Nonlinear Functional Analysis and Applications
Vol. 26, No. 3 (2021), pp. 611-628
ISSN: 1229-1595(print), 2466-0973(online)

https://doi.org/10.22771/nfaa.2021.26.03.11
http://nfaa.kyungnam.ac.kr/journal-nfaa

MONOTONIC OPTIMIZATION TECHNIQUES FOR SOLVING KNAPSACK PROBLEMS

Tran Van Thang[1], Jong Kyu Kim[2] and Won Hee Lim[3]

[1]Electric Power University, Hanoi, Vietnam
e-mail: thangtv@epu.edu.vn

[2]Department of Mathematics Education, Kyungnam University
Changwon, Gyeongnam, 51767, Korea
e-mail: jongkyuk@kyungnam.ac.kr

[3]Department of Mathematics Education, Kyungnam University
Changwon, Gyeongnam, 51767, Korea
e-mail: worry36@kyungnam.ac.kr

Abstract. In this paper, we propose a new branch-reduction-and-bound algorithm to solve the nonlinear knapsack problems by using general discrete monotonic optimization techniques. The specific properties of the problem are exploited to increase the efficiency of the algorithm. Computational experiments of the algorithm on problems with up to 30 variables and 5 different constraints are reported.

1. Introduction

We consider the knapsack problem which is the following nonlinear integer programming problem (shortly, $MNKP$):

$$\max \quad f(x) = \sum_{j=1}^{n} f_j(x_j) \tag{1.1}$$

[0]Received November 23, 2020. Revised March 15, 2021. Accepted March 20, 2021.
[0]2000 Mathematics Subject Classification: 90C90, 90C30, 49N15.
[0]Keywords: Lagrangian relaxation, discrete monotonic optimization, knapsack problem, quasi-conjugate duality.
[0]Corresponding author: Tran Van Thang(thangtv@epu.edu.vn).

$$\text{s.t.} \quad g_i(x) = \sum_{j=1}^{n} g_{ij}(x_j) \le w_i, \quad i = 1, 2, ..., m, \tag{1.2}$$

$$x \in X = \{x \in \mathbb{Z}^n : \ a_j \le x_j \le b_j, \quad j = 1, 2, ..., n\}, \tag{1.3}$$

where $\mathbb{Z}^n$ is the set of all integer points in $\mathbb{R}^n$, f_j are increasing functions on $[a_j, b_j]$ and g_{ij} are increasing lower semicontinuous functions on $[a_j, b_j]$ for any $i = 1, 2, ..., m,\ j = 1, 2, ..., n$, with $a_j \le b_j$ and $a_j, b_j \in \mathbb{Z}$ for any $j = 1, 2, ..., n$. The area of knapsack problems has an attracted attention from applications in economics, operations research. Recently, researchers have presented some methods to solve this type of problems.

In [4], Cooper used dynamic programming to solve problem in integers with a separable objective function. The monotonicity of the problem $(MNKP)$ here is not exploited. We can solve knapsack problem by continuous relaxation-based branch-and-bound [5] or hybrid DP/B&B methods [3, 6, 7, 12, 13, 14, 17, 19]. In the nonconcave case, however, computing upper bound of subproblem by this continuous relaxation method is usually not easy to have an exact number. Recently, the problem $(MNKP)$ has been solved by nonlinear Lagrangian methods [1, 2, 8, 9, 10, 15, 16]. In optimization theory, nonlinear Lagrangian methods are convergent, however, the computational is often difficult to implement.

In [11], Li, Sun, Wang and McKinnon proposed a convergent Lagrangian and domain cut method to solve knapsack problem $(MNKP)$. This method can be interpreted as an extension of the traditional branch-and-bound method. At each iteration, a box candidate for subdivision may be branched into $2n+1$ new subboxes and the dual search method is applied to the each newly generated subbox.

In this paper we apply the polyblock algorithm and the branch-reduction-and-bound algorithm for general discrete monotonic optimization ([21]) to solve $(MNKP)$. The specific properties of the problem are exploited to increase the efficiency of the algorithm. Specifically, the separability and the monotonicity of problem $(MNKP)$ help us to compute an upper bound of a subproblem by the Lagrangian relaxation method. Due to the problem $(MNKP)$ is an integer programming, the separation cut and reduction cut are realized simpler in the general case.

The paper is organized as follows. In sections 2 we review some necessary concepts, results from monotonic optimization as presented in [21] and prove some specific properties. Section 3, will present the polyblock approximation algorithm for $(MNKP)$. The branch-reduction-and-bound algorithm for $(MNKP)$ is presented in section 4. Finally in section 5, the computational experiments of branch-reduction-and-bound algorithm on problems with up to 30 variables and 5 constraints are reported.

2. Separation cut and reduction cut

In monotonic optimization ([21]), the separation cut and reduction cut are fundamental steps of polyblock algorithm and branch-reduce-and-bound algorithm. At a given iteration of the algorithm, it reduces a polyblock (box) to smaller polyblock (box) without losing any feasible solution currently still of interest.

A set $G \subset [a, b]$ is called normal if $x \in G$ whenever $a \leq x \leq x^{'} \in G$.

Proposition 2.1. (cf. [20]) *If f is an increasing function on $\mathbb{R}^n_+$ then the set $G = \{x \in \mathbb{R}^n_+ | \; f(x) \leq 1\}$ is normal and it is closed if f is lower semicontinuous.*

Definition 2.2. (cf. [21]) Let $A \subset [a, b]$. The normal hull of A, written $A^{\rceil}$, is defined by $A^{\rceil} = \cup_{z \in A}[a, z]$. If A is finite, then $A^{\rceil}$ is called a polyblock and the vector $z \in A$ is said to be vertex of the polyblock.

For the rest of this paper, denote by e^i the vector satisfying $e^i_i = 1$ and $e^i_j = 0$, for all $j \neq i$.

Lemma 2.3. (cf. [21]) *Let $x \in [a, b]$ satisfy $a < x < b$. Then, the set $[a, b] \setminus (x, b]$ is a polyblock with vertices $z^i = b - (b_i - x_i)e^i, \; i = 1, 2, ..., n$.*

Proposition 2.4. (cf. [21]) *The maximum of an increasing function $f(x)$ over a polyblock is achieved at a proper vertex of this polyblock.*

Consider the problem $(MNKP)$. Set $a = (a_1, a_2, ..., a_n), \; b = (b_1, b_2, ..., b_n)$. Since f_j and g_{ij} are increasing functions on $[a_j, b_j]$ for any $i = 1, 2, ..., m, \; j = 1, 2, ..., n$, without loss of generality, we may assume that $[a, b] \subset \mathbb{R}^n_+$.

For the rest of this paper, we define

$$\tilde{g}(x) = \max\{g_i(x) - w_i | \; j = 1, 2, ..., m\}.$$

Since g_i is a lower semicontinuous increasing function on $[a, b]$ for every $i = 1, 2, .., m$, the function $\tilde{g}$ is lower semicontinuous increasing on $[a, b]$. By Proposition 2.1, $G = \{x \in [a, b] | \; \tilde{g}(x) \leq 0\}$ is a closed and normal subset on $[a, b]$. Then, the problem $(MNKP)$ can be rewritten as following:

$$\max\{f(x) | \; x \in G \cap X\}. \tag{2.1}$$

This is discrete monotonic optimization problem which has been studied in [21].

