Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 12, No. 2, pp. 355–366, April 2016

REFORMULATIONS FOR PROJECT PORTFOLIO SELECTION PROBLEM CONSIDERING INTERDEPENDENCE AND CARDINALITY*

Xingmei Li, Yao-Huei Huang, Shu-Cherng Fang and Zhibin Deng†

Abstract: In this paper, we propose a new model for project portfolio selection problem (PPSP) considering both of the project interdependency and cardinality constraint. The proposed model can be reformulated as a mixed integer programming problem by using the conventional linearization technique. To further improve the computational efficiency, we also develop a new linearization technique and new reformulations for the proposed model. Numerical experiments are conducted to compare the efficiency of different reformulations and the results show that the new reformulations derived from the proposed linearization technique can be an efficient way to solve large-size PPSP.

Key words: *project portfolio selection problem, project Interdependency, cardinality constraint, reformulation, linearization.*

Mathematics Subject Classification: *90B50, 90C10, 90C90.*

1 Introduction

Project Portfolio Selection Problem (PPSP) selects the most suitable subset of projects from a set of project candidates to ensure an effective and efficient use of substantial resources. It is a challenging problem with high-practical relevance. Following the pioneering work of [14], PPSP has attracted a considerable amount of attentions in various fields, such as research and development [1], capital budgeting [3] and information systems/information technologies [16].

In order to reflect the reality, models in the literature often focus on some important factors, such as project interdependency [4], cardinality constraint [11], precedence relationship [12, 18], employee competence [7], divisibility [11]. In this paper, we are mainly concerned with *project interdependency* and *cardinality constraint*. It is worth pointing out that other factors could also be incorporated into the model proposed in this paper.

Interdependency. Benefit interdependency and *resource interdependency* are two main types of project interdependency. Benefit interdependency occurs when two or more interdependent projects are selected and executed. In this case, an extra benefit called *synergistic*

*Research supported by National Natural Science Foundation of China Grant #11501543, #61105127, #71171079, #71572154, Fundamental Research Funds for the Central Universities of China Grant #268SWJTU15WCX01, UCAS Scientific Research Foundation for Young Faculty Grant #Y551037Y00, and United States Army Research Office Grant #W911NF-15-1-0223.

†Corresponding author.

benefit, which is over the sum of the benefits derived from each individual project, is fulfilled [4]. Resource interdependency arises from sharing resources among various projects. For example, the implementation of two or more related projects at the same time may require less resources than if they are implemented separately [16].

The project interdependency can be modeled by the product of several binary variables, each of which denotes whether a project is chosen or not. This leads to an integer polynomial programming model for PPSP. For example, benefit and resource interdependency among three projects A, B and C requires a cubic term $x_A x_B x_C$, where x_A, x_B and x_C are binary variables. Due to the complexity of polynomial functions and integral constraints involved in these models, heuristic algorithms are often chosen in most literatures for the purpose of efficiency [2, 21], although the quality of the solutions is not guaranteed. On the other hand, in order to find global optimal solutions, Watters [20] developed a linearization technique to reformulate these models into linear integer programming problems. Glover and Woolsey [5] then extended Watters' work by providing rules to replace the polynomial cross-product terms with continuous variables instead of integer variables. This technique leads to a mixed integer linear programming (MILP) reformulation [8]. To the best of our knowledge, there is no other efficient linearization techniques than the one proposed in [5] to tackle this type of models in general. One key objective of this paper is to propose a new and more efficient linearization reformulation.

Cardinality constraint. Several researchers have pointed out that the failure of some project portfolios is a consequence of that too many projects have been selected such that their implementations exceed the capacity of a company [15]. Therefore, it is necessary to limit the total number of projects in an optimal portfolio. The number of projects consisted in a portfolio is called *cardinality.* Yu et al. [21] considered an equality constraint on cardinality, while others imposed an inequality constraint on cardinality [7, 22].

To our best knowledge, few literatures consider both project interdependency and cardinality constraint in PPSP at the same time. In these literatures, either a genetic algorithm is applied to solve the problem [21], or the way of describing the project interdependency is different [19, 22]. In this paper, we propose a new model for PPSP considering the benefit and resource interdependency and cardinality constraint at the same time.

The rest of the paper is organized as follows. In Section 2, a new model of PPSP considering both project interdependency and cardinality constraint is presented. A reformulation using linearization technique in [5] is derived in Section 3. Then, a new linearization technique is presented for providing a new reformulation in Section 4. Computational experiments are executed in Section 5 to highlight the efficiency of the proposed linearization technique. Conclusions are given in Section 6.

