Control
and Cybernetics

VOL. 17 (1988) No. 4

Maximin resource allocation problem with several constraints

by

WALDEMAR CZUCHRA

Merchant Marine Academy
Gdynia, Poland

This paper presents an extension of Brown's problem for the allocation of a single resource to a given number of variables to maximize the value of the smallest tradeoff function. Instead of single constraint in Brown's problem several number of constraints on sums of resource quantities are considered. The description of algorithms for strictly increasing and continuous tradeoff functions with all continuous and mixed, continuous and integer variables are presented. An illustrative example is included.

1. Introduction

Brown [1] developed the method of resource allocation for the following problem

$$F^*(x_1^*, \ldots, x_N^*) = \max \min_{n \in \mathcal{N}} \{f_n(x_n)\}, \tag{1a}$$

$$\sum_{n \in \mathcal{N}} x_n \leqslant h, \tag{1b}$$

$$x_n \geqslant 0, \quad n \in \mathcal{N}, \quad \mathcal{N} = \{1, 2, \ldots, N\}. \tag{1c}$$

This paper extends Brown's problem for the case of several constraints. Therefore the problem under consideration can be stated as

$$F^*(x_1^*, \ldots, x_N^*) = \max \min_{n \in \mathcal{N}} \{f_n(x_n)\}, \tag{2a}$$

$$\sum_{n \in \mathcal{D}_r} x_n \leqslant h_r, \quad r \in \mathcal{R}, \quad \mathcal{R} = \{1, 2, \ldots, R\}, \tag{2b}$$

$$x_n \geqslant 0, \quad n \in \mathcal{N}. \tag{2c}$$

The meanings of notations in (2a)–(2c) are as follows

1. N is the total number of variables.
2. $\mathscr{N}$ is the set of first N positive integers.
3. R is the total number of constraints.
4. $\mathscr{R}$ is the set of first R positive integers.
5. x_n is the quantity of the resource allocated to variable n.
6. f_n is strictly increasing and continuous tradeoff function.
7. $\mathscr{D}_r \subset \mathscr{N}$ is the set containing the numbers of variables of the constraint r.
8. $h_r > 0$ is the maximum quantity of the resource that can be allocated to variables of $\mathscr{D}_r$.

It has been assumed that $\mathscr{D}_r$ and h_r, $r \in \mathscr{R}$ are defined in such way that no constraint could be replaced by another one or no h_r changed to smaller value (if for example $\mathscr{D}_r \subset \mathscr{D}_q$ and $h_r > h_q$ then h_r could be replaced by h_q).

The problem (2) has been considered by Dutta and Vidyasagar [2] in more general form. They have proposed an algorithm for the problem having nonlinear constraints instead of linear in (2). This means that their method, converting constrained minimax problem to a sequence of unconstrained minimization of least-squares type objective function, can be applied in this case. However a gradient optimization technique has to be applied in their method to do the unconstrained optimization at each step of the sequence. Therefore it is difficult to assesss a computational complexity of that method. This question is important specially for large problems (great number of variables). The method proposed here has polynomial computational complexity, does not require any auxiliary procedure and is very simple to code.

The extension of problem (1) was inspired by the work by Mjelde [4] who considers similar extension of problem solved earlier by Luss and Gupta [3] where the objective functions considered are sums of tradeoff functions. While the method developed by Mjelde requires that $\mathscr{D}_r$, $r \in \mathscr{R}$, form a tree when ordered by the inclusion relation, algorithms presented here allow $\mathscr{D} = \{\mathscr{D}_1, \ldots, \mathscr{D}_R\}$ be any nonredundant collection of sets $\mathscr{D}_r$, $r \in \mathscr{R}$.

As an illustration of the problem (2) we can consider the problem of distribution of funds to increase a degree of environment purity in different regions. We expect to gain $f_n(x_n)$ degrees of purity if fund x_n is allocated to region n, $n \in \mathscr{N}$. The constraints on sums of x_n may be imposed by technological or geographical factors connected with various sets of regions. The aim of a decision-maker is to maximize the smallest magnitude of the degree of environment purity among all regions. Other applicational areas can be easily found, see for example [1], [4].

