ECONTECHMOD. AN INTERNATIONAL QUARTERLY JOURNAL – 2016, Vol.5, No.4, 29-36

COMBINATORIAL APPROACHES TO THE CAPITAL-BUDGETING PROBLEM

O. Pichugina

Kharkiv National University of Radio Electronics; e-mail: pichugina_os@mail.ru

Received September 22.2016: accepted October 20.2016

Abstract. Optimization approaches, combinatorial and continuous, to a capital-budgeting problem (CBP) are presented. This NP-hard problem, traditionally modelled as a linear binary problem, is represented as a biquadratic over an intersection of a sphere and a supersphere. This allows applying nonlinear optimization to it. Also, the method of combinatorial and surface cuttings (MCSC) is adopted to (CBP). For the single constrained version (1CBP), new combinatorial models are introduced based on joint analysis of the constraint, objective function, and feasible region. Equivalence of (1CBP) to the multi-choice knapsack problem (MCKP) is shown. Peculiarities of Branch&Bound techniques to (1CBP) are described.

Key words: capital-budgeting problem, integer programming, knapsack problem, combinatorial optimization, Branch and Bound.

INTRODUCTION

Nowadays, attraction of investment funds are relevant more than ever [1]. However, even more important is their rational management [2]. Capital-budgeting modelling - is a universal tool that allows applying optimization techniques to the current management of (possibly) thousands of capital projects that yields the greatest return on investment and satisfies specified financial, regulatory and project relationship requirements [3, 4], as well as to carry out a rational long-term planning [5]. In general, vision of the potential cash flows is necessary in a direct management, the same as at the stage of developing business plans.

Consider the following capital-budgeting problem (CBP) [5]: select potential investments out of the set $\mathbf{X}=\{X_i\}_{i=\overline{1,n}}$ maximizing total contribution from all investments without exceeding the limited availability of resources $\mathbf{R}=\{R_j\}_{j=\overline{1,m}}$ if partial investments are not permitted and are given: a) limits $b_j, j=\overline{1,m}$ on the resources, b) contribution c_i resulting from the investment X_i, $i=\overline{1,n}$, c) the amount a_{ji} of resource R_j required for the investment X_i, $i=\overline{1,n}, j=\overline{1,m}$.

The limited resources might be cash, manpower, time, etc., the investment decisions – a choice among possible plant locations, selecting a configuration of capital equipment, picking a set of research&development projects, and so on.

Another scenario for (CBP) [5] is a long-range planning. In this case:

a) m is the number of periods of planning;

b) $\mathbf{R}$ - are the periods;

c) a_{ji} is the net cash flow from the investment X_i in the period R_j, $i=\overline{1,n}, j=\overline{1,m}$;

d) b_j represents the incremental exogenous cash flow in the period R_j, $j=\overline{1,m}$.

All the parameters a_{ji}, c_i, b_j can be arbitrary integers. For instance, in the long-range planning (CBP)-version, $a_{ij}>0$ if the investment X_i requires additional cash in the period R_j, $a_{ij}<0$ if the investment X_i generates cash in the period R_j, while $a_{ij}=0$ if it neither requires nor generates cash. Also, $b_j>0$ if additional funds are made available in the period R_j, $b_j<0$ if funds are withdrawn in this period, otherwise, $b_j=0$. Finally, if $c_i>0$ the investment X_i is beneficial, if $c_i<0$ then it is harmful, otherwise, X_i is neutral.

If a plan of the investments denote:

$$x=(x_i)_{i\in J_n}:\ J_n=\{1,...,n\},$$

$$x_i=\begin{cases}1 \text{ if the investment } \mathrm{R}_i \text{ accepted,}\\ 0 \text{ if the investment } \mathrm{R}_i \text{ is rejected,}\end{cases} \tag{1}$$

then the problem is formalized as follows: find a boolean vector (1) maximizing $z=c^T x$ subject to constraints that the funds required for investment are enough for the whole planning horizon.

Let x^* is an optimal plan, then the mathematical model of (CBP) is [3, 5]:

$$z^*=max\ c^T x,\ x^*=\arg max\ c^T x, \tag{2}$$

$$x\in B_n=\{0,1\}^n, \tag{3}$$

$$a_j^T x\le b_j,\ j\in J_m, \tag{4}$$

where:

$$a_j, c\in \mathrm{R}^n,\ b_j\in \mathrm{R},\ j\in J_m. \tag{5}$$

If $m>1$ it is a multiply constrained (CBP), (mCBP), if $m=1$ - is single constrained (1CBP) which looks like (2), (3), subject to

$$a^T x \le b, \quad (6)$$

$$a, c \in \mathrm{R}^n,\ b \in \mathrm{R}. \quad (7)$$

Formulas (2)-(5) is a particular case of integer programs, namely, it is a linear constrained binary program. Exactly it is solvable with help of branch&bound (B&B), cutting plane methods, or a combination of both - branch&cut techniqies. Also, it can be solved approximately by heuristics such as tabu search, hill climbing, simulated annealing, evolutionary and genetic algorithms, as well as asymptotically by asymptotic integer algorithms [4-9].