2 A New PPSP Model

Suppose there are N projects to be selected from, and the decision variable x_i denotes whether project i is included in the portfolio ($x_i = 1$) or not ($x_i = 0$). In other words, a project portfolio can be represented by the vector $x = (x_1, \ldots, x_N)$ and the cardinality of a portfolio x is $I(x) = \sum_{i=1}^{N} x_i$. Let $r_i \geq 0$ be the benefit derived from implementing project i alone, $r_{i,j} \geq 0$ be the additional benefit derived from implementing projects i and j together, and $r_{i,j,k} \geq 0$ be the additional benefit derived from implementing projects i, j and k together. In project portfolio selection, a decision-maker is often faced with the problem of selecting a small subset of projects based on some certeria. Here, we assume the decision-maker only concerns the benefit of the project portfolio, i.e., to maximize

$$B(x) = \sum_{i=1}^{N} r_i x_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} r_{i,j} x_i x_j + \sum_{i=1}^{N-2} \sum_{j=i+1}^{N-1} \sum_{k=j+1}^{N} r_{i,j,k} x_i x_j x_k, \tag{2.1}$$

where the second and third terms in (2.1) account for the benefit interdependency of projects. We also assume that there are S different types of resources required by the candidate projects with $b^s > 0$ being the available amount of resource s. Each project i, if selected, requires an amount $d_i^s \geq 0$ of resource s for $s = 1, \ldots, S$. Let $d_{i,j}^s \geq 0$ be the amount of resource s shared by projects i and j, and $d_{i,j,k}^s \geq 0$ be the amount of resource s shared by projects i, j and k, if they are selected. Then, the resource s used by a project portfolio x can be represented as (ref. [16])

$$R_s(x) = \sum_{i=1}^{N} d_i^s x_i - \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{i,j}^s x_i x_j + \sum_{i=1}^{N-2} \sum_{j=i+1}^{N-1} \sum_{k=j+1}^{N} d_{i,j,k}^s x_i x_j x_k \tag{2.2}$$

for $s = 1, \ldots, S$. By incorporating the interdependency and cardinality into the decision modelling, a new model for PPSP can be formulated as

$$\mathbf{PPSP_{New}} \quad \begin{cases} \text{Max} & B(x) & (2.3) \\ \text{subject to} & R_s(x) \leq b^s, \ s = 1, \ldots, S, & (2.4) \\ & I(x) = (\leq)\ m, & (2.5) \\ & x_i \in \{0, 1\}, \ i = 1, \ldots, N, & \end{cases}$$

where m is a positive integer representing the preassigned limit on cardinality. Note that, in model $\mathbf{PPSP_{New}}$, the cardinality constraint can be either an equality or inequality constraint. As pointed out in Section 1, both cases have appeared in the literature.

Remark 2.1. High-order polynomial terms can be used to account for the benefit and resource interdependency among over four projects. But these high-order polynomial terms can be reduced to the third order by using a linearization technique proposed in [16]. Therefore, we only consider the model with the highest order of three.

3 Model Reformulation using Glover and Woolsey's Linearization Technique

Model $\mathbf{PPSP_{New}}$ in Section 2 is a polynomial integer programming problem, thus difficult to be solved directly. Glover and Woolsey proposed a linearization technique to transform the polynomial binary programming problem into a mixed 0-1 integer linear programming problem [5]. By introducing auxiliary variables $y_{i,j} = x_i x_j$ and $z_{i,j,k} = x_i x_j x_k$ for $i, j, k = 1, \ldots, N$ and additional valid linear inequalities, functions $B(x)$ and $R_s(x)$ can be replaced by functions

$$\bar{B}(x, y, z) = \sum_{i=1}^{N} r_i x_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} r_{i,j} y_{i,j} + \sum_{i=1}^{N-2} \sum_{j=i+1}^{N-1} \sum_{k=j+1}^{N} r_{i,j,k} z_{i,j,k} \tag{3.1}$$

and

$$\bar{R}_s(x, y, z) = \sum_{i=1}^{N} d_i^s x_i - \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{i,j}^s y_{i,j} + \sum_{i=1}^{N-2} \sum_{j=i+1}^{N-1} \sum_{k=j+1}^{N} d_{i,j,k}^s z_{i,j,k}, \tag{3.2}$$

respectively, where $y = (y_{1,2}, \ldots, y_{1,N}, y_{2,3}, \ldots, y_{N-1,N}) \in \mathbb{R}^{\frac{N(N-1)}{2}}$ and $z = (z_{1,2,3}, \ldots, z_{1,N-1,N}, z_{2,3,4}, \ldots, z_{N-2,N-1,N}) \in \mathbb{R}^{\frac{N(N-1)(N-2)}{6}}$. Then the model $\mathbf{PPSP_{New}}$ can be transformed into the following equivalently mixed 0-1 integer linear programming problem:

$$\mathbf{PPSP_R} \begin{cases} \text{Max} \ \ \bar{B}(x,y,z) \\ \text{s.t.} \ \ I(x) = (\leq) m, \\ \qquad \bar{R}_s(x,y,z) \leq b^s, \ s = 1,\ldots,S, & (3.3) \\ \qquad y_{i,j} \leq x_i, \ y_{i,j} \leq x_j, \ y_{i,j} \geq x_i + x_j - 1, \ i,j = 1,\ldots,N, \ i<j, & (3.4) \\ \qquad z_{i,j,k} \leq x_i, \ z_{i,j,k} \leq x_j, \ z_{i,j,k} \leq x_k, \ z_{i,j,k} \geq x_i + x_j + x_k - 2, & (3.5) \\ \qquad\qquad i,j,k = 1,\ldots,N, \ i<j<k, \\ \qquad x_i \in \{0,1\}, \ y_{i,j}, z_{i,j,k} \geq 0, \ i,j,k = 1,\ldots,N, \ i<j<k. \end{cases}$$

