Review of Integrative Business & Economics Research

An Optimal Project Scheduling Model with Lump-Sum Payment

Shangyao Yan
Department of Civil Engineering, National Central University,
Chungli 32001, Taiwan
t320002@cc.ncu.edu.tw

Sin-Siang Wang
Department of Civil Engineering, National Central University,
Chungli 32001, Taiwan
jay05080@gmail.com

Miawjane Chen*
Department of Finance, National United University, Miaoli 36003, Taiwan
mjchen@nuu.edu.tw

Jzu-Chun Liu
Department of Civil Engineering, National Central University,
Chungli 32001, Taiwan
joyce90289@hotmail.com

ABSTRACT

Project scheduling is essential in enterprise operations. Not only does it increase working efficiency but also the enterprise's profit. In the past, there have been studies developing heuristic algorithms to efficiently solve the complicated multi-mode resource constrained project scheduling problem with discounted cash flows (MRCPSPDCF). These obtained solutions are approximate rather than optimal, making it difficult to evaluate their solution quality. In this study, an optimization model, embedded in a time-precedence network, is proposed to optimally solve the MRCPSPDCF with lump-sum payment. Mathematically, the model is formulated as an integer network flow problem with side constraints, which can be efficiently solved for optimality, using mathematical programming software. To evaluate the model performance, numerical tests are performed. The test results show the good performance of the model.

Keywords: network flow technique; MRCPSPDCF; lump-sum payment; time-precedence network

1. Introduction

Good project scheduling is very helpful to enterprise's operations. This is because it can significantly influence the working efficiency and the project profit. Nowadays, the complexity of the project scheduling problem has gradually increased, mainly due to the fast pace of global economic growth, the scarcity of resources, and the consideration of additional constraints on operations. Project scheduling strategies used in the past that do not consider the constraint on resource consumption, such as the critical path method or the program evaluation and review technique, are not suitable for solving such complicated problems.

Generally, for performing any project, resources must be efficiently utilized to avoid wastage. For this purpose, the resource constrained project scheduling problem (RCPSP) is proposed. The RCPSP is defined as NP-hard (Błazewicz et al., 1983). There are three different types of resources in the RCPSP, renewable, non-renewable, and doubly constrained resources. Renewable and non-renewable resources are constrained on a period basis and a project basis, respectively. Doubly constrained resources integrate the constraints on both renewable and non-renewable resources. There have been several past studies of RCPSPs; for example, see Bell and Park (1990), Demeulemeester et al. (1994), Simpson and Patterson (1996), Brucker et al. (1998), Chen and Weng (2009), and Bianco and Caramia (2011). These studies have mainly focused on the makespan minimization objective, subject to the related operation and resource constraints. In addition, when the scale of RCPSP increases, the proper heuristic algorithms must be selected to ensure efficient solution.

In actual operations, there are several different modes that can be selected for executing each activity. Each mode corresponds to a specific resource requirement and duration. Therefore, a time-resource trade-off problem is the result. This type of problem is called the multi-mode resource constrained project scheduling problem (MRCPSP), which is also characterized as NP-hard. There have been a number of MRCPSPs studied in the past; for example, see Mori and Tseng (1997), Sprecher and Drexl, (1998), Reyck and Herroelen (1999), Alcaraz et al. (2003), Bouleimen and Lecocq (2003), Jarboui et al. (2008), and Peteghem and Vanhoucke (2010). The objective of these studies is to minimize the makespan, subject to the related operation and resource constraints, with considering the multiple modes. In addition, due to the characteristic of being NP-hard, suitable heuristic algorithms must be utilized to solve realistically large problems.

From the aspect of finance, when executing any project, the enterprise mainly focuses on maximizing profit or benefiting shareholders. Therefore, except when considering the related constraints on the RCPSP and the MRCPSP, the concept of the discount rate should also be incorporated into the project scheduling problem. The result is the multi-mode resource constrained project scheduling problem with discounted cash flows (MRCPSPDCF), which is more difficult to solve than the RCPSP or the MRCPSP. Examples of past MRCPSPDCF studies include Özdamar and Dündar (1997), Ulusoy et al. (2001), Józefowska et al. (2002), Mika et al. (2005), and Seifi and Tavakkoli-Moghaddam (2008). The objective of these studies is mainly to maximize the net present value (NPV) of all cash flows for executing the project, subject to the related operation and resource constraints, with considering the multiple modes and the discount rate. Because the MRCPSPDCF is also NP-hard, suitable heuristic algorithms have to be employed to efficiently solve realistically large problems. Consequently, the solutions obtained from the above studies are approximate and are difficult to evaluate in terms of optimality.

In this study, we mainly deal with the MRCPSPDCF with lump-sum payment and expect to find its optimal solution. However, the MRCPSPDCF involves complicated analyses related to the operating time-windows, the precedence of the activities, and certain side constraints, making it difficult to solve. Although heuristic algorithms have often been used to solve such complicated problems, the obtained solutions are not optimal. In addition, other studies have employed dynamic network flow techniques, such as the time-space network approach, to solve similar complicated scheduling problems with good solution quality. For example, see Yan et al. (2006), Yan et al. (2007), and Yan et al. (2012). The complicated analyses associated with the multiple activities in the dimensions of time and space can be naturally and efficiently expressed by the time-space network approach. For this, a time-precedence network, referring to the time-space network, is thus proposed to clearly represent multiple activities with multiple time-windows and the precedence of these activities. However, from the perspective of network optimization, the proposed time-precedence network is predicted to be the proper approach for optimally solving the MRCPSPDCF. Therefore, the time-precedence network technique is utilized to build the MRCPSPDCF model.

