APPROXIMATE METHODS FOR CONVEX MINIMIZATION PROBLEMS WITH SERIES–PARALLEL STRUCTURE

ADI BEN-ISRAEL, GENRIKH LEVIN, YURI LEVIN, AND BORIS ROZIN

Abstract. Consider a problem of minimizing a separable, strictly convex, monotone and differentiable function on a convex polyhedron generated by a system of m linear inequalities. The problem has a series-parallel structure, with the variables divided serially into n disjoint subsets, whose elements are considered in parallel. This special structure is exploited in two algorithms proposed here for the approximate solution of the problem. The first algorithm solves at most $\min\{m, \nu - n + 1\}$ subproblems; each subproblem has exactly one equality constraint and at most n variables. The second algorithm solves a dynamically generated sequence of subproblems; each subproblem has at most $\nu - n + 1$ equality constraints, where ν is the total number of variables. To solve these subproblems both algorithms use the authors' Projected Newton Bracketing method for linearly constrained convex minimization, in conjunction with the steepest descent method. We report the results of numerical experiments for both algorithms.

1. Introduction

Consider an n–partite graph

$$G = \{\mathbf{V}, \mathbf{E}\} = G_1 \times G_2 \times \cdots \times G_n, \tag{1}$$

where the component G_i has a set $\mathbf{V}_i$ of ν_i **vertices**, and

$$\mathbf{V} = \mathbf{V}_1 \cup \mathbf{V}_2 \cup \cdots \cup \mathbf{V}_n, \tag{2}$$

the **edges** are all possible pairs

$$\mathbf{E} = \{(v_i, v_{i+1}) : v_i \in \mathbf{V}_i,\ v_{i+1} \in \mathbf{V}_{i+1},\ i = 1, \ldots, n-1\} \tag{3}$$

connecting successive components, and the **paths** of the graph are the n–tuples

$$\mathbf{W} = \{(v_1, v_2, \ldots, v_n) : v_i \in \mathbf{V}_i,\ i = 1, \ldots, n\}, \tag{4}$$

Date: March 27, 2006.
Key words and phrases. Convex Programming, Decomposition, Large-Scale Optimization.

i.e. each path contains precisely one vertex from each component. With each vertex $v \in \mathbf{V}$ is associated a real variable x_v and a function $f_v : \mathbb{R} \to \mathbb{R}$. The variables $\{x_v : v \in \mathbf{V}\}$ are collected in a vector $\mathbf{x} \in \mathbb{R}^\nu$, where

$$\nu = \sum_{i=1}^{n} \nu_i \tag{5}$$

is the number of vertices of G, and an objective function $F : \mathbf{R}^\nu \to \mathbb{R}$ is defined by

$$F(\mathbf{x}) = \sum_{v \in \mathbf{V}} f_v(x_v). \tag{6}$$

It is required to minimize $F(\mathbf{x})$ subject to constraints over all the paths from some subset $\mathbf{I} \subseteq \mathbf{W}$

$$\sum_{v \in W} x_v \geq b_W, \ W \in \mathbf{I}, \tag{7}$$

where the summation is over all vertices in the path W, and the constants b_W are given.

For example, consider a design problem for a series–parallel system under **reliability** constraints. Let each vertex v have a **survival probability** q_v and fail independently of other vertices. Then the survival probability of any path is the product of the probabilities q_v of its vertices. Suppose that path survival probabilities are required to exceed certain levels, say

$$\prod_{v \in W} q_v \geq \beta_W, \ W \in \mathbf{I},$$

which reduce to the linear constraints (7) by changing variables to

$$x_v = \log q_v, \ \forall \ v \in \mathbf{V}. \tag{8}$$

The methods developed in this paper are applicable to any such design problem which has a cost function of the form (6).

Optimization problems with a series–parallel structure as above appear in gearbox design where the vertices are gears, the components are axles carrying several gears, and a path is a selection of gears, one on each pair of adjoining axles, so as to control the output velocity.

Similar problems arise in transportation networks, where paths correspond to alternate routes.

In this paper, we study optimization problems for systems with series–parallel structure, see Section 2, and propose two algorithms for their approximate solution in Section 5. These algorithms use the authors' Projected Newton Bracketing (PNB) method for linearly constrained convex minimization [2] that we review in Section 3. Section 4 is devoted to the simplest special case of the problem. In Section 6 the results of numerical experiments are described. The validity and convergence of the proposed methods are discussed in Appendix.

2. Notation and preliminaries

For convenience we denote each vertex v by a double index ij, where i is the index of the component, and j is the index of the vertex in the component i, $i = 1, \ldots, n$, $j = 1, \ldots, \nu_i$.

The **system** in question can be described as follows:

- it consists of n **components** connected sequentially (i.e., in series),
- each component consists of several **vertices** connected in parallel,
- the i^{th} component has ν_i such vertices indexed by the set

$$G_i = \{i1, i2, \cdots, i\nu_i\},$$

- the $(ij)^{\text{th}}$ vertex is represented by a continuous variable x_{ij}, required to lie in an interval

$$X_{ij} = [\underline{x}_{ij}, \overline{x}_{ij}] \subseteq \mathbb{R},\ \underline{x}_{ij} < \overline{x}_{ij},$$

- the variables are grouped in a vector $\mathbf{x} \in \mathbb{R}^\nu$, partitioned in n subvectors

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{i\nu_i}) \in \mathbb{R}^{\nu_i},$$

  where

$$\nu = \sum_{i=1}^{n} \nu_i,$$

  and the vector $\mathbf{x}$ is restricted to the interval

$$X_0 = \bigotimes_{i=1}^{n} \bigotimes_{j=1}^{\nu_i} X_{ij} \subseteq \mathbb{R}^\nu,$$

- the index set of the vector $\mathbf{x}$ is

$$\mathbf{J} = \bigcup_{i=1}^{n} G_i = \{11, \ldots, 1\nu_1, 21, \ldots, 2\nu_2, \ldots, n1, \ldots, n\nu_n\},$$

- a **path** (through the system) is a selection of one vertex from each component, the k^{th} path is denoted $W_k = \{1j_1, 2j_2, \cdots, nj_n\}$,
- there are $p = \prod_{i=1}^{n} \nu_i$ paths, and the set of all paths is denoted $\mathbf{W}$, and
- the cost function $F : X_0 \to \mathbb{R}$ is a separable function of the variables x_{ij}, say

$$F(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{\nu_i} f_{ij}(x_{ij}),$$

  where each f_{ij} is a strictly convex increasing differentiable function.

Let $\mathbf{I}$ be a given subset of path indices of $\mathbf{W}$. For each $k \in \mathbf{I}$ we associate with the path W_k the (linear) function

$$g_k(\mathbf{x}) = \sum_{(ij)\in W_k} x_{ij},$$

and the constraint

$$g_k(\mathbf{x}) = \sum_{(ij)\in W_k} x_{ij} \geq b_k, \tag{9}$$

where b_k are given constants. For any $I \subseteq \mathbf{I}$ we define the sets

$$X(I) = \{\mathbf{x} \in \mathbb{R}^\nu : \ g_k(\mathbf{x}) \geq b_k, \ k \in I\}, \tag{10}$$

$$\overline{X}(I) = \{\mathbf{x} \in \mathbb{R}^\nu : \ g_k(\mathbf{x}) = b_k, \ k \in I\}, \tag{11}$$

and the corresponding problems

$$\min\{F(\mathbf{x}) : \ \mathbf{x} \in X(I) \cap X_0\}, \tag{$T(I)$}$$

$$\min\{F(\mathbf{x}) : \ \mathbf{x} \in \overline{X}(I)\} \tag{$\overline{T}(I)$}$$

with optimal solutions $\mathbf{x}^*(I)$ and $\overline{\mathbf{x}}^*(I)$ respectively.

The goal of this paper is to develop fast approximate methods for solving the problem $T(\mathbf{I})$.

Let $J(I) = \bigcup\limits_{k\in I} W_k$.