It is worthy to point out that y and z in the reformulation $\mathbf{PPSP_R}$ are continuous variables instead of 0-1 binary variables. The equivalence between $\mathbf{PPSP_{New}}$ and $\mathbf{PPSP_R}$ is guaranteed by the linear constraints (3.4) and (3.5), which enforce $y_{i,j} = x_i x_j$ and $z_{i,j,k} = x_i x_j x_k$ when $x_i \in \{0,1\}$. Although the polynomial terms disappear, the reformulation does not make $\mathbf{PPSP_R}$ easy to solve. One main reason is that $\mathbf{PPSP_R}$ introduces $\frac{3N(N-1)}{2} + \frac{2N(N-1)(N-2)}{3}$ linear inequality constraints, which may increase the computational burden in the bounding process of an MILP solver. This motivates us to propose a novel linearization technique that significantly reduces the number of linear constraints in the reformulation for efficient computations.

4 A New Linearization Technique for Reformulation

The key idea of the new linearization technique is based on the following observation. In order to guarantee that $y_{i,j} = x_j x_j$ and $z_{i,j,k} = x_i x_j x_k$ hold, two sets of linear constraints (3.4) and (3.5) are involved in $\mathbf{PPSP_R}$ to capture the relation between x and y, and x and z, respectively. To reduce the number of linear constraints, we could use only one set of linear constraints to enforce the relation of x, y and z simultaneously. The cardinality constraint $\sum_{i=1}^{N} x_i = m$ and the binary constraint $x_i \in \{0,1\}$ help us achieve this purpose. Specifically, observing that

$$x_i \sum_{\substack{j=1 \\ j \neq i}}^{N} x_j = (m-1) x_i \tag{4.1}$$

and

$$x_i \sum_{\substack{j=1 \\ j \neq i}}^{N} x_j \left(\sum_{\substack{k>j \\ k \neq i}}^{N} x_k \right) = \frac{(m-1)(m-2)}{2} x_i, \tag{4.2}$$

following from the fact that there are exact $\frac{(m-1)(m-2)}{2}$ ones in the summation $\sum_{\substack{j=1 \\ j \neq i}}^{N} x_j \left(\sum_{\substack{k>j \\ k \neq i}}^{N} x_k \right)$. Then, the product terms of $x_i x_j$ and $x_i x_j x_k$ can be linearized by auxiliary variables $y_{i,j}$ and $z_{i,j,k}$ according to the next theorem.

Theorem 4.1. *For vectors* $x = (x_1, x_2, \ldots, x_N) \in \{0,1\}^N$, $y = (y_{1,2}, y_{1,3}, \ldots, y_{1,N}, y_{2,3}, \ldots, y_{N-1,N}) \in [0,1]^{N(N-1)/2}$, *and* $z = (z_{1,2,3}, z_{1,2,4}, \ldots, z_{1,2,N}, z_{2,3,4}, \ldots, z_{N-2,N-1,N}) \in [0,1]^{N(N-1)(N-2)/6}$, *if*

$$\sum_{i=1}^{N} x_i = m, \tag{4.3}$$

$$\sum_{j>i}^{N} y_{i,j} + \sum_{j<i}^{N} y_{j,i} + \sum_{j>i}^{N}\sum_{k>j}^{N} z_{i,j,k} + \sum_{j<i}^{N}\sum_{k>i}^{N} z_{j,i,k} + \sum_{j<k}^{N}\sum_{k<i}^{N} z_{j,k,i} = \frac{m(m-1)}{2} x_i \quad \text{for } i = 1\ldots, N, \tag{4.4}$$

then $x_i x_j = y_{i,j}$ *and* $x_i x_j x_k = z_{i,j,k}$ *for* $i,j,k = 1,\ldots,N$ *and* $i<j<k$.

Proof. We prove this theorem by examining three cases.

C1: ($N > 1$, $m = 1$) In this case, only one of x_i's, $i = 1,...,N$, is equal to 1, while others are equal to zero. Hence $x_i x_j = x_i x_j x_k = 0$ for $i,j,k = 1,\ldots,N$ and $i<j<k$. Note that all $y_{i,j}$ and $z_{i,j,k}$'s are nonnegative and the right-hand side of (4.4) is 0, hence $y_{i,j} = z_{i,j,k} = 0$. Consequently, $y_{i,j} = x_i x_j$ and $z_{i,j,k} = x_i x_j x_k$.