The remainder of the paper is organized as follows. In Section 2, the problem is described. In Section 3, the modeling approach is proposed. In Section 4, numerical tests are conducted. In Section 5, some conclusions are given.

2. Problem description

The definition of the studied MRCPSPDCF is presented below. There is a project that includes a set of activities, renewable resources, and non-renewable resources as well as an expected completion period. An activity consists of a set of similar or related works. Activity preemption is not allowed for performing the project. Each activity must be carried out with one of several given modes, each consuming a specific unit of renewable and non-renewable resources as well as taking a specific duration. In addition, once a specific mode is selected to execute the activity, this mode will not be changed during execution of the activity. For renewable/non-renewable resources, there are constraints on use during each time period/project duration. Each resource has a unit expenditure cost.

The NPV criterion is considered by incorporating the concept of the discount rate into the project schedule. The main idea is to calculate the present value of cash inflow and outflow for each activity. In actual operations, there are four project payment types (i.e., the cash inflow), as proposed by Mika et al. (2005). The lump-sum payment (LSP) is where the client, who is the owner of the project, pays the total payment to the contractor, whose task is to perform the project, upon completion of the project. In the progress payment (PP) method, the client pays project payments to the contractor at periodic time intervals, such as the end of each week, until the project is finished. The payment at equal time interval (ETI) method specifies a number of payments for the project. Except for the last payment made when the project is finished, the other payments are made at equal time intervals during the project execution. The payment at activity completion times (PAC) method is where the contractor receives payment from the client at the finish of each activity. In general, from the viewpoint of the clients, too many incidences of cash inflow could make the payment procedure very complicated. To simplify the complexity of the payment process, the LSP payment type is the first choice for the clients. Therefore, in this study, we focus on the LSP payment type.

In addition, disbursement (i.e., cash outflow) for each activity could be at the start, halfway through, or at the end of an activity, depending on the contractor’s operating considerations. However, there would not be any significant difference in the NPV of cash outflow for activities with shorter durations when considering different cash outflows. For the sake of conservative planning, the cash outflow for each activity is set to begin at the start of the activity. The studied MRCPSPDCF mainly focuses on maximizing the NPVs of all cash flows, subject to the above-mentioned operation constraints.

3. The Model

The MRCPSPDCF model is built employing the proposed time-precedence network technique. In general, there are two kinds of networks used to express the project schedule. One is the activity-on-node (AoN) network where nodes represent activities and arcs express precedence restrictions; the other is the activity-on-arc (AoA) network where nodes represent time events and arcs express activities. The AoA network gives a distinct indication of the duration of each activity and the precedence of these activities. Thus, we refer to the AoA network to build the decision-flow time-precedence network.

There are several major elements in the modeling, the decision-flow time-precedence network, the permissible work period for activities, the flow adjustment coefficients for the arcs, and the mathematical formulation, as described below:

3.1. The decision-flow time-precedence network

We employ the time-precedence network technique to formulate the MRCPSPDCF model, as shown in Figure 1. In this network, the vertical axis represents the allowable period for project execution. The horizontal axis indicates the precedence points used to formulate the precedence constraints on the activities. The activity before a specific precedence point is the predecessor of the activity after the same precedence point, and vice versa. In addition, there are two major components, nodes and arcs, in the network. A node indicates a precedence point at a specific time, in addition to the supply and collection points used to individually express the start and end of a project. The arcs are used to build all possible movements of the activities in the network. There are five types of arcs described below.