A vector $\bar{x} \in G$ is an upper boundary of G if the cone $K_{\bar{x}} := \{x| \ \ x > \bar{x}\}$ constains no point $x \in G$. Denote by $\partial^+ G$ the set of all upper boundary points of G.

Proposition 2.5. (cf. [21]) *Let G be a closed normal set G in $[a, b]$, and $\bar{z} \in [a, b] \setminus G$. If $\bar{x} \in \partial^+ G$ satisfying $\bar{x} < \bar{z}$ then the cone $K_{\bar{x}} := \{x| \ \ x > \bar{x}\}$ constains $\bar{z}$ but is disjoint from G, that is, $K_{\bar{x}} \cap G = \emptyset$.*

We shall refer to the cone $K_{\bar{x}}$ as a separation cut with vertex $\bar{x}$ for G.

Consider a box $[p, q] \subset [a, b]$ satisfying $p, q \in \mathbb{Z}^n$. Given any point $x \in [p, q]$. As in [21], the lower X-adjustment of x is the point

$$\lfloor x \rfloor_X = \tilde{x} \text{ with } \tilde{x}_i = \max\{y_i : \ y \in X \cup \{p\}, \ y_i \leq x_i\}, \ i = 1, 2, ..., n \tag{2.2}$$

and the upper X-adjustment of x is the point

$$\lceil x \rceil_X = \hat{x} \text{ with } \hat{x}_i = \min\{y_i : \ y \in X \cup \{q\}, \ y_i \geq x_i\}, \ i = 1, 2, ..., n. \tag{2.3}$$

Let $[x_i]$ denote the integer number satisfying $[x_i] \leq x_i < [x_i] + 1$ for all $i = 1, 2, ..., n$. It is easy to prove following proposition.

Proposition 2.6. *Let $x \in [p, q] \subset [a, b]$. Then*

$$\tilde{x}_i = [x_i], \ \forall \ i = 1, 2, ..., n, \tag{2.4}$$

$$\hat{x}_i = \begin{cases} [x_i] & \textit{if } [x_i] = x_i, \\ [x_i] + 1 & \textit{otherwise}, \end{cases} \quad i = 1, 2, ..., n. \tag{2.5}$$

Proposition 2.7. *Let $[p, q] \subset [a, b]$ satisfy $p, \ q \in \mathbb{Z}^n$. Then $\lfloor x \rfloor_X \in [p, q] \cap X$ and $\lceil x \rceil_X \in [p, q] \cap X$ for any $x \in [p, q]$.*

Proof. Since $p_i \in \mathbb{Z}$ and $p_i \leq x_i$, we have $p_i \leq \tilde{x}_i$ by (2.4). This together with $\tilde{x}_i \leq x_i \leq q_i$ implies that $p_i \leq \tilde{x}_i \leq q_i$ for any $i = 1, 2, .., n$. So, $\lfloor x \rfloor_X \in [p, q] \cap X$. We now show that $\lceil x \rceil_X \in [p, q] \cap X$. Since $q_i \in \mathbb{Z}$ and $x_i \leq q_i$, we have $\hat{x}_i \leq p_i$ by (2.5). This together with $p_i \leq x_i \leq \hat{x}_i$ implies that $q_i \leq \hat{x}_i \leq p_i$. Therefore, $\lceil x \rceil_X \in [p, q] \cap X$. $\square$

Proposition 2.8. (cf. [21]) *If x is the vertex of a separation cut for G, then $\lfloor x \rfloor_X$ is the vertex of a separation cut for the $G \cap X$, that is,, the cone $K_{\lfloor x \rfloor_X}$ is disjoint from $G \cap X$.*

Let $[p, q]$ be any box in $[a, b]$ satisfying $p, q \in X$. We consider the following problem:

$$\max\{f(x)| \ x \in G \cap [p, q]\}. \tag{2.6}$$

If a feasible solution of (2.6) is known with objective function value γ, then we would like to recognize whether or not the box $[p, q]$ contains a feasible solution to (2.6) with objective function value at least equal to γ.

Proposition 2.9. (i) *Let* $f(q) \geq \gamma$ *and*

$$p^{'} = q - \sum_{i=1}^{n} \alpha_i (q_i - p_i) e^i, \tag{2.7}$$

$$\alpha_i = \sup\{\alpha |\ 0 \leq \alpha \leq 1,\ f(q - \alpha(q_i - p_i)e^i) \geq \gamma\},\ i = 1, ..., n. \tag{2.8}$$

Then $\lceil p^{'} \rceil_X \in X$ *and the box* $[\lceil p^{'} \rceil_X, q]$ *still contains all feasible solutions with objective function value at least equal to* γ *of the problem* (2.6).

(ii) *Let* $\tilde{g}(p) \leq 0$ *and* $q^{'} = p + \sum_{i=1}^{n} \beta_i (q_i - p_i) e^i$, *where*

$$\beta_i = \sup\{\beta |\ 0 \leq \beta \leq 1,\ \tilde{g}(p + \beta(q_i - p_i)e^i) \leq 0\},\ i = 1, ..., n. \tag{2.9}$$

Then $\lfloor q^{'} \rfloor_X \in X$ *and the box* $[p, \lfloor q^{'} \rfloor_X]$ *still contains all feasible solutions with objective function value at least equal to* γ *of the problem* (2.6).

Proof. It suffices to prove (i) because the proof of (ii) is similar. From

$$p^{'} = q - \sum_{i=1}^{n} \alpha_i (q_i - p_i) e^i,$$

we have $p_i^{'} = \alpha_i p_i + (1 - \alpha_i) q_i$ with $0 \leq \alpha_i \leq 1$, for all $i = 1, 2, ..., n$. It follows that

$$p_i \leq p_i^{'} \leq q_i,\ \forall i = 1, 2, ..., n.$$

Hence, $p \leq p^{'} \leq q$. By Proposition 2.7, we have $\lceil p^{'} \rceil_X \in [p, q] \cap X$.

Consider any $x \in [p, q] \cap X$ satisfying $f(x) \geq \gamma$ and $\tilde{g}(x) \leq 0$ ($\tilde{g}(x) = \max\{g_i(x) - w_i |\ j = 1, 2, ..., m\}$). We now show that $x \geq p^{'}$. Suppose that $x \ngeq p^{'}$. Then there exists $i \in \{1, 2, ..., n\}$ such that

$$\begin{aligned} x_i < p_i^{'} &= \alpha_i p_i + (1 - \alpha_i) q_i \\ &= q_i - \alpha_i (q_i - p_i). \end{aligned}$$

So, there exists $\alpha \in (\alpha_i, 1]$ such that $x_i = q_i - \alpha(q_i - p_i)$. It follows that

$$q - (q_i - x_i)e^i = q - \alpha(q_i - p_i)e^i.$$

By virtue of the definition of α_i and last equality, we have

$$f(q - (q_i - x_i)e^i) = f(q - \alpha(q_i - p_i)e^i) < \gamma.$$

This together with $x \leq q - (q_i - x_i)e^i$ implies that

$$f(x) \leq f(q - \alpha(q_i - p_i)e^i) < \gamma.$$

It contradicts with $f(x) \geq \gamma$. So, $x \geq p^{'}$. From the definition of $\lceil p^{'} \rceil_X$ and $x \in X$, we have $x \geq \lceil p^{'} \rceil_X$. $\square$

