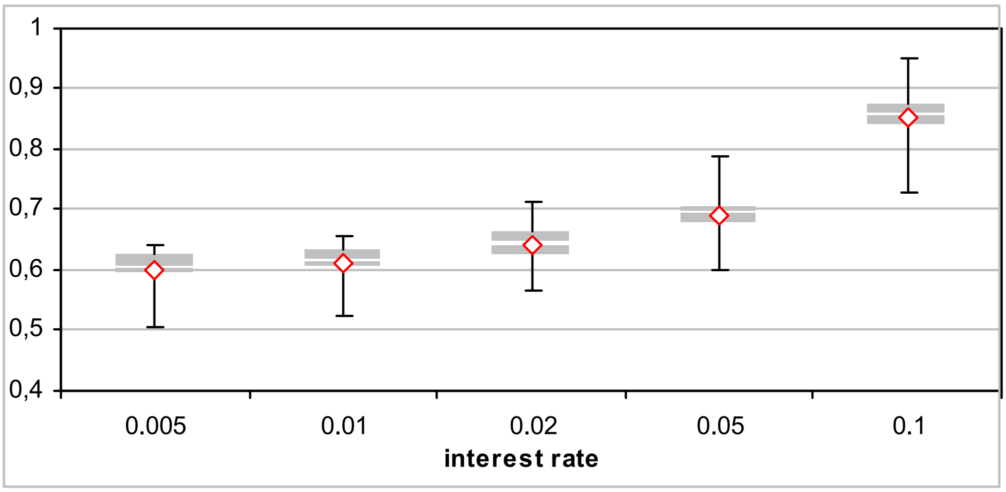

Figure 8 (b). Weighted bargaining values at different discount rate levels (β=0.25)

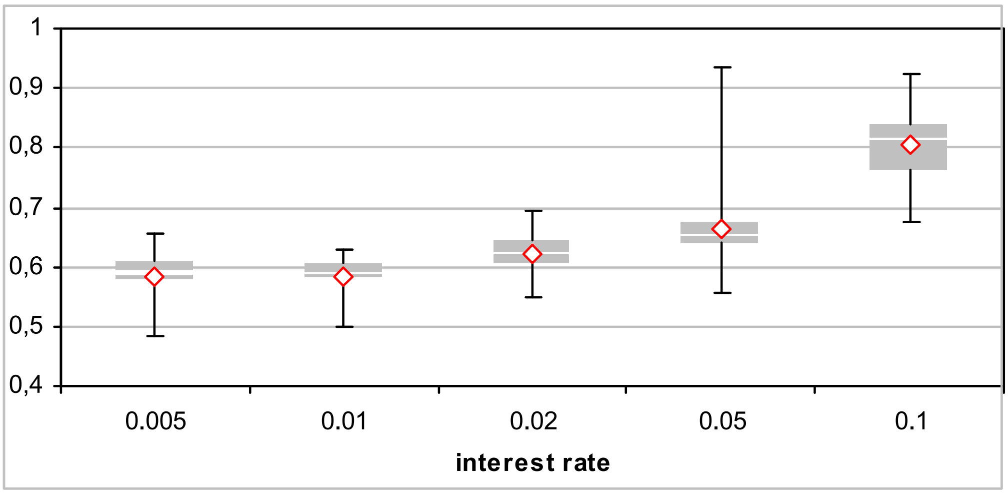

Figure 8 (c). Weighted bargaining values at different discount rate levels (β=0.50)