This paper contains the descriptions of two algorithms: first for the continuous problem (all variables x_n, $n \in \mathcal{N}$ are continuous) and second for the mixed integer problem (some variables x_n, $n \in \mathcal{I}$, $\mathcal{I} \subset \mathcal{N}$, must be positive integers). The numerical example of application of the first algorithm is presented. The paper ends with some final remarks.

2. Continuous problem

Let us consider the problem (2) with all variables x_n, $n \in \mathcal{N}$, being nonnegative real numbers and $R > 1$. It appears that the following theorem (Theorem 1 is stated and proved in the case of one constraint in (1)) holds: Theorem 1. *Let the variables x_n, $n \in \mathcal{N}$, be ordered and then renumbered so that*

$$f_1(0) \leqslant f_2(0) \leqslant \dots \leqslant f_N(0) \leqslant f_{N+1}(0), \tag{3}$$

where an extra variable $N+1$ is introduced and $f_{N+1}(0) = +\infty$. A feasible solution x^* of the problem (2) is optimal solution if and only if there exists an integer $k \in \mathcal{N}$, a real number λ_k and at least one integer $p \in \mathcal{R}$ such that the following conditions are satisfied:

$$x_n^* > 0, \qquad n \in \{1, \dots, k\} = \mathcal{N}_k, \tag{4a}$$

$$f_n(x^*) = \lambda_k, \qquad n \in \mathcal{N}_k, \tag{4b}$$

$$f_n(0) \geqslant \lambda_k, \qquad n \in \{k+1, \dots, N\} = \mathcal{N}_0, \tag{4c}$$

$$x_n^* = 0, \qquad n \in \mathcal{N}_0, \tag{4d}$$

$$\sum_{n \in \mathcal{D}_p} x_n^* = h_p, \tag{4e}$$

Proof:

Assume that x^* is a feasible solution that satisfies conditions (4). Let us consider any feasible solution $\hat{x}$ such that $\hat{x} \neq x^*$. If

$$\sum_{n \in \mathcal{D}_p} \hat{x}_n = h_p,$$

it follows from the relation (4a) that there exists a variable $m \in \mathcal{N}_k$ such that $\hat{x}_m < x_m^*$ (because $\hat{x}_l > x_l^*$ for some $l \in \mathcal{D}_p$, $l \neq m$ implies $\hat{x}_m < x_m^*$). This means that

$$f_m(\hat{x}_m) < f_m(x_m^*),$$

while the equation (4b) requires

$$f_m(\hat{x}_m) = \lambda_k.$$

Inspection of the conditions (4b) and (4c) shows that the optimal value of objective function is λ_k. Thus taking into account last two formulae it is obvious that $\hat{x}$ is not an optimal solution to (2).

Similar argumentation for the case

$$\sum_{n \in \mathscr{D}_p} \hat{x}_n < h_p,$$

shows that $\hat{x}$ is not the optimal solution to (2).
Thus x^* is the optimal solution because non-binding constraints (2b) do not influence the optimality.

Assume now that x^* is the optimal solution to (2). There exists always such an integer $k \in \mathscr{I}$ that

$$f_k(0) \leqslant F^* < f_{k+1}(0),$$

where F^* is the optimal value of objective function. We have to show, that there exists subset of zero allocations, i.e. $x_n^* = 0$ for $n \in \mathscr{I}_0$, where $\mathscr{I}_0 = \{k+1, \ldots, N\}$. This will be demonstrated by contradiction. Let $x_m^* > 0$ if $m \in \mathscr{I}_0$. Hence defining $\hat{x}$ such that

$$\hat{x}_n = \begin{cases} x_n^* + x_m^*/P & n \in \mathscr{I}_k \cap \mathscr{D}_p \\ x_n^* & n \in \mathscr{I}_k \text{ and } n \notin \mathscr{D}_p \\ 0 & n = m \\ x_n^* & n \in \mathscr{I}_0 - \{m\} \end{cases}$$

where $P = |\mathscr{D}_p|$, implies that for $n \in \mathscr{I}_k$

$$\min\{f_n(\hat{x}_n)\} > \min\{f_n(x_n^*)\} = F^*,$$

what contradicts with the assumption, because $\hat{x}$ gives better optimal value than x^*.