For box $[p,q]$ satisfying $\tilde{g}(p) \leq 0, f(q) \geq \gamma$. Clearly, the box $[\lceil p^{'}\rceil_X, q]$ defined in (i) is obtained from $[p,q]$ by cutting domain $\bigcup_{i=1}^n \{x|\ x_i < \hat{x}_i\}$ $(\lceil p^{'}\rceil_X = \hat{x})$ and the box $[p, \lfloor q^{'}\rfloor_X]$ defined in (ii) is obtained from $[p,q]$ by cutting $\bigcup_{i=1}^n \{x|\ x_i > \tilde{x}_i\}$ $(\lfloor p^{'}\rfloor_X = \tilde{x})$. The former cut is referred to as a lower cut for $[p,q]$ with vertex $\lceil p^{'}\rceil_X$ and the latter cut as an upper cut $[p,q]$ with vertex $\lfloor q^{'}\rfloor_X$. If $\tilde{g}(\lceil p^{'}\rceil_X) \leq 0$ and $\lfloor q^{'}\rfloor_X$ is the vertex of the upper cut for $[\lceil p^{'}\rceil_X, q]$, that is,

$$\lfloor q^{'}\rfloor_X = \lceil p^{'}\rceil_X + \sum_{i=1}^n \beta_i(q_i - \hat{p}_i)e^i,$$

where $\lceil p^{'}\rceil_X = \hat{p}$ and β_i is determined by (2.9). Then the box $[\lceil p^{'}\rceil_X, \lfloor q^{'}\rfloor_X]$ is called a γ-reduction of $[p,q]$, written $[\lceil p^{'}\rceil_X, \lfloor q^{'}\rfloor_X] = red_\gamma[p,q]$.

3. Branch-reduce-and-bound algorithm

In this section, we apply branch-reduce-and-bound algorithm to solve problem (MNPK). In each iteration, this algorithm is a procedure involving three basic operations: branching, reduction and bounding.

Assume at k-th iteration, we have the current best feasible solution x^k with the current best value $\gamma = f(x^k)$ a set of newly generated boxes that remain for exploration (note that the newly generated boxes have vertices in X). If $M = [p,q]$ is such a box, it is easily to verify that there exists a feasible solution $x \in [p,q]$ to (2.1) satisfying $f(x) \geq \gamma$ only if $\tilde{g}(p) \leq 0,\ f(q) \geq \gamma$. We now consider the following subproblem:

$$\max\{f(x)|\ x \in G \cap M \cap X\}. \tag{3.1}$$

I. Reduction. For every box $M = [p,q](p,q \in X)$, which is interested. This operation reduces $M = [p,q]$ to smaller box $red_\gamma[p,q]$ without losing any feasible solution currently still of interest. In general case ([21]), to compute $\lceil p^{'}\rceil_X$ and $\lfloor q^{'}\rfloor_X$, we must first calculate $p^{'}$ and $q^{'}$ by Proposition 2.9. In particular, we only can approximatively compute $p^{'}$ and $q^{'}$. In this paper, due to X is a subset in $\mathbb{Z}$, the $\lceil p^{'}\rceil_X$ and $\lfloor q^{'}\rfloor_X$ are realized without knowing $p^{'}$ and $q^{'}$ by following theorem.

Theorem 3.1. *Let $[p,q]$ be any box in $[a,b]$ satisfying $p,q \in X$, $\tilde{g}(p) \leq 0$ and $f(q) \geq \gamma$. Then,*

$$\lceil p^{'}\rceil_X = q - \sum_{i=1}^n \bar{\alpha}_i(q_i - p_i)e^i$$

where

$$\bar{\alpha}_i = \begin{cases} 0, & \textit{if} \quad q_i - p_i \neq 0 \\ \frac{1}{q_i - p_i} \max\{k_i |\ k_i \in \{0, 1, ..., q_i - p_i\},\ f(q - k_i e^i) \geq \gamma\}, & \textit{otherwise} \end{cases} \tag{3.2}$$

for $i = 1, ..., n$ *and*

$$\lfloor q' \rfloor_X = \lceil p' \rceil_X + \sum_{i=1}^{n} \bar{\beta}_i (q_i - \hat{p}_i) e^i,$$

where

$$\bar{\beta}_i = \begin{cases} 0, & \textit{if} \quad q_i - p_i \neq 0 \\ \frac{1}{q_i - \hat{p}_i} \max\{l_i |\ l_i \in \{0, 1, ..., q_i - \hat{p}_i\},\ \tilde{g}(\lceil p' \rceil_X + l_i e^i) \leq 0\}, & \textit{otherwise} \end{cases} \tag{3.3}$$

for $i = 1, ..., n$.

Proof. Since $p \leq \lceil p' \rceil_X \leq q$, there exists $\bar{\alpha}_i \in [0, 1],\ i = 1, 2, ..., n$ such that

$$\lceil p' \rceil_X = q - \sum_{i=1}^{n} \bar{\alpha}_i (q_i - p_i) e^i.$$

For $i \in \{1, 2, ..., n\}$, from last equality we have ($\hat{p} = \lceil p' \rceil_X$)

$$\hat{p}_i = q - \alpha_i (q_i - p_i) e^i. \tag{3.4}$$

If $q_i - p_i = 0$, it follows that $q_i = p_i$. By (2.5), we have $\hat{p}_i = q_i$. If $q_i - p_i \neq 0$, from $\hat{p}_i \in \mathbb{Z}$ and (3.4) it follows that $\bar{\alpha}_i (q_i - p_i) \in \mathbb{Z}$. We now show that

$$\bar{\alpha}_i = \max\{\alpha |\ \alpha \in [0, 1],\ \alpha(q_i - p_i) \in \mathbb{Z},\ f(q - \alpha(q_i - p_i) e^i) \geq \gamma\}. \tag{3.5}$$

Indeed, suppose that there exists a real number α_i^* in $[0, 1]$ satisfying

$$\alpha_i^* (q_i - p_i) \in \mathbb{Z},\ f(q - \alpha_i^* (q_i - p_i) e^i) \geq \gamma$$

such that $\alpha_i^* > \bar{\alpha}_i$. From (2.7), (2.8) and $f(q - \alpha_i^* (q_i - p_i) e^i) \geq \gamma$ we have $\alpha_i^* \leq \alpha_i$. It follows that $\bar{\alpha}_i < \alpha_i^* \leq \alpha_i$. Therefore,

$$\begin{aligned} p_i' &= q_i - \alpha_i (q_i - p_i) \\ &\leq q_i - \alpha_i^* (q_i - p_i) \\ &< q - \alpha_i (q_i - p_i) e^i \\ &= \hat{p}_i. \end{aligned} \tag{3.6}$$

From (3.6), $q_i - \alpha_i^* (q_i - p_i) \in \mathbb{Z}$ and $\hat{p} = \lceil p' \rceil_X$, it follows that $\hat{p}_i \leq q_i - \alpha_i^* (q_i - p_i)$. This conflicts with (3.6). Hence, we have (3.5). Let $\alpha \in [0, 1]$ satisfy

$$\alpha(q_i - p_i) \in \mathbb{Z},\ f(q - \alpha(q_i - p_i) e^i) \geq \gamma.$$

Setting $k_i := \alpha(q_i - p_i)$, we have $k_i \in \{0, 1, ..., q_i - p_i\}$. This together with (3.5) implies that

$$\bar{\alpha}_i = \frac{1}{q_i - p_i} \max\{k_i |\ k_i \in \{0, 1, ..., q_i - p_i\},\ f(q - k_i e^i) \geq \gamma\}.$$

Hence, $\lceil p' \rceil_X = q - \sum_{i=1}^n \bar{\alpha}_i (q_i - p_i) e^i$ with $\bar{\alpha}_i$ satisfying (3.2).