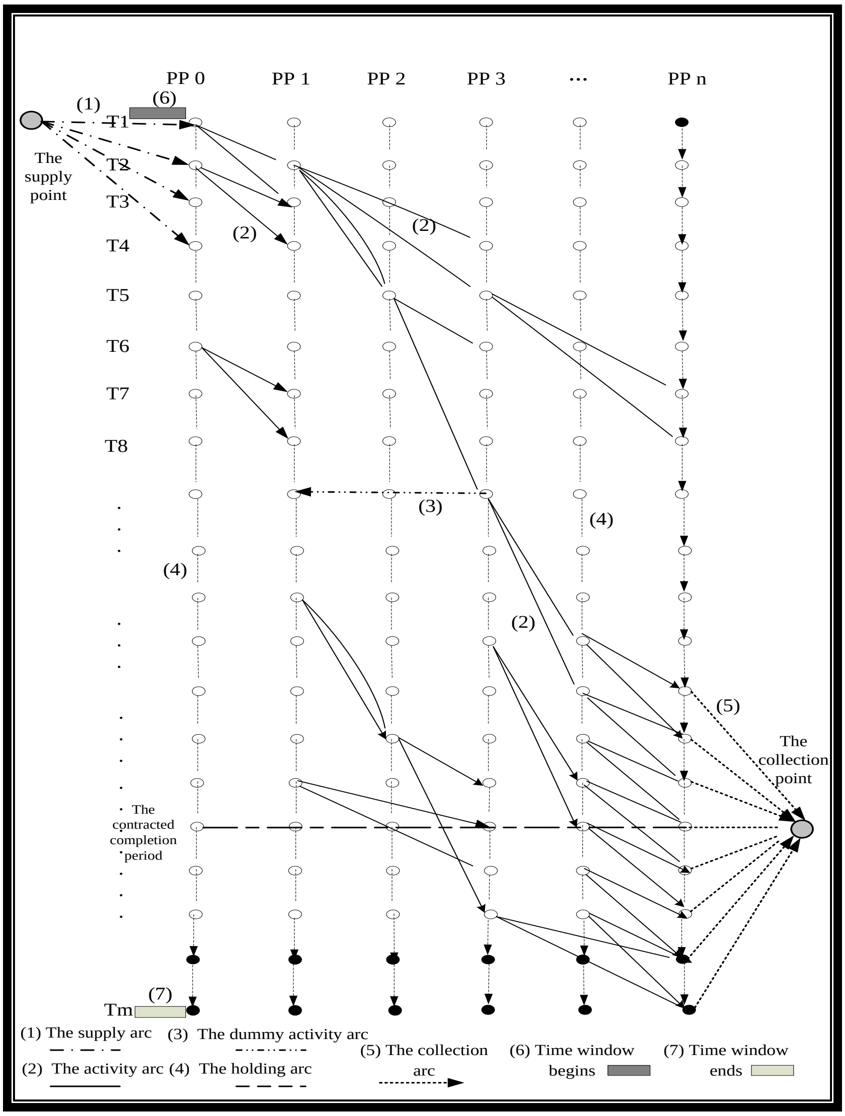

Figure 1 Decision-flow time-precedence network

3.1.1. The supply arc

A supply arc, see (1) in Figure 1, joins the supply point to the first precedence point at every time point. Its function is to decide on the suitable starting time for the project. The arc cost is zero. The arc flow is a binary variable. The arc flow's upper bound is one, meaning that the project begins at the corresponding time point. The arc flow's lower bound is zero, indicating that the project does not start at the corresponding time point.

3.1.2. The activity arc

An activity arc, see (2) in Figure 1, links two different precedence points. Its function is to indicate an activity with a specific mode and a starting time. In this network, all possible activity arcs are set (e.g., to cope with the precedence of the activities, there are four permissible work periods for each activity, which will be discussed in detail in Section 3.2.). Each activity arc has a specific time block, which is the specific mode duration selected to perform this activity. The arc cost is the discount value of the cash outflow for performing the activity. Note that the cash outflow time in terms of the activity duration can be different for each activity and should be determined according to practices. However, there would not be any significant difference in the NPV of cash outflow for activities with shorter durations. Thus, for the sake of conservative planning, the cash outflow for each activity can be set to begin at the start of the activity. The arc cost here has a negative impact on the objective value (i.e., it is the loss to the objective value). The arc flow is a binary variable. The arc flow's upper bound is one, which indicates that the specific mode is selected for performing the activity. The arc flow's lower bound is zero, which means that the specific mode is not chosen to perform that activity.

3.1.3. The dummy activity arc

A dummy activity arc, see (3) in Figure 1, connects the end precedence point of an activity arc at a specific time point to another precedence point at the same time point. Its function is to indicate an activity with a specific mode, with respect to the linked activity arc, and to guarantee the precedence of activities. All possible dummy activity arcs are set in this network. The arc cost is set to be zero and the arc flow is a binary variable. The arc flow's upper bound is one, indicating that this specific mode is picked to perform the dummy activity that connects two precedence points. The arc flow's lower bound is zero, meaning that this specific mode is not chosen for executing the dummy activity.

An example, obtained from the project scheduling problem library (PSPLIB), as shown in Table 1, is utilized to illustrate how to build the dummy activity arc. We first construct the AoA network based on the data associated with the example, and then modify this network by adding the dummy activity arc. The steps are discussed in detail below.

Table 1 Information associated with activities

(a) Activity No.	(b) Successors	(c) Predecessors	(d) Precedence point (PP) No.	(e) Precedence point (PP) No. associated with the successors
1	2, 3	0	PP 0	PP 1
2	5, 6, 8, 9	1	PP 1	PP 3, PP 4, PP 6
3	4, 5	1	PP 1	PP 2, PP 3
4	6, 8	3	PP 2	PP 4
5	7	2, 3	PP 3	PP 5
6	7	2, 4	PP 4	PP 5
7	10	5, 6	PP 5	PP 7
8	10	2, 4	PP 4	PP 7
9	10	2	PP 6	PP 7
10	--	7, 8, 9	PP 7	--

Note: the first two columns shadowed in gray are provided from a PSPLIB example and the symbol "--" indicates that the data does not exist.

Step 1: Use the successors for all activities in the example to search the predecessors for each activity. For instance, look at column (b) in Table 1. Since activity 1 is the starting activity, activity 1 has no predecessor. In addition, since activity 2 is the successor of activity 1, activity 1 is the predecessor of activity 2. The predecessors associated with the remaining activities can also be designated based on the same method.

Step 2: Number the corresponding precedence points (PPs) for each activity in sequence, from activity 1 to activity 10. The first PP is numbered PP 0. Then, when numbering the corresponding PP for the next activity, its predecessors must be compared with those of the already numbered activities. If its predecessors are the same as those of the numbered one, then its corresponding PP is numbered the same as that one; otherwise, it is assigned a larger number. For instance, look at column (c) in Table 1. Since activity 1 has no predecessor, its corresponding PP is numbered PP 0. Since activities 2 and 3 have the same predecessor to each other but different from that of activity 1, their corresponding PP is numbered PP 1. The corresponding PPs associated with the remaining activities can also be numbered based on the same method.

