A Preference-Based Bi-Objective Approach to the Payment Scheduling Negotiation Problem with the Extended r-Dominance and NSGA-II

Wei-neng Chen and Jun Zhang (Corresponding Author)
Department of Computer Science, Sun Yat-sen University, P.R. China
Key Laboratory of Digital Life, Ministry of Education, P.R. China
Key Laboratory of Software Technology, Education Dept. of Guangdong Province, P.R. China
junzhang@ieee.org

ABSTRACT
This paper addresses a complicated problem in project management termed the payment scheduling negotiation problem (PSNP). The problem is a practical extension of the classical multi-mode resource constrained project scheduling problem (MRCPSP) and it considers the financial aspects of both the project client and contractor in a contracting project. The client and contractor negotiate with each other to determine an optimal payment schedule and an activity schedule so as to maximize their net present values (NPVs). As the NPV of the client and the NPV of the contractor are conflicting objectives, this paper first formulates the PSNP as a bi-objective optimization problem. To solve this problem effectively, a non-dominated sorting genetic algorithm II (NSGA-II) approach is proposed. In the negotiation, the client and contractor may have two preferences: the ideal NPVs for the client and the contractor, and the optimization degree of the activity schedule. In order to tackle these preferences, this paper further introduces a new dominance relation named the extended r-dominance (er-dominance) relation. The er-dominance relation extends the r-dominance relation and is able to deal with multiple preferences described by aspiration functions. Experimental results show that by incorporating the NSGA-II with the er-dominance, the proposed approach is promising for the PSNP.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]:Problem Solving, Control Methods and search-Heuristic methods;

General Terms
Algorithms, Experimentation.

Keywords
Multi-objective Optimization, Preference, NSGA-II, Payment Scheduling, Project Scheduling

GECCO'12, July 7-11, 2012, Philadelphia, Pennsylvania, USA.

1. INTRODUCTION
The resource constrained project scheduling problem (RCPSP) is important and challenging in project management [1]. It involves scheduling the activities of a project subject to precedence and resource constraints. The classical RCPSP has been proven to be NP-complete [2]. In a more complicated and practical form of RCPSP which termed the multi-mode RCPSP (MRCPSP), each activity can be implemented in different alternative processing modes [3]. Kolisch [4] proved that finding a feasible solution for the MRCPSP with more than one nonrenewable resource is already NP-complete. Due to the importance and difficulty of the problem, RCPSP has been widely studied and various methods including exact algorithms [5], heuristic approaches [6] and metaheuristic approaches [7][8] have been proposed.

Traditional studies of RCPSP usually aim at minimizing the makespan of a project [1]. During the last decade, the research into considering financial aspects in project scheduling has attracted increasing attention [9]-[11]. The most commonly used financial criterion is the net present value (NPV) of discounted cash flows [12]. It is evaluated by summating the present values of all cash inflows (positive values) and cash outflows (negative values). With the NPV criterion, the objective function becomes nonlinear, making the scheduling problem even more complicated. Several works have been done on proposing metaheuristic approaches to the MRCPSP with the NPV criterion, e.g., the genetic algorithm (GA) approach by Ulusoy [9], the simulated annealing (SA) and tabu search (TS) approaches by Mika [10], and the ant colony optimization (ACO) approach by the authors [11].

In the above-mentioned studies, the optimization objective is to maximize the NPV of the project contractor. However, in a contracting project, cash flows are involved by two players: client and contractor. The client determines a schedule for payments, while the contractor devises a schedule for processing the activities of the project. Both the schedules significantly influence the cash inflows and outflows of the two players. To achieve optimal NPVs for both the client and contractor, the two players have to negotiate with each other to find satisfying schedules. This negotiation activity results in a practical and very complicated problem in project scheduling – the payment scheduling negotiation problem (PSNP).

According to the latest survey [3], the PSNP has only been considered in three papers. Ulusoy [13] first introduced the payment scheduling negotiation problem and developed a double-loop GA approach. The double-loop GA uses an outer-loop GA to optimize the payment schedule of the project for the interest of the project client. To evaluate the fitness of each payment schedule generated by the outer-loop GA, an inner-GA is used to optimize the activity schedule and maximize the NPV of the project contractor. Although the working process of the double-loop GA fits the negotiation process of the client and contractor in a contracting project, the algorithm is too time-consuming. Because each objective function evaluation in the outer-loop GA is an inner-loop GA, for medium to large scale projects, the double-loop GA has to run for weeks to return a solution. In [14][15], He *et al*. proposed simulated annealing based approaches to the multimode project payment scheduling problem. But they relaxed the project model by ignoring resource constraints, which makes the model less practical. Overall, designing practical optimization models and effective algorithms for PSNP is still an emerging topic and more works are needed to help the client and contractor plan payment and activity schedules more efficiently [3].

In this paper, we intend to propose a novel approach to the PSNP by viewing the problem as a bi-objective optimization problem. Multi-objective optimization is a rapid developing research topic in optimization and operations research [16]. Different from single-objective optimization, multi-objective optimization treats various optimization objectives of the problem simultaneously. Instead of obtaining a single solution, the goal of multi-objective optimization is to find a set of tradeoff solutions to facilitate decision making. In the PSNP, the project client has to determine the payment schedule and the project contactor has to determine the activity schedule. Both of these two players want to optimize their own cash flows, but the NPV of the client and the NPV of the contractor are usually conflicting. Therefore, it is natural to regard the maximization of the client's NPV and that of the contractor's NPV as two optimization objectives. The payment schedule (time and amount of payments) and the activity schedule (when and how to implement activities) are the decision variables to be optimized. In this way, the PSNP can be formulated as a bi-objective optimization problem. The benefit of the bi-objective formulation is that it can consider the interests of the client and contractor in a more fair and comprehensive way. As it provides a set of tradeoff solutions, the client and contractor can have better information to negotiate and achieve maximum integrated utility.