By the same above proof, we show that $\lfloor q' \rfloor_X = \lceil p' \rceil_X + \sum_{i=1}^n \bar{\beta}_i (q_i - p_i) e^i$ with $\bar{\beta}_i$ satisfying (3.3). □

II. Bounding. In this operation, we need compute a upper bound $\mu(M)$ such that

$$\mu(M) \geq \gamma(M) = \max\{f(x) |\ x \in G \cap M \cap X\}.$$

Due to the $f(x)$ is increasing in $[p, q]$ so $f(q)$ is a upper bound of problem (3.1). In general case, we can always take $\mu(M) = f(q)$ when a better bound is expensive to compute. In this paper, due to the separability of objective, subjective functions and $X \subset \mathbb{Z}^n$, we can obtained a upper bound better that $f(q)$ by using Lagrangian relaxation method or separation cut which is a operation in polyblock algorithm ([21]).

If $q \in G$ then q is a feasible solution and

$$\max\{f(x) |\ x \in G \cap M \cap X\} = f(q).$$

Hence, we take $\mu(M) = f(q)$. Suppose that $q \notin G$. We now compute a upper bound by separation cut. Denote

$$X_k = \{x \in X |\ f(x) > \gamma\},$$

$\overline{q}$ is the first point of G on the line segment joint q to p and $\tilde{q} = \lfloor \overline{q} \rfloor_{X_k}$ (using formula (2.3) for $X := X_k$). For $i = 1, 2, ..., n$, set

$$\begin{aligned}\Delta_i = \big\{\alpha_i | \alpha_i \in \{0, 1, ..., q_i - p_i\}, t_i = \frac{\alpha_i}{q_i - p_i}, f(q - (q_i - p_i - \alpha_i) e^i) > \gamma, \\ \tilde{g}(p + t_i (q - p)) \leq 0\big\}. \qquad (3.7)\end{aligned}$$

Proposition 3.2. *Let $q \notin G$. Then, the polyblock $[p, q] \setminus (\tilde{q}, q]$ still contains $G \cap M \cap X_k$, $\tilde{q} \in G \cap X$ and*

$$\tilde{q} = p + \sum_{i=1}^n \tilde{\alpha}_i e^i, \qquad (3.8)$$

where $\tilde{\alpha}_i = \max\big\{\alpha_i |\ \alpha_i \in \Delta_i\big\}$ for every $i = 1, 2, ..., n$, (agree that $\max \emptyset = 0$).

Proof. By Proposition 2.8, the polyblock $[p, q] \setminus (\lfloor \overline{q} \rfloor_{X_k}, q]$ still contains $G \cap M \cap X_k$. Set

$$x^* = p + \sum_{i=1}^{n} \tilde{\alpha}_i e^i. \tag{3.9}$$

Since $\tilde{\alpha}_i \in \{0, 1, ..., q_i - p_i\}$, we have $x_i^* \in \mathbb{Z}$ and $p_i \leq x_i^* \leq q_i \ \forall i = 1, 2, ..., n$. So, $x^* \in X$. Since $\tilde{q} = \lfloor \overline{q} \rfloor_{X_k}$, we have $\tilde{q} \leq \overline{q}$. It follows that $\tilde{g}(\tilde{q}) \leq \tilde{g}(\overline{q}) \leq 0$ (because $\overline{q} \in G$). So, $\tilde{q} \in G \cap X$.

We next prove (3.8). For $i \in \{1, 2, ..., n\}$. If $q_i = p_i$, by (3.7), we have $\tilde{\alpha}_i = 0$, and so $x_i^* = p_i$. From (2.3) and $q_i = p_i$, it follows that $\tilde{q}_i = p_i$. So $\tilde{q}_i = x_i^*$.

If $q_i > p_i$ and $\Delta_i = \emptyset$, then $\tilde{\alpha}_i = 0$, and so $x_i^* = p_i$ by (3.9). Using Proposition 2.7, we have $\tilde{q} \in [p, q] \cap X_k$, it follows that $\tilde{q}_i \geq p_i = x_i^*$.

Now, we suppose that $\tilde{q}_i > p_i = x_i^*$. Since $\tilde{q} = \lfloor \overline{q} \rfloor_{X_k}$ and (2.3), there exists $x \in [p, q] \cap X_k$ such that $x_i = \tilde{q}_i$. Setting $\alpha = \tilde{q}_i - p_i$, then

$$\begin{aligned} q - (q_i - p_i - \alpha)e^i &= q - (q_i - \tilde{q}_i)e^i \\ &\geq x. \end{aligned}$$

From the last inequality and $x \in X_k$, it follows that

$$f(q - (q_i - p_i - \alpha)e^i) \geq f(x) > \gamma. \tag{3.10}$$

Since $\overline{q}$ is the first point of G on the line segment joint q to p, one has $p \leq \overline{q} \leq q$, so there is number $\bar{t} \in [0, 1]$ such that $\overline{q} = p + \bar{t}(q - p)$. From the defination of $\lfloor \overline{q} \rfloor_{X_k}$, we have $\tilde{q}_i \leq \overline{q}_i$. This together with $\alpha = \tilde{q}_i - p_i$ implies that

$$p_i + t(q_i - p_i) \leq p_i + \bar{t}(q_i - p_i),$$

where $t = \frac{\alpha}{q_i - p_i}$. It follows that $t \leq \bar{t}$, and so

$$p + t(q - p) \leq p + \bar{t}(q - p) = \overline{q}.$$

By the increaseness of $\tilde{g}$ we have

$$\tilde{g}(p + t(q - p)) \leq \tilde{g}(\overline{q}) \leq 0.$$

This together with (3.10) implies that $\alpha \in \Delta_i$, which contradicts the fact that $\Delta_i = \emptyset$. Therefore, $\tilde{q}_i = x_i^*$.