Typically, in (CBP) there are present two types of investments: beneficial, that required resources, and harmful, that generates cash. In this case, (1CBP) is reducible to a knapsack problem, (KP), (0-1KP) [9-12]: n objects with positive values (profits, utilities) c_i and weights a_i ($i \in J_n$) are given and a knapsack of a capacity b is formed from them with maximal total value (profit, utility).

It's mathematical model is (2), (3), (6)

$$a, c \in \mathrm{R}^n_{++},\ b \in \mathrm{R}_{++}.$$

Similarly, (mCBP) becomes the multiple constrained (KP) (mKP) [10,11] if (4) are knapsack constraints [13], i.e., there is holds:

$$a_j,\ c \in \mathrm{R}^n_{++},\ b_j \in \mathrm{R}_{++},\ j \in J_m. \quad (8)$$

Detecting (KP)-type problems among (CBP) allows applying various solution approaches specific to (KP) exactly and approximately. Among exact approaches are dynamic programming (DP), (B&B), and hybridizations of both; the integer hull search with cutting planes and tightening constraints. Among approximate are heuristics, reduction and asymptotic methods, e.g., greedy and fully polynomial time approximation schemes [9-13].

OBJECTIVES

The purpose of the paper is to present new approaches to (CBP) based on analysis of properties of nonlinear functions, as well as peculiarities of all components of the problem – the feasible discrete set, constraints, and objective function.

THE ANALYSIS OF RECENT RESEARCHES AND PUBLICATIONS

In recent years, heuristic evolutionary and genetic algorithms have been intensively developed in integer programming, in particular, for (KP) [14].

Recent investigations concerned, primarily, methods specific to various KP generalizations such as the multidimensional and multi-objective (KP) [15], generalized assignment and quadratic (KP) [16].

A great success was achieved in approximate (KP)-solving. As reported in [14], instances of dimensions up to 100000 are solvable by DP, greedy and genetic algorithms, of which the first is exact and last shows better results than greedy. Note that execution time of DP is, on average, 10 times more than of the approximate ones. At the same time, B&B handle problems with at most 60000 variables.

From our point of view, a promising way to solve exactly large-size (CBP) is in constructing its new Euclidean combinatorial models [17, 18] on B_n-subsets, investigating properties of the subsets, then applying them in optimization. The optimization approaches can be combinatorial, such as branch&bound and branch&cut techniques [19, 20], as well as continuous based on functional representations of these sets [19, 20] and their inscription into a hypersphere. Among the continuous approaches are cutting plane techniques [21] and equivalent unconstrained reformulations based on extensions of objective functions [19, 20].

THE MAIN RESULTS OF THE RESEARCH

Introduce some terminologies.

A numerical 1-multiset (or a multiset) [17] is a collection of numbers:

$$G = \{g_i\}_{i \in J_n} :\ g_i \in R,\ i \in J_n. \quad (9)$$

Without loss of generality, we can assume that its elements are ordered:

$$g_i \le g_{i+1},\ i \in J_{n-1}. \quad (10)$$

A multiset is defined by a set $S(G)$ of its different elements, a basis, and multiplicities, a G-primary specification $[G]$ [18]:

$$S(G) = \{e_i\}_{i \in J_k} :\ e_i < e_{i+1},\ i \in J_{k-1}; \quad (11)$$

$$[G] = (n_i)_{i \in J_k} : n_i - \text{is a multiplicity of } e_i. \quad (12)$$

Now, G is representable as follows [18]:

$$G = \left\{ e_j^{n_j} \right\}_{j \in J_k} : \sum_{j=1}^{k} n_j = n. \quad (13)$$

A 2-multiset is a collection of 2-tuples:

$$G = \{g_i\}_{i \in J_n} :\ g_i = \begin{pmatrix} g_i^1 \\ g_i^2 \end{pmatrix} \in R^2,\ i \in J_n. \quad (14)$$

Here, we assume that the tuples are ordered lexicographically:

$$g_i \le^{lex} g_{i+1},\ i \in J_{n-1}, \quad (15)$$

implying that:

$$\forall i \in J_{n-1}\ g_i^1 \le g_{i+1}^1;\ \text{if}\ g_i^1 = g_{i+1}^1,\ g_i^2 \le g_{i+1}^2. \quad (16)$$

Similarly to a 1-multiset, different tuples of a 2-multiset (14) form its basis $S(G)$ whose elements are strictly lexicographically ordered:

$$S(G) = \{e_j\}_{j \in J_k}\ :\ e_j = \begin{pmatrix} e_i^1 \\ e_i^2 \end{pmatrix} \in R^2,\ j \in J_k; \quad (17)$$
$$e_j \prec e_{j+1} \Leftrightarrow e_j \le^{lex} e_{j+1},\ e_j \ne e_{j+1}.$$

In terms of e_j-coordinates, this means that:

$$\forall i \in J_{k-1}\ e_i^1 \le e_{i+1}^1;\ \text{if}\ e_i^1 = e_{i+1}^1,\ e_i^2 < e_{i+1}^2. \quad (18)$$

Now, similar to a 1-multiset, a G-primary specification is defined by (12) and the 2 multiset is representable in the form (13).