We will consider the non-trivial case, when all vertices of the set $\mathbf{J}$ are used in the collection of paths $\mathbf{I}$, and the problem cannot be decomposed into independent subproblems:

$$\bigcup_{k\in\mathbf{I}} W_k = \mathbf{J} \text{ and } \bigcup_{k\in I'} W_k \bigcap \bigcup_{k\in\mathbf{I}\setminus I'} W_k \neq \emptyset \text{ for any } I' \subset \mathbf{I}.$$

We also make the following assumptions:

- $$\sum_{(ij)\in W_k} \overline{x}_{ij} > b_k, \ k \in \mathbf{I}, \tag{A1}$$

  which implies that the feasible set of the problem $T(\mathbf{I})$ is non-empty and satisfies Slater's regularity conditions;

- the reduction of any variable to its lower bound when all other variables have feasible values leads to violation of all of the constraints corresponding to the set $\mathbf{I}$,

  $$\underline{x}_{sp} + \sum_{(ij)\in W_k\setminus\{(sp)\}} \overline{x}_{ij} < b_k \text{ for any } k \in \mathbf{I}, \ (sp) \in W_k, \tag{A2}$$

  which implies that the lower bound constraints cannot be binding in any feasible solution, and

- for any $(sp) \in J(I)$, $I \subseteq \mathbf{I}$ when x'_{sp} exceeds its upper bound $\overline{x}_{sp}$ then value $f_{sp}(x'_{sp})$ increases so fast that it exceeds $\sum_{(ij)\in J(I)} f_{ij}(x_{ij})$ for any feasible $\mathbf{x}$,

$$\sum_{(ij)\in J(I)} f_{ij}(x_{ij}) < \sum_{(ij)\in J(I)} f_{ij}(x'_{ij}), \text{ for any } I \subseteq \mathbf{I} \text{ and } x, x' \in \overline{X}(I) \text{ such that}$$
$$x_{ij} \in X_{ij} \text{ for all } (ij) \in J(I), \text{ and } x'_{sp} \notin X_{sp} \text{ for some } (sp) \in J(I), \tag{A3}$$

which implies that for any $I \subseteq \mathbf{I}$ such that $\overline{X}(I) \bigcap X_0 \neq \emptyset$, a solution of a problem $\overline{T}(I)$ cannot be achieved at any infeasible $\mathbf{x}'$.

The problem in question $T(\mathbf{I})$ is a special linearly constrained convex minimization problem. It is close, in particular, to convex multicommodity network flow problems [8], and some general solution techniques for the latter problems can also be applied to solve the problem $T(\mathbf{I})$. However, to create more efficient solution methods (in particular, for large scale instances), the peculiarities of the problem structure should be used. The structure of constraints of the problem $T(\mathbf{I})$ differs from those of multicommodity network flow problems. This structure is essential for special methods proposed in this paper.

In the proposed algorithms for the problem $T(\mathbf{I})$ solution, a sequence of auxiliary subproblems of type $\overline{T}(I)$ needs to be solved. To solve these subproblems, we use the Projected Newton Bracketing method (PNB) [2] in conjunction with the steepest descent method. We review the PNB method in the next section.

To describe the proposed approaches for solving $T(\mathbf{I})$ we introduce the following definitions.

Definition 1. (a) A set $I \subseteq \mathbf{I}$ is called a *covering* if $\bigcup_{k\in I} W_k = \mathbf{J}$.

(b) A covering I is called *extreme* if $\overline{\mathbf{x}}^*(I) = \mathbf{x}^*(I)$.

Define for any $k \in \mathbf{I}$ a binary vector

$$\mathbf{a}^{(k)} = (a_{11}^{(k)}, \ldots, a_{1\nu_1}^{(k)}, a_{21}^{(k)}, \ldots, a_{2\nu_2}^{(k)}, \ldots, a_{n1}^{(k)}, \ldots, a_{n\nu_n}^{(k)}),$$

where

$$a_{ij}^{(k)} = \begin{cases} 1 & \text{if } (ij) \in W_k, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2. A set $I \subseteq \mathbf{I}$ is called *linearly independent* if the set of vectors $\{\mathbf{a}^{(k)} : k \in I\}$ is linearly independent.

Let:

- $A(I)$ be the $|I| \times \nu$ matrix with rows $\mathbf{a}^{(k)},\ k \in I$;
- $\overline{\nu} = \max\{\nu_i : i = 1, ..., n\}$.

The following remarks give some properties of the coverings defined above.

Remark 1. A subset $I \subseteq \mathbf{I}$ is a covering if and only if $t_{ij}(I) = \sum_{k \in I} a_{ij}^{(k)} \geq 1$ for all $(ij) \in \mathbf{J}$. The number of elements in a covering is at least $\overline{\nu}$.

The number of elements in a linearly independent covering does not exceed $\nu - n + 1$ (see Theorem 2 in Appendix). For any linearly independent covering I the set $\overline{X}(I) \neq \emptyset$ (see Corollary 2 in Appendix).

Remark 2. Since $F(\mathbf{x})$ is strictly convex and increasing in each variable, it is also strictly increasing in each variable. Therefore, some of the constraints of $X(\mathbf{I})$ are binding in the optimal solution (recall that lower bound constraints cannot be binding by (A2)), and $\mathbf{x}^*(\mathbf{I})$ belongs to some facet of a convex polyhedral set $X(\mathbf{I})$.

Remark 3. If a covering $I \subseteq \mathbf{I}$ is linearly independent and $\mathbf{x} = \overline{\mathbf{x}}^*(I)$ then (see, for example, [4]) there exists a unique vector $\lambda = (\lambda_k : k \in I) \in \mathbb{R}^{|I|}$ such that

$$\nabla F(\mathbf{x}) = \sum_{k \in I} \lambda_k a^{(k)}. \tag{12}$$

A covering I is an extreme covering if and only if $\lambda \geq 0$ (see, for example,[4],[9]).

Remark 4. If in the covering I for any $k \in I$ there exists $(ij) \in W_k$ such that $t_{ij}(I) = 1$ then I is an extreme linearly independent covering.

Really, a covering I in this case is a linearly independent covering because $\mathbf{a}^{(k)}$ can not be a linear combination of other vectors of the set $\{\mathbf{a}^{(s)} :\ s \in I\}$. Moreover for any $k \in I$ take place $g_k(\overline{\mathbf{x}}^*(I)) = b_k$ because if there exists $s \in I$ such that $g_s(\overline{\mathbf{x}}^*(I)) > b_s$ then the value $F(\overline{\mathbf{x}}^*(I))$ can be decreased by decreasing some x_{ij} such that $(ij) \in W_s$ and $t_{ij}(I) = 1$ (with regard to (A2)). Consequently, $\overline{\mathbf{x}}^*(I) = \mathbf{x}^*(I)$ and such covering is an extreme covering.

3. The PNB method

The problem

$$\min f(\mathbf{x})$$

$$\text{s.t.} \quad A\mathbf{x} = \mathbf{b},$$

is specified by the triple $\{f, A, \mathbf{b}\}$, where $S = \{\mathbf{x} : A\mathbf{x} = \mathbf{b}\}$ is assumed nonempty and $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function, differentiable and bounded below with an attained infimum $f_{\min}$ on S.

An iteration of the PNB method begins with a feasible solution $\mathbf{x}$, and an interval $[L, U]$, called a *bracket*, containing the minimum value $f_{\min}$,

$$L \leq f_{\min} \leq U. \tag{13}$$

The upper bound is $U := f(\mathbf{x})$, where $\mathbf{x}$ is the current iterate. An initial lower bound L^0 is assumed known. At each iteration the bracket $[L, U]$ is reduced, either by lowering U or by raising L.

If the bracket is sufficiently small, say

$$U - L < \epsilon \tag{14}$$

then the current $\mathbf{x}$ is declared optimal, and computations stop.

For each non-terminal step, select $0 < \alpha < 1$ and define $M \in [L, U]$:

$$M := \alpha U + (1 - \alpha)L.$$

Let $P_{N(A)}$ be an orthogonal projection operator (see, e.g., [1]) on $N(A) = \{\mathbf{x} : A\mathbf{x} = 0\}$.