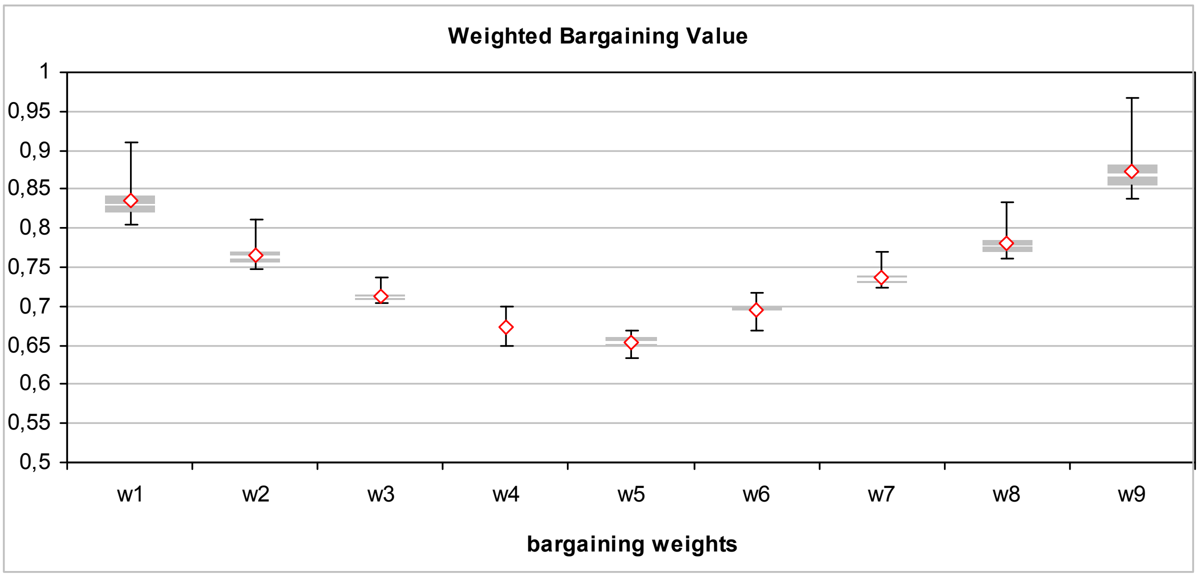

Figure 9. Weighted bargaining values vs. bargaining power weights (γ=0.020)