A set $\overline{S}_k^n(G)$ is a <u>set</u> of n-combinations with repetitions from the multiset (13) with $[G] = (k^n)$ [22]. Its elements are ordered n-samples from G whose coordinates are ordered non-decreasingly:

$$\Big\{\overline{S}_k^n(G) = x \in R^n\ :\ x_i \in S(G),\ i \in J_n; \quad (19)$$
$$x_i \le x_{i+1},\ i \in J_{n-1}\Big\}.$$

A convex hull of (19) is a <u>polytope</u> $\overline{Q}_k^n(G)$ of n-combinations with repetitions [22] which is a n-simplex:

$$\overline{Q}_k^n(G) = conv\overline{S}_k^n(G) = \quad (20)$$
$$\left\{x \in R^n\ :\ x_1 \ge e_1,\ x_n \le e_k;\ x_i \le x_{i+1}, i\right\}.$$

After eliminating the constraint on ordering x-coordinates, $\overline{S}_k^n(G)$, $\overline{Q}_k^n(G)$ become a <u>set</u> $\overline{E}_k^n(G)$ and a polytope $\overline{\Pi}_k^n(G)$ of permutations with repetitions, respectively [18]:

$$\overline{E}_k^n(G) = \left\{x \in R^n\ :\ x_i \in S(G),\ i \in J_n\right\}, \quad (21)$$

$$\overline{\Pi}_k^n(G) = \left\{x \in R^n\ :\ \mathbf{e}_1 \le x \le \mathbf{e}_k\right\}. \quad (22)$$

A particular case of (21), (22) are the Boolean set and unit hypercube [19, 20]:

$$B_n = \{0,1\}^n = \overline{E}_2^n\left(\{0^n, 1^n\}\right),$$
$$PB_n = [0,1]^n = \overline{\Pi}_2^n\left(\{0^n, 1^n\}\right).$$

If, in a zero-one multiset, multiplicities of $0,1$ can be restricted:

$$G = \left\{0^{\eta_1}, 1^{\eta_2}\right\} : 1 \le \eta_1, \eta_2 \le n,\ \eta = \eta_1 + \eta_2 \ge n, \quad (23)$$

the corresponding B_n - subset is a Boolean permutation set $B_n(\eta_2)$ [20] if $\eta = n$ and it is a Boolean partial permutation set $B_n(n - \eta_1, \eta_2)$ [20] if $\eta > n$:

$$\forall a \in R\ \mathbf{a} = (a)_{i \in J_n}$$
$$B_n(\eta_2) = \left\{x \in B_n\ :\ x^T\mathbf{1} = \eta_2\right\}, \quad (24)$$
$$B_n(n - \eta_1, \eta_2) = \left\{x \in B_n\ :\ n - \eta_1 \le x^T\mathbf{1} \le \eta_2\right\}.$$

Convex hulls of the sets (24) are $n-1$-hypersimplex and n - hypersimplex [20]:

$$\Delta_{n,\eta_1,\eta_2} = \left\{x \in B_n\ :\ n - \eta_1 \le x^T\mathbf{1} \le \eta_2\right\}. \quad (25)$$

A particular case of (25) is a unit n-simple<u>x</u>:

$$\Delta_{n,\eta_2} = \left\{x \in B_n\ :\ x^T\mathbf{1} = \eta_2\right\},$$

$$\Delta_{n,0,1} = \text{conv}\ B_n(0,1) = \left\{x \ge \mathbf{0} : x^T\mathbf{1} \le 1\right\}. \quad (26)$$

One more B_n-subset is a Boolean set of combinations with repetitions $\overline{S}_2^n\left(\{0^n, 1^n\}\right)$.

The Cartesian product of combinatorial sets is called a <u>set</u> of the sets' tuples.

Let J_n be partitioned into l subsets:

$$J_n = \bigcup_{j=1}^{l} I_j,\ |I_j| = n_j > 0,\ j \in J_l,\ \sum_{j=1}^{l} n_j = n, \quad (27)$$

$$\overline{n} = (n_j)_{j \in J_l},\ \overline{2} = (2^l), \quad (28)$$

Then, for instance,

$$\bar{\mathbf{S}}_{\bar{2}}^{\bar{n}}\left(\left\{0^n,1^n\right\}\right)=\underset{j=1}{\overset{l}{\otimes}}\ \bar{\mathrm{S}}_2^{n_j}\left(\left\{0^{n_j},1^{n_j}\right\}\right), \quad (29)$$

$$\mathbf{B}_{\bar{n}}\left(0,1\right)=\underset{j=1}{\overset{l}{\otimes}}\ B_{n_j}\left(0,1\right) - \quad (30)$$

are a set of tuples of the 0-1 combinations with repetitions and a 0-1 set of tuples sum to at most 1, respectively.