For a projected gradient direction

$$\mathbf{d} = P_{N(A)} \nabla f(\mathbf{x}), \tag{15}$$

do one iteration of the directional Newton method [6]

$$\mathbf{x}_+ := \mathbf{x} - \frac{f(\mathbf{x}) - M}{\langle \nabla f(\mathbf{x}), \mathbf{d} \rangle} \mathbf{d}, \tag{16}$$

as if to solve $f(\mathbf{x}) = M$. This guarantees that $\mathbf{x}$ lies in S at all iterates if the initial solution is in S. If $\mathbf{d} = \mathbf{0}$, i.e., if $\nabla f(\mathbf{x}) \perp N(A)$, then the current $\mathbf{x}$ is optimal.

The new value $f(\mathbf{x}_+)$ then allows narrowing the bracket $[L, U]$ to obtain a new bracket $[L_+, U_+]$, as follows:

$$\text{Case 1: if } f(\mathbf{x}_+) \;<\; f(\mathbf{x}) \quad \text{then} \quad U_+ := f(\mathbf{x}_+), L_+ := L, \tag{17a}$$

$$\text{Case 2: if } f(\mathbf{x}_+) \;\geq\; f(\mathbf{x}) \quad \text{then} \quad L_+ := M \;, U_+ := U, \mathbf{x}_+ := \mathbf{x}. \tag{17b}$$

In either case the bracket is reduced, the *reduction ratio* is

$$\frac{U_+ - L_+}{U - L} = \begin{cases} \dfrac{f(\mathbf{x}_+) - L}{f(\mathbf{x}) - L} & \text{in Case 1,} \\ 1 - \alpha & \text{in Case 2.} \end{cases} \tag{18}$$

Similar to the Newton Bracketing method [7], the PNB method is valid if the level sets of the problem are not "too narrow". A typical sufficient condition is: let f be a quadratic function

$$f(\mathbf{x}) = \tfrac{1}{2}\,\mathbf{x}^T Q\mathbf{x} - c^T\mathbf{x} + \gamma\,, \tag{19}$$

where the matrix Q is positive definite. Then the PNB method is valid for minimizing quadratic function f if the *condition number* of the matrix $P_{N(A)}QP_{N(A)}$ is sufficiently small,

$$\mathrm{cond}(P_{N(A)}QP_{N(A)}) \;\le\; \frac{1}{7-\sqrt{48}} \approx 13.92820356, \tag{20}$$

see [2, Section 8.1].

Note, that PNB method does not guarantee obtaining the problem solution, if the level sets of the problem are "too narrow". In this case, it can happen that $L > f_{\min}$ for current L.

4. The simplest special case

Consider a special case of the problem $T(\mathbf{I})$, when $\mathbf{I} = \mathbf{W}$, and $b_k = b$ for all $k \in \mathbf{W}$. The following property of a solution $\mathbf{x}^*$ of the problem $T(\mathbf{W})$ allows us to find this solution in a simple and efficient way.

Theorem 1. If $b_k = b$ for any $k \in \mathbf{W}$ then a solution $\mathbf{x}^* = \mathbf{x}^*(\mathbf{W})$ of the problem $T(\mathbf{W})$ is such that

- for any $i = 1, ..., n$, $x^*_{ij} = x^*_{i1}$ for all $j = 2, ..., \nu_i$,
- $g_k(\mathbf{x}^*) = b$ for all $k \in \mathbf{W}$.

Proof. Without loss of generality, assume $x^*_{i1} = \min\{x^*_{ij} : j = 1, ..., \nu_i\}$ for all $i = 1, ..., n$ and $W_1 = (11, ..., i1, ..., n1)$. Since $g_k(\mathbf{x}^*) \ge g_1(\mathbf{x}^*)$ for all $k \in \mathbf{W}$, and $\mathbf{x}^*$ is on the border of a polyhedral set $X(\mathbf{I})$ then $g_1(\mathbf{x}^*) = b$. Assume $x^*_{i'j'} > x^*_{i'1}$ for some $(i'j') \in \mathbf{J}$. Denote by I a set of all paths in $\mathbf{W}$ containing a vertex $\nu_{i'j'}$. Obviously, $g_k(\mathbf{x}^*) > b$ for all $k \in I$. Consider a vector $\mathbf{x}^0$, obtained from $\mathbf{x}^*$ by replacing $x^*_{i'j'}$ by $x^0_{i'j'} = x^*_{i'1}$. A vector $\mathbf{x}^0$ remains feasible due to condition (A2), $x^*_{i'1} > \underline{x}^*_{i'j'}$.

The following statements hold:

- $g_k(\mathbf{x}^*) = g_k(\mathbf{x}^0) \ge b$ for all $k \in \mathbf{W} \setminus I$,
- $g_k(\mathbf{x}^*) > g_k(\mathbf{x}^0) \ge g_1(\mathbf{x}^*) = b$ for all $k \in I$, and according to (A2) $x^0_{i'j'} > \underline{x}_{i'j'}$,
- $F(\mathbf{x}^*) > F(\mathbf{x}^0)$.

This contradicts to the assumption that $\mathbf{x}^*$ is a solution of the problem $T(\mathbf{W})$. Therefore, there is no $(i'j') \in \mathbf{J}$ such that $x^*_{i'j'} > x^*_{i'1}$. □

Let

$$\phi_i(x_{i1}) = \sum_{j=1}^{\nu_i} f_{ij}(x_{i1}).$$

Corollary 1. Under conditions of Theorem 1, the solution of the problem $T(\mathbf{W})$ can be reduced to finding a vector $(x^*_{i1} : i = 1, ..., n)$, which minimizes a function

$$\Phi(x_{11}, ..., x_{n1}) = \sum_{i=1}^{n} \phi_i(x_{i1}) \tag{21}$$

under a single linear equality constraint

$$\sum_{i=1}^{n} x_{i1} = b. \tag{22}$$

The problem (21)–(22) can be solved using the PNB method with the initial lower bound $L^0 = \Phi(a^0_{11}, ..., a^0_{n1})$, where $a^0_{i1} = \max\{\underline{x}_{ij} : j = 1, ..., \nu_i\}$, and the initial solution $(\frac{b}{n}, ..., \frac{b}{n})$ (assuming $\frac{b}{n} \le \min\{\overline{x}_{ij} : j = 1, ..., \nu_i\}$ for all $i = 1, ..., n$).

5. Approximate solution

An approximate solution of the problem $T(\mathbf{I})$ can be obtained by different heuristic methods. In this section we present two such algorithms. Each algorithm reduces the solution of the initial problem $T(\mathbf{I})$ to the solution of a sequence of subproblems of $\overline{T}(I)$ type, $I \subseteq \mathbf{I}$. For solving each of the subproblems we use the author's PNB method followed by (if further improvement is necessary) the steepest descent method. The improvement is needed when the level sets of the subproblem are "too narrow" (see Section 3).

5.1. The first algorithm in question maintains, for each path k, the sets of variables Ω_k which can still be modified in subsequent iterations. In each iteration, it selects a path (constraint) ℓ with the smallest slack and optimizes the variables in the corresponding set Ω_ℓ in the space of this path's equality constraint. These variables are then removed from all Ω-sets and remain fixed in subsequent iterations. The algorithm terminates when all variables are fixed and solves at most $\min\{|\mathbf{I}|, \nu - n + 1\}$ subproblems. Each of these subproblems contains at most n variables and a single binding constraint (9).

Algorithm 1 (Approximate solution of the problem $T(\mathbf{I})$)**.**

1. Set $I = \mathbf{I}$, $x_{ij} = \overline{x}_{ij}$ for all $(ij) \in \mathbf{J}$, $\Omega_k := W_k$ and $d_k := b_k$ for all $k \in I$, $Z = \emptyset$.

2. Choose

$$\ell \in \operatorname{Argmin}\Big\{\Delta_k = \sum_{(ij)\in\Omega_k} x_{ij} - d_k : k \in I\Big\}. \tag{23}$$

If $\Delta_\ell = 0$ then go to Step 5.