If $q_i > p_i$ and $\Delta_i \neq \emptyset$. Setting $\tilde{t}_i = \frac{\tilde{\alpha}_i}{q_i - p_i}$, we have

$$f(q - (q_i - p_i - \tilde{\alpha}_i)e^i) > \gamma, \tag{3.11}$$

$$\tilde{g}(p + \tilde{t}_i(q - p)) \leq 0. \tag{3.12}$$

From (3.11), we have

$$q - (q_i - p_i - \tilde{\alpha}_i)e^i \in X_k. \tag{3.13}$$

From (3.12), we have

$$p + \tilde{t}_i(q - p) \in G. \tag{3.14}$$

Now, we show that $x_i^* \leq \overline{q}_i$. Assume the contrary that $x_i^* > \overline{q}_i$, then

$$p_i + \tilde{t}_i(q_i - p_i) > p_i + \bar{t}(q_i - p_i),$$

and so $\tilde{t}_i > \bar{t}$. It follows that $p + \tilde{t}_i(q - p) \geq \overline{q}$ and $p + \tilde{t}_i(q - p) \neq \overline{q}$. This together with (3.14) contradicts the fact that $\overline{q}$ is the first point of G on the line segment joint q to p. Hence, $x_i^* \leq \overline{q}_i$. This together with (3.9), $x_i^* \in \mathbb{Z}$ and $\tilde{q} = \lfloor \overline{q} \rfloor_{X_k}$ implies that

$$\tilde{q}_i \geq x_i^* = p_i + \tilde{\alpha}_i.$$

Suppose that $\tilde{q}_i > x_i^*$. Setting $\alpha^* = \tilde{q}_i - p_i$, from last inequality, we have $\alpha^* > \tilde{\alpha}_i$. It follows that

$$q - (q_i - p_i - \alpha^*)e^i \geq q - (q_i - p_i - \tilde{\alpha}_i)e^i,$$

and so,

$$f(q - (q_i - p_i - \alpha^*)e^i) > \gamma. \tag{3.15}$$

Setting $t^* = \frac{\alpha^*}{q_i - p_i}$, by the same above proof, we also show that

$$\tilde{g}(p + t^*(q - p)) \leq \tilde{g}(\overline{q}) \leq 0. \tag{3.16}$$

From (3.15) and (3.16), it follows $\alpha^* \in \Delta_i$. This together with $\alpha^* > \tilde{\alpha}_i$ conflicts with $\tilde{\alpha}_i = \max\left\{\alpha_i |\ \alpha_i \in \Delta_i\right\}$. Hence, $\tilde{q} = x^*$. $\square$

In general case ([21]), in order to compute $\tilde{q}$, the first we must compute approximately $\overline{q}$. In this paper, due to the problem $(MNKP)$ is an integer programming, we can exactly calculate $\tilde{q}$ without knowing $\overline{q}$ by above proposition.

Since G is a normal set and $\overline{q}$ is the first point of G on the line segment joint q to p, then $\overline{q}$ is an upper boundary of G. By Lemma 2.5, $\overline{q}$ is the vertex of a separation cut for the G. By Proposition 2.8, $\tilde{q}$ is the vertex of a separation cut for the $G \cap X$, and so, the polyblock $[p, q] \setminus (\tilde{q}, q]$ still contains all feasible solution x in $[p, q]$ satisfying $f(x) \geq \gamma$. Using Lemma 2.3, we have that the vertex set of polyblock $[p, q] \setminus (\tilde{q}, q]$ is

$$\left\{ q + (\tilde{q}_i - q_i)e^i |\ i \in \{1, 2, ..., n\} \right\}.$$

By Proposition 2.4, we have

$$\begin{aligned} \max\{f(x)|\ x \in G \cap M \cap X_k\} &\leq \max\{f(x)|\ x \in [p, q] \setminus (\tilde{q}, q]\} \qquad (3.17)\\ &\leq \max\{f(q + (\tilde{q}_i - q_i)e^i)|\ i \in \{1, 2, ..., n\}\}. \end{aligned}$$

So, we have that $\max\{f(q + (\tilde{q}_i - q_i)e^i)|\ i \in \{1, 2, ..., n\}\}$ is a upper bound of (3.1). Since $q + (\tilde{q}_i - q_i)e^i \in [p, q],\ \forall i \in \{1, 2, ..., n\}$, we have

$$\max\{f(q + (\tilde{q}_i - q_i)e^i)|\ i \in \{1, 2, ..., n\}\} \leq f(q). \tag{3.18}$$

Next, we show that the upper bound of (3.1) can be provided by Lagrangian relaxation method. The problem (3.1) can be rewritten as following:

$$\max\{f(x)|\ g(x) \le w,\ x \in M \cap X\}. \tag{3.19}$$

where $g(x) = (g_1(x), g_2(x), ..., g_m(x)),\ w = (w_1, w_2, ..., w_m)$. The Lagrangian relaxation of (3.19) is defined as follows

$$d(\lambda) = \max\{L(x, \lambda)|\ x \in M \cap X\}, \tag{3.20}$$

where $L(x, \lambda) = f(x) - \lambda^T(g(x) - w),\ \forall \lambda \in \mathbb{R}^m_+$.

It is well known that the following weak duality always holds

$$d(\lambda) \ge f(x)\ \forall x \in G \cap M \cap X, \quad \forall \lambda \in \mathbb{R}^m_+.$$

It follows that $d(\lambda) \ge \max\{f(x)|\ x \in G \cap M \cap X\}\ \forall \lambda \in \mathbb{R}^m_+$ and so $d(\lambda)$ is a upper bound of $f(x)$ over $G \cap M \cap X$ for each $\lambda \in \mathbb{R}^m_+$.

The duality problem of (3.19) is defined by

$$\min\{d(\lambda)|\ \lambda \ge 0\}. \tag{3.21}$$

Let $d^* = \min\{d(\lambda)|\ \lambda \ge 0\}$. Then d^* is a upper bound of $f(x)$ over $G \cap M \cap X$. By this way and (3.17), we can take

$$\mu(M) = \min\left\{d^*, \max\left\{f(q + (\tilde{q}_i - q_i)e^i)|\ i \in \{1, 2, ..., n\}\right\}\right\}.$$

From (3.18) and last equality, it follows that $\mu(M) \le f(q)$.

The problem (3.21) is a convex optimization problem. So, we can use subgradient method to solve this problem ([18]). To minimize $d(\lambda)$ over $\mathbb{R}^m_+$, the subgradient method uses the iteration $\lambda^{s+1} = \lambda^s - t^s\eta^s$ where η^s is any subgradient of d at λ^s, $t^s > 0$ is the s-th step size satisfying $t^s \to 0$ and $\sum_{s=1}^{\infty} t^s \to +\infty$.

At each s-th iteration, we need compute the $d(\lambda^s)$ by solving the Lagrangian relaxation problem:

$$d(\lambda^s) = \max\{L(x, \lambda^s)|\ x \in M \cap X\}. \tag{3.22}$$

Suppose that x^{λ^s} is an optimal solution of (3.22) corresponds to λ^s. It easy to check that vector $w - g(x^{\lambda^s})$ is a subgradient of d at λ^s. So, the subgradient method updates the multipliers by

$$\lambda_i^{s+1} = \max\{0, \lambda_i^s - t^s\eta_i/||\eta||\},\ i = 1, 2, ., m.$$

We have

$$L(x, \lambda^s) = \sum_{j=1}^{n}\left(f_j(x_j) - \sum_{i=1}^{m}\lambda_i^s g_{ij}(x_j)\right) + (\lambda^s)^T w.$$

Therefore, in order to solve the (3.22), we solve n following one-dimensional problems

$$\max\left\{f_j(x_j) - \sum_{i=1}^{m} \lambda_i^s g_{ij}(x_j) |\ x_j \in [p_j, q_j] \cap \mathbb{Z}\right\}, \quad j = 1, 2, ..., n. \qquad (3.23)$$

It is a d.m problem in the form studied in [20] (the objective is difference of two increasing functions). Using the method proposed in the latter paper the problem (3.23) is then converted to nonlinear integer monotonic optimization problem in two-dimensional space. So, the our method take advantages fully the monotonicity of functions $f_j(x_j),\ j = 1, 2, ..., n$.