Step 3: Search for the corresponding PPs of successors for each activity according to the precedence of these activities. For instance, look at columns (b) and (d) in Table 1. Activity 1 has two successors, activities 2 and 3, whose corresponding PP is PP 1. PP 1 is the corresponding PP of successors for activity 1. The corresponding PPs of successors related to the remaining activities can also be searched for based on the same method.

Utilizing Steps 1-3, the AoA network can be built, as shown in Figure 2. However, in this network, considering the precedence of the activities, the corresponding PPs of successors for some activities, such as activities 2 and 3, may be different (i.e., an activity may be represented by multiple PP pairs). In addition, an activity in the time-precedence network is expressed by a specific PP pair. To express an activity with multiple PP pairs, the concept of the dummy activity arc is proposed. The construction of the dummy activity arc is presented in Step 4.

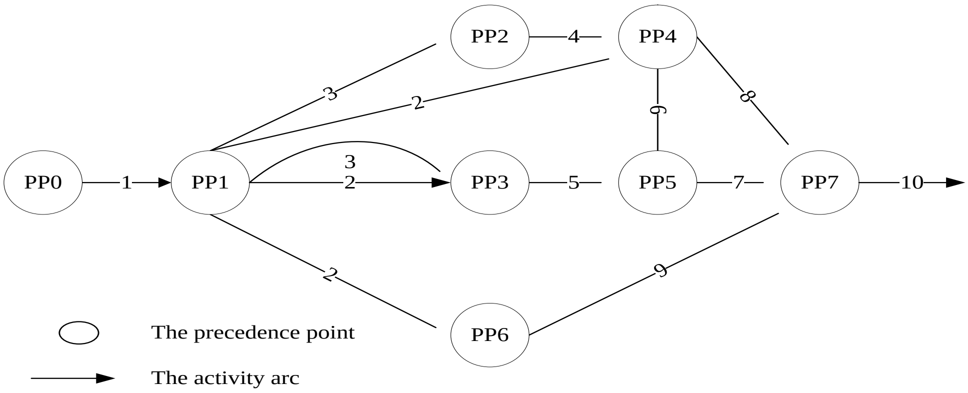

Figure 2 AoA network associated with the project schedule

Step 4: First, define the first PP pair for an activity with multiple PP pairs as the activity arc. Afterward, construct a dummy activity arc for each of the rest PP pairs by switching the starting node to the end node of the first one. For instance, as shown in Figure 2, there are three PP pairs, PP 1 - PP 3, PP 1 - PP 4, and PP 1 - PP 6, that can express activity 2. First, define the first PP pair, PP 1 - PP 3, as the activity arc. Then, switch the starting node for the second and third PP pairs (i.e., PP 1) to the end node of the first one (i.e., PP 3). As a consequence, dummy activity arcs 2′ and 2″ are constructed, as shown in Figure 3. Thus, the activity with multiple PP pairs can be reduced to one activity arc with one PP pair and a number of dummy activity arcs, each substituting one of the rest PP pairs. The dummy activity arc associated with the remaining activities with multiple PP pairs can also be constructed based on the same method.

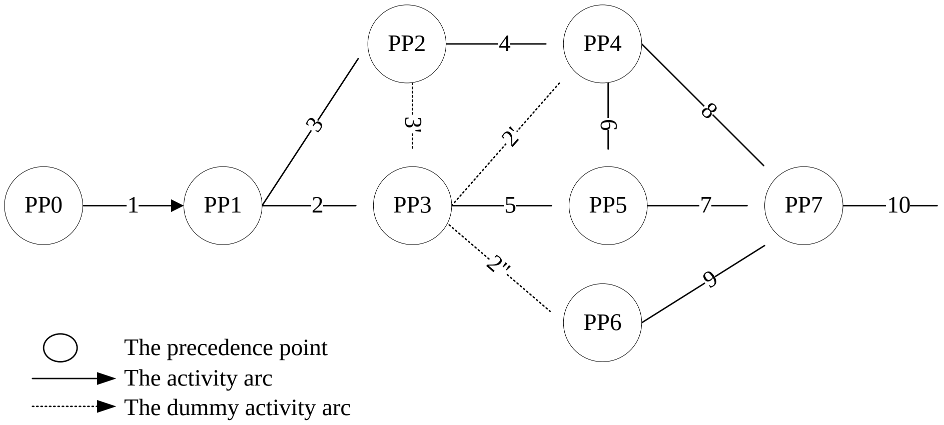

Figure 3 AoA network with dummy activities associated with the project schedule

3.1.4. The holding arc

A holding arc, see (4) in Figure 1, connects two precedence points with two adjacent time points at the same precedence point. Its function is to determine suitable starting times for later activities for a precedence point, which mainly ensures the precedence of activities. The arc cost is set to be zero and the arc flow is a non-negative integer variable. The arc flow's upper bound is infinity, implying that there is no constraint on the number of starting time determinations for activities later than the associated time window at this precedence point. The arc flow's lower bound is zero, showing that there is no starting time determination for activities later than the associated time window at this precedence point.