In order to solve the PSNP, this paper also develops an evolutionary multi-objective optimization (EMO) approach. Because evolutionary computation algorithms can provide a set of solutions in a single run, and they are insensitive to the mathematical properties of the objective functions of the problem, multi-objective evolutionary algorithms (MOEAs) have been found to be suitable and very promising for solving multi-objective optimization problems (MOPs) [16][17]. Some well-known MOEAs include the non-dominated sorting GA II (NSGA-II) [16], the strength Pareto EA II (SPEA2) [18], the Pareto archived evolutionary strategy [19], and the MOEA based on decomposition (MOEA/D) [20]. In this paper, we develop an EMO approach to the PSNP based on NSGA-II. Because the algorithm can take both the optimization objectives into account in a single run, compared with the double-loop GA algorithm [13], the computational cost of the EMO approach can be significantly reduced.

Traditional EMO approaches usually aim at approximating the whole Pareto front (PF) of a problem. However, in practice of the PSNP, not all the solutions in the PF are of sense for the client and contractor. For example, the payment schedule that arranges all the payments to the beginning of the project is good for the contractor, but is unacceptable for the client. Only a subset of solutions in the PF that are acceptable and preferred by both the client and the contractor belong to the region of interest (ROI) [17] of the problem. To focus on searching for the ROI instead of the whole PF, we further develop a preference-based EMO for the PSNP. Preference-based EMO is a new direction in EMO research and has attracted increasing emphasis in recent years [21]-[22]. Usually, preferences can be integrated in EMO algorithms in three ways: a priori, a posteriori, and interactively [17]. Preference information is usually given by weights of objectives, solution ranking, reference point, reference direction and preference thresholds [17]. In PSNP practice, the project client and contractor may specify their ideal NPVs. Such ideal NPVs can be viewed as a reference point of the problem. In addition, given a certain payment schedule, the contractor will always try to optimize the activity schedule to maximize his NPV. Therefore, the solutions with better optimized activity schedules are usually more stable and preferred in negotiation. In order to address both preferences, this paper further introduces an extended r-dominance (er-dominance) relation based on the r-dominance relation developed by Said *et al.* [17]. The er-dominance relation is capable of tackling multiple preferences described by aspiration functions. By incorporating the NSGA-II approach with the er-dominance, the proposed approach is able to deal with the two kinds of preferences simultaneity. Experimental results on 20 instances demonstrate the effectiveness of the proposed approach.

The rest of this paper is organized as follows. Section 2 introduces the PSNP. Section 3 presents the bi-objective approach based on the NSGA-II. Section 4 further proposes the preference-based EMO approach by introducing the extended r-dominance relation. Experimental results are given in Section 5. Conclusions are finally drawn in Section 6.

2. PROBLEM DESCRIPTION

In the traditional research on project scheduling with financial aspect, only the NPV of the contractor is considered. But in the PSNP, both the client and the contractor are involved. The client needs to determine a payment schedule, i.e., the timing and the amount of the payments that he is going to pay to contractor for project implementation. The contractor needs to determine an activity schedule for processing the activities of the project. Both the payment and activity schedules have a significant influence on the NPVs of the client and the contractor. Both the client and the contractor intend to maximize their own NPVs. But the increase of one's NPV usually induces the decrease of the other one's NPV. The detailed description for this problem is as follows.

2.1 Project Network

The project network defines the precedence relations among the activities of the project. A project network is typically a directed acyclic graph. There are two types of commonly-used project networks, i.e., the activity-on-node (AoN) network and the activity-on-arc (AoA) network. In this paper, the AoA network is adopted for project description [11].

An AoA network $G=(E,A)$ is a graph where the node set $E=\{e_1, e_2, \ldots, e_m\}$ corresponds to the set of events of the project and the arc set $A=\{a_1, a_2, \ldots, a_n\}$ corresponds to the set of activities. Here m is the number of events and n is the number of tasks in the project. Based on the AoA network, the precedence constraint of the project is defined as follows: a) Event e_i takes place as soon as all direct predecessor activities of e_i have finished. b) Only after the occurrence of e_i can the direct successor activities of e_i start.

2.2 Resource Constraints

Renewable resource constraint is considered in this paper. We assume that R types of resources are used in the project. For each time period during the processing course of the project, the number of the available resources of the k-th type (i=1,2,…,R) is limited by r_k.

2.3 Multi-Mode

Each activity a_i(i=1,2,…,n) can be processed by any mode out of a finite alternative mode set $M_i = \{m_{i1}, m_{i2}, ..., m_{i|M_i|}\}$, where m_{ij} is the j-th processing mode for a_i and $|M_i|$ is the total number of available modes for a_i [11].

Different modes for a_i represent the time/cost/resource tradeoffs for the processing of a_i. That is, to implement a_i, different modes may consume different amounts of time, cost and resources. For a mode m_{ij}, we denote its duration as $m_{ij}.d$, cost as $m_{ij}.c$, and consumption of the k-th resource as $m_{ij}.r_k$. To schedule projects with multi-mode, we need to map each activity a_i to an execution mode from M_i for activity processing.