3. Find a solution $(x^*_{ij} : (ij) \in \Omega_\ell)$ of an auxiliary problem

$$\sum_{(ij)\in\Omega_\ell} f_{ij}(x_{ij}) \to \min, \tag{24}$$

$$\sum_{(ij)\in\Omega_\ell} x_{ij} = d_\ell. \tag{25}$$

4. Set

$$x_{ij} := x^*_{ij} \quad \text{for all} \quad (ij) \in \Omega_\ell\,.$$

5. Set

$$d_k := d_k - \sum_{(ij)\in\Omega_\ell \bigcap \Omega_k} x_{ij} \quad \text{and} \quad \Omega_k := \Omega_k \backslash \Omega_\ell \quad \text{for all} \quad k \in I,$$

$$I := I\backslash\{k : \Omega_k = \emptyset\};$$

$$Z := Z \cup \{\ell\}.$$

6. If $I = \emptyset$, an approximate solution of $T(\mathbf{I})$ has been obtained. Otherwise, go to Step 2 (the next iteration).

For solving the auxiliary problem (24)-(25) the PNB method is used with an initial lower bound

$$L^0 = \sum_{(ij)\in\Omega_\ell} f_{ij}(\underline{x}_{ij}),$$

and an initial solution from the set

$$\{(x_{ij} : (ij) \in \Omega_\ell) \in \prod_{(ij)\in\Omega_\ell} X_{ij}, \sum_{(ij)\in\Omega_l} x_{ij} = d_\ell\}.$$

The solution obtained by this method is further improved (if necessary) by the steepest descent method.

The number of iterations in Algorithm 1 is at most $\min[|\mathbf{I}|, \nu - n + 1]$, at each iteration we obtain a vector $\mathbf{x} \in X(\mathbf{I}) \cap X_0$, and a set Z obtained at the terminal iteration is a linearly independent covering (see Theorem 3 in Appendix).

A solution obtained by Algorithm 1 can be improved by Algorithm 2.

5.2. Algorithm 2 implements a controlled partial enumeration of facets of the polyhedron $X(\mathbf{I})$ with the minimization of the objective function on each current facet so that the obtained objective function values decrease. Then, the algorithm checks each obtained point $\mathbf{x}$ if it can be accepted as an approximate solution of the initial problem. It starts with some linearly independent covering Z and $\mathbf{x} \in \overline{X}(Z) \cap X(\mathbf{I}) \cap X_0$. At each iteration, the current point $\mathbf{x} \in X(\mathbf{I}) \cap X_0$, and we check if $\mathbf{x}$ is a solution of the auxiliary problem $\overline{T}(Z)$ for a current covering Z, i.e. $\mathbf{x} = \overline{\mathbf{x}}^*(Z)$. (The test is accomplished by attempting to represent the gradient of $F(\mathbf{x})$ as a linear combination of normal vectors $\mathbf{a}^{(k)}$, $k \in Z$.) One of the following three scenarios is possible. First, if $\mathbf{x} = \overline{\mathbf{x}}^*(Z)$, and all constraints of the problem $\overline{T}(Z)$ are essential in the problem $T(Z)$ (i.e. $\mathbf{x} = \overline{\mathbf{x}}^*(Z) = \mathbf{x}^*(Z)$), the algorithm stops with the obtained approximate solution $\mathbf{x}$ of the initial problem. Second, if $\mathbf{x} \neq \overline{\mathbf{x}}^*(Z)$, the algorithm attempts to improve the current point $\mathbf{x}$, or increase a lower bound of the objective function of the initial problem using the modified step of the PNB method. If this attempt fails, the algorithm tries to improve this $\mathbf{x}$ by means of the steepest descent method. Third, if $\mathbf{x} = \overline{\mathbf{x}}^*(Z)$, and for some elements $q \in Z$ the constraints $g_q(\mathbf{x}) \geq b_q$ are inessential for the problem $T(Z)$ then one such element q is temporarily removed from the covering Z. One iteration of the modified PNB method is attempted. It computes the maximum step size towards the proposed point while maintaining feasibility. If a constraint corresponding to some element $r \in \mathbf{I} \setminus Z$ prevents the move (maximum step size is zero), then the algorithm adds r to the current covering Z and moves to the next iteration. The algorithm keeps track of the set Y of elements q deleted from Z to prevent cycling.

Let $\mathbf{d}(\mathbf{x}, Z) = P_{N(A(Z))} \nabla F(\mathbf{x})$.

Algorithm 2.

1. Let $Z = Z^0$ be a linearly independent covering and $\mathbf{x} = \mathbf{x}^0 \in \overline{X}(Z) \cap X(\mathbf{I}) \cap X_0$.
   Set $U := F(\mathbf{x})$, $L := L_0 = \sum\limits_{(ij) \in \mathbf{J}} f_{ij}(\underline{x}_{ij})$ and $Y := \emptyset$.
2. Attempt to find $\lambda = (\lambda_k : k \in Z) \in \mathbb{R}^{|Z|}$ such that $\nabla F(\mathbf{x}) = \sum\limits_{k \in Z} \lambda_k \mathbf{a}^{(k)}$.
   If such λ exists, go to Step 3. Otherwise, go to Step 4.
3. Choose $q \in \operatorname{Argmin}\{\lambda_k : \sum\limits_{l \in Z} \lambda_l \mathbf{a}^{(l)} = \nabla F(\mathbf{x}),\ k \in Z \setminus Y\}$.
   If $\lambda_q \geq 0$, then $\mathbf{x}$ is an approximate solution, stop.
   Otherwise, set $Z := Z \setminus \{q\}$, $Y := Y \cup \{q\}$ and $L = L_0$.
4. Define $\alpha \in (0, 1)$, and set $M := \alpha U + (1 - \alpha) L$.

5. Calculate $\mathbf{d}(\mathbf{x}, Z)$. If $\mathbf{d}(\mathbf{x}, Z) \approx 0$, go to Step 3.
   Otherwise, set $\mathbf{x}_+ := \mathbf{x} - \dfrac{U - M}{\langle \nabla F(\mathbf{x}), \mathbf{d}(\mathbf{x}, Z)\rangle}\, \mathbf{d}(\mathbf{x}, Z)$.

6. Define $\tilde{\alpha} = \max\{h \in [0,1] : g_k(\mathbf{x}(h)) \geq b_k, k \in \mathbf{I}\backslash Z\}$, where $\mathbf{x}(h) = h\mathbf{x}_+ + (1-h)\mathbf{x}$.

7. If $\tilde{\alpha} = 0$, choose $r \in \{k \in \mathbf{I}\backslash Z : g_k(\mathbf{x}_+) < b_k, g_k(\mathbf{x}) = b_k\}$ and set $Z := Z \cup \{r\}$.
   Otherwise, set $Y := \emptyset$ and go to Step 9.

8. If $Y \neq \emptyset$ and $\mathbf{a}^{(q)}$ is linearly dependent on the system of vectors $\{\mathbf{a}^{(k)} : k \in Z\}$, go to Step 3.
   Otherwise, go to Step 5.

9. Set $\mathbf{x}' := \tilde{\alpha}\mathbf{x}_+ + (1 - \tilde{\alpha})\mathbf{x}$.

10. If $F(\mathbf{x}') < F(\mathbf{x})$, set $U := F(\mathbf{x}')$ and $\mathbf{x} := \mathbf{x}'$. Otherwise, set $L := \tilde{\alpha}M + (1 - \tilde{\alpha})U$.
    If $U - L > \varepsilon$, go to Step 4. If $\mathbf{d}(\mathbf{x}, Z) \approx 0$, go to Step 3.

11. Starting with the initial point $\mathbf{x}$ find by the steepest descent method a point $\mathbf{x}_s \in \overline{X}(Z) \cap X(\mathbf{I})$ such that $F(\mathbf{x}_s) \leq F(\mathbf{x})$ and $\mathbf{d}(\mathbf{x}_s, Z) \approx 0$ or $\langle \mathbf{a}^{(r)}, \mathbf{d}(\mathbf{x}_s, Z)\rangle < 0$ for some $r \in \mathbf{I}\backslash Z$.
    Set $L := L_0$. If $\mathbf{x}_s \neq \mathbf{x}$ then set $\mathbf{x} := \mathbf{x}_s, U := F(\mathbf{x}), Y := \emptyset$, otherwise set $Z := Z \cup \{r\}$. If $\mathbf{d}(\mathbf{x}_s, Z) \approx 0$ then go to Step 3, otherwise go to Step 4.