3.1.5. The collection arc

A collection arc, see (5) in Figure 1, joins the last precedence point at every time point to the collection point. Its function is to decide on the ending time for the project and to assure the flow conservation of arcs in the network. The arc benefit (or the negative arc cost) is the discount value of all cash inflows for the project to finish, plus a reward/penalty value determined with respect to the ending of the project within/without the contracted completion period. The arc benefit here has a positive impact on the objective value (i.e., it is the addition to the objective value). In the maximization problem, the reward/penalty value is positive/negative. The arc flow is a binary variable. The arc flow’s upper bound is one, indicating that the project ends at the corresponding time point. The arc flow’s lower bound is zero, implying that the project does not end at the corresponding time point.

3.2. The permissible work periods for activities

In actual operations, there are several permissible work periods for each activity, such as earliest starting, earliest ending, latest starting, and latest ending times. These are used to satisfy the precedence of activities. These four types of times are discussed below.

3.2.1. The earliest starting and ending times for activities

The earliest starting time for each activity can be obtained based on the concept of the forward pass calculation (FPC). The earliest ending time for each activity can be obtained by taking the shortest mode duration for each activity plus its earliest starting time. Look at the PSPLIB example shown in Table 2. It is proposed to illustrate the calculation of the earliest starting and ending times for each activity. For simplicity, the unit for mode duration for each activity in the example is assumed to be one day. For instance, look at columns (c) and (e) in Table 2. For activity 1, the shortest mode duration is zero and there is no predecessor. Therefore, the earliest starting and ending times for activity 1 are the starting of day 0 (DS 0). The shortest mode duration for activities 2, 3, and 4 is one day and their predecessor is activity 1 whose earliest ending time is DS 0. Therefore, the earliest starting and ending times for these three activities are DS 1 and the ending of day 1 (DE 1), respectively. If a specific activity has multiple earliest starting times, then the maximum one is chosen in order to guarantee completion of its predecessors. The earliest starting and ending times for the dummy activity are the same as the earliest ending time for its corresponding activity. The earliest starting and ending times for the remaining activities can also be calculated based on the same method.

Table 2 Four kinds of work duration associated with activities

(a) Activity No.	(b) Successors	(c) Shortest mode duration (day)	(d) Longest mode duration (day)	(e) Predecessors	(f) Dummy activity	(g) Earliest starting time	(h) Earliest ending time	(i) Latest starting time	(j) Latest ending time
1	2, 3, 4	0	0	0	--	DS 0	DS 0	DE 5	DE 5
2	5, 11	1	6	1	2′	DS1 (DE1)	DE1 (DE1)	DS10 (DE15)	DE15 (DE15)
3	5, 11	1	10	1	3′	DS1 (DE1)	DE1 (DE1)	DS6 (DE15)	DE15 (DE15)
4	9, 11	1	10	1	4′	DS1 (DE1)	DE1 (DE1)	DS30 (DE39)	DE39 (DE39)
5	6	1	10	2, 3	--	DS2	DE2	DS16	DE25
6	7, 8, 10	3	4	5	--	DS3	DE5	DS26	DE29
7	9	5	10	6	--	DS6	DE10	DS30	DE39
8	9	2	7	6	--	DS6	DE7	DS33	DE39
9	12	7	9	4, 7, 8	--	DS11	DE17	DS40	DE48
10	12	3	5	6	--	DS6	DE8	DS44	DE48
11	12	4	6	2, 3, 4	--	DS2	DE5	DS43	DE48
12	--	0	0	9, 10, 11	--	DS18	DS18	DE48	DE48

Note: the first four columns shadowed in gray are provided from a PSPLIB example, the symbol "--" indicates that the data does not exist, and the data included in brackets is associated with the starting and ending times for the dummy activity

3.2.2. The latest starting and ending times for activities

The latest ending time for each activity can be obtained based on the concept of the backward pass calculation (BPC). The latest starting time for each activity can be obtained by taking the latest ending time for each activity minus its longest mode duration. An example (the same as in Section 3.2.1) as shown in Table 2 is proposed to show the calculation of the latest starting and ending times for each activity. For instance, as can be seen from columns (b) and (d) in Table 2, there is no successor for activity 12, and its longest mode duration is zero. Therefore, the latest starting and ending times for activity 12 are DE 48 (i.e., the longest project execution duration that can be obtained utilizing the label correcting algorithm). The successor for activities 11, 10, and 9 is activity 12 (whose latest starting time is DE 48), so their latest ending time is DE 48. In addition, because the longest mode durations for these three activities are 6 days, 5 days, and 9 days, respectively, their latest starting times are DS 43, DS 44, and DS 40, respectively. If a specific activity has multiple latest starting times, then the minimum one is selected in order to accelerate the execution of its successors. The latest starting and ending times for the dummy activity are the same as the latest ending time for its corresponding activity. The latest starting and ending times for the remaining activities can also be calculated based on the same method.

3.3. The flow adjustment coefficients for arcs

The number of inflow and outflow arcs for a specific node in the network may be inconsistent when the precedence of the activities is considered. This violates flow conservation at the node. To remedy this problem, the inflow and outflow arcs for each node must use the appropriate flow adjustment coefficient (called *u* hereafter). An example is shown in Figure 4 of the calculation of *u* for arcs in the AoA network given a supply and a demand. For instance, for a1, the number of inflow and outflow arcs to its end node, PP 1, is 1 (i.e., a1) and 2 (i.e., a2 and a3), respectively. Therefore, the appropriate *u* is 2(= 2/1). The calculation of *u* for the remaining arcs can also be carried out with the same method. Note that the *u* is 1 for the supply, collection, and holding arcs. The reason is that the supply and collection arcs mainly guarantee flow conservation for all arcs in the network, and the holding arc mainly indicates the holding of determinations at a precedence point within a time window.