Now assume that there exists an integer $s \in \mathscr{I}_k \cap \mathscr{D}_p$, where $|\mathscr{D}_p| \geqslant 2$ (what always will hold if not all $\mathscr{D}_r$, $r \in \mathscr{R}$ are trivial, i.e. at least two-element sets), such that

$$f_s(x_s^*) > F^*.$$

This means, that the optimal value of objective function can be increased by redistribution of some excess resource in allocation x_s^*. Let define

$$\hat{x}_n = \begin{cases} x_n^* - \Delta & n = s, \\ x_n^* + \Delta/(S-1) & n \in \mathscr{I}_k \cap \mathscr{D}_p - \{s\}, \\ 0 & n \in \mathscr{I}_0, \end{cases}$$

where $S = |\mathscr{D}_p|$ and Δ is chosen such that still $f_s(\hat{x}_s) > F^*$. This results in

$$\min\{f_n(\hat{x}_n)\} > F^*,$$

for $n \in \mathscr{I}_k$, what gives contradiction. The next implication is that for $n \in \mathscr{I}_k$

$$f_n(x_n^*) = F^*,$$

what ends the proof. ■

The theorem 1 allows to propose the following solution procedure for problem (2).

Algorithm 1.

Step 1. Define $\mathscr{A} = \{a_n : a_n = f_n(0),\ n \in \mathcal{N}\}$ and $\mathscr{B} = \{b_n : b_n = a_m;\ b_n \leqslant b_{n+1};\ n, m \in \mathcal{N}$ and $b_{N+1} = \infty\}$. Set $k = N$.

Step 2. Calculate for $n \in \mathcal{N}$

$$x'_n = \begin{cases} f_n^{-1}(b_k) & \text{if} \quad f_n^{-1}(b_k) \geqslant 0, \\ 0 & \text{if} \quad f_n^{-1}(b_k) < 0, \end{cases}$$

and for $r \in \mathscr{R}$

$$h'_r = h_r - \sum_{i \in \mathscr{D}_r} x'_i.$$

If $h'_r \geqslant 0$, $r \in \mathscr{R}$, go to Step 3. Otherwise $k = k-1$ and repeat Step 2.

Step 3. Set $k = k+1$. Define $\mathcal{N}_0 = \{n : a_n \geqslant b_k,\ n \in \mathcal{N}\}$, $\mathcal{N}_k = \mathcal{N} - \mathcal{N}_0$ and replace $\mathscr{R}$ by $\mathscr{R} - \Delta\mathscr{R}$, where $\Delta\mathscr{R} = \{r : \mathscr{D}_r \subset \mathcal{N}_0,\ r \in \mathscr{R}\} \cup \{r : \mathscr{D}_r \cap \mathcal{N}_k = \mathscr{D}_q \cap \mathcal{N}_k$ and $h_r > h_q,\ r, q \in \mathscr{R},\ r \neq q\}$. Define $\mathscr{D}'_r = \mathscr{D}_r \cap \mathcal{N}_k$, for $r \in \mathscr{R}$.

Step 4. For each $r \in \mathscr{R}$ determine F'_r from the equation

$$\sum_{n \in \mathscr{D}'_r} f_n^{-1}(F'_r) = h_r.$$

Step 5. Calculate optimal value

$$F^* = \min_{r \in \mathscr{R}} \{F'_r\},$$

and assign the following optimal values to the variables $n \in \mathcal{N}$

$$x_n^* = \begin{cases} f_n^{-1}(F^*) & \text{if} \quad n \in \mathcal{N}_k \\ 0 & \text{if} \quad n \in \mathcal{N}_0 \end{cases}$$

An optimal solution has been found. STOP.