Remark 5. Step 11 is necessary since the level sets of the problem for the current Z can be "too narrow", and the PNB method can stop at $\mathbf{x}$ such that $|\mathbf{d}(\mathbf{x}, Z)| >> \varepsilon$ (see Section 3).

With the assumption of finiteness of the steepest descent method for a given accuracy, Algorithm 2 is finite. The set Z obtained by Algorithm 2 is a linearly independent covering, a vector $\mathbf{x} \approx \overline{\mathbf{x}}^*(Z) \in X(\mathbf{I}) \cap X_0$, and $F(\mathbf{x}) \leq F(\mathbf{x}^0)$ (see Theorem 4 in Appendix).

6. Numerical experiments

The goal of the numerical experiments is to test the performance of Algorithms 1, 2 and estimate its dependence on the problem size as well as the applicability of the PNB-method to problems of considered class. The algorithms were tested on the following class of problems $T(I)$:

- the functions $f_{ij}(x_{ij})$ are exponential, $f_{ij}(x_{ij}) = a_{ij}e^{c_{ij}x_{ij}}$, where $a_{ij}, c_{ij} > 0$;
- $x_{ij} \in X_{ij} = [0,1]$ for all $(ij) \in \mathbf{J}$;
- $b_k \in (n-1, n)$ for all $k \in I$.

These problems appear in optimization of engineering systems, in particular, the multi-unit transmission systems. The controlled parameters are survival probabilities of system elements, and the dependence of objective functions (e.g. system cost or system mass) on these parameters can be represented by an exponential function. Besides, the controlled parameters have the same interval of possible values that can be reduced to $[0, 1]$ by a linear transformation. The parameters b_k are chosen so that the conditions (A1) and (A2) hold.

We use Algorithms 1 and 2 to solve a set of random problems of the considered type. These problems are grouped in collections of 10. Problems from the same collection are characterized by the same parameters n, ν and $|I|$ and differ in the set of constraints I and the values of $\nu_i, b_k, a_{ij}, c_{ij}$. The coefficients a_{ij} and c_{ij} are generated in a random way so that the condition (A3) holds. A set of constraints I for each generated problem is chosen so that I is a covering.

Let $x^*(A1)$ denote an approximate solution of a problem in question obtained by Algorithm 1. When $x^*(A1)$ is improved by Algorithm 2, denote an improved approximate solution by $x^*(A2)$.

To solve each problem from a collection we use the following sequence:

- find an approximate solution $x^*(A1)$;

- find an improved solution $x^*(A2)$;

- find an exact solution $x^*(I)$ of the problem $T(I)$ by the special method based on the idea of constraints relaxation [3].

The subproblems (24)–(25) formed in Algorithm 1 were solved in two stages:

- solving by the PNB-method followed by verification of optimality of the obtained solution;

- improvement of a non-optimal solution by the steepest descent method in the corresponding manifold.

The following characteristics were calculated (for each collection of 10 problems):

- t_{A1} - the average processing time (sec) for the search of $x^*(A1)$;

- $N(A1)$ - the average number of subproblems solved by Algorithm 1;

- $N_{PNB}(A1)$ - the average total number of iterations of the PNB-method executed by Algorithm 1;

- $\xi(A1)$ - the relative inaccuracy $\frac{F(x^*(A1))-F(x^*(I))}{F(x^*(I))}$ of $F(x^*(A1))$ with respect to the optimal value $F(x^*(I))$;

- $M\xi(A1)$ - the average of $\xi(A1)$;

- $D\xi(A1)$ - the standard deviation of $\xi(A1)$;

- $\psi(A1)$ - the proportion of subproblems in Algorithm 1 successfully solved by the PNB-method;

- $M\psi(A1)$ - the average of $\psi(A1)$;

- $D\psi(A1)$ - the standard deviation of $\psi(A1)$;

- t_{A2} - the average processing time (sec) for the search of $x^*(A2)$ by Algorithm 2 with the initial approximation $x^*(A1)$;

- $N(A2)$ - the average number of subproblems solved by Algorithm 2;

- $N_{PNB}(A2)$ - the average total number of iterations of the PNB-method executed by Algorithm 2;

- $\xi(A1-2)$ - the relative inaccuracy $\frac{F(x^*(A1))-F(x^*(A2))}{F(x^*(A2))}$ of $F(x^*(A1))$ with respect to the value $F(x^*(A2))$;

- $M\xi(A1-2)$ - the average of $\xi(A1-2)$;

- $D\xi(A1-2)$ - the standard deviation of $\xi(A1-2)$;

- $\xi(A2)$ - the maximum relative inaccuracy $\frac{F(x^*(A2))-F(x^*(I))}{F(x^*(I))}$ of $F(x^*(A2))$ obtained by Algorithm 2 with respect to the optimal value $F(x^*(I))$.

The results of experiments for Algorithms 1 and 2 are presented in Tables 1 and 2.

We next present some conclusions from the obtained results.

1. The average number $N(A1)$ of subproblems solved by Algorithm 1 increases proportionally to $\ln \nu$ with coefficient of proportionality approximately equal to 1.6-1.8. Similarly, $N(A1)$ increases proportionally to $\ln |I|$ with coefficient of proportionality approximately equal to 0.79-0.82.

2. The average aggregate number $N_{PNB}(A1)$ of iterations of the PNB-method executed by Algorithm 1 increases proportionally to $\ln \nu$ with coefficient of proportionality approximately equal to 1.1-1.15. Similarly, $N_{PNB}(A1)$ increases proportionally to $\ln |I|$ with coefficient of proportionality approximately equal to 0.8-0.82.

3. The average inaccuracy $M\xi(A1)$ of solution values obtained by Algorithm 1 with respect to the optimal solution values varies in the range [0.095;1.29] and decreases with the increase of $|I|$.

4. Algorithm 2 finds $x^*(I)$ in almost all problems.

5. Algorithm 2 requires considerably more calculations than Algorithm 1.

7. Conclusion

The numerical experiments have confirmed the applicability of both Algorithms 1 and 2 for convex minimization problems with series-parallel structure and revealed that the average number of subproblems and processing time increase sufficiently slowly with the problem size. Algorithm 1, relatively

n	ν	$\|I\|$	$N(A1)$	$N_{PNB}(A1)$	$M\xi(A1)$	$D\xi(A1)$	$M\psi(A1)$	$D\psi(A1)$	t_{A1}
4	8	4	3.7	95.1	0.095	0.00548	0.65	0.525	0.005
4	8	10	4.4	117.3	0.19319	0.05342	0.38	0.216	0
4	8	16	4.7	131.1	0.13482	0.01050	0.29	0.189	0.011
4	12	10	7.2	235.8	0.39968	0.11351	0.3941	0.266	0
4	12	40	8.4	291.7	0.41758	0.08809	0.1944	0.0463	0
4	12	80	8.8	309.8	0.30657	0.05298	0.1389	0.0309	0
4	16	13	9.2	349.8	0.78805	0.58112	0.4736	0.091	0
4	16	100	12.4	486.9	0.57372	0.19361	0.1324	0.03978	0
4	16	256	12.9	508.4	0.4815	0.08657	0.0859	0.00725	0
4	20	16	11.7	489.1	0.90747	0.67132	0.48788	0.0186	0.011
4	20	200	16.4	697.8	0.40708	0.06190	0.09853	0.0105	0.01
4	20	625	17.0	722.3	0.47080	0.30287	0.05882	0	0.015
8	16	10	5.2	296.3	1.00718	0.78521	0.735	0.21358	0.005
8	16	100	7.8	433.5	0.51401	0.1980	0.69484	0.19616	0.006
8	16	256	8.3	456.2	0.50564	0.14801	0.71389	0.32569	0.011
8	24	17	8.3	481.2	1.18416	1.2828	0.79115	0.14515	0.011
8	24	1000	14.2	787.4	0.90196	0.47199	0.70743	0.13787	0.006
8	24	6561	16.8	905.7	0.44885	0.06135	0.64779	0.28162	0.0143
8	28	21	10.7	620.4	1.25808	0.77459	0.80255	0.11006	0.033
8	28	1000	17.7	974.2	1.28899	0.68464	0.71842	0.12482	0.033
8	28	20000	20.8	1117.6	0.65316	0.1878	0.53762	0.05665	0.56
12	24	13	5.5	320.3	0.17434	0.08213	0.930	0.12989	0.023
12	24	1500	11.5	634.0	0.65477	0.31668	0.72514	0.11463	0.035
12	24	4096	12.3	668.6	0.60108	0.28123	0.57622	0.34395	0.111
12	28	20	7.5	431.9	0.65229	0.49537	0.92083	0.07024	0.017
12	28	2000	13.9	769.5	0.96424	0.52089	0.69714	0.23861	0.055
12	28	16000	16.2	875.8	0.93806	0.35559	0.65132	0.21941	0.636
16	32	17	5.5	324.5	0.59418	0.33907	0.91905	0.12154	0.029
16	32	1000	10.8	606.8	0.77479	0.07568	0.78273	0.07307	0.068
16	32	20000	15.2	828.3	0.71435	0.16382	0.69911	0.38962	0.512

