Control
and Cybernetics
VOL. 18 (1989) No. 1

Maximin resource allocation problem with several constraints

by

WALDEMAR CZUCHRA

Merchant Marine Academy
Gdynia, POLAND

This paper presents an extension of the 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 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

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

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

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

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

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

$$\sum_{n\in D_r} x_n \leqslant h_r,\ r \in R,\ r = \{1,2,\ldots,R\} \tag{2b}$$

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

The meanings of parameters in (2a)—(2c) are as follows

1. N is the total number of variables.
2. N is the set of first N positive integers.
3. R is the total number of constraints.
4. 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 contiuous tradeoff function.
7. $D_r \subset N$ 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 D_r.

It has been assumed that D_r and h_r, $r \in R$ 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 $D_r \subset D_q$ 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 a more general form. They have proposed an algorithm for the problem having nonlinear constraints instead of linear ones in (2). This means that their method, converting constrained minimax problem to a sequence of unconstrained minimization of a least-sqaures 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 assess the computational complexity of that method. This question is important especially 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 of 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 D_r, $r \in R$, form a tree when ordered by the inclusion relation, algorithms presented here allow $D = \{D_1,...,D_R\}$ be any nonredundant collecion of sets D_r, $r \in R$.

As an illustration of the problem (2) we can consider the problem of distribution of funds to increase the degree of environmental purity in different regions. It is supposed that a gain of $f_n(x_n)$ in terms of degree of purity arises when x_n is allocated to region n, $n \in N$. 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 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, $n \in N$ are continuous) and second for the mixed integer problem (some variables x_n, $n \in J$, $J \subset N$, must be positive integers). The numerical example of application of the first algorithm is presented. The paper ends with some final remarks.

2. The continuous problem

Let us consider the problem (2) with all variables x_n, $n \in N$, being nonnegative real numbers and R $>$ 1. It turns out that the following theorem (Theorem 1 is stated and proved for the case of one constraint in [1]) holds:

THEOREM 1. *Let the variables x_n, $n \in N$, be ordered and then renumbered so that*

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

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

$$x_n^* > 0 \qquad n \in \{1,\ldots,k\} = N_k \tag{4a}$$

$$f_n(x^*) = \lambda_k \qquad n \in N_k \tag{4b}$$

$$f_n(0) \geqslant \lambda_k \qquad n \in \{k+1,\ldots,N\} = N_0 \tag{4c}$$

$$x_n^* = 0 \qquad n \in N_0 \tag{4d}$$

$$\sum_{n \in D_p} x_n^* = h_p \tag{4e}$$

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

$$\sum_{n \in D_p} \hat{x}_n = h_p$$

it follows from the relation (4a) that there exists a variable $m \in N_k$ such that $\hat{x}_m < x_m^*$ (because $\hat{x}_l > x_l^*$ for some $l \in D_p$, $l \neq m$ 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 the last two formulae it is obvious that $\hat{x}$ is not an optimal solution to (2).

Similar argumentation for the case

$$\sum_{n \in D_p} \hat{x}_n < h_p$$

shows that $\hat{x}$ is not an 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 $k \in N$ 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 a subset of zero allocations, i.e. $x_n^* = 0$ for $n \in N_0$, where $\mathrm{N}_0 = \{k+1, ..., \mathrm{N}\}$. This will be demonstrated by contradiction. Let $x_m^* > 0$ if $m \in N_0$. Hence, definition of $\hat{x}$ such that

$$\hat{x}_n = \begin{cases} x_n^* + x_m^*/P & n \in N_k \cap D_p \\ x_n^* & n \in N_k \text{ and } n \notin D_p \\ 0 & n = m \\ x_n^* & n \in N_0 - \{m\} \end{cases}$$

where $P = |D_p|$, implies that for $n \in N_k$

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

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

Assume that there exists an integer $s \in N_k \cap D_p$, where $|D_p| \geqslant 2$ (what will always hold if not all D_r, $r \in R$ are trivial, i.e. are at least two-element sets), such that

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

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

$$\hat{x}_n = \begin{cases} x_n^* - \Delta & n = s \\ x_n^* + \Delta/(S-1) & n \in N_k \cap D_p - \{s\} \\ 0 & n \in N_0 \end{cases}$$

here $S = |D_p|$ and Δ is chosen so that there is $f_s(\hat{x}_s) > F^*$. This results in

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

for $n \in n_k$, which leads to contradiction. The next implication is that for $n \in N_k$

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

ending the proof.

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

Algorithm 1.

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

STEP 2. Calculate for $n \in N$

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

and for $r \in R$

$$h_r' = h_r - \sum_{i \in D_r} x_i'$$

If $h_r' \geqslant 0$, $r \in R$, go to Step 3. Otherwise $k = k - 1$ and repeat Step 2.