2.4 The Payment Model

In the considered problem, the commonly-used payment at event occurrences (PEO) [9] model is adopted. In this payment model, payments are occurred at events. To build a payment schedule, the client has to specify a payment list

$$\text{payment list}: (pay_1, pay_2, ..., pay_m) \tag{1}$$

where pay_i means the percentage of payments that the client plans to pay to the contractor at event i. The payment list must satisfy the following two constraints:

$$\sum_{i=1}^{m} pay_m = 1 \tag{2}$$

$$pay_1 \geq prepayRate \tag{3}$$

Here, the equation (2) means that the summation of all the payment percentages must equal to 1, so that the total payment equals to the total value U of the project in the contract. The inequation (3) means that the payment at the beginning of the project (event e_1) must be not smaller than a predefined prepayment rate $prepayRate$.

2.5 NPV of the Client

After the payment schedule and the project schedule have all been determined, the NPV of the client can be estimated.

We denote the start time of each activity a_i(i=1,2,…,n) as $a_i.st$, and the time when the event e_i(i=1,2,…,m) occurs is denoted as $e_i.t$. The NPV of the client NPV_{client} is given by

$$NPV_{client} = -\sum_{i=1}^{m} pay_i \cdot U \cdot \exp(-\alpha \cdot e_i.t) + revenue \cdot \exp(-\alpha \cdot e_m.t) \tag{4}$$

Here the first item of the right side of the equation means the cash outflows of the client due to payments. The second item means the expected *revenue* of the client after the project is complete. It is important to note that all the cash inflows and outflows have to be discounted to their present values according to the discounted rate α.

2.6 NPV of the Contractor

For the contractor, his cash inflows are derived from the payments from the client. The cash outflows are mainly caused by the expenditure of activity processing. Thereby, suppose the start time of a_i is $a_i.st$ and the time when e_i occurs is $e_i.t$, the NPV of the contractor NPV_{cont} is given by

$$NPV_{cont} = \sum_{i=1}^{m} pay_i \cdot U \cdot \exp(-\alpha \cdot e_i.t) - \sum_{i=1}^{n} m_{ij}.c \cdot \exp(-\alpha \cdot a_i.st) \tag{5}$$

where m_{ij} is the mode for processing a_i in the project schedule, and $m_{ij}.c$ is the cost of m_{ij}.

2.7 Optimization Goal

Based on the above discussions, the goal of the PSNP is to find a payment schedule in the form of (1) and an activity schedule that maps each activity a_i to an available execution mode $m_{ij} \in M_i$ and specifies the start time and end time of all the activities, so that the NPV of the client given in (4) and the NPV of the contractor given in (5) are maximized.

3. THE BI-OBJECTIVE APPROACH WITH THE NSGA-II

3.1 Background of Multi-objective Optimization

To present the proposed approach, here we first briefly introduce the background and basic concepts in multi-objective optimization. Let us suppose

$$\text{maximize } F(x) = (f_1(x), f_2(x), ..., f_K) \quad x \in \Omega \tag{6}$$

is a MOP with K objectives, where x is the decision variable to be optimized, and Ω is the definition domain of x. As the objectives usually contradict each other, it is impossible to find a solution that optimizes all the objectives simultaneously. Therefore, the goal of MOP is usually to find a set of tradeoff solutions that balance the objectives. A commonly used criterion for evaluating the quality of a solution is the *Pareto dominance* relation.

Definition 1 (*Pareto Dominance*): Let $u, v \in \Omega$ are two feasible solutions to the problem, u is said to Pareto dominate v if and only if $f_i(u) \geq f_i(v)$ for all objectives (i=1,2,…,K), and $f_j(u) > f_j(v)$ for at least one objective $j \in \{1,2,...,K\}$.

Based on this definition, if u dominates v, we usually consider u is better than v. If there is no other solution from Ω that can dominate a solution x^*, we call x^* a *Pareto optimal* solution. The set of all Pareto optimal solutions is call the *Pareto set (PS)*. The objective function vectors of all the Pareto optimal solutions form the *Pareto front (PF)* [20].

Taking the bi-objective formulation of the PSNP for example, we show the objective function vectors of some feasible solutions in Fig. 1. In the figure, the point A Pareto dominates B, as A yields better NPVs for both the client and the contractor. The points A and C are Pareto equivalent, as A achieves better NPV for the contractor, and C achieves better NPV for the client. The points marked by solid black circles in the figure form the PF of the instance.

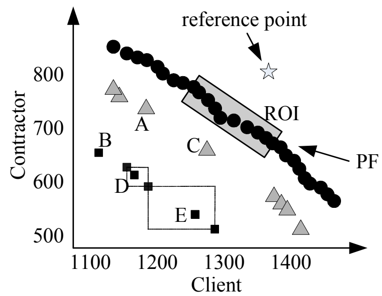


Figure1. An illustration of multi-objective optimization

3.2 Background of NSGA-II

To find a set of solutions for a MOP, we usually need to address two issues: 1) how to improve the quality of the solutions so that they converge to the Pareto optimal set; and 2) how to maintain the diversity of the solutions so that they cover the whole PF. Various MOEAs have been developed for solving MOPs. One famous MOEA is the non-dominated sorting GA II (NSGA-II) proposed by Deb *et al*. [16].