TABLE 1. The results of numerical experiments for Algorithm 1

quickly, finds an approximate solution that is fairly close to the optimal one. Algorithm 2 requires considerably more calculations than Algorithm 1 but finds the optimal solution of the problem in almost all instances. Unfortunately, the use of the PNB method for considered problems does not guarantee optimality of the obtained solutions since the level sets of the problems can be "too narrow". Nevertheless, its use in conjunction with the steepest descent method allows to reduce the processing time of the proposed methods.

ACKNOWLEDGEMENT

We thank the referees for their constructive suggestions, and A. Suvorov for the implementation of the proposed algorithms. The research of the second and fourth authors was partly supported by the International Science and Technology Center (Project B-986). The research of the third author was supported by Natural Sciences and Engineering Research Council of Canada (grant number 261512-04).

n	ν	$\|I\|$	$N(A2)$	$N_{PNB}(A2)$	$M\xi(A1-2)$	$D\xi(A1-2)$	$\xi(A2)$	t_{A2}
4	8	4	4.1	71.4	0.095	0.005477	0	0
4	8	10	5.8	103.6	0.19319	0.05342	0	0
4	8	16	7.3	127.2	0.13482	0.01050	0	0
4	12	10	9.3	171.6	0.39968	0.11351	0	0.011
4	12	40	13.5	250.6	0.41758	0.08809	0	0.011
4	12	80	16.4	288.8	0.30657	0.05298	0	0.011
4	16	13	12.0	269.5	0.78805	0.58112	0.000074	0.038
4	16	100	24.7	485.6	0.57372	0.19361	0	0.044
4	16	256	33.8	655.1	0.4815	0.08657	0	0.088
4	20	16	14.9	323.7	0.90747	0.67132	0.000367	0.005
4	20	200	36.8	712.7	0.40708	0.06190	0	0.1
4	20	625	55.4	1147.7	0.47080	0.30287	0	0.337
8	16	10	7.0	258.8	1.00714	0.785164	0.008104	0.022
8	16	100	16.0	433.8	0.51401	0.1980	0	0.049
8	16	256	21.4	584.7	0.50564	0.14801	0	0.11
8	24	17	11.9	330.5	1.18415	1.2828	0.001888	0.033
8	24	1000	64.3	1700.0	0.90196	0.47199	0	1.306
8	24	6561	111.5	3137.0	0.44884	0.061345	0	13.512
8	28	21	15.9	415.7	1.25782	0.774195	0.004697	0.054
8	28	1000	84.9	2291.9	1.28899	0.68464	0	1.764
8	28	20000	198.6	6000.6	0.65318	0.1878	0	81.896
12	24	13	8.4	238.1	0.17428	0.082147	0.001315	0.054
12	24	1500	51.2	1409.2	0.65477	0.31668	0	1.964
12	24	4096	71.6	2070.3	0.60108	0.28123	0	7.37
12	28	20	10.2	295.5	0.65228	0.49537	0.000035	0.06
12	28	2000	72.6	1998.1	0.96424	0.52089	0	3.937
12	28	16000	164.2	5125.3	0.93806	0.35559	0	72.403
16	32	17	7.7	245.2	0.59413	0.339105	0.001712	0.08
16	32	1000	47.0	1253.8	0.77479	0.075638	0	1.548
16	32	20000	172.6	5437.8	0.71435	0.16382	0	122.045

TABLE 2. The results of numerical experiments for Algorithm 2

REFERENCES

[1] A. Ben-Israel and T.N.E. Greville, *Generalized Inverses: Theory and Applications*, Springer, New York, 2003.

[2] A. Ben-Israel and Y. Levin, The Newton Bracketing Method for the Minimization of Convex Functions subject to Affine Constraints, *Discrete Applied Mathematics*, *Special Issue in Memory of Leo Khachiyan*, forthcoming.

[3] A. Ben-Israel, Y. Levin, G. Levin and B. Rozin, A Decomposition Approach to the Minimization of a Separable Convex Function on a Convex Polyhedron generated by a Parallel-Consecutive Structure, *Abstracts of the 18th International Symposium on Mathematical Programming, Copengagen, Denmark, August 18-22,2003, p.62.*

[4] V. Karmanov, *Mathematical Programming*, Mir, Moscow, 1989.

[5] L. Lasdon, *Optimization Theory for Large Systems*, Macmillan, New York, 1970.

[6] Y. Levin and A. Ben-Israel, Directional Newton Methods in n Variables, *Math. of Computation* **71**(2002), 237–250.

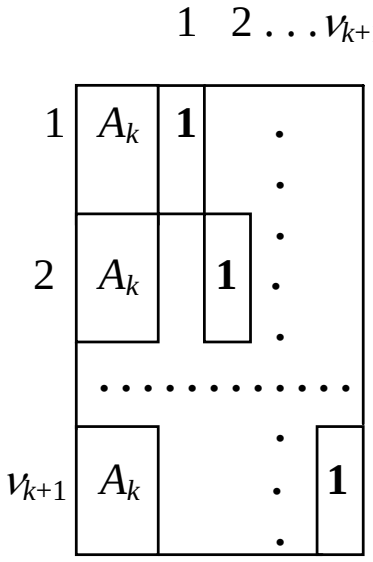

FIGURE 1. The structure of the matrix A_{k+1}.

[7] Y. Levin and A. Ben-Israel, The Newton Bracketing Method for Convex Minimization, *Comput. Optimiz. and Appl.* **21**(2002), 213–229.

[8] A. Ouorou, P. Mahey and J.-Ph. Vial, A Survey of Algorithms for Convex Multicommodity Flow Problems, *Management Science*, **46**(2000), 126–147.

[9] J. Rosen, The Gradient Projection Method for Nonlinear Programming, Part I, Linear Constraints, *J.Soc.Indust. and Appl. Math.* **8**(1960), 184–217.

8. Appendix. Basic theorems

The algorithms proposed in the previous section are based on the following results.

Theorem 2. A number of elements in a linearly independent covering does not exceed $\nu - n + 1$.

Proof. It is sufficient to show that the rank of the matrix $A(\mathbf{W})$ is equal to $\nu - n + 1$. We prove this statement by induction on the number n of sets G_i and the number ν_i of elements in the current set G_i. Let $A_k = A(\mathbf{W})$ for $n = k$. A_k consists of $p_k = \prod_{i=1}^{k} \nu_i$ rows and $s_k = \sum_{i=1}^{k} \nu_i$ columns. Figure 1 presents the structure of the matrix A_{k+1} for $k > 1$, where **1** denotes a submatrix which consists of the single column of ones.

For $n = 1$ the theorem holds, since A_1 is an identity $\nu_1 \times \nu_1$ matrix.

We assume that the theorem holds for $n = k$, and prove for $n = k + 1$.

For $\nu_{k+1} = 1$ the rank of the matrix A_{k+1} coincides with the rank of the matrix A_k, since the column **1** is a sum of the first ν_1 columns of A_k. Since the rank of the matrix A_k is $\sum_{i=1}^{k} \nu_i - k + 1$ (or $\sum_{i=1}^{k+1} \nu_i - (k+1) + 1$), the theorem is true for $\nu_{k+1} = 1$.

Suppose that the theorem holds for $\nu_{k+1} = l$ and show that it also holds for $\nu_{k+1} = l + 1$.