STEP 3. Set $k = k + 1$. Define $N_0 = \{n : a_n \geqslant b_k, n \in N\}$, $N_k = N - N_0$ and replace R by $R - \Delta R$, where $\Delta R = \{r : D_r \subset T_0, r \in R\} \cup \{r : D_r \cap N_k = D_q \cap N_k \text{ and } h_r > h_q, r, q \in R, r \neq q\}$.
Define $D_r' = D_r \cap N_k$, for $r \in R$.

STEP 4. For each $r \in R$ determine F_r' from the equation

$$\sum_{n \in D_r'} f_n^{-1}(F_r') = h_r$$

STEP 5. Calculate optimal value

$$F^* = \min_{r \in R} \{F_r'\}$$

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

$$x_n^* = \begin{cases} f_n^{-1}(F^*) & \text{if } n \in N_k \\ 0 & \text{if } n \in N_0 \end{cases}$$

An optimal solution has been found. STOP.

Some comments are necessary:

1. The set B consists of ordered elements of the set A with the last additional element being $b_{N+1} = \infty$.
2. The set ΔR contains the constraints which can be eliminated because some variables (those of N_0) take zero values in the optimal solution.
3. If explicit expressions $F_r = g_r(h_r)$, where g_r denotes a given function, cannot be derived 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

$$l_r = \max_{n \in D_r'} \{f_n(0)\}$$

$$u_r = \min_{n \in D_r'} \{f_n(h_r)\}$$

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

$$N_{\min} = \bigcup_{r \in R_{min}} D_r$$

Thus variables x_n, $n \in N_k - N_{min}$ can be increased by $\Delta x_n^* \geqslant 0$ without violation of any constraint and without an increase of the value of F^*, where for $r \in R - R_{\min}$

$$\sum_{n \in D_r \cap (N_k - N_{\min})} \Delta x_n^* = h_r - \sum_{n \in D_r'} x_n^*$$

The optimality of the solution x_n^*, $n \in N$, is established by the following theorem.

THEOREM 2. *Algorithm* 1 *gives the optimal solution to the continuous problem* (2).

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

The first task of the algorithm is to determine an integer k so as to partition the set N into two sets N_k and N_0, such that $N_k \cup N_0 = N$. This is done iteratively, by a process of trial and error starting with $k = N_0$ and determinimg 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 N can be partitioned. Next, zero variables can be eliminated from the sets D_r, $r \in R$ (this is done in Step 3). Owing to this the sets D_r contain only nonzero variables. Assuming temporarily that all constraints (2b) are binding constraints we find in Step 4 at least one of the potential objective optimal values. Minimum of these values is in fact the optimal

value F^*. This allows finding of the optimal allocations x_n^*, $n \in N$ (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, which ends the proof.

The final considerations in the above proof make it possible 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 illustrated by the following example with $N = R = 4$, $f_1(x_1) = .5 + .5 \ln(x_1 + 2)$, $f_2(x_2) = = 1 + .5 \ln(x_2 + 1)$, $f_3(x_3) = 1 + \ln(x_3 + 3)$, $f_4(x_4) = 1 + 2\ln(x_4 + 2)$ and constraints $x_1 + x_2 + x_3 + x_4 \leqslant 3$, $x_1 + x_2 \leqslant 2$, $x_1 + x_3 + x_4 \leqslant 2$, $x_2 \leqslant 1$, $x_n \geqslant x_n \geqslant 0$ for $n \in N$.

Sets A and B obtained in Step 1 are $A = \{.846,1,2.098,2.386\}$, $B = \{.846,1,2.098,2.386,\infty\}$. 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 a yet another 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 $N_0 = \{3,4\}$, $N_k = \{1,2\}$, $\Delta R = \{1\}$, $R = \{2,3,4\}$, $D_2' = \{1,2\}$, $D_3' = \{1\}$, $D_4' = \{2\}$. The values of F_r' calculated in Step 4 are $F_2' = 1.148$, $F_3' = 1.346$, 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. The mixed integer problem

The algorithm presented below for the problem (2) in which some variables x_n, $n \in J$ and $J \subset N$ are positive integers and some x_n, $n \in C$, $J \cup C = N$, $J \cap C = \emptyset$ 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 constrains in problem (1) and (2). Therefore the sets M_r, $r \in R$, are introduced. The rest of calculations are the same or analogical and can be easily explained.

Algorithm 2.

STEP 1. Allowing all x_n, $n \in J$ be nonintegers solve the problem (2) using Algorithm 1. Let x_n for $n \in J \cup C$ represent continuous solution.

STEP 2. Set

$$F' = \min_{n \in J} f_n([x_n])$$

where $[x]$ is the largest integer not greater than x. Let S contain the variable numbers n, $n \in J$ such that $f_n([x_n]) = F'$.