C2: ($N > 2$, $m = 2$) Without loss of generality, assume $x_{i'} = x_{j'} = 1$ for some $i' < j'$ and other x_i's are zero. According to the discussion in C1, it follows that $z_{i,j,k} = x_i x_j x_k = 0$ for all $i<j<k$ and $i,j,k = 1,...,N$, because at lease one of i, j and k is not in $\{i', j'\}$. Then equation (4.4) can be further simplified as $\sum_{j>i}^{N} y_{i,j} + \sum_{j<i}^{N} y_{j,i} = x_i$. Similar discussion leads to $y_{i,j} = x_i x_j = 0$ for all $i<j$, except that $i = i'$ and $j = j'$, and $y_{i',j'} = x_{i'} x_{j'} = 1$.

C3: ($N > 3$, $m \geq 3$) Denote $\mathcal{A} = \{i | x_i = 1\} \subseteq \{1,2,\ldots,N\}$. By (4.3), there are m elements in $\mathcal{A}$. Similar to the discussion in C2, we have $y_{i,j} = x_i x_j = 0$ if any one of i, j is not in $\mathcal{A}$. $z_{i,j,k} = x_i x_j x_k = 0$ if any one of i, j and k is not in $\mathcal{A}$. For any $i' \in \mathcal{A}$, the value of the right-hand side of (4.4) is $\frac{m(m-1)}{2}$. Since there are $m-1$ possible nonzero terms like $y_{i',j'}$, $y_{j',i'}$ and $\frac{(m-1)(m-2)}{2}$ possible nonzero terms like $z_{i',j',k'}$, $z_{j',i',k'}$ and $z_{j',k',i'}$ for j', $k' \in \mathcal{A}$ on the left-hand side, and all $y_{i,j}$'s and $z_{i,j,k}$'s are no more than 1, it follows that $y_{i',j'} = x_{i'} x_{j'} = 1$ and $z_{i',j',k'} = x_{i'} x_{j'} x_{k'} = 1$ for $i', j', k' \in \mathcal{A}$ and $i' < j' < k'$.

□

Referring to Theorem 4.1, Model $\mathbf{PPSP_{New}}$ with the equality cardinality constraint $I(x) = m$ can be reformulated as

$$\mathbf{PPSP_{REQ}} \begin{cases} \text{Max} & \bar{B}(x,y,z) \\ \text{s.t.} & (3.3), (4.3)-(4.4), \\ & x_i \in \{0,1\},\ 0 \leq y_{i,j}, z_{i,j,k} \leq 1,\ i,j,k = 1,\ldots,N,\ i<j<k. \end{cases}$$

In order to address the case of inequality cardinality constraint, we introduce a new auxiliary integer variable Q defined as

$$Q = \sum_{t=1}^{m} u_t t, \tag{4.5}$$

where $u_t \in \{0,1\}$ for $t = 1,\dots,m$ and

$$\sum_{t=1}^{m} u_t = 1. \tag{4.6}$$

In fact, Q represents the cardinality of a portfolio since Q is an integer between 1 and m, and $Q = t$ when $u_t = 1$ for some $1 \le t \le m$. By replacing m in (4.4) with Q, the right-hand side of (4.4) becomes $\frac{Q(Q-1)}{2}x_i$, which is a nonlinear term. The following theorem shows that this nonlinear term can also be linearized by introducing some linear inequalities.

Theorem 4.2. *For a set of binary variables $x_i \in \{0,1\}$, $i = 1,\dots,N$, such that $\sum\limits_{i=1}^{N} x_i \le m$ where $0 < m < N$, a set of positive continuous variables Φ_i for $i = 1,\dots,N$, a set of binary variables u_t for $t = 1,\dots,m$, and an integer variable $Q = \sum\limits_{t=1}^{m} tu_t$, the nonlinear product term $\frac{Q(Q-1)}{2}x_i$ can be linearized as Φ_i by the following linear system:*

$$\sum_{i=1}^{N} x_i = \sum_{t=1}^{m} tu_t, \tag{4.7}$$

$$\sum_{t=1}^{m} u_t = 1, \tag{4.8}$$

$$\sum_{t=1}^{m} \frac{t(t-1)}{2}u_t + \frac{m(m-1)}{2}(x_i - 1) \le \Phi_i \le \sum_{t=1}^{m} \frac{t(t-1)}{2}u_t + \frac{m(m-1)}{2}(1 - x_i), \tag{4.9}$$

$$\Phi_i \le \frac{m(m-1)}{2}x_i. \tag{4.10}$$

Proof. From (4.7) and (4.8), we know that $1 \le \sum\limits_{i=1}^{N} x_i \le m$. Since there exists a unique $u_{t'}$ being active (i.e., $u_{t'} = 1$), the product term $\frac{Q(Q-1)}{2}$ can be re-expressed as $\sum\limits_{t=1}^{m} \frac{t(t-1)}{2}u_t$. Moreover,

(i) If $x_i = 1$, then $\Phi_i = \sum\limits_{t=1}^{m} \frac{t(t-1)}{2}u_t$ follows from (4.9);

(ii) If $x_i = 0$, then $\Phi_i = 0$ follows from (4.10).

Therefore, the nonlinear product term $\frac{Q(Q-1)}{2}x_i$ can be represented by Φ_i via the linear system (4.7)-(4.10). □

It is important to note that the binary variables u_t for $t = 1,\dots,m$ in (4.8) can be relaxed as non-negative variables by applying the technique in [9] and [10]. It helps us reduce the computational burden by using only $\lceil \log_2 m \rceil$, the largest integer no bigger than $\log_2 m$ binary variables.