Observe that for any $t = 1, ..., p_k$, a row $\mathbf{a}^{(lp_k+t)}$ of the matrix A_{k+1} is linearly independent of its rows $\mathbf{a}^{(q)}, q = 1, ..., lp_k$. At the same time, $\mathbf{a}^{(lp_k+t)} = \mathbf{a}^{(lp_k+1)} + \mathbf{a}^{(t)} - \mathbf{a}^{(1)}$ for any $t = 2, ..., p_k$. Therefore, the new block of rows appended to A_{k+1} when ν_{k+1} is increased by 1 only adds one row which is independent of the rows contained in the block above, and the rank of A_{k+1} is $\sum_{i=1}^{k+1} \nu_i - (k+1) + 1$ for $\nu_{k+1} > 1$. □

Corollary 2. For any linearly independent covering I the set $\overline{X}(I) \neq \emptyset$.

The validity of the statement follows from the fact that the number of variables ν always exceeds the maximum possible number $\nu - n + 1$ of constraints in I.

Thereby, the system of constraints of each auxiliary problem $\overline{T}(I)$ is consistent and contains at most $\nu - n + 1$ linearly independent constraints. So the solution $\overline{\mathbf{x}}^*(I)$ of the problem $\overline{T}(I)$ always exists and belongs to X_0 if $\overline{X}(I) \bigcap X_0 \neq \emptyset$ (follows from assumption (A3)).

Theorem 3. (a) The number of iterations in Algorithm 1 is at most $\min[|\mathbf{I}|, \nu - n + 1]$.

(b) At each iteration of Algorithm 1, $\mathbf{x} \in X(\mathbf{I}) \bigcap X_0$.

(c) A set Z obtained at the terminal iteration is a linearly independent covering.

Proof. **Part (a).** At the initial iteration the set Ω_ℓ contains exactly n elements from $\mathbf{J}$, which are fixed at this iteration. At each subsequent iteration of the algorithm at least one new element from $\mathbf{J}$ is fixed. Since $|\mathbf{J}| = \nu$, the number of iterations does not exceed $\nu - n + 1$. On the other hand, the number of iterations is at most $|\mathbf{I}|$, since at each iteration the current set $I \subseteq \mathbf{I}$ is reduced by at least one element. Thus, the number of iterations is at most $\min[|\mathbf{I}|, \nu - n + 1]$.

Part (b). For an initial solution $\mathbf{x} = (\overline{x}_{ij} : (ij) \in \mathbf{J})$ all inequalities (9) of the problem $T(\mathbf{I})$ are satisfied (see Assumption (A1)).

As ℓ is selected by (23) at each iteration and $\{x_{ij} = x^*_{ij} : (ij) \in \Omega_\ell\}$ is a solution of the problem (24)–(25), then

$$\sum_{(ij)\in\Omega_\ell} x_{ij} - d_\ell = 0,$$

and for all $k \in I$ the following statements hold:

(i) $\sum\limits_{(ij)\in\Omega_k} x_{ij} - d_k \geq 0 \quad \Longrightarrow \quad \sum\limits_{(ij)\in W_k} x_{ij} - b_k \geq 0;$

(ii) $\sum\limits_{(ij)\in\Omega_k} x_{ij} - d_k \geq \sum\limits_{(ij)\in\Omega_\ell} x_{ij} - d_\ell = 0$; and

(iii) $\sum\limits_{(ij)\in\Omega_\ell} (\overline{x}_{ij} - x_{ij}) \geq \sum\limits_{(ij)\in\Omega_\ell\cap\Omega_k} (\overline{x}_{ij} - x_{ij}).$

Therefore, at each iteration $\sum\limits_{(ij)\in W_k} x_{ij} \geq b_k$ for all $k \in I$.

The validity of the statement $x_{ij} \leq \overline{x}_{ij}$ for all $(ij) \in \Omega_\ell$ follows directly from Assumption (A3). Then $x_{ij} \geq \underline{x}_{ij}$ for all $(ij) \in \Omega_\ell$ follows from Assumption (A2). Finally, at each iteration $\mathbf{x} \in X(I) \cap X_0$.

Part (c). At each iteration of the algorithm, Z is a linearly independent set, since each new inserted constraint ℓ contains at least one variable x_{ij} such that $(ij) \notin \bigcup\limits_{j\in Z} W_j$. Obviously, at the terminal iteration $\bigcup\limits_{j\in Z} W_j = \mathbf{J}$, and Z is a linearly independent covering. □

Let $\mathbf{d}(\mathbf{x}, Z) = P_{N(A(Z))}\nabla F(\mathbf{x})$.

Lemma 1. Let Z be a linearly independent covering, $q \in Z$ is such that $\lambda_q < 0$ in the expansion

$$\nabla F(\overline{\mathbf{x}}^*(Z)) = \sum_{k\in Z} \lambda_k \mathbf{a}^{(k)}. \tag{26}$$

Then $\langle \mathbf{a}^{(q)}, \mathbf{d}(\mathbf{x}, Z\backslash\{q\})\rangle < 0$.

The validity of this statement follows, in particular, from [9].

Theorem 4. Let Z^0 be a linearly independent covering and $\mathbf{x}^0 \in \overline{X}(Z^0) \cap X(\mathbf{I}) \cap X_0$. Then Algorithm 2 finds in a finite number of iterations a linearly independent covering Z and $\mathbf{x} \in X(\mathbf{I}) \cap \overline{X}(Z) \cap X_0$ such that $F(\mathbf{x}) \leq F(\mathbf{x}^0)$ and $F(\mathbf{x}) \approx F(\overline{\mathbf{x}}^*(Z))$ or $\mathbf{d}(\mathbf{x}, Z) \approx 0$.

Proof. 1. An initial set Z is, by definition, a linearly independent covering and $\mathbf{x} \in \overline{X}(Z) \cap X(\mathbf{I}) \cap X_0$. Let us show that at any subsequent iteration and at any step of the algorithm the current Z and $\mathbf{x}$ satisfy these conditions, and at any Step 3 $F(\mathbf{x}) \approx F(\overline{\mathbf{x}}^*(Z))$ or $\mathbf{d}(\mathbf{x}, Z) \approx 0$.

The set Z can be modified only at Steps 3,7 or 11 of the algorithm, and $\mathbf{x}$ - at Steps 10 and 11. We consider these modifications next.

Let at the current Step 3 Z be a linearly independent covering, $\mathbf{x} \in \overline{X}(Z) \cap X(\mathbf{I}) \cap X_0$ and $F(\mathbf{x}) \approx F(\overline{\mathbf{x}}^*(Z))$ or $\mathbf{d}(\mathbf{x}, Z) \approx 0$ (i.e. $\nabla F(\mathbf{x}) \approx \sum\limits_{k\in Z} \lambda_k \mathbf{a}^{(k)}$ for some $\lambda = (\lambda_k : k \in Z)$). Suppose that a set $Z\backslash\{q\}$ formed at Step 3 is not a covering and $\lambda_q < 0$. Then there exists $(ij) \in W_q$ such that $(ij) \notin W_k$ for any $k \in Z\backslash\{q\}$. Since $F(\mathbf{x})$ is an increasing function in components of $\mathbf{x}$, then $\lambda_q = \nabla F_{ij}(\mathbf{x}) > 0$.

This contradicts to the condition $\lambda_q < 0$. Consequently, $Z\backslash\{q\}$ is a linearly independent covering. It is also obvious that $\mathbf{x} \in \overline{X}(Z\backslash\{q\}) \cap X(\mathbf{I}) \cap X_0$.

At Step 7, $g_k(\mathbf{x}) = g_k(\mathbf{x}_+) = b_k$ for all $k \in Z$, and $g_r(\mathbf{x}_+) < g_r(\mathbf{x}) = b_r$ for r which is added to the set Z at this step. Therefore, the set $Z \cup \{r\}$ formed at this step is a linearly independent covering if Z is a linearly independent covering and moreover $\mathbf{x} \in \overline{X}(Z \cup \{r\}) \cap X(\mathbf{I}) \cap X_0$.

Obviously, under transition to Step 3 from Steps 2, 5, 10 or 11 for current $\mathbf{x}$ and Z the value $F(\mathbf{x}) \approx F(\overline{\mathbf{x}}^*(Z))$ or $\mathbf{d}(\mathbf{x}, Z) \approx 0$.