Preliminary stage. It is known [10], (1CBP) is reducible to (KP) and after this transformation the problem dimension (possibly) decreases. Recall its stages and then modify them for (mCBP) and apply.

1. Denote

$$I^0=\left\{i:a_i\ge 0,c_i\le 0\right\},I^1=\left\{i:a_i\le 0,c_i\ge 0\right\}$$
$$I^+=\left\{i:a_i,c_i>0\right\},\ I^-=\left\{i:a_i,c_i<0\right\}; \quad (31)$$

$$n^{[\cdot]}=\left|I^{[\cdot]}\right|,\ \sum_{[\cdot]\in\{0,1,+,-\}}n^{[\cdot]}=n. \quad (32)$$

2. Assign $x_i^*=\begin{cases}0,\ i\in I^0,\\ 1,\ i\in I^1,\end{cases}$ and reduce the dimension to $n'=n-n^0-n^1$.

3. Introduce new variables:

$$y_i=\begin{cases}x_i,\ i\in I^+,\\ 1-x_i,\ \forall i\in I^-.\end{cases}.$$

Now, (1CBP) is equivalent to (KP):

$$z^*=max\sum_{i\in I^+\cup I^-}\left|c_i\right|y_i+\sum_{i\in I^1\cup I^-}c_i, \quad (33)$$

$$y_i\in\left\{0,1\right\},\ i\in I^+\cup I^-. \quad (34)$$

$$\sum_{i\in I^+\cup I^-}\left|a_i\right|y_i\le b-\sum_{i\in I^1\cup I^-}a_i, \quad (35)$$

For (mCBP), (31), (32) become:

$$I^0=\left\{i:c_i\le 0,\ a_{ji}\ge 0,j\in J_m\right\},$$
$$I^1=\left\{i:c_i\ge 0,\ a_{ji}\le 0,j\in J_m\right\}, \quad (36)$$
$$I^-=\left\{i\notin I^0:c_i<0\right\},\ \bar{I}=J_n\setminus\left\{I^0,I^1,I^-\right\}.$$

Formulas (33)-(35) are transformed into:

$$z^*=max\sum_{i\in\bar{I}\cup I^-}\left|c_i\right|y_i+\sum_{i\in I^1\cup I^-}c_i, \quad (37)$$

$$y_i\in\left\{0,1\right\},\ i\in\bar{I}\cup I^-, \quad (38)$$

$$\sum_{i\in\bar{I}}a_{ji}y_i-\sum_{i\in I^-}a_{ji}y_i\le b_j-\sum_{i\in I^1\cup I^-}a_{ji},\ j\in J_m. \quad (39)$$

The problem (37)-(39) is (mKP) if, in (36),

$$\bar{I}=I^+=\left\{i:c_i>0,\ a_{ji}\ge 0,j\in J_m\right\}, \quad (40)$$

otherwise, it is a general linear binary problem (referred to as (mCBP) again).

Approaches to (mCBP). For this general case, we recommend the following continuous approaches:

1. The method of combinatorial and surface cuttings (MCSC) [21] where a sphere

$$S:\sum_{i=1}^{n}\left(x_i-\frac{1}{2}\right)^2=\frac{n}{4} \quad (41)$$

circumscribed around B_n [19, 20] is used. (41) implies that B_n is polyhedral-spherical [20], therefore we use two continuous relaxations of (mCBP) – spherical and polyhedral [19, 20]:

$$z^S=\max_{x\in S}\ c^Tx,\ x^S=\arg\max_{x\in S}\ c^Tx, \quad (42)$$

$$z^P=\max_{x\in P}\ c^Tx,\ x^P=\arg\max_{x\in P}\ c^Tx, \quad (43)$$

$$P=convB_n\cap\left\{x:a_j^Tx\le b_j,\ j\in J_m\right\}=$$
$$=\left\{x\in\left[0,1\right]^n:a_j^Tx\le b_j,\ j\in J_m\right\}. \quad (44)$$

Assume that $n'=dim P=n$, otherwise, a projection onto n' dimensional space is performed.

Outline (MCSC) in application to (mCBP):

- solve the linear program (43);

if $x^P\in B_n$, then $x^*=x^P$, otherwise, form a right cut for x^P:

  - choose n P-edges intersecting at x^P: $\left\{l_i=\left[x^p,x^i\right]\right\}_{i\in J_n}:\left\{x^i\right\}_i\subset vertP$;
  - use the relaxation (42) extending the edges toward x^i-x^p, $i\in J_n$, up to an intersection with S and get $Y=\left\{y^i\right\}_i\subset S$;
  - construct, trough Y, a hyperplane $\Pi=\left\{x:a_{m+1}^Tx=b_{m+1}\right\}$ and a cut of x^P $D_{m+1}=\left\{x:a_{m+1}^Tx\le b_{m+1}\right\}:a_{m+1}^Tx^P>b_{m+1}$;

- add D_{m+1} to (44), set $m=m+1$, and repeat all these steps iteratively.

