WORKING PAPER

Problems of Multiobjective Mathematical Programming and the Algorithms of their Solution

Victor Volkovich

August 1989
WP-89-57

IIASA
International Institute
for Applied Systems Analysis

Problems of Multiobjective Mathematical Programming and the Algorithms of their Solution

Victor Volkovich

August 1989
WP-89-57

Glushkov Institute of Cybernetics of the Ukr. SSR Academy of Sciences, Kiev
U.S.S.R.

Working Papers are interim reports on work of the International Institute for Applied Systems Analysis and have received only limited review. Views or opinions expressed herein do not necessarily represent those of the Institute or of its National Member Organizations.

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

Foreword

Development of interactive Decision Support Systems requires new approaches and numerical algorithms for solving Multiple Objective Optimization Problems. These algorithms must be robust and efficient and applicable to possibly a broad class of problems. This paper presents the new algorithm developed by the author. The algorithm consists of two steps: (a) reduction of the initial Multiple Objective Optimization Problem into a system of inequalities, and (b) solving this set of inequalities by the iterative procedure proposed by the author. Due to its generality, the algorithm applies to various Multiple Criteria Optimization Problems, including integer optimization problems. The author presents several variants of the algorithm as well as results of numerical experiments.

Prof. A.B. Kurzhanski
Chairman,
Systems and Decision Sciences Program

Table of Contents

Page

Problems of Multiobjective Mathematical Programming and the Algorithms of their Solution

Victor Volkovich

Glushkov Institute of Cybernetics
of the Ukr. SSR
Academy of Sciences
Kiev
U.S.S.R.

1. A General Approach to Solving Multiobjective Programming Problems

Let it be necessary to choose a certain decision $x \in D_0$ by vector criterion $f = \{f_i(x)\}$, $i \in I$ where D_0 is the region of admissible solutions from which the choice is to be made.

$I = \{\overline{1,M}\}$ is a set of indices of objective functions and $I_1 = I_1 \cup I_2$ (I_1 is the index of objective functions which are being maximized; I_2 is the index of objective functions which are being minimized).

It is well-known that the solution of multiobjective programming can not give optimum each objective function and must be a compromise solution. For definition a compromise solution is necessary to execute heuristic procedures in originally staging of problems. There are different ways for that; we will consider one of them [1 & 2]. Let execute the next two heuristic procedures. First, to introduce transformations of the objective functions, which permit of comparing them with each other, of the following form

$$w_i(f_i(x)) = \begin{cases} \dfrac{f_i^0 - f_i(x),}{f_i^0 - f_{i\,\min}}, \forall_i \in I_1 \\[2ex] \dfrac{f_i(x) - f_i^0}{f_{i\,\max} - f_i^0}, \forall_i \in I_2 \end{cases} \tag{1}$$

It is first necessary, of course, first for each objective function optimal f_i^0 $\forall_i \in I_2$ and worst $f_{i\,\min}$, $\forall_i \in I_1$ and $f_{i\,\max}$, $\forall_i \in I_2$ values are calculated separately over the feasible region D_0. The transformation w_i measures the degree to which the i-th objective function value departs from the ideal value towards the worst feasible value for this function. We will call $w_i(f_i(x))$ the relative loss function for the i-th objective calculated over the feasible region. The dimensional space defined by all the relative loss function $w_i(f_i(x))$ will be designated as W. Second of them to introduce the preference on a set criterion functions in the numerical scale with the help of weighting coefficients $\rho \in Q^+ = \{\rho_i : \rho_i > 0, \forall_i \in I, \sum_{i \in I} \rho_i = 1\}$ particular, if the DM indicates this preference for assigning desirable values to each objective function $f_i^* \in [f_i^0 , f_{i\,\max}]$ if $i \in I_1$ and $f_i^* \in [f_{i\min} , f_1^0]$, if $f_i \in I_2$, the preference can be calculated by the following expression [1]:

$$\rho_i = \frac{\underset{\substack{j \in I \\ j \neq i}}{M} w_j^*}{\sum\limits_{q \in i} \prod\limits_{\substack{j \in I \\ j \neq q}} w_j^*} , \tag{2}$$

where w_j^* defines value $f_j^* \, \forall_i \in I$ in space W of transforming objective functions, and point $f^* = \{f_i^* , \forall_i \in I\}$ define point $w^* = \{w_i^* = w_i(f_i^*), \ \forall_i \in I\}$ in space W.

These heuristic procedures allow us to define a decision to the vector optimization problem to mean that compromise decision which belongs to the no-inferior set and lies in the direction defined by the section $\rho \in Q^+$ in space transforming objective function W.