To overcome the above two issues, NSGA-II introduces two methods, i.e., non-dominated sorting method and crowding-distance assignment. The non-dominated sorting method divides solutions into different non-domination levels based on the Pareto dominance relation among the solutions. For example, in Fig. 1, we suppose that the points marked by triangles and squares are the solutions found by the algorithm. In this case, the solutions marked by triangles belong to the first non-domination level, as no other solution found by the algorithm can dominate any of these solutions. The solutions marked by squares belong to the second non-domination level. Obviously, the solutions belonging to the first level are considered to be more promising than the solutions in higher levels. For the solutions in the same non-domination level, the crowding-distance assignment method is applied to estimate the density of solutions surrounding those solutions and the solutions with lower density are preferred. For example, in Fig. 1, the solution E is considered to be better than D as the density of the solutions surrounding E is much lower.

For detailed information of the non-dominated sorting and the crowding-distance assignment methods, please refer to [16]. Based on these methods, the NSGA-II has been found to be promising for solving MOPs. In this paper, we propose a NSGA-II based approach for the considered PSNP.

3.3 Encoding Scheme

In the considered PSNP, both the payment schedule and the project schedule have to be optimized. To represent all the decision variables to be optimized, a three-section encoding scheme is designed. A solution to the considered problem is encoded by

$$\begin{aligned}
&\text{payment list}: (share_1, share_2, ..., share_{20}) \\
&\text{activity list}: \ (act_1, act_2, ..., act_n) \\
&\text{mode list}: \quad (mod_1, mod_2, ..., mod_n)
\end{aligned} \tag{7}$$

Here, ($share_1$, $share_2$, …, $share_{20}$) is an integer encoding of the payment list given in (1). In this representation scheme, the total payment (except for the minimal prepayments) is divided into 20 shares. (We can also divide it into 30 or more shares. But 20 shares are usually enough for many projects [13].) Each share means 5% of the total payment. If we assign $share_i$ to the event e_j (denoted as $share_i$=e_j), it means that this 5% of the toal payment will occur at e_j. In this way, the presentation ($share_1$, $share_2$, …, $share_{20}$) can be transformed to the standard form of the payment list (i.e., the form of (1)) as follows:

$$pay_j = \begin{cases} \sum_{i=1}^{20} \Gamma_j(i) \cdot 5\% \cdot (1 - prepayRate) + prepayRate, & \text{if } j = 1 \\ \sum_{i=1}^{20} \Gamma_j(i) \cdot 5\% \cdot (1 - prepayRate), & \text{otherwise} \end{cases}$$

$$\text{where } \Gamma_j(i) = \begin{cases} 1, & \text{if } share_i = e_j \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

In (7), the activity list specifies the priority of activities to consume resources. The mode list specifies the processing mode of each activity. Based on the activity list and the mode list, the serial schedule generation scheme (SSGS) [8][11] can be applied to obtain the actual schedule (i.e., the start time and end time of each activity and the occurrence time of each event). (For more information about the SSGS, please refer to the references [8][11].)

3.4 Selection

According to Deb [16], the binary tournament selection operator is applied in the NSGA-II algorithm.

3.5 Crossover

In the proposed algorithm, individuals are selected to perform crossover with a probability *px*. The crossover operator can be performed on either the payment list, the activity list or the mode list randomly.

For the crossover on payment list and mode list, the classical one-point crossover operator is applied.

For the crossover on activity list, in order to guarantee that the newly generated activity lists can always obey the precedence constraints defined by the AoA network, a one-point order-based crossover operator is applied. The basic idea of this crossover operator is that a) for the positions before the randomly selected crossover point, the activities remain unchanged; and b) for the positions after the crossover point, the activities are rearranged based on their order of appearances in the other parent. An example of this crossover operator is illustrated in Figure. 2.

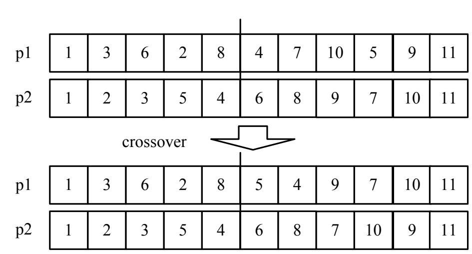


Figure 2. The crossover operator for activity list

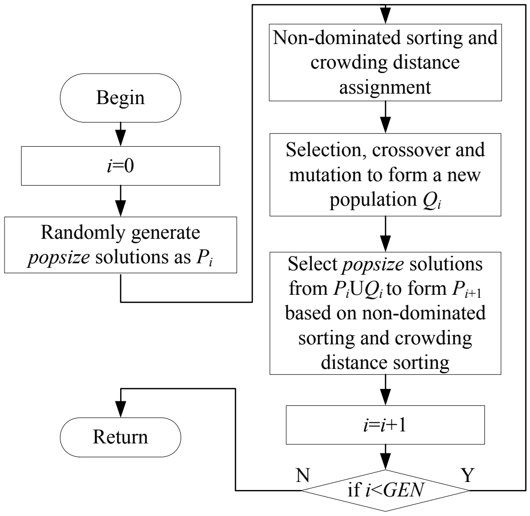


Figure 3. Flowchart of the NSGA-II algorithm

3.6 Mutation

In the algorithm, the genes are selected to perform mutation with a probability of *pm*.