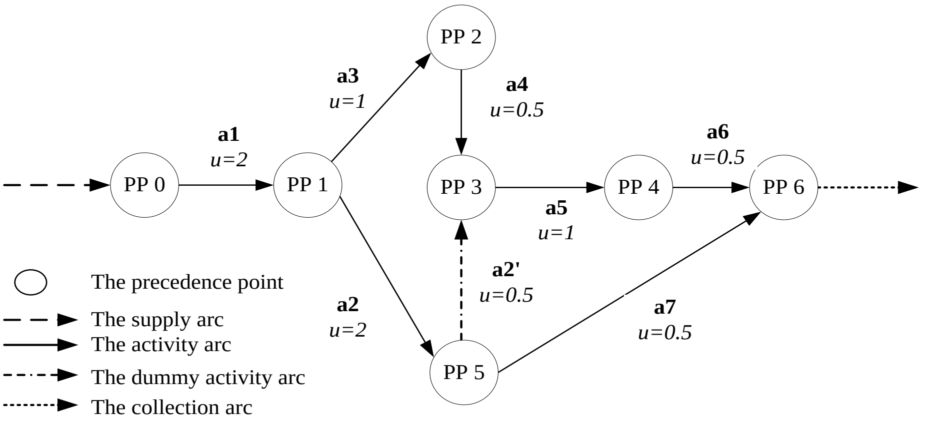

Figure 4 Example of demonstrating flow adjustment coefficients for arcs

3.4. Notations and symbols used in the MRCPSPDCF model

Decision variable:

y_{ijk} : the flow of the k^{th} arc for the node pair (i,j) in the network;

Parameters:

ICF_{ijk} : the discount value of the cash inflow for the project to finish, plus a reward/penalty value, of the k^{th} arc for the node pair (i,j), associated with a collection arc benefit, in the network;

PCF_{ijk} : the discount value of the cash outflow of the k^{th} arc for the node pair (i,j), associated with an activity arc cost, in the network;

re_{ijkl} : the amount of the l^{th} renewable resource consumed by the k^{th} arc for the node pair (i,j) in the network;

re_{ijko} : the amount of the o^{th} non-renewable resource consumed by the k^{th} arc for the node pair (i,j) in the network;

R_l : the amount of the l^{th} renewable resource that is available;

R_0 : the amount of the o^{th} non-renewable resource that is available;

p_{qi} : the number of predecessors for the node pair (q,i) in the network;

u_{qik} : the flow adjustment coefficient of the k^{th} arc for the node pair (q,i) in the network;

a : the a^{th} activity;

sp : the supply point in the network;

cp : the collection point in the network;

ub_{ijk} : the upper bound for the k^{th} arc of the node pair (i,j) in the network;

Sets:

N : the set of all nodes in the network;

A : the set of all activities in the network;

NP : the set of all node pairs for activities in the network;
NP_a : the set of all node pairs for the a^{th} activity in the network;
PA_{ij} : the set of all parallel activity arcs for the node pair (i,j) in the network;
B_{qi} : the set of node pairs of all predecessors for the node pair (q,i) in the network;
RE : the set of all types of renewable resources;
NRE : the set of all types of non-renewable resources;
T : the set of all time points in the network;
T_h : the set of all node pairs of activities at the h^{th} time point;
CA : the set of all collection arcs in the network.

3.5. Model formulation

The MRCPSPDCF model is formulated as follows:

Maximize NPV:

$$\sum_{ij\in CA}\sum_{k\in PA_{ij}} ICF_{ijk}\, y_{ijk} - \sum_{ij\in NP}\sum_{k\in PA_{ij}} PCF_{ijk}\, y_{ijk} \tag{1}$$

Subject to:

$$\sum_{j\in N}\sum_{k\in PA_{ij}} y_{ijk} - \sum_{q\in N}\sum_{k\in PA_{qi}} u_{qik}\, y_{qik} = \begin{cases} 1, & if \quad i = sp \\ 0, & others \\ -1, & if \quad i = cp \end{cases} \qquad \forall i \in N \tag{2}$$

$$\sum_{k\in PA_{qi}} p_{qi}\, y_{qik} \le \sum_{ij\in B_{qi}}\sum_{k\in PA_{ij}} y_{ijk} \qquad \forall (q,i) \in NP \tag{3}$$

$$\sum_{ij\in NP_a}\sum_{k\in PA_{ij}} y_{ijk} = 1 \qquad a \in A \tag{4}$$

$$\sum_{ij\in T_h}\sum_{k\in PA_{ij}} re_{ijkl}\, y_{ijk} \le R_l \qquad \forall h \in T, l \in RE \tag{5}$$

$$\sum_{ij\in NP}\sum_{k\in PA_{ij}} re_{ijko}\, y_{ijk} \le R_o \qquad o \in NRE \tag{6}$$

$$0 \le y_{ijk} \le ub_{ijk}, \text{ integer} \qquad \forall k \in PA_{ij}, \forall (i,j) \in NP \tag{7}$$