Note that: In this operation, we always obtain the feasible solution $\tilde{q}$. Moreover, we can obtain some feasible solutions in the set $\{x^{\lambda^s} |\ s = 1, 2, ...\}$. This is very useful for updating the current best feasible solution and current best value in branch-reduce-and-bound algorithm.

III. Branching. This operation is performed according to a standard bisection rule. At k-th iteration, let $[p, q] \subset [a, b]$ $(p, q \in \mathbb{Z})$ be a box candidate for subdivision. Compute the numbers

$$\delta([a, b]) = \max\{q_i - p_i |\ i = 1, 2, ..., n\} = q_{i_M} - p_{i_M}, r_{i_M} = \frac{1}{2}(q_{i_M} + p_{i_M})$$

and divide $[a, b]$ into two boxes

$$M_+ = \{x \in M |\ x_{i_M} \geq [r_{i_M}] + 1\},$$

$$M_- = \{x \in M |\ x_{i_M} \leq [r_{i_M}]\},$$

where $r_{i_M} \in \mathbb{Z}$ such that $r_{i_M} \leq r < r_{i_M} + 1$. Since the set

$$\{x \in M |\ [r_{i_M}] < x_{i_M} < [r_{i_M}] + 1\}$$

don't contains any elements of X, we have

$$M \cap X = M_+ \cup M_- \cap X.$$

It is easily to verify that the vertices of $M_+(M_-)$ are contained in X.

Algorithm:

Initialization. Let $P_1 = \{M_1\}$, $M_1 = [a, b]$, $R_1 = \emptyset$. Let a be the current best feasible solution, CBV $= f(a)$. Set $k := 1$.

Step 1. Let $P_k^{'} = \{red_\gamma[p, q] |\ [p, q] \in P_k\}$ for $\gamma =$ CBV. In particular delete every box $[p, q]$ such that $g(p) > 0$.

Step 2. For each box $M = red_\gamma[p, q] = [\lceil p^{'} \rceil_X, \lfloor q^{'} \rfloor_X] \in P_k^{'}$, compute a bound $\mu(M)$ satisfying $\mu(M) \leq f(\lfloor q^{'} \rfloor_X)$ and determine some feasible solutions in M.

Step 3. Let $S_k = P_k^{'} \cup R_k$. Update CBV, using the new feasible solutions encountered in Step 2, if any. Delete every $M \in S_k$ such that $\mu(M) \leq$ CBV and let R_{k+1} be the collection of remaining boxes.

Step 4. If $R_{k+1} = \emptyset$, then terminate: CBV is the optimal value and the feasible solution $\bar{x}$ with $f(\bar{x}) =$ CBV is an optimal solution.

Step 5. If $R_{k+1} \neq \emptyset$, let $M_k \in \text{argumax}\{\mu(M) | \ M \in R_{k+1}\}$. Divide M_k into two boxes according to the above described rule. Let P_{k+1} be the collection of these two subboxes of M_k.

Step 6. Increment k and return to Step 1.

Theorem 3.3. (see Theorem 17, [21]) *The algorithm terminates after finitely many iterations, yielding an optimal solution.*

4. Computational results

As an illustration, we present in this final some numerical examples of branch-reduce-and-bound algorithm for some problems. The algorithm was coded in Matlab 7.0 and the program was run on a PC Intel(R) (2.26GHz with 1.96GB of DDR RAM).

Example 4.1. Consider the problem:

$$\begin{aligned} \max \quad & f(x) = \frac{1}{2}x_1^2 + 5x_1 + 6x_2 \\ \text{s.t} \quad & g(x) = 6x_1 + x_2^2 \leq 23, \\ & x \in X = [1,5]^2 \cap \mathbb{Z}^2. \end{aligned}$$

For initialization we take $P_1 = \{M_1\}$, $M_1 = [a,\ b]$, $a = (1,\ 1)$, $b = (5,\ 5)$, $R_1 = \emptyset$ and let $x_{best} = a$ be the current best feasible solution with current best value CBV $= f(a) = \frac{23}{2}$. Set $k = 1$.

Iteration 1:

Step 1. $red_\gamma M_1 = [(1,\ 1), (3,\ 4)]$, $P_1^{'} = \{[(1,\ 1), (3,\ 4)]\}$.

Step 2. Determine feasible solution $\lfloor \pi_G(3,\ 4) \rfloor_{X_1} = (2,\ 3)$ and calculate the upper bound $\mu(red_\gamma M_1) = \max\{f(2,\ 4), f(3,\ 3)\} = 37.5$.

Step 3. $S_1 = P_1^{'} \cup R_1 = \{[(1,\ 1), (3,\ 4)]\}$, $x_{best} = (2,\ 3)$, CBV $= 30$, $R_2 = \{[(1,\ 1), (3,\ 4)]\}$. Since $R_2 \neq \emptyset$, go to Step 4.

Step 4. $M_2 = [(1,\ 1), (3,\ 4)]$, divide M_2 into two boxes: $M_2 = M_{21} \cup M_{22}$, $M_{21} = [(1,\ 1), (2,\ 4)]$, $M_{22} = [(3,\ 1), (3,\ 4)]$, $P_2 = \{M_{21}, M_{22}\}$, $R_2 = R_2 \setminus \{M_2\} = \emptyset$.

Step 5. $k = 2$.

Iteration 2:

Step 1. $red_\gamma M_{21} = [(2,\ 3), (2,\ 3)]$, $red_\gamma M_{22} = [(3,\ 2), (3,\ 2)]$,
$P_2' = \{[(2,\ 3), (2,\ 3)], [(3,\ 2), (3,\ 2)]\}$.

Step 2. Both $(2,\ 3)$ and $(3,\ 2)$ are feasible solution, $\mu(red_\gamma M_{21}) = 30$, $\mu(red_\gamma M_{21}) = 31.5$.

Step 3. $S_2 = \{[(2,\ 3), (2,\ 3)], [(3,\ 2), (3,\ 2)]\}$, $x_{best} = (3,\ 2)$, CBV $= 31.5$, $R_3 = \emptyset$.

Step 4. We have $R_3 = \emptyset$. So, $x_{opti} = (3,\ 2)^t$ is the optimal solution with optimal value $f(x_{opti}) = 31.5$.