- If a gene $share_i$ belonging to the payment list is selected to mutate, $share_i$ is randomly assigned to an event e_{ran}, where *ran* is a random number uniformly distributed in the set $\{1,2,...,m\}$.
- If a gene act_i belonging to the activity list is selected to mutate, we swap the values of act_i and act_{i+1} if the precedence constraints are not violated. Otherwise, the chromosome remains unchanged.
- If a gene mod_i belonging to the mode list is selected to mutate, the value of mod_i is reset to a random mode $m_{i,ran}$, where ran is a random number uniformly distributed in the set $\{1,2,...,|M_i|\}$.

3.7 Overall Flowchart of the NSGA-II

Based on the above encoding schemes and operators, the overall flowchart of the NSGA-II algorithm for the considered problem is given in Fig. 3. At the beginning, a population of *popsize* solutions are randomly generated as P_0. Then these solutions are evaluated and the non-dominated sorting and crowding-distance assignment method are applied on these solutions. The selection, crossover and mutation operators of GA are performed on these solutions to form a new population Q_0. Based on the results of non-dominated sorting and crowding-distance sorting, a number of *popsize* solutions are selected from the set $P_0 \cup Q_0$ to form a new generation of population P_1. These procedures run iteratively until the number of generations has reached a predefined maximum generation number *GEN*.

4. THE PREFERENCE-BASED APPROACH

4.1 Preferences in the PSNP

In the practice of multi-objective decision making, the decision makers usually do not concern the whole Pareto set of the problem, but only a preferred subset of the Pareto set. To facilitate decision making, the research into preference-based multi-objective optimization has attracted increasing attention in recent years [21]-[22]. Instead of wasting time on searching for the unpreferred Pareto optimal solutions, preference-based approaches only focus on searching for the ROI. In this way, they can usually provide better and more preferred solutions for decision making.

In the PSNP considered in this paper, there are two kinds of preference information:

4.1.1 Ideal NPVs

In a contracting project, the project client and contractor usually have their expected ideal NPVs. We denote the ideal NPVs of the client and the contractor as $ideal_{client}$ and $ideal_{cont}$, respectively. These two ideal NPVs ($ideal_{client}$, $ideal_{cont}$) can be viewed as a reference point in multi-objective decision making.

An example of the reference point and the related ROI of the PF is shown in Fig. 1. In the figure, the point marked by a star is the reference point. As the point is given by the ideal NPVs of the client and contractor, it is usually impossible to achieve the ideal values for both the players at the same time. Therefore, the reference point is usually out of the feasible area in the objective function vector space bounded by the PF. However, this reference point is of significance as it specifies a ROI for the problem. The solutions that are close to the reference point ($ideal_{client}$, $ideal_{cont}$) are considered to be more preferred in decision making.

4.1.2 Optimization Degree of the Activity Schedule

In addition to the ideal NPVs, the optimization degree of the activity schedule under a certain payment schedule is also preference information for decision making.

In the PSNP, the client determines the payment schedule given in the form of (1), and the contractor determines the activity schedule given by the activity list and the mode list (7). As both the client and the contractor want to optimize their own NPVs and achieve maximal integrated utility, it is reasonable that they cooperate together to negotiate and make the final decision of payment and activity schedules. However, in practice, the payment schedule is usually determined by the client first, and then the contractor determines the activity schedule. If the activity schedule is not well optimized, the resulting solution is not likely to be preferred, because a rational contractor can change the plan and use a better optimized activity schedule to obtain higher NPVs for his own. Therefore, in negotiation, only the solutions that optimize the activity schedule to an acceptable degree are preferred. Given a certain payment schedule, the problem of optimizing the activity schedule to maximize the NPV of the contractor is just the MRCPSP with discounted cash flows considered in traditional research [9]-[11]. To address this preference information, we should guarantee that the activity schedule in the solution is well optimized in terms of the MRCPSP with discounted cash flows.

As mentioned in Section I, the MRCPSP is strongly NP-hard. Thus it is not easy to judge how good the activity schedule is optimized. In order to model this preference information for decision making, we develop a criterion for evaluating the optimization degree of a solution based on the difference between the upper bound value of the contractor' NPV and the actual NPV achieved by the activity schedule. Given a solution x with a payment schedule PS (in the form of the payment list in (7)) and an activity schedule AS (in the form of the activity and mode lists in (7)), its optimization degree of the activity schedule is evaluated as follows.

Step 1): Evaluating the upper bound $NPV_{cont}^{upper/PS}$ of the contractor's NPV when the payment schedule PS is fixed.

The upper bound is obtained by

i) ignoring all the resource constraints in the problem;

ii) supposing all events happen at their earliest possible time, so that the payments occur as early as possible;

iii) supposing all activates begin at their latest start time without delaying the project, so that the expenditures occur as late as possible; and

iv) supposing every activity is executed by the mode with the lowest cost

Step 2): Calculating the difference between the upper bound $NPV_{cont}^{upper/PS}$ and the actual NPV of the contractor NPV_{cont}

$$OD(x) = NPV_{cont}^{upper/PS} - NPV_{cont} \tag{9}$$

If the optimization degree value $OD(x)$ is small, it means that the activity schedule in the solution x is well optimized. Otherwise, the activity schedule in x is not optimized well and thus the solution x is not preferred in the decision making.

4.2 The Extended r-Dominance

To consider preference information in multi-objective optimization, various preference-based MOEAs have been proposed recently [17][21][22]. One promising approach is to modify the Pareto dominance relation [17][21]. In a recent work of Said *et al*. [17], in order to incorporate the preference information given by a reference point, a novel dominance relation named the reference solution-based dominance (*r-dominance*) is developed.