Theorem 1. For solution $x_0 \in D_0$, such that $w_i(x_0) > 0$ for every $i \in I$, to be a non-inferior solution, it is sufficient for x_0 to be the only solution to the system of inequalities.

$$\rho_i w_i(x_0) \geq k_0 \text{ for every } i \in I \tag{3}$$

for the minimal value k^* of parameter k_0 for which this system is consistent.

Proof: Suppose that the opposite is true, i.e., that the only solution x_0 of the system of inequalities (1.14) where $k_0 = k^*$, is not an efficient solution. Then there must exist an alternative $x^1 \in 0$ such that $w_i(x^1) \leq \ w_i(x_0^*)$, for every $i \in I$, with at least one of these inequalities holding strictly. Multiplying these inequalities by $\rho_i > 0$, for every $i \in I$, we obtain $\rho_i w_i(x^1) \leq \rho_i w_i(x_0^*) \leq \ k_0^*$ with at least one of the left hand inequalities holding strictly. This implies that x^1 satisfies (3) for $k_0 = k_0^*$. But this contradicts the uniqueness of solution x_0. Therefore, x_0 must be a non-inferior solution.

This theorem forms the theoretical basis for the method of searching a compromise decision of multiobjective programming problems. The method starts by transforming the original problem into a system of inequalities consisting of (3) and the feasible region expressed as inequalities in space W. Then parameter $k_0 \in (0,1/M)$ is successively reduced and the system of inequalities is checked for consistency. The process of reducing k_0 and checking the inequalities for consistency continues until the inequalities are found to be inconsistent. Suppose that this occurs at the (l+1)th step. If $k_0^{(l)} - k_0^{(l+1)} \leq \epsilon$, where $\epsilon \geq 0$, then the procedure stops. (ϵ is chosen beforehand out of consideration of an acceptable solution time for the problem.) If $k_0^{(l)} - k_0^{(l+1)} > \epsilon$, the procedure continues with $k_0^{(l+2)}$ being chosen such that $k_0^{(l+1)} < k_0^{(l+2)} < k_0^l$. On finding a $k_0^{(l+j)}$ such that

$$k_0^l - k_0^{(l+j)} \leq \epsilon$$

the inequalities are solved for $k_0 = k_0^{(l+j)}$. If there is a unique solution x_0^*, then it is the best compromise alternative. At x_0^* all the weighted relative losses $\rho_i w_i(x_0^*)$ are exactly equivalent. If however the solution is not unique, then some global criteria, combining all the relative loss function, for example,

$$w(x) = \sum_{i \in I} \rho_i w_i(x) \tag{4}$$

must be optimized over the system of inequalities with $k_0 = k_0^{(l+j)}$. The result will be a unique compromise solution x_0^* for which all the weighted relative losses $\rho_i w_i(x_0^*)$ will be equivalent to within ϵ.

If we again return to the original objective function, to inequalities (3) is rewritten to the following form:

$$\begin{cases} f_i(x) \geq f_i^0 - \dfrac{k_0}{P_i}(f_i^0 - f_{i\,\mathrm{min}}), \ \forall_i \in I_1 \\ f_i(x) \leq f_i^0 + \dfrac{k_0}{P_i}(f_{i\,\mathrm{max}} - f_i^0), \ \forall_i \in I_2 \\ x \in D_0 \end{cases} \tag{5}$$

As before, we must look for a unique solution x^k for which the system of inequalities (5) will be consistent for minimal value parameter k_0. If the solution is not unique, we must indicate higher global criteria (4), which can be rewritten in the original objective function in the following form:

$$W(x) = \sum_{i \in I_1} \rho_i \frac{f_i^0 - f_i(x)}{f_i^0 - f_{i\,\mathrm{max}}} + \sum_{i \in I_2} \rho_i \frac{f_i(x) - f_i^0}{f_{i\,\mathrm{max}} - f_1^0} \tag{6}$$

As far as the parameter k_0 limits the value of objective function, we will name this method the Method of Constraints. The solution procedure component is a generally applicable method for solving the reformulated problem by checking the consistency of the inequalities which comprise its constraint set.

There is a different way we can find indicating higher decision solving the problem of the next form

$$\min_{x \in D_0} \{ W(x) = \max_{i \in I} \rho_i w_i(f_1(x))\} \tag{7}$$

where $w_i(f_1(x))$ is defined in expression (1).

To acquire a clearer understanding of the algorithm, let's consider a 2-dimensional illustrative example depicted in Figure 1.