Some comments are necessary:

1. The set $\mathscr{B}$ consists of ordered elements of the set $\mathscr{A}$ with the last additional element being $b_{N+1} = \infty$.
2. The set $\Delta\mathscr{R}$ contains the constraints which can be eliminated because some variables (those of $\mathcal{N}_0$) take zero values in the optimal solution.
3. If explicit expressions $F'_r = g_r(h_r)$, where q_r denotes a given function, are not possible to derive in Step 4, then numerical methods to determine F'_r are needed. Therefore two starting points l_r and u_r that define the interval for F'_r, i.e. $F'_r \in \langle l_r, u_r \rangle$, can be calculated using

$$l_r = \max_{n \in \mathscr{D}'_r} \{f_n(0)\}$$

$$u_r = \min_{n \in \mathscr{D}'_r} \{f_n(h_r)\}$$

4. In general not all constraints are — in the optimal point x_n^*, $n \in \mathcal{N}$, obtained by the algorithm — binding constraints. Define $\mathcal{R}_{\min} = \{r: F_r' = F^*, r \in \mathcal{R}\}$ and

$$\mathcal{N}_{\min} = \bigcup_{r \in \mathcal{R}_{\min}} \mathcal{D}_r.$$

Thus variables x_n, $n \in \mathcal{N}_k - \mathcal{N}_{\min}$ can be increased by $\Delta x_n^* \geqslant 0$ without violating any constraint and without increasing the value of F^*, where for $r \in \mathcal{R} - \mathcal{R}_{\min}$

$$\sum_{n \in \mathcal{D}_r \cap (\mathcal{N}_k - \mathcal{N}_{\min})} \Delta x_n^* = h_r - \sum_{n \in \mathcal{D}_r'} x_n^*.$$

The optimality of the solution x_n^*, $n \in \mathcal{N}$, is settled by the following theorem.

Theorem 2. *Algorithm* 1 *produces the optimal solution to the continuous problem* (2).

Proof.

We have to show that the Algorithm 1 finds the solution x_n^*, $n \in \mathcal{N}$ satisfying the conditions (4) and terminates in finite number of steps.

The first task of the algorithm is to determine an integer k to partition the set $\mathcal{N}$ into two sets $\mathcal{N}_k$ and $\mathcal{N}_0$, such that $\mathcal{N}_k \cup \mathcal{N}_0 = \mathcal{N}$. This is done iteratively, by a process of trial and error starting with $k = N$ and determining x_n' using the formula in Step 2. If all $h_r' \geqslant 0$ the process is stopped because the optimal value k has been found and set $\mathcal{N}$ can be partitioned. Next, zero variables can be eliminated from the sets $\mathcal{D}_r$, $r \in \mathcal{R}$ (this is done in Step 3). Owing to this the sets $\mathcal{D}_r'$ contain only nonzero variables. Assuming temporarily that all constraints (2b) are binding constraints we find in Step 4 at least one potential objective optimal values. Minimum of these values is in fact the optimal value F^*. This allows to find the optimal allocations x_n^*, $n \in \mathcal{N}$ (Step 5).

The way in which k, F^* and x^* are determined ensures that the optimal solution satisfies (4).

The number of computations in each step of the algorithm is bounded from above by $N \log N$ in step 1, NR in step 2, max $\{R^2, N\}$ in step 3, R in step 4 and N in step 5. This means that the number of computations in the whole algorithm is finite, what ends the proof. ■

The final considerations in the above proof enable to evaluate the computational complexity of Algorithm 1 as equal to $O(N, Q)$, where $Q = \max\{\log N, R^2\}$.

Computations performed by Algorithm 1 can be ilustrated by the following example with $N = R = 4$, $f_1(x_1) = .5 + .5ln(x_1 + 2)$, $f_2(x_2) = 1 + .5ln(x_2 + 1)$, $f_3(x_3) = 1 + ln(x_3 + 3)$, $f_4(x_4) = 1 + 2ln(x_4 + 2)$ and constraints $x_1 + x_2 + x_3 + x_4 \leqslant 3$, $x_1 + x_2 \leqslant 2$, $x_1 + x_3 + x_4 \leqslant 2$, $x_2 \leqslant 1$, $x_n \geqslant 0$ for $n \in \mathcal{N}$.

Sets $\mathcal{A}$ and $\mathcal{B}$ obtained in Step 1 are $\mathcal{A} = \{.846,\ 1,\ 2.098,\ 2.386\}$, $\mathcal{B} = \{.846,\ 1,\ 2.098,\ 2.386,\ \infty\}$. For $k=4$ calculations in Step 2 result in $x'_1 = 41.492$, $x'_2 = 15$, $x'_3 = 1$; $x'_4 = 0$ what gives $h'_1 = -54.492$, $h'_2 = -54.492$, $h'_3 = -40.492$, $h'_4 = -14$. This causes that Step 2 is repeated for $k=3$. Algorithm may enter Step 3 after once more repetition of Step 2 because finally for $k=2$ we obtain $x'_1 = .718$, $x'_2 = x'_3 = x'_4 = 0$ and all $h'_r > 0$, $r = 1, 2, 3, 4$.