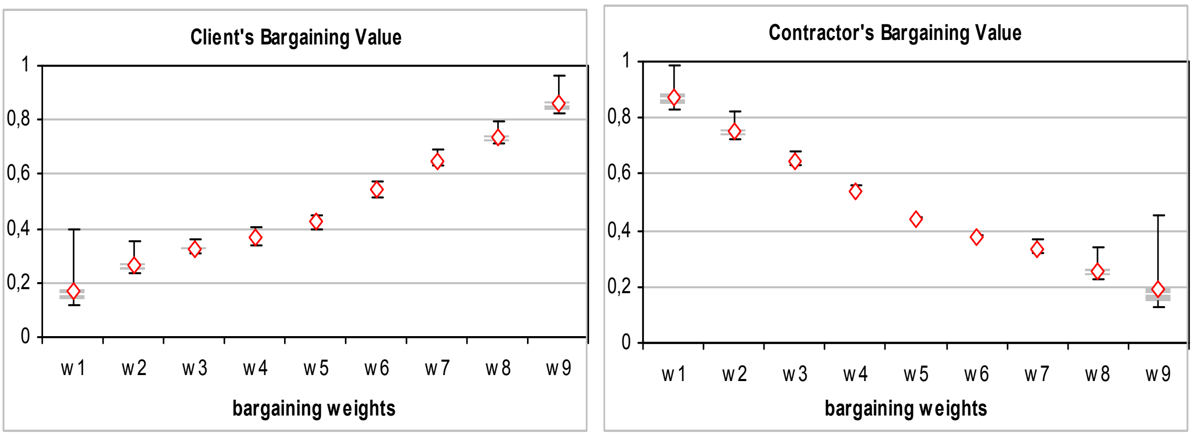

Figure 10. Bargaining values of the players vs. bargaining power weights

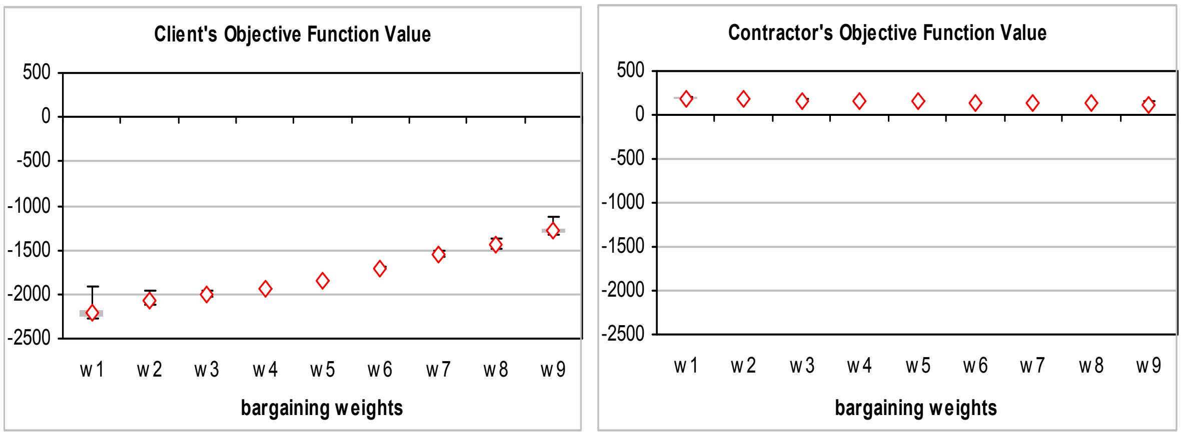

Figure 11. Objective function values of the players vs. bargaining power weights

Table 1. Percentage of feasibility hits for each problem set

	14-activity problems	20-activity problems	32-activity problems
Number of mode lists generated for each activity list	60	100	140
Number of idle time lists generated for each activity list	60	100	140
Percentage of feasible solution hits	16%	10%	7%

Table 2. Iteration counts for each problem set

	14-activity problems	20-activity problems	32-activity problems
maxIter	1000	2000	5000

Table 3. Average percent deviations from the optimal solutions and *t*-test results

	% deviations						**T-Test** (p=0.05)		
	SA			**GA**					
	14 activity network	20 activity network	32 activity network	14 activity network	20 activity network	32 activity network	14 activity network	20 activity network	32 activity network
Progress payments at 10 time periods	6%	10%	16%	5%	9%	15%	**S**	**S**	**S**
Progress payments at 5 time periods	4%	8%	14%	3%	7%	13%	**S**	**S**	**S**
Payments at activity completion	1%	6%	12%	1%	5%	12%	**IS**	**IS**	**IS**

S: Superiority of GA results over SA results is statistically significant
IS: Superiority of GA results over SA results is statistically insignificant

Table 4. Average CPU times

	CPU times for SA (seconds)	**CPU times for GA** (seconds)
14-activity problems	63	74
20-activity problems	239	265
32-activity problems	529	564

Table 5. Mode structure of the activities in the example problem

		activities											
		2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
mode 1	**duration**	3	2	3	5	3	3	4	1	5	2	4	3
	cost	130	200	440	110	560	260	98	390	400	250	340	189
	resource 1	10	7	3	6	1	8	2	2	3	0	0	9
	resource 2	0	0	0	0	0	0	0	0	0	10	1	0
mode 2	**duration**	8	6	3	6	3	5	5	4	8	7	8	3
	cost	80	90	210	78	460	179	45	289	320	147	218	103
	resource 1	9	0	0	4	0	7	0	0	0	0	9	0
	resource 2	0	8	8	0	7	0	7	7	8	9	0	6
mode 3	**duration**	10	10	5	8	4	9	8	9	10	10	9	10
	cost	67	55	100	34	290	98	23	160	170	110	178	65
	resource 1	7	0	0	0	0	0	1	1	0	0	4	0
	resource 2	0	6	4	6	6	10	0	0	7	8	0	4