The model is formulated as an integer network flow problem with side constraints, in which the objective function (1) is to maximize the NPV of all cash flows for executing the project, i.e., the total arc benefit of all collection arcs minus the total arc cost of all activity arcs. The discount cash inflow plus a reward/penalty value for the project to finish, i.e., a collection arc benefit, can be represented as $ICF = {(I+v)}\big/{(1+\alpha)^t}$, where I, v, t, and α indicate the cash inflow for the project to finish, a reward/penalty value, the time for the project to finish, and the discount rate, respectively. The discount cash outflow for each activity, i.e., an activity arc cost, can be represented as $PCF = {Ea_i}\big/{(1+\alpha)^{t_i}}$, where Ea_i, t_i, and α indicate the cash outflow for the a^{th} activity at the time for the i^{th} time-precedence point, the time for the i^{th} time-precedence point, and the discount rate, respectively. Equation (2) guarantees flow conservation at every node in the network. Equation (3) satisfies the precedence of the activities. Equation (4) shows that each activity is performed using one mode. Equations (5) and (6) control the use of the renewable and the non-renewable resources, respectively. Equation (7) assures the range and integrality of the arc flows.

It should be mentioned that the model can be modified to be used for the PAC payment type, meaning that the time for the cash inflow can be changed from the finish of the project to the finish of each activity (Chen, et al. 2012). To do this, the activity arc cost and the collection arc benefit in the network must be suitably modified. For the details, please refer to Chen, et al. (2012). An example referring to Table 2 is executed to demonstrate whether these two models can optimally solve the maximal NPV problem. The test results are confirmed manually. It is found that the NPVs obtained from the two models are optimal (i.e., the maximal NPV problem can be optimally solved by the proposed models).

4. Numerical tests

To demonstrate the performance of the models, numerical tests, as in the PSPLIB examples, are conducted. All the necessary programs are developed with the C computer language, coupled with CPLEX 11.1 mathematical programming solver. An AMD Dual Core Processor 4450e 2.30GHz CPU with 2.0GB of RAM operating in the environment of Microsoft Window XP is used to solve the problem.

4.1. Input data

The test problem contains 1 project with 20 activities, each performed in one of three modes, and a contracted completion period of 18 time units (a time unit can be a month, a quarter year, half a year, or a year, according to the planner’s considerations). Based on the BPC, the longest project execution duration (i.e., the analysis period) is 53 time units. The type of renewable resources is 2, R1 and R2, according to the available amounts for 14 and 16 per time unit. The type of non-renewable resources is 2, N1 and N2, according to the available amounts of 60 and 68 during the analysis period. To save space, for a detailed description of the successors and the mode data (e.g., duration and resource consumption) for each activity, the reader can refer to the file, md330_.bas, from the PSPLIB. For the setting of the dummy activity arcs and the permissible work periods for the activities, the reader can refer to Sections 3.1.3 and 3.2, respectively.

All cost parameters to the activities are generated by us, mainly due to the fact that these parameters are not provided by the PSPLIB. The total contract payment for the project is 4,000,000 monetary units (a monetary unit can be USD, EURO, or NTD, based on real practices). The unit expenditure cost for R1/R2/N1/N2 is 13,065/5,823/11,628/5,538 monetary units. With the LSP payment type, the cash inflow is the expense of completing all activities in the project, and the cash outflow for each activity is the expenditure for the mode selected to perform the activity at the start of the activity for the sake of conservative planning. With the PAC payment type, the cash inflow/outflow for each activity is the expenditure for finishing the activity/the mode selected to perform the activity at the start of the activity. The reward and the penalty values, used for calculating the collection arc costs, are 40,000 monetary units per time unit. According to the interest rates and the risk considerations for the contractor carrying out the project, the discount rate is set to be 0.03 per time unit.

In this study, one decision-flow network is built to show the potential movements of decisions for 20 activities in the test problem. Each precedence point during the analysis period can be divided into 54 time-precedence points based on the setting of the time interval of one time unit. The model contains 901 nodes, 2,721 arcs, and 6,499 constraints, in which there are 901 constraints ensuring flow conservation at each node, 2,721 constraints assuring the range and integrality of the arc flows, and 2,877 side constraints for considering the precedence of activities and the use of resources.

4.2. Output results

The output results for the model using the LSP payment type demonstrate that the optimal NPV, obtained directly with CPLEX within 149.93 seconds of CPU time, is 427,740 monetary units. The makespan required to finish the project is 20 time units. The assignment of all resources meets the usage restrictions. In addition, the output results for the model using the PAC payment type demonstrate that the optimal NPV, obtained directly with CPLEX within 6.46 seconds of CPU time, is 1,117,300 monetary units. The makespan required to complete the project is 23 time units. The assignment of all resources satisfies the usage constraints. Based on these output results, it can be found that the PAC is better than the LSP in terms of maximizing the NPV. This is because the discount cash inflow for the LSP is lower than that for the PAC, based on the difference of the payment time. In general, the earlier the payment time, the higher the discount cash inflow, and vice versa. That is, when executing a project, it is better for enterprises to select the PAC payment type in order to maximize the NPV.