Lemma 4.3. *Given a positive integer m where $0 < m < N$, let $g_{w,t}$ be the binary number satisfying the equations of $1 + \sum\limits_{w=1}^{h} 2^{w-1}g_{w,t} = t$ for $t = 1,...,m$ where $h = \lceil \log_2 m \rceil$. Also let vector $u = (u_1,\dots,u_m) \in [0,\infty]^m$ and a binary vector $\lambda = (\lambda_1,\dots,\lambda_h)$, if*

$$\sum_{t=1}^{m} u_t = 1, \tag{4.11}$$

and

$$\lambda_w = \sum_{t=1}^{m} g_{w,t} u_t \text{ for } w = 1, \ldots, h, \tag{4.12}$$

then $u_t \in \{0, 1\}$ *for* $t = 1, ..., m$.

(Proof follows Theorem 1 of [9]).

Theorem 4.4. *For a given integer* m $(0 < m < N)$*, let* $M = \frac{m(m-1)}{2}$*,* $h = \lceil \log_2 m \rceil$ *and the values of* $g_{w,t}$ *are binary numbers satisfying* $1 + \sum_{w=1}^{h} 2^{w-1} g_{w,t} = t$ *for* $t = 1, ..., m$*. For a set of positive variables* Φ_i*,* $i = 1, \ldots, N$*, binary vectors* $x = (x_1, x_2, \ldots, x_N) \in \{0, 1\}^N$ *with* $\sum_{i=1}^{N} x_i \le m$*,* $\lambda = (\lambda_1, \ldots, \lambda_h) \in \{0, 1\}^h$*, a non-negative vector* $u = (u_1, \ldots, u_m) \in [0, \infty]^m$*, bounded vectors* $y = (y_{1,2}, y_{1,3}, \ldots, y_{1,N}, y_{2,3}, \ldots, y_{N-1,N}) \in [0, 1]^{N(N-1)/2}$ *and* $z = (z_{1,2,3}, z_{1,2,4}, \ldots, z_{1,2,N}, z_{1,3,4}, \ldots, z_{N-2,N-1,N}) \in [0, 1]^{N(N-1)(N-2)/6}$*, if*

$$\sum_{j>i}^{N} y_{i,j} + \sum_{j<i}^{N} y_{j,i} + \sum_{j>i}^{N}\sum_{k>j}^{N} z_{i,j,k} + \sum_{j<i}^{N}\sum_{k>i}^{N} z_{j,i,k} + \sum_{j<k}^{N}\sum_{k<i}^{N} z_{j,k,i} = \Phi_i \quad \text{for } i = 1, \ldots, N, \tag{4.13}$$

$$\lambda_w = \sum_{t=1}^{m} g_{w,t} u_t \quad \text{for } w = 1, \ldots, h, \tag{4.14}$$

$$\sum_{t=1}^{m} u_t = 1, \tag{4.15}$$

$$\sum_{i=1}^{N} x_i = \sum_{t=1}^{m} t u_t, \tag{4.16}$$

$$\sum_{t=1}^{m} \frac{t(t-1)}{2} u_t + M(x_i - 1) \le \Phi_i \le \sum_{t=1}^{m} \frac{t(t-1)}{2} u_t + M(1 - x_i) \quad \text{for } i = 1, \ldots, N, \tag{4.17}$$

and

$$\Phi_i \le M x_i \quad \text{for } i = 1, \ldots, N, \tag{4.18}$$

then $x_i x_j = y_{i,j}$ *and* $x_i x_j x_k = z_{i,j,k}$ *for* $i, j, k = 1, \ldots, N$ *and* $i < j < k$.

Proof. From equations (4.14) and (4.15), it follows that the vector u is binary according to Lemma 4.3. Then the conditions in Theorem 4.2 hold, and expressions (4.15)-(4.18) imply $\Phi_i = \frac{Q(Q-1)}{2} x_i$ for $i = 1, \ldots, N$. Consequently, equation (4.13) leads to $x_i x_j = y_{i,j}$ and $x_i x_j x_k = z_{i,j,k}$ for $i, j, k = 1, \ldots, N$ and $i < j < k$ according to Theorem 4.1. □

Theorem 4.4 indicates that if the cardinality of a project portfolio is restricted by $\sum_{i=1}^{N} x_i \le m$, then the product terms of $x_i x_j$ and $x_i x_j x_k$ can be linearized as $y_{i,j}$ and $z_{i,j,k}$ by using $\lceil \log_2 m \rceil$ binary variables (i.e., λ_w), $N(N-1)/2 + N(N-1)(N-2)/6$ bounded variables (i.e., $y_{i,j}$ and $z_{i,j,k}$) and N non-negative variables (i.e., Φ_i) in $2 + 4N + \lceil \log_2 m \rceil$ linear constraints (i.e.,(4.13)-(4.18)).