Table 6. Player with the worse weighted bargaining value

Discount rate γ=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	Client	Client	Client	Contractor	Client	Client

Table 7. Objective function values at different profit margin levels

Objective Function Values		β=0.10	β=0.25	β=0.50
γ=0.005	client	-2650	-3075	-4175
	contractor	286	591	1096
γ=0.010	client	-2339	-2756	-3516
	contractor	229	497	951
γ=0.100	client	-228	-346	-542
	contractor	-58	53	178

Table 8. Objective function values at different discount rate levels

Objective Function Value		γ=0.005	γ=0.01	γ=0.02	γ=0.05	γ=0.10
β=0.10	client	-2650	-2339	-1846	-950	-228
	contractor	286	229	149	31	-58
β =0.25	client	-3075	-2756	-2233	-1413	-346
	contractor	591	497	341	203	53
β =0.50	client	-4175	-3516	-2802	-1720	-542
	contractor	1096	951	730	387	178

Table 9. Summary of the bargaining power weighted tests (γ=0.05, β=0.10)

			Bargaining Power Weight Value	Bargaining Value	Weighted Bargaining Value	Objective Function Value f_A, f_B
w1	client	w(A)	0.1	0.186	0.845	-1055
	contractor	w(B)	0.9	0.827	0.843	50
w2	client	w(A)	0.2	0.313	0.793	-1015
	contractor	w(B)	0.8	0.765	0.807	46
w3	client	w(A)	0.3	0.376	0.746	-995
	contractor	w(B)	0.7	0.663	0.750	40
w4	client	w(A)	0.4	0.423	0.709	-980
	contractor	w(B)	0.6	0.559	0.706	34
w5	client	w(A)	0.5	0.491	0.701	-958
	contractor	w(B)	0.5	0.496	0.704	30
w6	client	w(A)	0.6	0.559	0.705	-937
	contractor	w(B)	0.4	0.437	0.718	26
w7	client	w(A)	0.7	0.644	0.735	-910
	contractor	w(B)	0.3	0.375	0.745	23
w8	client	w(A)	0.8	0.761	0.804	-873
	contractor	w(B)	0.2	0.299	0.785	18
w9	client	w(A)	0.9	0.839	0.854	-849
	contractor	w(B)	0.1	0.199	0.851	12

Table 10. Schedules at different bargaining power values

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

Table 11. Weighted bargaining values vs. bargaining power weights

Weighted Bargaining Values		w1	w2	w3	w4	w5	w6	w7	w8	w9
γ=0.005	client	0.810	0.738	0.672	0.619	0.583	0.616	0.672	0.734	0.819
	contractor	0.811	0.720	0.678	0.608	0.578	0.620	0.667	0.726	0.824
γ =0.010	client	0.817	0.743	0.682	0.631	0.606	0.634	0.679	0.754	0.840
	contractor	0.826	0.752	0.694	0.626	0.611	0.632	0.682	0.734	0.829
γ =0.020	client	0.831	0.759	0.705	0.659	0.648	0.677	0.716	0.755	0.842
	contractor	0.838	0.758	0.704	0.668	0.645	0.660	0.705	0.751	0.838
γ=0.050	client	0.845	0.793	0.746	0.709	0.701	0.705	0.735	0.804	0.854
	contractor	0.843	0.807	0.750	0.706	0.704	0.718	0.745	0.785	0.851
γ =0.100	client	0.948	0.910	0.880	0.852	0.836	0.860	0.883	0.912	0.954
	contractor	0.957	0.904	0.881	0.847	0.839	0.852	0.880	0.917	0.953

Table 12. Direction of change of the objective function value with changing parameters

	profit margin *(β)* increases	discount rate *(γ)* increases	bargaining power weight *w(A)* increases	bargaining power weight *w(B)* increases
objective function value of the client *($f_A(x)$)*	↓	↑	↑	↓
objective function value of the contractor *($f_B(x)$)*	↑	↓	↓	↑

Table 13. 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 = 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 = 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 = 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