Calculate

$$x_n' = \begin{cases} [f_n^{-1}(F')] & \text{if} \quad n \in J \\ f_n^{-1}(F') & \text{if} \quad n \in C \end{cases} \tag{5}$$

where $[x]$ is the smallest integer not less than x. Define for $x \in R$ sets $M_r = \{n: n \in J \cap D_r \text{ and } f_n(x_n') = F'\}$ and $M = \bigcup_{r \in R} M_r$. Calculate for $r \in R$

$$h_r' = \sum_{n \in J \cap D_r} x_n' + \sum_{n \in C \cap D_r} x_n' \tag{6}$$

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

STEP 3. Set $x_n^* = x_n'$ for all $n \in J \cup C$. STOP.

STEP 4. Set $x_n^* = x_n' + 1$ for all $n \in M$ and $x_n^* = x_n'$ for $n \in N - M$. STOP.

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

The calculations performed in and the 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^*, $n \in N$ *to the mixed integer problem* (2).

Proof. (Analogous to the proof for the 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 J are allowed to be continuous, then the continuous solution to the appriopriate continuous problem gives the value of objective function $\overset{\circ}{F}$ which is the upper bound to the optimal value F^* of the objective function of the 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 J. Thus, value of F' computed in the step 2 is the lower bound to the optimal value of the integer problem, i.e. $F' \leqslant F^* \leqslant \overset{\circ}{F}$ But we desire to have the value of F^* being as close to $\overset{\circ}{F}$ as possible without violation of any constraint 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 $n \in S$ should be equal to $[x_n] + 1$. But $f_n([x_n]) + 1$ is greater than $\overset{\circ}{F}$. Hence, we have to check whether the value of variables from the set S should be $[x_n]$ or $[x_n] + 1$. Using (5) we determine x_n' for $n \in J$ such that it is possible to attain more than F'. But solution x_n', $n \in N$ computed this way may not be feasible. That is why we calculate next the smallest possible sums of allocations with all integer constraints being satisfied and the value of objective function being at least F' for $n \in N$. These sums are denoted h_r' in the algorithm and are determined for all constraints $r \in R$, using the formula (6). The set M_r. $r \in R$ consists of those variables of J for which $f_n(x_n') = F'$ and $M_r \subset D_r$. To obtain a solution with the objective function greater than F', each variable in $M = \bigcup_{r \in R} M_r$ has to be increased by 1. This will result in an increase of the sum of allocations from the value h_r' to the value $h_r' + |M_r|$, $r \in R$. Thus, $h_r' + |M_r|$ are the smallest sums possible such that all variables $n \in J$ can be set so that their tradeoff function values are greater than F', while all continuous variables $n \in C$ have tradeoff function values equal to F', i.e. $f_n(x_n') = F'$ for all $n \in C$. Hence if there exists at least one constraint $r \in R$ such that $h_r' + |M_r| > h_r$ then $x_n^* = x_n'$, $n \in N$ is an optimal solution because it is not possible to gain more than F' (see Step 3). If $h_r' + |M_r| = h_r$ for at least one $r \in R$, while for all $r \in R$ the inequalities $h_r' + |M_r| \leqslant h_r$ hold and the set of continuous variables is not empty $C \neq \emptyset$, then $x_n^* = x_n'$, $n \in N$ is also an optimal solution, because the continuous variables $n \in C$ 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 function values greater than F'. Of course, if $C = 0$ and $h_r' + |M_r| = h_r$, then the variables from the set M can be increased by 1. Thus the optimal solution is $x_n^* = x_n' + 1$ for $n \in M$ and $x_n^* = x_n'$ for $n \in N - M$ (Step 4). If none of these cases is valid it means that $h_r' + |M_r| < h_r$ for all $r \in R$ and obviously the objective function is greater than F' and optimal values of variables in S are $x_n' + 1$. This means that variables from 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 a lower number of integer variables. The whole solution procedure is repeated for the new problems formed this way until the set J is empty. By this process all optimal allocations x_n^*, $n \in N$ are obtained one after another. The process is finite because the set J is finite and each iteration removes at least one variable from J.

We have shown that the algorithm is finite and that it determines the optimal allocation x_n^*, $n \in N$ with the optimal value of the problem being

$$F^* = \min_{n \in N} \{f_n(x_n^*)\}$$

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

The computational complexity of the Algorithm 2 is easily evaluated using the observation that it 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 0 (N) or 0 (R). It is worth while to note that its computational complexity does not depend on max $\{h_r\}$, $r \in R$.

4. Concluding remarks

Brown [1] considers more types of functions f_n. For the linear case he develops an algorithm a little bit simpler than for the 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 other such 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, September 1988.

Acknowledgment

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

Minimaksowe zadanie rozdziału zasobów z wieloma ograniczeniami

W pracy przedstawiono rozwinięcie problemu Browna rozdziału zasobów pomiędzy określoną liczbę 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 колличествам переменных с целью максимизации из значений функции цели, соответствующих этим переменным. Браун рассматривает случай с одним ограничением по сумме количества предоставленного ресурса. В свою очередь алгоритмы, описанные в работе, допускают произвольное конечное число этого типа ограничений. Они разработаны для непрерывных и дискретных переменных. Представлен пример иллюстрирующий один из описанных алгоритмов.