5. Conclusions

In this study, an MRCPSPDCF model utilizing the LSP payment type is developed based on the proposed time-precedence network technique. In addition, to increase the flexibility of the model, the cash inflow to the project can be changed from the LSP to the PAC type. Mathematically, the models are formulated as integer network flow problems with side constraints and can be optimally solved utilizing CPLEX mathematical programming solver. To demonstrate the performance of the models, numerical tests, referring to the PSPLIB examples, are conducted. The size of the test problem includes 901 nodes, 2,721 arcs, and 6,499 constraints. The test results indicate that the two models can be optimally solved (i.e., finding the optimal NPV), within a reasonable time, using CPLEX. In addition, it can be seen that the PAC is better than the LSP for maximizing the NPV mainly because the discount cash inflow for the LSP is lower than that for the PAC, based on the difference of the payment time. Finally, the main focus in the proposed models is on the LSP and the PAC payment types, and can not be directly applied to other payment types (e.g., PP or ETI). Therefore, how to develop suitable network flow models for other payment types could be a topic of future research. In addition, there could be fluctuation in the interest rates for the cash flow throughout the project duration, depending on the total economic environment, leading to fluctuation in the discount rate. How to incorporate a fluctuating discount rate into the model could be another direction of future research.

Acknowledgements

This research was supported by a grant (NSC-101-2410-H-239-002) from the National Science Council of Taiwan.

References

[1] Alcaraz, J., Maroto, C., and Ruiz, R. (2003). Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms. *Journal of the Operational Research Society*, 54, 614-626.

[2] Bell, C. A. and Park, K. (1990). Solving resource-constrained project scheduling problems by a* search. *Naval Research Logistics*, 37, 61-84.

[3] Bianco, L. and Caramia, M. (2011). Minimizing the completion time of a project under resource constraints and feeding precedence relations: a Lagrangian relaxation based lower bound. 4OR: A Quarterly Journal of Operations Research, 9(4), 371-389.

[4] Błazewicz, J., Lenstra, J.K., and Rinnooy Kan, A.H.G. (1983). Scheduling subject to resource constraints. Discrete Applied Mathematics, 5, 11-24.

[5] Bouleimen, K. and Lecocq, H. (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. European Journal of Operational Research, 149, 266-281.

[6] Brucker, P., Knust, S., Schoo, A., and Thiele,O. (1998). A branch and bound algorithm for the resource-constrained project scheduling problem. European Journal of Operational Research, 107, 272-288.

[7] Chen, M., Yan, S., Wang, S.S. and Liu, C.L. (2012), “A Network Flow Model for the Multi-Mode Resource Constrained Project Scheduling Problem with Discounted Cash Flows,” Working paper, National United University.

[8] Chen, P.H. and Weng, H. (2009). A two-phase GA model for resource-constrained project scheduling. Automation in Construction, 18 (4), 485-498.

[9] Demeulemeester, E., Herroelen, W., Simpson, W.P., Baroum, S., Patterson, J.H., and Yang, K.-K.(1994). On a paper by Christofides et al. for solving the multiple-resource constrained, single project scheduling problem. European Journal of Operational Research, 76, 218-228.

[10] Jarboui, B., Damak, N., Siarry, P., and Rebai, A. (2008). A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. Applied Mathematics and Computation, 195, 299-308.

[11] Józefowska, J., Mika, M., Waligóra, G., and Węglarz, J.(2002). Multi-mode resource-constrained project scheduling problem with discounted positive cash flows—simulated annealing vs. tabu search. Foundations of Computing and Decision Sciences, 27, 97-114.

[12] Mika, M., Waligóra, G., and Węglarz, J. (2005). 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, 164, 639-668.

[13] Mori, M. and Tseng, C.C. (1997). A genetic algorithm for multi-mode resource constrained project scheduling problem. European Journal of Operational Research, 100, 134-141.

[14] Özdamar, L. and Dündar, H. (1997). A flexible heuristic for a multi-mode capital constrained project scheduling problem with probabilistic cash inflows. Computers and Operations Research, 24, 1187-1200.

[15] Peteghem, V.V. and Vanhoucke, M. (2010). A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. European Journal of Operational Research, 201, 409-418.

[16] Reyck, B.D. and Herroelen, W. (1999). The multi-mode resource-constrained project scheduling problem with generalized precedence relations. European Journal of Operational Research, 119. 538-556.

[17] Seifi, M. and Tavakkoli-Moghaddam, R. (2008). A new bi-objective model for a multi-mode resource-constrained project scheduling problem with discounted cash flows

and four payment models. International Journal of Engineering Transactions A: Basics, 21, 347-359.

[18] Simpson III, W. P. and Patterson, J.H. (1996). A multiple-tree search procedure for the resource-constrained project scheduling problem. European Journal of Operational Research, 89, 525-542.

[19] Sprecher, A. and Drexl, A. (1998). Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm. European Journal of Operational Research, 107, 431-450.

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

[21] Yan, S., Chen, C.H., Chen, H.Y., and Lou, T.C. (2007). Optimal scheduling models for ferry companies under alliances. Journal of Marine Science and Technology, 15, 53-66.

[22] Yan, S., Chen, S.C., and Chen, C.H. (2006). Air cargo fleet routing and timetable setting with multiple on-time demands. Transportation Research Part E, 42, 409-430.

[23] Yan, S., Wang, S.S., and Wu, M.W. (2012). A model with a solution algorithm for the cash transportation vehicle routing and scheduling problem. Computers and Industrial Engineering, 63, 464-473.