Referring to Theorem 4.4, model $\mathbf{PPSP_{New}}$ with inequality cardinality constraint $I(x) \leq m$ can be reformulated as

$$\mathbf{PPSP_{RIEQ}} \begin{cases} \text{Max} & \bar{B}(x,y,z) \\ \text{s.t.} & (3.3), (4.13) - (4.18), \\ & 0 \leq \Phi_i,\ i = 1, \ldots, N, \\ & x_i,\ \lambda_w \in \{0,1\},\ i = 1, \ldots, N,\ w = 1, \ldots, h, \\ & 0 \leq y_{i,j},\ z_{i,j,k} \leq 1,\ i,j,k = 1, \ldots, N,\ i < j < k. \end{cases}$$

Table 1 summarizes the required numbers of binary variables, continuous variables and linear constraints in reformulations $\mathbf{PPSP_R}$, $\mathbf{PPSP_{REQ}}$ and $\mathbf{PPSP_{RIEQ}}$. These three reformulations require about the same number of binary variables and continuous variables, while the proposed reformulations $\mathbf{PPSP_{REQ}}$ and $\mathbf{PPSP_{RIEQ}}$ involve much fewer linear constraints than $\mathbf{PPSP_R}$. Hence, it is expected that the proposed reformulations have the potential to solve large-size instances.

Table 1: Comparison of problem sizes for $\mathbf{PPSP_R}$, $\mathbf{PPSP_{REQ}}$ and $\mathbf{PPSP_{RIEQ}}$.

Reformulation	# of 0-1 variables	# of continuous variables	# of linear constraints
$\mathbf{PPSP_R}$	N	$N(N-1)/2+$ $N(N-1)(N-2)/6$	$1 + S + 3N(N-1)/2+$ $4N(N-1)(N-2)/6$
$\mathbf{PPSP_{REQ}}$	N	$N(N-1)/2+$ $N(N-1)(N-2)/6$	$1 + S + N + N(N-1)/2+$ $N(N-1)(N-2)/6$
$\mathbf{PPSP_{RIEQ}}$	$N + \lceil \log_2 m \rceil$	$m + N + N(N-1)/2+$ $N(N-1)(N-2)/6$	$2 + S + 4N + \lceil \log_2 m \rceil +$ $N(N-1)/2 + N(N-1)(N-2)/6$

5 Computational Experiments

In this section, we first discuss the procedure to generate test instances and describe all instances with various combinations of N and m. In order to compare the efficiency of different reformulations, each test instance is first solved by the commercial software `Lingo` [13] and then by the mixed-integer linear programming solver `Gurobi` [6] with default options for the reformulations $\mathbf{PPSP_R}$, $\mathbf{PPSP_{REQ}}$ and $\mathbf{PPSP_{RIEQ}}$, sequentially. All the experimental tests are conducted on a PC equipped with the Intel Core 2 Duo CPU 1.87GHz, 4GB RAM and Windows 7 (64 bit) operating system. The running time for all instances are limited to 3 CPU hours.

5.1 Generation and description of instances

We set $S = 5$ for all instances. Other parameters, including resource limit (b^s), resource requirements (d_i^s, $d_{i,j}^s$, $d_{i,j,k}^s$) and benefits ($r_i, r_{i,j}, r_{i,j,k}$), are randomly generated as follows.

(i) b^s is a random integer uniformly distributed over $[0.05G^s, 0.1G^s]$, where $G^s = \sum_i d_i^s - \sum_i \sum_{j>i} d_{i,j}^s + \sum_i \sum_{j>i} \sum_{k>j} d_{i,j,k}^s$ for $s = 1, \ldots, 5$.

(ii) d_i^s, $d_{i,j}^s$, $d_{i,j,k}^s$ are random integers uniformly distributed over $[1, 10]$, $[5, 20]$ and $[10, 50]$, respectively, for $i,j,k = 1, \ldots, N$, $i < j < k$ and $s = 1, \ldots, 5$, such that $d_i^s < d_{i,j}^s < d_{i,j,k}^s$ (ref. [16]).

(iii) r_i, $r_{i,j}$ and $r_{i,j,k}$ are random integers uniformly distributed over $[10, 100]$, $[50, 200]$ and $[100, 500]$, respectively, for $i, j, k = 1, \ldots, N$ and $i < j < k$.

We generate three types of instances using the above procedure. The first type, called *Small-PPSP*, has 50 random instances with $(N, m) = (16, 3)$. The second type, called *Medium-PPSP*, includes 50 random instances with $(N, m) = (32, 5)$. The third type, called *Large-PPSP*, has 50 random instances with $(N, m) = (64, 7)$. Table 2 describes the details of these three types of instances.