Example 4.2.

$$\begin{aligned} \max \quad & f(x) = \sum_{j=1}^{20} (c_j x_j - d_j x_j^2) \\ \text{s.t} \quad & g(x) = Ax \le w, \\ & x \in X = [1,5]^{20} \cap \mathbb{Z}^{20}, \end{aligned}$$

where
$c^t = (300, 111, 123, 300, 121, 298, 143, 300, 134, 299, 178, 176, 157, 298, 254, 134, 176, 300, 300, 300)$,
$d^t = (1, 10, 9, 1, 8, 1, 7, 1, 10, 1, 9, 7, 10, 1, 9, 8, 9, 1, 1, 1)$,
$w = 0.7A \times h, h^t = (5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5)$,

$$A = \begin{pmatrix} 4 & 45 & 47 & 8 & 41 & 34 & 8 & 9 & 43 & 25 & 27 & 23 & 4 & 25 & 23 & 6 & 8 & 44 & 1 & 24 \\ 31 & 5 & 34 & 33 & 26 & 8 & 5 & 9 & 33 & 25 & 7 & 35 & 32 & 5 & 28 & 33 & 31 & 21 & 41 & 29 \\ 21 & 35 & 4 & 23 & 6 & 28 & 35 & 29 & 43 & 35 & 37 & 45 & 22 & 35 & 38 & 3 & 21 & 41 & 3 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}_{5\times 20}.$$

Since $c_j/2d_j \ge 5\ \forall j = 1, 2, ..., n$, the function $f_j = c_j x_j - d_j x_j^2$ is increasing on $[1, 5]$ for every $j = 1, 2, ..., n$. Therefore, f is increasing on $[1, 5]^{20}$. The algorithm terminates after 230 iterations with the computational time is 13.2 seconds, yielding 16714 is the optimal value and the optimal solution is

$$x_{opti} = (5, 2, 4, 5, 2, 5, 3, 5, 1, 5, 3, 1, 2, 5, 5, 1, 1, 5, 5, 5)^t.$$

Example 4.3.

$$\begin{aligned} \max \quad & f(x) = \sum_{j=1}^{20} (c_j x_j + d_j x_j^2) \\ \text{s.t} \quad & g(x) = Ax \le w, \\ & x \in X = [1,5]^{20} \cap \mathbb{Z}^{20}, \end{aligned}$$

where $c^t = (30, 11, 23, 30, 21, 28, 43, 3, 14, 29, 17, 16, 17, 29, 50, 34, 16, 4, 30, 3)$,
$d^t = (1, 10, 9, 1, 8, 1, 7, 1, 10, 1, 9, 7, 10, 1, 9, 8, 9, 1, 1, 1)$,
$w = 0.7A \times h, h^t = (5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5)$,

$$A = \begin{pmatrix} 4 & 45 & 47 & 8 & 41 & 34 & 8 & 9 & 43 & 25 & 27 & 23 & 4 & 25 & 23 & 6 & 8 & 44 & 1 & 24 \\ 31 & 5 & 34 & 33 & 26 & 8 & 5 & 9 & 33 & 25 & 7 & 35 & 32 & 5 & 28 & 33 & 31 & 21 & 41 & 29 \\ 21 & 35 & 4 & 23 & 6 & 28 & 35 & 29 & 43 & 35 & 37 & 45 & 22 & 35 & 38 & 3 & 21 & 41 & 3 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}_{5\times 20}.$$

The function $f_j = c_j x_j + d_j x_j^2$ is increasing on $[1, 5]$ for every $j = 1, 2, ..., n$. Therefore, f is increasing on $[1, 5]^{20}$. The algorithm terminates after 389 iterations with the computational time is 15.9 seconds, yielding 3990 is the optimal value and the optimal solution is

$$x_{opti} = (2, 5, 5, 1, 5, 1, 5, 1, 5, 1, 5, 1, 5, 5, 5, 5, 5, 1, 5, 1)^t.$$

Example 4.4.

$$\begin{aligned} \max \quad & f(x) = \sum_{j=1}^{20} (c_j x_j + d_j (x_j - e_j)^3) \\ & g(x) = Ax \leq w, \\ & x \in X = [1, 5]^{20} \cap \mathbb{Z}^{20}, \end{aligned}$$

where $c^t = (33, 1, 42, 3, 23, 4, 13, 18, 3, 2, 32, 21, 1, 1, 24, 14, 25, 32, 3, 25)$,
$d^t = (1, 10, 3, 1, 8, 1, 7, 1, 10, 1, 3, 7, 10, 1, 9, 2, 9, 1, 1, 1)$,
$e^t = (23, 4, 5, 2, 1, 3, 4, 5, 3, 2, 4, 2, 5, 3, 2, 1, 2, 3, 5)$,
$w = 0.7A \times h, h^t = (5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5)$,

$$A = \begin{pmatrix} 4 & 5 & 4 & 8 & 1 & 4 & 8 & 9 & 3 & 2 & 2 & 3 & 4 & 5 & 3 & 6 & 8 & 4 & 1 & 4 \\ 31 & 5 & 34 & 33 & 26 & 8 & 5 & 9 & 33 & 25 & 7 & 35 & 32 & 5 & 28 & 33 & 31 & 21 & 41 & 29 \\ 21 & 35 & 4 & 23 & 6 & 28 & 35 & 29 & 43 & 35 & 37 & 45 & 22 & 35 & 38 & 3 & 21 & 41 & 3 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}_{5\times 20}.$$

The function f is increasing on $[1, 5]^{20}$ and f is not convex and concave on $[1, 5]^{20}$. The algorithm terminates after 330 iterations the computational time is 9.6 seconds, yielding 2590 is the optimal value and the optimal solution is

$$x_{opti} = (5, 2, 5, 1, 5, 1, 5, 3, 4, 1, 5, 3, 5, 2, 5, 2, 5, 5, 1, 4)^t.$$

Example 4.5.

$$\begin{aligned} \max \quad & f(x) = \sum_{j=1}^{30} -\frac{d_j}{x_j} \\ \text{s.t} \quad & g(x) = Ax^1 + Bx^2 \leq w, \\ & x \in X = [1,5]^{30} \cap \mathbb{Z}^{30}, \end{aligned}$$

where $x = \begin{pmatrix} x^1 \\ x^2 \end{pmatrix}$, $x^1,\ x^2 \in \mathbb{R}^{15}$,
$d^t = (19, 2, 1, 11, 18, 16, 17, 11, 20, 1, 2, 1, 2, 3, 4, 1, 1, 18, 19, 1, 3, 14, 1, 19, 18, 19, 14, 1, 18, 14)$,
$w = 0.7A \times h, h^t = (5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5)$,

$$A = \begin{pmatrix} 4 & 5 & 7 & 8 & 1 & 3 & 8 & 9 & 4 & 5 & 7 & 2 & 4 & 2 & 2 \\ 31 & 5 & 34 & 33 & 26 & 8 & 5 & 9 & 33 & 25 & 7 & 35 & 32 & 5 & 2 \\ 21 & 5 & 4 & 23 & 6 & 28 & 35 & 29 & 43 & 35 & 3 & 45 & 22 & 35 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 30 & 29 & 28 & 27 & 26 & 25 & 24 & 23 & 22 & 21 & 20 & 19 & 18 & 17 & 16 \end{pmatrix}_{5\times 15},$$

$$B = \begin{pmatrix} 6 & 8 & 4 & 1 & 4 & 5 & 1 & 4 & 1 & 3 & 12 & 7 & 3 & 7 & 9 \\ 33 & 31 & 21 & 41 & 29 & 4 & 45 & 47 & 8 & 41 & 34 & 8 & 9 & 3 & 5 \\ 3 & 21 & 41 & 3 & 9 & 3 & 26 & 8 & 5 & 9 & 33 & 25 & 7 & 35 & 32 \\ 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 \\ 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}_{5\times 15}.$$

The function $f_j = -\frac{d_j}{x_j}$ is increasing on $[1, 5]$ for every $j = 1, 2, ..., n$. Therefore, f is increasing on $[1, 5]^{20}$. The algorithm terminates after 2553 iterations with the computational time is 112.2 seconds, yielding -69.0833 is the optimal value and the optimal solution is

$$x_{opti} = (5, 3, 2, 4, 5, 5, 5, 4, 5, 1, 3, 1, 2, 2, 4, 2, 1, 4, 5, 2, 4, 4, 2, 5, 5, 5, 4, 2, 4, 4)^t.$$

Next, we present the computational result for problem which is problem $(MNKP)$ with the nonlinear constraints and the objective is not concave.