We iteratively construct the feasible region by imposing increasingly tighter partitions along the search ray. These partitions are obtained from the constraints defined by $\rho_i w_i(x) \leq k_0^{(r)}$, $i = 1,2$, where k_0 is successively reduced until the remaining feasible region is sufficiently small to allow identification of a best compromise solution. Notice that decreasing k_0 reduces all the weighted relative losses and thereby reduces the feasible region into an increasingly smaller area. Thus, if we define $\Omega^{(p)}$ as the constructed feasible region at the p-th iteration, then as $k_0 \to$'0 all the relative loss functions approach zero, i.e., the objective functions approach their optimal values. On the other hand, as $k_0 \to \frac{1}{M}, \Omega_\rho \to G$.

The best compromise solution C^* is that feasible point for which the weighted relative losses are both equivalent and minimal, that is

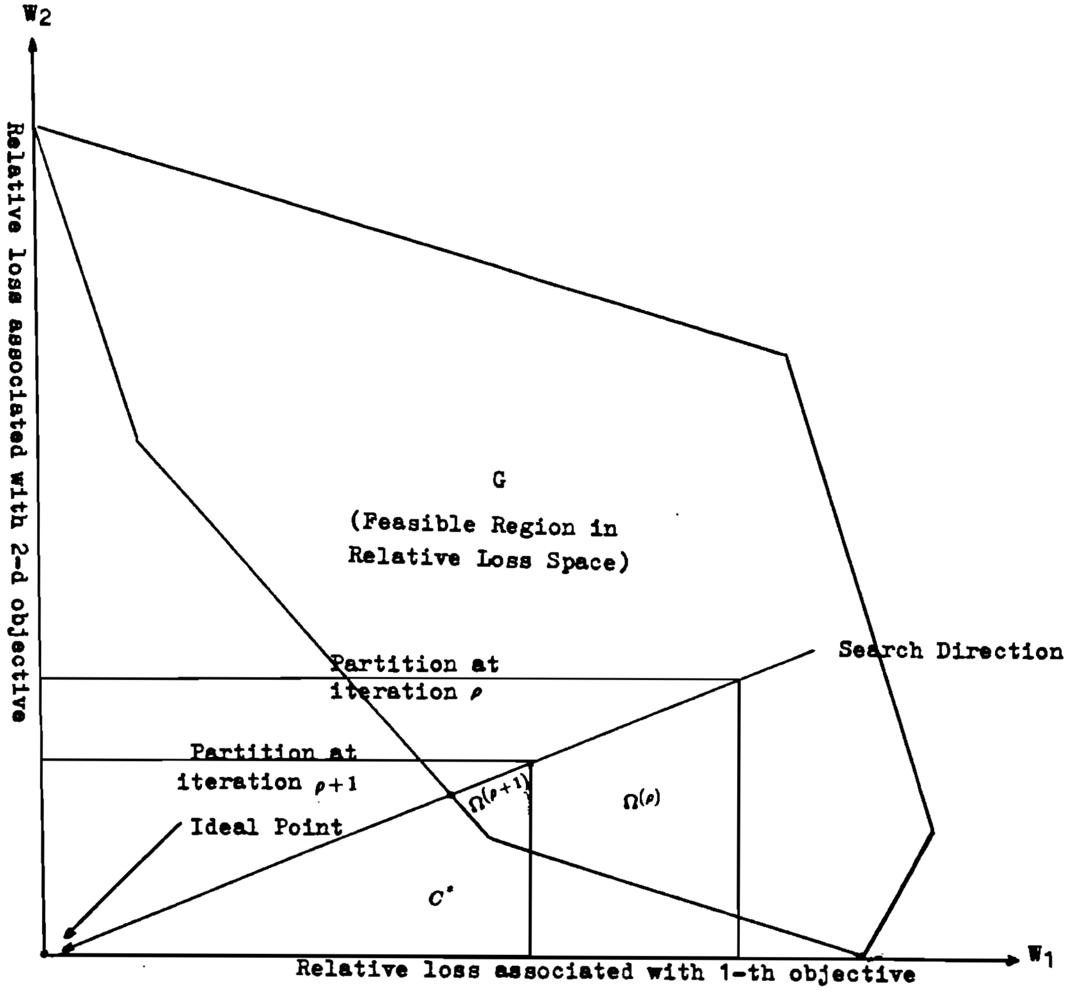
$$\rho_1 w_1(x) = \rho_2 w_2(x) = \cdots = \rho_M w_M(x) = k_{0\,\min}$$

Graphically, C^* is that feasible point which is closest to the ideal point along the search ray.

To find the best compromise solution, the Method of Constraints seeks the lowest value of $k_0^{(r)}$ for which the intersection of G and $\Omega_\rho^{(r)}$ is not empty. $G \cap \Omega \neq 0$ as long as the inequalities defining the problem's constraint set remain consistent. The method derives its name from its iterative imposition of tighter constraints $\rho_i w_i(x) \leq k_0^{(r)}$ on the original feasible region G.