Definition 2 (*r-Dominance*): Let $u, v \in \Omega$ are two feasible solutions, and *ref* is a reference point, u is said to r-dominate v if and only if one of the following statements holds true:

i) u Pareto dominates v;

ii) u and v are Pareto equivalent and $D(u, v, ref)<-\delta$, where $\delta \in [0,1]$ is a parameter called the *non-r-dominance threshold*, and the function $D(u, v, ref)$ is defined as

$$D(u,v,ref) = \frac{Dist(u,ref) - Dist(v,ref)}{Dist_{\max} - Dist_{\min}} \tag{10}$$

$$Dist_{\max} = \max_{x \in POP} Dist(x,ref) \tag{11}$$

$$Dist_{\min} = \min_{x \in POP} Dist(x,ref) \tag{12}$$

where POP is the population of solutions, $Dist(x, ref)$ is the *weighted Euclidean distance* (WED) between the solution x and the reference point *ref* [22], and $Dist_{\max}$ and $Dist_{\min}$ are the maximum and the minimum WED between the individuals in the population POP and the reference point *ref*, respectively. $Dist(x, ref)$ is given by

$$Dist(x,ref) = \sqrt{\sum_{i=1}^{K} \omega_i \left(\frac{f_i(x) - f_i(ref)}{f_i^{\max} - f_i^{\min}} \right)^2}, \ \sum_{i=1}^{K} \omega_i = 1 \tag{13}$$

where $f_i^{\max}$ and $f_i^{\min}$ are the upper bound and the lower bound of the i-th objective function, respectively, and ω_i is the weight for the i-th objective.

In [17], Said *et al*. also proved the compatibility and completeness of the r-dominance with respect to the Pareto dominance. In addition, by incorporating the r-dominance relation with the NSGA-II, they showed that the extend of the obtained ROIs can be easily controlled by tuning the parameter δ, and the resulted algorithm managed to achieve very promising results compared with other preference-based MOEAs. However, the definition of the r-dominance relation only considers the preference information given by the reference point. In order to consider both the two kinds of preference information for the PSNP, in this paper, we further extend the r-dominance relation to the *extended r-dominance relation (er-dominance)*.

Definition 3 (*er-Dominance*): Let $u, v \in \Omega$ are two feasible solutions to the problem, $AF=\{g_1, g_2, \ldots, g_Q\}$ is a set of aspiration functions, and each of the aspiration function g_j: $\Omega \rightarrow R^+$ is a mapping from the solution space to the real space, specifying one type of preference information for the decision maker (if $g_j(u)<g_j(v)$, we consider u is preferred in terms of the preference information g_j), u is said to er-dominate v if and only if one of the following two statements holds true:

i) u Pareto dominates v;

ii) u and v are Pareto equivalent, $ED(u, v, g_j)\leq\delta$ holds true for all $g_j \in AF$, and $ED(u, v, g_k)<-\delta$ holds true for at least one aspiration function $g_k \in AF$, where $\delta \in [0,1]$ is a parameter and $ED(u, v, g_j)$ is defined as

$$ED(u,v,g_j) = \frac{g_j(u) - g_j(v)}{g_j^{\max} - g_j^{\min}} \tag{14}$$

$$g_j^{\max} = \max_{x \in POP} g_j(x) \tag{15}$$

$$g_j^{\min} = \min_{x \in POP} g_j(x) \tag{16}$$

Based on this definition, we set

$$g_1(x) = Dist(x,ref) \tag{17}$$

$$g_2(x) = OD(x) \tag{18}$$

where $Dist()$ is defined in (13), and $OD()$ is defined in (9). If we set $AF=\{g_1\}$, the preference information given by the reference point in considered. The resulting er-dominance relation is just equivalent to the r-dominance relation in Definition 2. If we set $AF=\{g_1, g_2\}$, then both the two types of preference information are considered. By incorporating this er-dominance relation with the NSGA-II approach, we can address both the preferences for the PSNP.

5. EXPERIMENTAL RESULTS

In order to validate the proposed approach, we study the performance of the following five approaches in the experiment: 1) the double-loop GA proposed by Ulusoy [13] (denoted as D-GA for short); 2) the NSGA-II using the Pareto dominance relation (denoted as PD); 3) the NSGA-II using the r-dominance relation (denoted as r-D); 4) the NSGA-II using the er-dominance relation with $AF=\{g_2\}$ (denoted as er-D(OD)); and 5) the NSGA-II using the er-dominance relation with $AF=\{g_1, g_2\}$ (denoted as er-D(both)). These five approaches are compared on 20 randomly-generated project instances. We name these random instances as “ins1” to “ins20”. The sizes of these instances are shown in Table 1. The parameters in the instances, including the costs, durations and resources of processing modes are all randomly generated. In the experiment, the parameters of the double-loop GA are set according to [13]. The parameters of the NSGA-II and its extensions with the r-dominance and the er-dominance are set empirically as listed in Table 1. In addition, we set the non-r-dominance threshold $\delta = 0.5$ in the comparison. The weights for evaluating the weighted Euclidean distance are set to $\omega_1 = \omega_2 = 0.5$, as both the objective functions (i.e., the client’s NPV and the contractor’s NPV) are in the same magnitude. For every NSGA-II based approach, we perform 20 independent runs on each instance. Because the double-loop GA involves thousands of inner-loop GAs per run, it is very time-consuming and takes several days to perform a single run. We run it for five independent times on each instance.