Table 2: Details of the three types of test instances

Instance Type	# Instances	(N, m)
Small-PPSP	50	$(16, 3)$
Medium-PPSP	50	$(32, 5)$
Large-PPSP	50	$(64, 7)$

5.2 Numerical Results

In this section, we display the numerical results for different reformulations. It turns out that LINGO can not solve any type of instance for the original formulation $\mathbf{PPSP_{New}}$ within 3 CPU hours. Hence, we do not show the results for $\mathbf{PPSP_{New}}$ here. This fact shows that the randomly generated instances, even the type of Small-PPSP, are not easy to solve. On the other hand, the main purpose of the experiment is to compare the performance of MILP solver on the reformulations $\mathbf{PPSP_{R}}$, $\mathbf{PPSP_{REQ}}$ and $\mathbf{PPSP_{RIEQ}}$. We show these results based on two cases: with equality cardinality constraint or with inequality cardinality constraint.

5.2.1 Test results for $\mathbf{PPSP_{R}}$ with $I(x) = m$ and $\mathbf{PPSP_{REQ}}$

The computational results for reformulations $\mathbf{PPSP_{R}}$ with equality cardinality constraint and $\mathbf{PPSP_{REQ}}$ are reported in Table 3. GUROBI is unable to solve any instance of type Large-PPSP within 3 CPU hours for $\mathbf{PPSP_{R}}$ while it solves all instances within 100 seconds for $\mathbf{PPSP_{REQ}}$. The running time for solving $\mathbf{PPSP_{REQ}}$ is much shorter than that for $\mathbf{PPSP_{R}}$, especially for the instances of types of Medium-PPSP and Large-PPSP. This is reasonable because the number of linear constraints for $\mathbf{PPSP_{REQ}}$, as shown in the column "# of linear constraints" of Table 3, is one order of magnitude less than $\mathbf{PPSP_{R}}$ while the number of variables are the same for these two reformulations.

Table 3: Computational results for $\mathbf{PPSP_{R}}$ with $I(x) = m$ and $\mathbf{PPSP_{REQ}}$.

(N, m)	Reformulation	# of 0-1 variables	# of continuous variables	# of linear constraints	Avg. CPU time (sec.)	Std. dev. CPU time(sec.)
(16, 3)	$\mathbf{PPSP_{R}}$	16	680	2,606	3.10	0.95
	$\mathbf{PPSP_{REQ}}$	16	680	702	2.90	0.70
(32, 5)	$\mathbf{PPSP_{R}}$	32	5,456	21,334	1587.34	395.79
	$\mathbf{PPSP_{REQ}}$	32	5,456	5,494	52.73	17.90
(64, 7)	$\mathbf{PPSP_{R}}$	64	43,680	172,710	-	-
	$\mathbf{PPSP_{REQ}}$	64	43,680	43,750	73.45	19.71

"-": indicates the running time for GUROBI exceeds 3 CPU hours for all instances.

5.2.2 Test results of $\mathbf{PPSP_R}$ with $I(x) \leq m$ and $\mathbf{PPSP_{RIEQ}}$

The computational results for reformulations $\mathbf{PPSP_R}$ with inequality cardinality constraint and $\mathbf{PPSP_{RIEQ}}$ are reported in Table 4. Similar to the results in Table 3, none of the instances of type Large-PPSP can be solved by GUROBI within 3 hours. Large-PPSP cannot be solved within 3 hours only "for reformulation $\mathbf{PPSP_R}$". While it takes about half an hour on average to solve instances of type Large-PPSP for reformulation $\mathbf{PPSP_{RIEQ}}$, the average running time for instances of types Small-PPSP and Medium-PPSP is less than 20 seconds. Moreover, the running time for $\mathbf{PPSP_{RIEQ}}$ is at least one order of magnitude smaller than $\mathbf{PPSP_R}$ because the number of linear constraints in $\mathbf{PPSP_{RIEQ}}$ is much fewer than the one in $\mathbf{PPSP_R}$.

Table 4: Computational results for $\mathbf{PPSP_R}$ with $I(x) \leq m$ and $\mathbf{PPSP_{RIEQ}}$.

(N, m)	Reformulation	# of 0-1 variables	# of continuous variables	# of linear constraints	Avg. CPU time (sec.)	Std. dev. CPU time(sec.)
(16, 3)	$\mathbf{PPSP_R}$	16	680	2,606	7.34	1.71
	$\mathbf{PPSP_{RIEQ}}$	18	699	772	0.29	0.09
(32, 5)	$\mathbf{PPSP_R}$	32	5,456	21,334	931.45	234.51
	$\mathbf{PPSP_{RIEQ}}$	35	5,493	5,631	13.56	4.15
(64, 7)	$\mathbf{PPSP_R}$	64	43,680	172,710	-	-
	$\mathbf{PPSP_{RIEQ}}$	67	43,751	44,017	1,756.32	580.18

"-": indicates the running time for GUROBI exceeds 3 hours for all instances.

6 Conclusion

In this paper, we have proposed a new project portfolio selection model which considers both of the project interdependency and cardinality constraints simultaneously. The model is first reformulated as a mixed integer linear programming problem by using the linearization technique in [5]. To further enhance the computational efficiency, we have developed a new linearization technique to construct new reformulations involving significant fewer linear constraints. In terms of the running time and the ability to solve instances of relativly large size, our computational results indicate that the new reformulations derived from the proposed linearization technique have the potential to solve PPSP with large number of project candidates.