Example 4.6.

$$\begin{aligned} \max \quad & f(x) = \sum_{j=1}^{20} (c_j x_j + d_j (x_j - e_j)^3) \\ \text{s.t} \quad & g_1(x) = \sum_{j=1}^{20} (r_j x_j + u_j x_j^2) \leq 20000, \\ & g_2(x) = Ax \leq w, \\ & x \in X = [1,5]^{20} \cap \mathbb{Z}^{20}, \end{aligned}$$

where $c^t = (33, 13, 42, 30, 23, 24, 13, 18, 31, 22, 32, 21, 15, 15, 24, 14, 25, 32, 13, 25)$,
$d^t = (1, 10, 3, 1, 8, 1, 7, 1, 10, 1, 3, 7, 10, 1, 9, 2, 9, 1, 1, 1)$,
$e^t = (2, 3, 4, 5, 2, 1, 3, 4, 5, 3, 2, 4, 2, 5, 3, 2, 1, 2, 3, 5)$,
$r^t = (300, 111, 123, 300, 121, 298, 143, 300, 134, 299, 178, 176, 157, 298, 254, 134, 176, 300, 300, 300)$,
$u^t = (111, 10, 294, 112, 8, 199, 7, 134, 10, 1, 9, 7, 300, 31, 9, 8, 279, 164, 215, 121)$,
$w = 0.7A \times h, h^t = (5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5)$,

$$A=\begin{pmatrix} 31 & 5 & 34 & 33 & 26 & 8 & 5 & 9 & 33 & 25 & 7 & 35 & 32 & 5 & 28 & 33 & 31 & 21 & 41 & 29 \\ 21 & 35 & 4 & 23 & 6 & 28 & 35 & 29 & 43 & 35 & 37 & 45 & 22 & 35 & 38 & 3 & 21 & 41 & 3 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 20 & 19 & 18 & 17 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}.$$

The function f and g_1 are increasing on $[1, 5]^{20}$ and f is not concave on $[1, 5]^{20}$. The algorithm terminates after 907 iterations the computational time is 88.2 seconds, yielding 1979 is the optimal value and the optimal solution is

$$x_{opti} = (1, 5, 1, 1, 5, 1, 5, 1, 5, 1, 5, 3, 1, 2, 5, 3, 5, 1, 1, 1)^t.$$

Acknowledgments: This work was supported by the Basic Science Research Program through the National Research Foundation(NRF) Grant funded by Ministry of Education of the republic of Korea (2018R1D1A1B07045427).

References

[1] P.N. Anh and L.D. Muu, *Lagrangian duality algorithms for finding a global optimal solution to mathematical programs with affine equilibrium constraints,* Nonlinear. Dynam. Syst. Theory, **6** (2006), 225-244.
[2] P.N. Anh and L.D. Muu, *Contraction mapping fixed point algorithms for multivalued mixed variational inequalities on network, in Optimization with Multivalued Mappings,* Eds: S. Dempe and K. Vyacheslav, Springer, 2006.
[3] P.N. Anh and T.V. Thang, *Optimality condition and quasi-conjugate duality with zero gap in nonconvex optimization,* Optim. Letters, **14** (2020), 2021-2037.
[4] M.W. Cooper, *The use of dynamic programming for the solution of a class of nonlinear programming problems,* Nav. Res. Logist. Q., **27** (1980), 89-95.
[5] O.K. Gupta and A. Ravindran, *Branch and bound experiments in convex nonlinear integer programming,* Manag. Sci. **31** (1985), 1533-1546.
[6] J.K.Kim, T.M. Tuyen and M.T. Ngoc Ha, *Two projection methods for solving the split common fixed point problem with multiple output sets in Hilbert spaces,* Numerical Funct. Anal. Optim., **42** (8)(2021), 973-988, https://doi.org/10.1080/01630563.2021.1933528
[7] F. Korner, *A hybrid method for solving nonlinear knapsack problems,* Eur. J. Oper. Res. **38** (1989), 238-241.
[8] D. Li and X.L. Sun, *Success guarantee of dual search in nonlinear integer programming: P-th power Lagrangian method,* J. Glob. Optim., **18** (2000), 235-254.
[9] D. Li and X.L. Sun, *Nonlinear Integer Programming,* Springer, New York, 2006.
[10] D. Li and D.J. White, *P-th power Lagrangian method for integer programming,* Ann. Oper. Res., **98** (2000), 151-170.

[11] D. Li, X.L. Sun, J. Wang and K.I.M. McKinnon, *Convergent Lagrangian and domain cut method for nonlinear knapsack problems,* Comput. Optim. Appl., **42** (2009), 67-104.

[12] R.E. Marsten and T.L. Morin, *A hybrid approach to discrete mathematical programming,* Math. Program., **14** (1978), 21-40.

[13] P.T. Thach and T.V. Thang, *Problems with rerource allocation constraints and optimization over the efficient set,* J. Glob. Optim., **58** (2014), 481-495.

[14] T.V. Thang, *Conjugate duality and optimization over weakly efficient set,* Acta Math. Vietnamica, **42**(2) (2017), 337-355.

[15] J.P. Penot and M. Volle, *Quasi-conjugate duality,* Math. Oper. Res., **15**(4) (1990), 597-625.

[16] T.D. Quoc, P.N. Anh and L.D. Muu, *Dual extragradient algorithms to equilibrium problems,* J. Glob. Optim., **52** (2012), 139-159.

[17] T. Rockafellar, *Conjugate duality and optimization,* Siam for Industrial and Applied Mathematics Philadelphia, University of Washington, Seattle, 1974.

[18] N.Z. Shor, *Minimization Methods for Non-differentiable Functions,* Springer Series in Computational Mathematics., Springer, 1985.

[19] T. V. Thang and N. D. Truong, *Conjugate duality for concave maximization problems and applications,* Nonlinear Funct. Anal. Appl., **25**(1) (2020), 161-174.

[20] H. Tuy, *Monotonic optimization: Problems and solution approaches,* SIAM J. Optim., **11** (2000), 464-494.

[21] H. Tuy, M. Minoux and N.T. Hoai-Phuong, *Discrete Monotonic Optimization With Application to A Discrete Location Problem,* SIAM J. Optim., **17** (2006), 78-97.