Application of Step 3 gives $\mathcal{N}_0 = \{3, 4\}$, $\mathcal{N}_k = \{1, 2\}$, $\Delta\mathcal{R} = \{1\}$, $\mathcal{R} = \{2, 3, 4\}$, $\mathcal{D}'_2 = \{1, 2\}$, $\mathcal{D}'_3 = \{1\}$, $\mathcal{D}'_4 = \{2\}$. The values of F'_r calculated in Step 4 are $F'_2 = 1.148$, $F'_3 = 1.193$, $F'_4 = 1.346$, and consequently, Step 5 gives optimal solution $F^* = 1.148$ for allocations $x^*_1 = 1.655$, $x^*_2 = .345$, $x^*_3 = x^*_4 = 0$.

3. Mixed integer problem

The algorithm presented below for the problem (2) in which some variables x_n, $n \in \mathcal{I}$ and $\mathcal{I} \subset \mathcal{N}$ are positive integers and some x_n, $n \in \mathcal{C}$, $\mathcal{I} \cup \mathcal{C} = \mathcal{N}$, $\mathcal{I} \cap \mathcal{C} = \emptyset$ are nonnegative reals differs slightly in first two steps from the original algorithm developed by Brown [1]. The difference in Step 1 is that Algorithm 1 is applied instead of Brown's algorithm. The difference in Step 2 is caused by different number of constraints in problem (1) and (2). Therefore the sets $\mathcal{U}_r$, $r \in \mathcal{R}$, are introduced. The rest of calculations are the same or analogical and can be easily explained.

Algorithm 2.

Step 1. Allowing all x_n, $n \in \mathcal{I}$ be nonintegers solve problem (2) using Algorithm 1. Let x_n for $n \in \mathcal{I} \cup \mathcal{C}$ represent continuous solution.

Step 2. Set

$$F' = \min_{n \in \mathcal{I}} f_n(\lfloor x_n \rfloor),$$

where $\lfloor x \rfloor$ is the largest integer not greater than x. Let $\mathcal{S}$ contain the variable numbers n, $n \in \mathcal{I}$ such that $f_n(\lfloor x_n \rfloor) = F'$.

Calculate

$$x'_n = \begin{cases} \lceil f_n^{-1}(F') \rceil & \text{if} \quad n \in \mathcal{I} \\ f_n^{-1}(F') & \text{if} \quad n \in \mathcal{C} \end{cases} \tag{5}$$

where $\lceil x \rceil$ is the smallest integer not less than x. Define for $r \in \mathcal{R}$ sets $\mathcal{U}_r = \{n : n \in \mathcal{I} \cap \mathcal{D}_r \text{ and } f_n(x'_n) = F'\}$ and $\mathcal{U} = \bigcup_{r \in \mathcal{R}} \mathcal{U}_r$. Calculate for $r \in \mathcal{R}$

$$h'_r = \sum_{n \in \mathcal{I} \cap \mathcal{D}_r} x'_n + \sum_{n \in \mathcal{C} \cap \mathcal{D}_r} x'_n. \tag{6}$$

If $h'_r+|\mathscr{U}_r| > h_r$, where $|\mathscr{U}_r|$ means the number of elements of $\mathscr{U}_r$, for at least one $r\in\mathscr{R}$, then go to Step 3. If $h'_r+|\mathscr{U}_r| \leqslant h_r$ for all $r\in\mathscr{R}$, $h'_r+|\mathscr{U}_r| = h_r$ for at least one $r\in\mathscr{R}$ and $\mathscr{C} = \emptyset$, then go to Step 4. Otherwise go to Step 5.

Step 3. Set $x^*_n = x'_n$ for all $n\in\mathscr{I}\cup\mathscr{C}$. STOP.

Step 4. Set $x^*_n = x'_n+1$ for all $n\in\mathscr{U}$ and $x^*_n = x'_n$ for $n\in\mathscr{N}-\mathscr{U}$. STOP.