Acknowledgments

The authors would like to sincerely thank the anonymous reviewers for their thoughtful and valuable comments which have significantly improved the quality of this paper.

References

[1] A.G. Beged-Dov, Optimal assignment of research and development projects in a large company using an integer programming model, *IEEE Trans. Eng. Manag.* 12 (1965) 138–142.

[2] R. Bhattacharyya, P. Kumar and S. Kar, Fuzzy R & D portfolio selection of interdependent projects, *Comput. Math. Appl.* 62 (2011) 3857–3870.

[3] R.L. Carraway and R.L. Schmidt, An improved discrete dynamic programming algorithm for allocating resources among interdependent projects, *Manage. Sci.* 37 (1991) 1195–1200.

[4] W. Cho, M.J. Shaw and H.K. Dharma, The effect of synergy enhancement on information technology portfolio selection, *Inf. Technol. Manag.* 14 (2013) 125–142.

[5] F. Glover and E. Woolsey, Converting the 0-1 polynomial programming problem to a 0-1 linear program, *Oper. Res.* 22 (1974) 180–182.

[6] GUROBI, Gurobi Optimizer, version 5.6.2, <http://www.gurobi.com>, 2014.

[7] W.J. Gutjahr, S. Katzensteiner, P. Reiter, C. Stummer and M. Denk, Competence-driven project portfolio selection, scheduling and staff assignment, *Cent. Eur. J. Oper. Res.* 16 (2008) 281–306.

[8] S.H. Ji, X.J. Zheng and X.L. Sun, An improved convex 0-1 quadratic program reformulation for quadratic knapsack problems, *Pac. J. Optim.* 8 (2012) 75–87.

[9] H.-L. Li, Y.-H. Huang and S.-C. Fang, A logarithmic method for reducing binary variables and inequality constraints in solving task assignment problems, *INFORMS J. Comput.* 25 (2013) 643–653.

[10] H.-L. Li, Y.-H. Huang and S.-C. Fang, Linear reformulation of polynomial discrete programming for fast computation, to appear in *INFORMS J. Comput.* (2016).

[11] X.M. Li, S.-C. Fang, Y. Tian and X.L. Guo, Expanded model of project portfolio selection problem with divisibility, time profile factors and cardinality constraints, *J. Oper. Res. Soc.* 66 (2015) 1132–1139.

[12] X.M. Li, S.-C. Fang, X.L. Guo, Z.B. Deng and J.X. Qi, An extended model for project portfolio selection with project divisibility and interdependency, *J. Syst. Sci. Syst. Eng.* 25 (2016) 119–138.

[13] LINGO, LINDO System Inc., Release 9.0, <http://www.lindo.com>, 2004.

[14] J.H. Lorie and L.J. Savage, Three problems in rationing capital, *J. Bus.* 28 (1955) 229–239.

[15] M. Nowak, Project portfolio selection using interactive approach, *Procedia Eng.* 57 (2013) 814–822.

[16] R. Santhanam and G.J. Kyparisis, A decision model for interdependent information system project selection, *European J. Oper. Res.* 89 (1996) 380–399.

[17] J.A. Sefair and A.L. Medaglia, Towards a model for selection and scheduling of risky projects, in *Systems and Information Engineering Design Symposium*, IEEE, 2005, pp. 158–164.

[18] V.V. Servakh and S.L. Sukhikh, Hybrid algorithm for scheduling with regard for reinvestment of profits, *Autom. Remote Control* 65 (2004) 449–455.

[19] C. Stummer and K. Heidenberger, Interactive R & D portfolio analysis with project interdependencies and time profiles of multiple objectives, *IEEE T. Eng. Manage.* 50 (2003) 175–183.

[20] L.J. Watters, Reduction of integer polynomial programming problems to zero-one linear programming problems, *Oper. Res.* 15 (1967) 1171–1174.

[21] L. Yu, S.Y. Wang, F.H. Wen and K.K. Lai, Genetic algorithm-based multi-criteria project portfolio selection, *Ann. Oper. Res.* 197 (2012) 71–86.

[22] A. Zuluaga, J.A. Sefair and A.L. Medaglia, Model for the selection and scheduling of interdependent projects, in *Systems and Information Engineering Design Symposium*, IEEE, 2007, pp. 1–7.

Manuscript received 28 September 2014
revised 25 December 2014
accepted for publication 1 April 2015

Xingmei Li
School of Economics and Management
North China Electric Power University
Beijing 102206, China
E-mail address: xingmeil@163.com

Yao-Huei Huang
School of Economics and Management
Southwest Jiaotong University
Chengdu 610031, China
E-mail address: yaohuei.huang@gmail.com

Shu-Cherng Fang
Industrial and Systems Engineering Department
North Carolina State University
Raleigh, NC 27695-7906, USA
E-mail address: fang@ncsu.edu

Zhibin Deng
School of Economics and Management
University of Chinese Academy of Sciences
Key Laboratory of Big Data Mining and
Knowledge Management
Chinese Academy of Sciences
Beijing 100190, China
E-mail address: zhibindeng@ucas.ac.cn