2. The Lagrangian and penalty methods based on the following functional representations of B_n [19]:

(R1):
$$f_1(x)=\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i^2 = 0,$$
$$f_2(x)=\sum_{i=1}^{n}\left(x_i-0.5\right)^4 - 0.625n = 0;$$

(R2): $$f_1(x)\le 0,\ f_2(x)\le 0.$$

If $f_0(x)=-c^T x$, $f_{j+2}(x)=a_j^T x - b_j,\ j\in J_m$, then an equivalent problem to (mCBP) is:

$$F(x,\lambda)=f_0(x)+\sum_{j=1}^{m+2}\lambda_j f_j(x)\to min, \tag{45}$$

$$x\in R^n,\ \lambda\in R_+^{m+2}. \tag{46}$$

Formulas (45), (46) is solvable numerically [23] and yields a local minimum for (mCBP).

Another approach is incorporate all constraints into a penalty function [23], e.g.:

$$\Phi(x,\mu)=f_0(x)+$$
$$\mu\left(f_1^2(x)+f_2^2(x)+\sum_{j=3}^{m+2} min\left(0,-f_j(x)\right)^2\right)\to min$$

solvable numerically for increasing sequence of $\mu\in R$, and get also a local minimum for (mCBP). The Lagrangian $F(x,\lambda)$- and penalty minimization techniques can be combined by the augmented Lagrangian method [23].

Approaches to (1CBP). First, transform the problem into (KP). Assume that we deal with (2)-(4),(8). Preliminary, check the following: a) $\sum_i a_i > b$, otherwise, (4) does not work; b) multiplicities of items allow to put the whole group G_j of items of the same weight e_j in the knapsack: $n_j \le \lfloor b/e_j \rfloor,\ j\in J_l$, where:

$$A=\{a_i\}_i=\left\{e_j^{n_j}\right\}_{j\in J_l}$$

$$a_i \le a_{i+1},\ i\in J_{n-1};\ e_j < e_{j+1},\ j\in J_{l-1}. \tag{47}$$

Otherwise, $\forall\, j\in J_l\ n_j' = n_j - \lfloor b/e_j \rfloor$ items of G_j with the smallest values are eliminated from A; c) a capacity of the knapsack does not require each specific G_j: $\sum_{i=1}^{n} a_i - n_j e_j > b$, $\forall j\in J_l$, otherwise, $n_j'' = \left| \left(b-\sum_{i=1}^{n} a_i\right)/e_j \right|$ items of G_j with the largest values are placed in the knapsack.

To get initial feasible solutions x^{**}, lower z^l and upper z^u bounds on z^*, determine values $r_i = c_i/a_i,\ \forall i$ of the profit per unit weight [10]

$$J_n=\{i_j\}_j : r_{i_j}\ge r_{i_{j+1}},\ j\in J_{n-1}.$$

Now, the knapsack x^{**} is filled with the items with the largest r_i-values: $x_{i_j}^{**}=1,\ j\le j_0,\ x_{i_j}^{**}=0,\ j\ge j_0$.

A polyhedral relaxation (43) solution [10]:

$$x_{i_j}^P=1,\ j\le j_0;\ x_{i_j}^P=0,\ j>j_0+1;$$
$$x_{i_{j_0}+1}^P=\left(b-\sum_{j=1}^{j_0} a_{i_j}\right)/a_{i_{j_0}+1}.$$

x^{**}, x^P yield the bounds $z^l=c^T x^{**}=\sum_{j=1}^{j_0} c_{i_j}$,
$z^u=\lfloor z^P \rfloor,\ z^P=c^T x^P=z^l+x_{i_{j_0}+1}^P\cdot c_{i_{j_0}+1}^P$.

Now, a two-sided knapsack constraint [24]:

$$z^l+1\le c^T x\le z^u \tag{48}$$

can be added to (KP).

Another two-sided knapsack constraint - on a number k of items in x^* - may be added:

$$k_1\le k = x^T\mathbf{1}\le k_2. \tag{49}$$

The bounds k_1, k_2 can be found by filling the knapsack with the heaviest and lightest items, respectively:

$$k_2 : \sum_{i=1}^{k_2} a_i\le b,\ \sum_{i=1}^{k_2+1} a_i > b; \tag{50}$$

$$k_1 : \sum_{i=1}^{k_1} a_{n-i+1}\le b,\ \sum_{i=1}^{k_1+1} a_{n-i+1} > b. \tag{51}$$

In terms of the Boolean partial permutation set (see (24)). Now, our (KP) is reformulated as a linear constrained combinatorial problem (referred as (KP.C1)) (2), (6), (48),

$$x\in B_n(k_1,k_2). \tag{52}$$

Notice, (52) may considerably reduce the search domain in comparison with (3).