Step 5. Set $x^*_n = x'_n+1$ for $n\in\mathscr{S}$ and replace h_r by $h_r-\sum_{n\in\mathscr{S}} x^*_n$ for $r\in\mathscr{R}$ and $\mathscr{I}$ by $\mathscr{I}-\mathscr{S}$. If $\mathscr{I} = \emptyset$ then STOP. Otherwise go to Step 1.

The calculations performed in and features of the solutions produced by the Algorithm 2 are discussed in the proof of the following theorem.

Theorem 3. *Algorithm 2 finds the optimal solution x^*_n, $n\in\mathscr{N}$ to the mixed integer problem* (2).

Proof. (Analogous to the proof for single constraint mixed integer problem (1)).

Calculations performed in Step 1 and in the beginning of Step 2 are aimed at determining lower and upper bounds for the optimal value of the objective function. If the integer variables from the set $\mathscr{I}$ are allowed to be continuous, then the continuous solution to the appropriate continuous problem gives the value of objective function $\mathring{F}$, which is upper bound to the optimal value F^* of objective function of mixed integer problem. A feasible solution to the integer problem can be obtained from the continuous solution by dropping fractional parts of those variables which belong to $\mathscr{I}$. Thus, value F' computed in the step 2 is the lower bound to the optimal value of the integer problem, i.e. $F' \leqslant F^* \leqslant \mathring{F}$. But we desire to have value F^* being so close to $\mathring{F}$ as it is possible without no violating any constraints of the problem. Hence, if the optimal value of the objective function for the integer problem would have been greater than F', then all variables $n\in\mathscr{S}$ should be equal to $\lfloor x_n\rfloor+1$. But $f_n(\lfloor x_n\rfloor)+1$ is greater than $\mathring{F}$. So we have to check if the value of variables from the set $\mathscr{S}$ should be $\lfloor x_n\rfloor$ or $\lfloor x_n\rfloor+1$. Using (5) we determine x'_n for $n\in\mathscr{I}$ such that it is possible to attain more than F'. But this way computed solution x'_n, $n\in\mathscr{N}$ may not be feasible. That is why we next calculate the smallest possible sums of allocations so that all integer constraints are satisfied and the value of objective function is at least F' for $n\in\mathscr{N}$. These sums are denoted h'_r in the algorithm and are determined for all constraints $r\in\mathscr{R}$, using the formula (6). The set $\mathscr{U}_r$, $r\in\mathscr{R}$ consists of those variables of $\mathscr{I}$ for which $f_n(x'_n) = F'$ and $\mathscr{U}_r \subset \mathscr{D}_r$. To obtain a solution with the objective function greater than F', each variable in $\mathscr{U} = \bigcup_{r\in\mathscr{R}} \mathscr{U}_r$ has to be increased by 1. This will result in increase of the sum of allocations from the value h'_r to the value $h'_r+|\mathscr{U}_r|$, $r\in\mathscr{R}$. Thus, $h'_r+|\mathscr{U}_r|$ are the smallest sums possible such that all variables $n\in\mathscr{I}$ can be set so that their tradeoff function values are greater than F', while all continuous variables $n\in\mathscr{C}$ have tradeoff function values equal to F', i.e. $f_n(x'_n) = F'$ for all $n\in\mathscr{C}$.