The experimental results are showed in Table 2. In the table, “avg client” is the mean NPV of the client averaged over all the non-Pareto-dominated solutions found by an algorithm. “avg cont” is that of the contractor. To facilitate comparison, we report the ratios between the averaged NPV of the client (contractor) and the reference value $ideal_{client}$ ($ideal_{cont}$) in the table. We also plot the objective function vectors found by these algorithms on the instances ins1 and ins11 in Fig. 4. From Table 2 and Fig. 4, it can be seen that the “PD” scheme fails to get satisfying NPVs for the contractor. Because the NPV optimization problem for the contractor in the traditional MRCPSP model is strongly NP-hard [4], the optimization for the contractor’s NPV is a relatively more difficult objective in the PSNP. Without preference information, it is not easy for the NSGA-II approach with the conventional Pareto dominance relation to drill into the objective of the contractor’s NPV to yield satisfying solutions. On the other hand, because the strategy in the double-loop GA is to first optimize the contractor’s NPV and then optimize the client’s NPV, it cannot guarantee to yield Pareto optimal solutions in all cases. Therefore, compared to the double-loop GA, the proposed er-D(both) scheme is able to get higher integrated utility for the client and the contractor. For example, for the instances ins11-20, the “avg client” and the “avg cont” are 98% and 81%, which are all higher than the “90%” and “72%” yielded by the double-loop GA.

Table 1. Configurations for the experiment

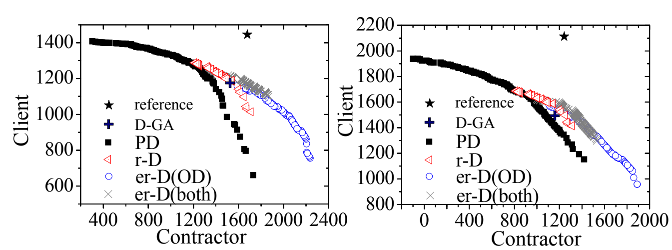
instance	*n*	*m*	*popsize*	*px*	*pm*	*GEN*
ins1-ins10	40	22	100	0.9	1/40	2500
ins11-ins20	48	32	100	0.9	1/48	2500

Table 2. Resullts on the 20 instances

instance		D-GA	PD	r-D	er-D(OD)	er-D(both)
averaged on the ten instances ins1-10	avg client	84%	94%	88%	71%	84%
	avg cont	86%	64%	83%	103%	94%
	avg Dist	1.00	2.41	1.05	1.82	1.07
	avg OD	1.00	1.60	1.37	0.99	1.04
averaged on the ten instances ins11-20	avg client	90%	110%	107%	83%	98%
	avg cont	72%	48%	62%	95%	81%
	avg Dist	1.00	1.92	1.00	1.25	1.05
	avg OD	1.00	1.64	1.29	0.96	1.01

We also compare these approaches from the view point of the satisfaction of preferences. In Table 2, “avg Dist” means the averaged distance between the solutions found by the algorithm and the reference point, and “avg OD” means the averaged optimization degree of the activity schedule. These two criteria correspond to the two preferences considered for the PSNP. To facilitate comparison, we use the averaged distance and the averaged optimization degree found by the double-loop GA as the norms. The ratios between the averaged distance (averaged optimization degree) found by the algorithm in comparison and the one found by the double-loop GA are reported in the Table. From these results, we can see that the averaged distances found by the schemes “PD” and “er-D(OD)” are significantly longer than other approaches, because these two schemes do not consider the reference point in the dominance relation. Similarly, the averaged optimization degrees found by the schemes “PD” and “r-D” are significantly larger than other approaches, because these two schemes do not consider the optimization degree of the activity schedule in the dominance relation. By using the proposed er-dominance relation with $AF=\{g_1, g_2\}$ to consider both preferences, the scheme “er-D(both)” manages to find acceptable averaged distances and averaged optimization degrees. In other words, these results show that the proposed er-dominance relation is effective in dealing with both the preferences.

Overall, compared to the double-loop GA, the proposed approach manages to find solutions with higher integrated utility. Compared to the NSGA-II with the Pareto dominance relation and the r-dominance relation, the proposed approach with the er-dominance relation can better deal with both of the preferences in the problem. In addition, the double-loop GA has to run for several days to get a single solution. But the proposed approach just needs to run for an hour to get a set of promising solutions for negotiation. These results reveal that the proposed approach is promising for the PSNP.



(a) ins1 (b) ins11

Figure 4. Comparison between the objective vectors obtained by different approaches

6. CONCLUSION

In this paper, the payment scheduling negotiation problem in project scheduling has been addressed. We formulate the problem as a bi-objective optimization problem. An extended r-dominance relation is proposed to address the two kinds of preferences in the problem. By incorporating the er-dominance relation with the NSGA-II, a preference-based EMO approach is developed. Experimental results demonstrate that the proposed approach is effective.

In future study, we plan to develop other metaheuristic approaches like ant colony optimization (ACO) [24] and particle swarm optimization (PSO) [25][26] for the problem. Adaptively tuning the parameters of the algorithm is also a desirable research topic [27]-[29].

7. ACKNOWLEDGMENTS

This work was supported in part by the National Science Fund for Distinguished Young Scholars No.61125205, National Natural Science Foundation of China No.61070004, NSFC Joint Fund with Guangdong under Key Project U0835002.

8. REFERENCES

[1] P. Brucker, A. Drexl, R. Mohring, K. Neumann, E. Pesch, "Resource-constrained project scheduling: notation, classification, models and methods", *European Journal of Operational Research*, 112, pp. 3–41, 1999.