A specifics of B&B for (KP.C1). As $B_n(k_1,k_2)$ is decomposed into $B_n(k)$-sets:

$$B_n(k_1,k_2) = \bigcup_{k=k_1}^{k_2} B_n(k), \quad (53)$$

the traditional for binary problems branching scheme based on fixing a coordinate [4-11] (we refer it to as Scheme 1) can be combined with another one (Scheme 2), based on analysis of integrity of $k^P = \mathbf{e}^T y^P$. We recommend the following: if $k^P \in Z$, then (Scheme 1) is applied, otherwise, (Scheme 1) is used. It is based on the fact that the feasible region is divisible into branches:

$$B = \left\{x:\ \mathbf{e}^T \mathrm{x} \le \left\lfloor \mathrm{k}^P \right\rfloor\right\},\ B' = \left\{y:\ \mathbf{e}^T \mathrm{x} \ge \left\lceil \mathrm{k}^P \right\rceil\right\}.$$

Now, (KP.C1) is decomposed into two (KP.C1)-subproblems of the same dimension: (2), (6), (48) on $B_n\left(k_1,\left\lfloor \mathrm{k}^P \right\rfloor\right)$, $B_n\left(\left\lceil \mathrm{k}^P \right\rceil,k_2\right)$, respectively. Since $x^P \notin B,B'$, the polyhedral relaxations on two hypersimplexes (25) need to be solved.

For these subproblems, infeasibility of (KP.C1) and irredundancy of constraints (2), (6) are easily verified. It is due to $B_n(k_1,k_2)$ is a kind of the partial permutation set and a linear problem over (53) is solved explicitly [18].

Remark. Ordering (47) allows (possibly) adding new constraints to (KP.C1). Namely:

$$\forall i \in J_{n-1} \text{ if } j < i : c_j \ge c_i \Rightarrow x_i \le x_j, \quad (54)$$

implying a priority of an item that is neither heavier not less valuable than another one.

A model (KP.C2). The observation (54) allows to order variables within each G_j. For that, a 2-multiset $AC = \left\{(a_i,c_i)^T\right\}_{i\in J_n}$ of the items weights and values are ordered: $(a_i,c_i)^T \le^{lex} (a_{i+1},c_{i+1})^T,\ i \in J_{n-1}$.

Now, from (54), there follows: $\forall j \in J_l :$ $n_l > 1\ x_i \le x_{i+1},\ i \in J_{n_j^0} \setminus J_{n_{j-1}^0}$, where $n_0^0 = 0$, $n_j^0 = \sum_{i=1}^{j} n_i,\ j \in J_l$. With (53), this implies (see (19)) that

$$\forall j \in J_l :\ \bar{x}_j = (x_i)_{i\in J_{n_j^0} \setminus J_{n_{j-1}^0}} \in \bar{S}_2^{n_j}\left(\left\{0^{n_j},1^{n_j}\right\}\right), \quad (55)$$

Respectively, according to (27)-(29),

$$x \in \bar{\mathbf{S}}_{\bar{2}}^{n}\left(\left\{0^n,1^n\right\}\right). \quad (56)$$

A new (KP)-model (referred as (KP.C2)) is a linear constrained problem (2),(6),(48),(49), (56) on the set of tuples of 0-1-combimations with repetitions.

Notice a peculiarity of B&B for (KP.C2) that (Scheme 1) of fixing a variable within each $G_j : n_j > 1$, leads to decomposition of the problem into two subproblems of the dimension $n-1,\ n-n_j$. Thus, considering large-size groups first are expected to discard the branches faster.

A model (KP.C3). One more combinatorial model of (KP) will be formed based on the following proposition:

Proposition 1. A linear program (2),(4)

$$x \in \bar{\mathrm{Q}}_k^n(G), \quad (57)$$

is equivalent to a linear problem:

$$z'^* = max\ c'^T y,\ y'^* = \arg max\ c'^T y, \quad (58)$$

$$y \ge \mathbf{0}, \quad (59)$$

$$y^T \mathbf{e} \le e_k - e_1, \quad (60)$$

$$A' y \le b'. \quad (61)$$

Proof. By (20), the polytope $\bar{\mathrm{Q}}_k^n(G)$ is n-simplex given by a system:

$$x_1 \ge e_1. \quad (62)$$

$$x_i \le x_{i+1},\ i \in J_{n-1}. \quad (63)$$

$$x_n \le e_k. \quad (64)$$

Introduce a change of variables:

$$y_1 = x_1 - e_1, \quad (65)$$

$$y_i = x_i - x_{i-1},\ i \in J_n \setminus \{1\}. \quad (66)$$

Formula (65) transforms the (62) into $y_1 \ge 0$, (66) with (63) yields $y_i \ge 0,\ i \in J_n \setminus \{1\}$ Hence (59) holds. The inverse change of variables is:

$$x_i = \sum_{j=1}^{i} y_j + e_1,\ i \in J_n. \quad (67)$$

By (67), the constraint (64) becomes:

$$x_n = \sum_{j=1}^{n} y_j + e_1 \le e_k \text{ or } \sum_{j=1}^{n} y_j \le e_k - e_1$$

that is (60). Transform the constraints (4) in the form:

$$\sum_{i=1}^{n} a_{ji} x_i \le b_j ,\ j \in J_m . \quad (68)$$

Applying (67) to (68), we obtain:

$$\sum_{i=1}^{n} a_{ji}\left(\sum_{j'=1}^{i} y_{j'} + e_1\right) = e_1 \sum_{i=1}^{n} a_{ji} + \sum_{i=1}^{n}\sum_{i'=1}^{i} a_{ji} y_{i'} =$$

$$= e_1 \sum_{i=1}^{n} a_{ji} + \sum_{i=1}^{n} y_i \sum_{i'=i}^{n} a_{ji'} \le b_j ,\ j \in J_m ,$$

wherefrom, (61) is derived with

$$A' = \left(a'_{ji}\right)_{m\times n},\ b' = \left(b'_j\right)_m : b'_j = b_j - e_1 \sum_{i=1}^{n} a_{ji},$$

$$a'_{ji} = \sum_{i'=i}^{n} a_{ji'} ,\ i \in J_n ,\ j \in J_m . \quad (69)$$

Similarly, (2) becomes (58) with

$$c' = \left(c'_i\right)_n :\ c'_i = \sum_{i'=i}^{n} c_{i'} ,\ i \in J_n . \quad (70)$$

Corollary 1. A linear program (2), (4)

$$x \in \overline{\mathrm{Q}}_k^n\left(\left\{0^n, 1^n\right\}\right), \quad (71)$$

is equivalent to a linear program (58), (61), (69), (70),

$$b' = b ,\quad y \in \Delta_{n,0,1} . \quad (72)$$

If, in the corollary, we move on to a vertex set of $\overline{\mathrm{Q}}_k^n\left(\left\{0^n, 1^n\right\}\right)$, (71), (72) are transformed into:

$$x \in \overline{\mathrm{S}}_k^n\left(\left\{0^n, 1^n\right\}\right),$$

$$y \in B_n(0,1) . \quad (73)$$

Corollary 2. A linear program (58), (61), (73) is equivalent to $n+1$-dimension linear multi-choice knapsack problem (MCKP) [10]:

$$z'^{*} = max\ \bar{c}^T \bar{y},\ \bar{y}'^{*} = \arg max\ \bar{c}^T \bar{y}, \quad (74)$$

$$y \in B_{n+1}(1),$$

$$\bar{y} = (y_i),\ \bar{c} = \left(c'_i\right) \in R^{n+1} : y_{i+1} = 1 - \sum_{i=1}^{n} y_i ,$$

subject to (61).

Proposition 2. A linear program (2), (6), (56) is equivalent to linear constrained over (30).

Proof. Combine the linear constraints (6), (48), (49) of (KP.C3) into a system (4) with $m = 5$. Decompose this problem into l subproblems corresponding to each group G_j:

$$z = c^T x = \sum_{j=1}^{l} \bar{c}_j^T \bar{x}_j ,\ \bar{c}_j = \left(c_i\right)_{i \in J_{n_j^0} \setminus J_{n_{j-1}^0}} ,\ j \in J_l ,$$

$$a_i^T x = \sum_{j=1}^{l} \bar{a}_{ij}^T \bar{x}_j \le b_i ,\ \bar{a}_{ij} = \left(a_{ij}\right)_{i \in J_{n_j^0} \setminus J_{n_{j-1}^0}} \quad \forall i,j .$$

Applying Corollary 1 to vectors (55), they are transformed into

$$\bar{y}_j = B_{n_j}(0,1),\ j \in J_l , \quad (75)$$

Subject to five common linear constraints, for all these groups, representable in the form (61). By (30), the combinatorial constraints (73) are combined into:

$$y \in \mathbf{B}_{\bar{n}}(0,1) . \quad (76)$$

The model (58), (61), (74) (referred to as (KP.C3)) is (KP) equivalent reformulation on the 0-1 set of tuples sum to at most 1.

Remark. Further application Corollary 2 to (KP.C3) transforms it into (MCKP) of the dimension $n+l$. Thus we found an equivalent reformulation of (KP) as (MCKP). Now, techniques specific to (MCKP) [9-11] can be applied to the standard (KP), as well as to it's another generalization - (1CBP).

CONCLUSIONS

1. New optimization approaches to the capital-budgeting problem (CBP) are presented. They are based on biquadratic functional representation of B_n. Two of them are continuous and one – combinatorial. These are: an exact cutting plane (MCSC), an approximate based on (CBP)-reformulation as a nonlinear unconstrained problem, and exact – B&B, respectively. The continuous methods are extendable into most (KP)-generalization including nonlinear.