Hence if there exists at least one constraint $r \in \mathcal{R}$ such that $h'_r + |\mathcal{U}_r| > h_r$ then $x^*_n = x'_n$, $n \in \mathcal{N}$ is an optimal solution because it is not possible to gain more than F' (see Step 3). If $h'_r + |\mathcal{U}_r| = h_r$ for at least one $r \in \mathcal{R}$, while for all $r \in \mathcal{R}$ the inequalities $h'_r + |\mathcal{U}_r| \leqslant h_r$ hold and the set of continuous variables is not empty $\mathcal{C} \neq \emptyset$, then $x^*_n = x'_n$, $n \in \mathcal{N}$ is also optimal solution, because the continuous variables $n \in \mathcal{C}$ cannot be increased, without violating constraints, so that their tradeoff function values are greater than F', despite the fact that the integer variables can be increased to obtain greater function values than F'. Of course, if $\mathcal{C} \neq \emptyset$ and $h'_r + |\mathcal{U}_r| = h_r$, then the variables from the set $\mathcal{U}$ can be increased by 1. Thus the optimal solution is $x^*_n = x'_n + 1$ for $n \in \mathcal{U}$ and $x^*_n = x'_n$ for $n \in \mathcal{N} - \mathcal{U}$ (Step 4). If none of these cases is valid it means that $h'_r + |\mathcal{U}_r| < h_r$ for all $r \in \mathcal{R}$ and obviously the objective function is greater than F' and optimal values of variables in $\mathcal{S}$ are $x'_n + 1$. This means that variables from $\mathcal{S}$ can be eliminated from the old problem and the limits h_r on sum of allocations can be reduced (Step 5). This new problem has less number of integer variables. The whole solution procedure is repeated for new problems formed this way until the set $\mathcal{I}$ is empty. By this process all optimal allocations x^*_n, $n \in \mathcal{N}$ are obtained one after the other. The process is finite because the set $\mathcal{I}$ is finite and each iteration removes at least one variable from $\mathcal{I}$.

We have shown that the algorithm is finite and determines the optimal allocation x^*_n, $n \in \mathcal{N}$ with the optimal value of the problem being

$$F^* = \min_{n \in \mathcal{N}} \{f_n(x^*_n)\},$$

such that $F^* \leqslant \mathring{F}$. This ends the proof. ■

The computational complexity of the Algorithm 2 is easily evaluated using the observation that is depends on the maximum number of Algorithm 1 calls. This number is at most N. Thus the computational complexity of Algorithm 2 is $O(N^2 Q)$ because all other steps of this algorithm are $O(N)$ or $O(R)$. It is worthwhile to note that is computational complexity does not depend on $\max\{h_r\}$, $r \in \mathcal{R}$.

4. Concluding remarks

Brown [1] considers more types of functions f_n. For linear case he develops a little bit simpler algorithm than for non-linear case because the solution of the equation in Step 4 can be derived as a closed-form expression. Since Algorithm 1 can be also applied to linear functions the modification of Linear Algorithm [1] is omitted herein.

The modifications of Algorithm 1 and 2 for piecewise linear functions, piecewise nonlinear functions and any functions will be obvious when reader confronts Brown's paper.

References

[1] Brown J. R. The knapsack sharing problem. *Opns Res.*, **27** (1979), 341–355.

[2] Dutta S. R. K., Vidyasagar M. New algorithms for constrained minimax optimization. *Math. Prog.*, **13** (1977), 140–155.

[3] Luss H., Gupta S. Allocation of effort resources among competing activities. *Opns. Res.*, (**23**) 1975, 360–366.

[4] Mjelde K. M. Resource allocation with tree constraints. *Opns Res.*, (**31**) 1983, 881–890.

[5] Mjelde K. M. Max-min resource allocation. *BIT*, (**23**) 1983, 529–537.

Received, October 1987.

Acknowledgment

The author would like to thank the unknown referee for his careful revision and comments which improved the earlier version of the paper.

Minimaksowe zadanie rozdziału zasobów z wieloma ograniczeniami

W pracy przedstawiono rozwinięcie problemu Browna rozdziału zasobów pomiędzy określoną ilość zmiennych w celu maksymalizacji najmniejszej z wartości funkcji celu odpowiadających tym zmiennym. Brown rozważa przypadek z jednym ograniczeniem na sumę ilości przydzielonego zasobu. Natomiast algorytmy opisane w pracy dopuszczają dowolną skończoną liczbę tego typu ograniczeń. Opracowano je dla zmiennych ciągłych oraz dyskretnych. Załączono przykład ilustrujący jeden z opisanych algorytmów.

Минимаксная задача распределения ресурсов со многими ограничениями

В работе представлено развитие задачи Брауна распределения ресурсов между определенным числом переменных с целью максимизации наименьшей из величин функции цели, соответствующих этим переменным. Браун рассматривает случай с одним ограничением по сумме количества отведенных ресурсов. В свою очередь алгоритмы, описанные в работе, допускают произвольное конечное число этого типа ограничений. Они разработаны для непрерывных и дискретных переменных. Прилагается пример, иллюстрирующий один из описанных алгоритмов.