[2] J. Blazewicz, J.K. Lenstra, A.H.G. Rinnooy Kan, "Scheduling subject to resource constraints," *Discrete Applied Mathematics*, vol. 5, pp. 11-24, 1983.

[3] J. Węglarz, J. Józefowska, M. Mika, and G. Waligóra, "Project scheduling with finite or infinite number of activity processing modes – a survey," *European Journal of Operational Research*, 208, pp. 177–205, 2011.

[4] R. Kolisch, "Project scheduling under resource constraints, efficient heuristics for several problem classes," *Physica*, Heidelberg, 1995.

[5] B.D. Reyck, W. Herroelen, "A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations," *European Journal of Operational Research,* vol. 111, no. 1, pp. 152-174, Nov., 1998.

[6] V. Valls, S. Quintanilla, and F. Ballestin, "Resource-constrained project scheduling: a critical activity reordering heuristic," *European Journal of Operational Research,* vol. 149, no. 2, pp. 282-301, Sep., 2003.

[7] L. Őzdamar, "A genetic algorithm approach to a general category project scheduling problem," *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol. 29, no. 1, pp. 44-59, 1999.

[8] D. Merkle, M. Middendorf, H. Schmeck, "Ant colony optimization for resource-constrained project scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 333-346, 2002.

[9] G. Ulusoy, "Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows", *Annals of Operations Research*, vol. 102, pp. 237-261, 2001.

[10] M. Mika, G. Waligóra, J. Węglarz, "Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models," *European Journal of Operational Research,* vol. 164, no. 3, pp. 639-668, August, 2005.

[11] W.-N. Chen, J. Zhang, H. Chang, R.-Z. Huang, and O. Liu "Optimizing Discounted Cash Flows in Project Scheduling - An Ant Colony Optimization Approach," *IEEE Transactions on System, Man, and Cybernetics, Part C*, vol. 40, no. 1, pp. 64-77, 2010.

[12] A.H. Russell, "Cash flows in networks," *Management Science*, vol. 16, pp. 357–373, 1970.

[13] G. Ulusoy, S. Cebelli, "An equitable approach to the payment scheduling problem in project management," *European Journal of Operational Research*, vol. 127, pp.262-278, 2000.

[14] Z.W. He and Y. Xu, "Multi-mode project payment scheduling problems with bonus-penalty structure," *European Journal of Operational Research*, vol. 189, pp. 1191-1207, 2008.

[15] Z.W. He, N. Wang, T. Jia, and Y. Xu, "Simulated annealing and tabu search for multimode project payment scheduling," *European Journal of Operational Research*, vol. 198, no. 3, pp. 688-696, 2009.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[17] L.B. Said, S. Bechikh, and K. Ghédira, "The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making," *IEEE transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 801-818, 2010.

[18] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the streagth Pareto evolutionary algorithm," Comput. Eng. Networks Lab. (TIK), Swiss Fed. Inst. Technol. (ETH), Zurich, Swozerland, Tech. Rep. 103, May 2001.

[19] J.D. Knowles and D. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149-172, 2000.

[20] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[21] K. Deb, A. Sinha, J. Korhonen, and J. Wallenius, "An interactive evolutionary multiobjective optimization method based on progressively approximated value functions," *IEEE transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 723-739, 2010.

[22] J. Branke, T. Kaussler, and H. Schmeck, "Guidance in evolutionary multiobjetive optimziation," *Adv. Eng. Sofw*., vol. 32, no. 6, pp. 499-507, Jun. 2001.

[23] K. Deb, J. Sundar, U. Bhaskara, and S. Chaudhuri, "Reference point based multiobjective optimization using evolutionary algorithms," *Int. J. Comput. Intell. Res. (IJCIR)*, vol. 2, no. 3, pp. 273-286, 2006.

[24] W.-N. Chen and Jun Zhang, "Ant Colony Optimization Approach to Grid Workflow Scheduling Problem with Various QoS Requirements", *IEEE Transactions on Systems, Man, and Cybernetics--Part C: Applications and Reviews*, vol. 31, no. 1, pp.29-43, Jan 2009.

[25] W.-N. Chen, Jun Zhang, Ying Lin, Ni Chen, Zhi-hui Zhan, Henry Chung, Yun Li, and Yu-hui Shi, "Particle Swarm Optimization with an Aging Leader and Challengers," *IEEE Transactions on Evolutionary Computation*, in press, 2012

[26] W.-N. Chen, Jun Zhang, Henry Chung, W.L., Zhong, W.G. Wu and Y.H., Shi, "A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems", *IEEE Transactions on Evolutionary Computation*, vol.14, no.2, pp.278-300, April 2010

[27] Jun Zhang, Henry Chung and W.L., LO, "Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms", *IEEE Transactions on Evolutionary Computation*, vol.11, no.3, Page. 326-335, June 2007

[28] Jun Zhang, Zhi-hui Zhan, Ying Lin, Ni Chen, Yue-jiao Gong, Henry S.H. Chung, Yun Li and Yu-hui Shi, "Evolutionary Computation Meets Machine Learning: A Survey", *IEEE Computational Intelligence Magazine*, pp.68-75, Nov. 2011

[29] Z.H. Zhan, Jun Zhang, Y. Li and Henry Chung, "Adaptive Particle Swarm Optimization", *IEEE Transactions on Systems, Man, and Cybernetics--Part B*. vol. 39, no. 6, pp. 1362-1381, Dec. 2009