2. A possibility of reducing a feasible region of (1CBP) depending on a presence of repetitions in c-coefficients was studied and three equivalent combinatorial models of (1CBP) were obtained – on $B_n(k_1,k_2)$, $\overline{\mathbf{S}}_2^{\bar{n}}\left(\left\{0^n,1^n\right\}\right)$, and $\overline{\mathbf{B}}_{\bar{n}}(0,1)$. A new branching scheme based on B_n-decomposition into $B_n(k)$-sets are recommended to (1CBP).

REFERENCES

1. **Paranchuk S., Korbutyak A., 2013.** Problems investment support innovative development of the national economy and solutions. Econtechmod. An Internetional Quaterly Journal, 2(4), 53–60.

2. **Podolchak N., Melnyk L. and Chepil B., 2014**. Improving administrative management costs using optimization modeling. Econtechmod. An Internetional Quaterly Journal, 1(1), 75–80.

3. **Markowitz H., Manne A., 1957**. On the Solution of Discrete Programming Problems. Econometrica, (1), 84-110.

4. **Hoffman K., Padberg M., 2001**. Combinatorial and Integer Optimization. Encyclopedia of Operations Research & Management Science, 94–102.

5. **Bradley S., Hax A. and Magnanti T., 1977.** Applied Mathematical Programming. Reading, Mass: Addison-Wesley, 716.

6. **Schrijver A., 1998**. Theory of Linear and Integer Programming. Chichester, New-York: Wiley, 484.

7. **Wolsey L. and Nemhauser G., 1999**. Integer and Combinatorial Optimization (1 edition). New York: Wiley-Interscience, 763.

8. **Papadimitriou C., Steiglitz K., 2013**. Combinatorial Optimization: Algorithms and Complexity. New York: Dover Publications, 528.

9. **Taha H., 2014.** Integer Programming: Theory, Applications, and Computations. New York: Academic Press, 392.

10. **Martello S. and Toth P., 1990**. Knapsack Problems: Algorithms and Computer Implementations. Chichester, New York: Wiley, 308.

11. **Kellerer H., Pferschy U., Pisinger D., 2010.** Knapsack Problems (Softcover reprint of hardcover 1st ed. 2004 edition). Berlin, New York: Springer, 548.

12. **Hiroshi I., 2014**. Some Thoughts on the 2-Approximation Algorithm for Knapsack Problems: A Survey. Far East Journal of Applied Mathematics, 87(3), 257–267.

13. **Escudero L., Martello S. and Toth P., 1998**. On tightening 0-1 programs based on extensions of pure 0-1 knapsack and subset-sum problems. Annals of Operations Research, 81(1–4), 379–404.

14. **Shaheen A., Sleit A., 2016.** Comparing between different approaches to solve the 0/1 Knapsack problem. International Journal of Computer Science and Network Security, 16(7), 1–10.

15. **Lust T., Teghem J., 2012**. The multiobjective multidimensional knapsack problem: a survey and a new approach. International Transactions in Operational Research, 19(4), 495–520.

16. **Öncan T., 2007**. A Survey of the Generalized Assignment Problem and Its Applications. INFOR, 45(3), 123–141.

17. **Stoyan Y., Yakovlev S., 1986**. Mathematical models and optimization methods in Geometric Design. Kiev: Naukova Dumka, 268. (in Russian).

18. **Stoyan Y., Yemets O., 1993**. Theory and methods of Euclidean combinatorial optimization. Kiev: ISSE, 188. (in Ukrainian).

19. **Pichugina O., Yakovlev S., 2016**. Continuous Approaches to the Unconstrained Binary Quadratic Problems. In: Bélair, J. et al. (eds.) Mathematical and Computational Approaches in Advancing Modern Science and Engineering, Springer International Publishing, Switzerland, 689-700.

20. **Pichugina O., Yakovlev S., 2016**. Convex Extensions and Continuous Functional Representations in Optimization, with Their Applications. J. Coupled Syst. Multiscale Dyn. 4(2), 129–152.

21. **Pichugina O., 2016.** Surface and Combinatorial Cuttings in Euclidean Combinatorial Optimization Problems. Mathematical and computer modeling. The Series: Physics and Mathematics. 1(13), 144–160. (in Russian).

22. **Pichugina O., 1996.** The methods and algorithms for a solution of some problems of optimization on arrangements and combinations (Ph.D. dissertation). Institute for Mechanical Engineering Problems, Kharkiv, Ukraine. (in Russian).

23. **Bertsekas D., 1999**. Nonlinear Programming, 2nd edn. Belmont: Athena Scientific, 708.

24. **Trick M., 2003.** A Dynamic Programming Approach for Consistency and Propagation for Knapsack Constraints. Annals of Operations Research, 118(1–4), 73–84.