Consider a transition to Step 3 from Step 8. Let $\mathbf{x}_l$ and Z_l denote the current $\mathbf{x}$ and Z after the modification at Step $l \in \{3, 7\}$, $A(Z) = \{\mathbf{a}^{(k)} : k \in Z\}$. According to the algorithm $\mathbf{x}_3 = \mathbf{x}_7 = \mathbf{x}$, $Z_3 \subset Z_7$, and $\nabla F(\mathbf{x})$ is a linear combination of $A(Z_3 \cup \{q\})$. Since $\mathbf{a}^{(q)}$ is a linear combination of $A(Z_7)$, $\nabla F(\mathbf{x})$ is also a linear combination $A(Z_7)$, i.e. $F(\mathbf{x}) \approx F(\overline{\mathbf{x}}^*(Z_7))$ or $\mathbf{d}(\mathbf{x}, Z_7) \approx 0$.

For $\mathbf{x}'$ obtained at Step 9 and current $\mathbf{x}$ and Z, the segment $[\mathbf{x}, \mathbf{x}'] \subseteq \overline{X}(Z) \cap X(\mathbf{I})$ (see Steps 6,7,9) and the set $\{\mathbf{x}'' \in [\mathbf{x}, \mathbf{x}'] | F(\mathbf{x}'') \leq F(\mathbf{x})\} \subseteq X_0$ (according to Assumptions (A1)-(A3)). Thus, $\mathbf{x} \in X_0$ after Step 10. Similarly, it can be shown that $\mathbf{x} \in X_0$ after Step 11.

According to Assumptions (A1-A3), the value $L = L_0$ assigned at Steps 1, 3 and 11 satisfies to the condition $L \leq F(\overline{\mathbf{x}}^*(Z))$ for any current Z. Moreover, if the condition $L \leq F(\overline{\mathbf{x}}^*(Z))$ holds for the current Z at some step of the algorithm, then it holds for all subsequent steps while L is unmodified (at Step 10), since at Step 7 Z can only be expanded.

When analyzing the possible modification of L for the current Z at Step 10, it is necessary to distinguish two cases. In the first case when the level sets are not "too narrow" the modified value $L = \tilde{\alpha}M + (1 - \tilde{\alpha})U \leq F(\overline{\mathbf{x}}^*(Z))$ for the current Z, since $0 < \tilde{\alpha} \leq 1$ and $\mathbf{x}' = \mathbf{x} - \tilde{\alpha}((U - L)/\langle \nabla F(\mathbf{x}), \mathbf{d}(\mathbf{x}, Z)\rangle)\mathbf{d}(\mathbf{x}, Z)$ for the current $\mathbf{x}$ (see Section 3). Thus, under transition to Step 3 from Step 10 in this case, $F(\mathbf{x}) \approx F(\overline{\mathbf{x}}^*(Z))$ and $\mathbf{d}(\mathbf{x}, Z) \approx 0$.

Consider the second case (i.e. when level sets can be "too narrow" for some current Z). Then it is possible for such Z that $L > F(\overline{\mathbf{x}}^*(Z))$ at Step 10 in some iteration of the algorithm. This can result in transition from Step 10 to Step 11 with $U - L \leq \varepsilon$ and $F(\mathbf{x}) > F(\overline{\mathbf{x}}^*(Z)) + \varepsilon$. Then at Step 11 the algorithm finds (if it is possible) an improved $\mathbf{x} \in \overline{X}(Z) \cap X(\mathbf{I}) \cap X_0$, the values U, L are modified, and a transition to Step 3 is made if $\mathbf{d}(\mathbf{x}, Z) \approx 0$ or to Step 4, otherwise. Under transition to Step 4, the current Z can be modified in the similar way as at Step 7. Consequently, such modified Z is a linearly independent covering.

In summary, all sets Z obtained in the algorithm are linearly independent coverings, $\mathbf{x} \in \overline{X}(Z\backslash\{q\}) \cap X(\mathbf{I}) \cap X_0$ and under transition to Step 3 the value $F(\mathbf{x}) \approx F(\overline{\mathbf{x}}^*(Z))$ or $\mathbf{d}(\mathbf{x}, Z) \approx 0$ for the current $\mathbf{x}$ and Z.

2. Next we show the finiteness of the algorithm.

Let at the current Step 8 the value $\tilde{\alpha} = 0$, $Y \neq \emptyset$ and $\mathbf{a}^{(q)}$ is linearly independent of the system $A(Z)$. Then for the current $\mathbf{x}$ and Z the following conditions hold:

a) $\nabla F(\mathbf{x})$ is not a linear combination of $A(Z)$,

b) in the expansion $\nabla F(\mathbf{x}) = \sum\limits_{k \in Z \cup \{q\}} \lambda_k \mathbf{a}^{(k)}$ the coefficient $\lambda_r = 0$ for the current r, and for all $k \in Z\backslash\{r\}$ the values of λ_k coincide with their values for the previous Z.

Thus, according to Lemma 1, $\langle \mathbf{a}^{(q)}, \mathbf{d}(\mathbf{x}, Z\backslash\{q\}) \rangle < 0$ for the current $\mathbf{x}$ and Z. Consequently, the constraint $g_q(\mathbf{x}) \geq b_q$ is inessential at this point $\mathbf{x}$ and can be removed.

Otherwise, given $\tilde{\alpha} = 0$ after the transition to Step 3, the algorithm either stops (if $\lambda_k \geq 0$ for all $k \in Z\backslash Y$) or the next q such that $\lambda_q < 0$ is removed from the current set $Z\backslash Y$ and added to Y. Afterwards, while the value $\tilde{\alpha} = 0$ remains unmodified, this q can be inserted in Z at some subsequent Step 7 but cannot subsequently be removed from $Z\backslash Y$ without previous modification of $\tilde{\alpha}$ (see Steps 7,8 and 3). Thus, due to finiteness of the set $\mathbf{I}$, the number of iterations of the algorithm with fixed $\tilde{\alpha} = 0$ is finite. Particularly, the number of transitions to Step 3 with unmodified $\mathbf{x}$ is finite.

For unmodified Z, due to finiteness of the PNB method, the number of iterations in the cycle between Steps 4 and 11 is finite. As in that cycle Z can only be expanded due to the finiteness of the set $\mathbf{I}$, the total number of iterations in this cycle is also finite.

Consequently, the number of steps of the algorithm between two adjacent transitions to Step 3 with different values of $\mathbf{x}$ is finite. As we showed above, at Step 3 $F(\mathbf{x}) \approx F(\overline{\mathbf{x}}^*(Z))$ or $\mathbf{d}(\mathbf{x}, Z) \approx 0$ for the current $\mathbf{x}$ and Z. Due to the strict convexity of $F(\mathbf{x})$ and decreasing values $F(\mathbf{x})$ while $\mathbf{x}$ is modified in the algorithm, Z cannot be repeated under subsequent transitions to Step 3. So, the number of transitions to Step 3 is finite due to the finiteness of the set $\mathbf{I}$ (and consequently, the finiteness of the number of linearly independent coverings Z from $\mathbf{I}$). Consequently, Algorithm 2 is finite.

Since $F(\mathbf{x})$ decreases when $\mathbf{x}$ is modified, $F(\mathbf{x}) \leq F(\mathbf{x}^0)$ for the terminal $\mathbf{x}$. □

Adi Ben-Israel, RUTCOR–Rutgers Center for Operations Research, Rutgers University, 640 Bartholomew Rd, Piscataway, NJ 08854-8003, USA
E-mail address: bisrael@rutcor.rutgers.edu

Genrikh Levin, Operations Research Laboratory, United Institute of Informatics Problems, National Academy of Science of Belarus, Minsk, Belarus
E-mail address: levin@newman.bas-net.by

Yuri Levin, School of Business, Queen's University, Kingston, Ontario, Canada K7L 3N6
E-mail address: ylevin@business.queensu.ca

Boris Rozin, Operations Research Laboratory, United Institute of Informatics Problems, National Academy of Science of Belarus, Minsk, Belarus
E-mail address: rozin@newman.bas-net.by