In this approach we are not only concerned with a few objective functions and the region of admissible solutions, but also notice that each step in this interactive procedure must have effective algorithm checking the consistency of the system inequalities (5). We consider the solving of different algorithms multiobjective programming problems based on this approach.

Figure 1.


W2
Relative loss associated with 2-d objective
G
(Feasible Region in Relative Loss Space)
Search Direction
Partition at iteration ρ
Partition at iteration $\rho+1$
$\Omega^{(\rho+1)}$
$\Omega^{(\rho)}$
Ideal Point
C^*
Relative loss associated with 1-th objective
W1

2. The Algorithms Solving Multiobjective Mathematical Programming Problems

We write a general form of problem with the set of linear objective function

$$f = \{f_i(x) = \sum_{j \in J} c_j^i x_j\} \ , j \in J = \{\overline{1,n}\}, \ i \in I \tag{8}$$

linear constraints in general view

$$\sum_{j \in J} a_{ij} x_j \leq b_i, \ i \in Q = \{\overline{1,m}\} \tag{9}$$

and constraints on each variables

$$d_{j(l)} \leq x_j \leq d_{j(u)} \ , \forall_j \in J \tag{10}$$

There is a set J of index variables n, Q a set of index constraints, dimension m, $c_j^i \ \forall_i \in I, \forall_j \in I$, a_{ij} , $\forall_j \in J$, $i \in Q$, b_i, $\forall_i \in Q$ is corresponding coefficients, $d_{j(l)}$ - the level boundary of change variable x_j,$d_{j(u)}$ – the upper boundary of change variable x_j in the original statement problem.

If x_j is continuous we usually have a linear programming problem with set objective function and will denote it by MLP. If x_j is non-continuous we have multiobjective linear programming problem with integer variables and will denote it by MILP. In particular, if $d_{j(l)} = 0$, and $d_{j(u)} = 1$ and variables are non-continuous, we will have multiobjective linear programming problem with 0-1 integer variables (Bulev Variables) and will denote it by MBLP.

Consider MLP problem. We will assume that the definition of the above heuristic procedure was done. We know the value $f_i^0 = \sum_{j \in J} c_j^i \, x_j^{0(i)}$, $\forall_i \in I$, and

$$f_{i\,(\min)} = \sum_{j \in J} c_j^i x_{j(\min)}^{(i)} \ \ \forall_i \in I_1 \ , \ f_{i(\max)} = \sum_{j \in J} c_j^i x_{j(\max)}^{(i)}$$

where

$$X^{0(i)} = \arg\max(\min) = \sum_{j \in J} c_j^i \, x_j \ , \ x \in D_0 \ \ \forall i \in I_i \ (\forall i \in I_2) \ ,$$

and

$$x_{\min}^{(i)} = \arg\min_{x} \ \sum_{j \in I} c_j^i \, x_j \ , \ x \in D_0 \ , \ \forall_i \in I_1 \ ,$$

$$x_{\max}^{(i)} = \arg\max_{x} = \sum_{j \in I} c_j^i \, x_j \ , \ x \in D_0 \ , \forall_i \in I_2$$

D_0- region of admissible solutions denoted constraints (9) and (10). Using the Method of Constraints reformulation of the MLP problem we get [3].

$$\sum_{j\in J} c_j^i x_j \geq \sum c_j^i x_j^{0(i)} - \frac{k_0}{\rho_i} \left(\sum_{j\in I} c_j^i x_j^{0(i)} - \sum_{j\in I} c_j^i x_{j(\min)}^{(j)} \right) \forall_i \in I_1 \tag{11}$$

$$\sum_{i\in J} c_j^i x_j \leq \sum c_j^i x_j^{0(i)} + \frac{k_0}{\rho_i} \left(\sum_{j\in I} c_j^i x_{j(\max)}^i - \sum_{i\in J} c_j^i x_j^{0(i)} \right) \forall_i \in I_2$$

$$\sum_{j\in I} a_{i_j}, x_j \leq b_i \quad i \in Q$$

$$d_{j(l)} \leq x_j \leq d_{j(u)} \,, \quad i \in J$$

We must look for unique solution x^k for which this system will be consistent to minimal value parameter k_0. We cannot consider interactive procedure with parameter k_0 in this case and reformulated these problems to new problem linear programming.

$$\min_{x\,,x_{n+1}} \; x_{n+1} = K_0$$

with constraints.

$$\sum_{j\in I} a_{ij} x_j + a_{in+1}\, x_{n+1} \leq d_i \quad \forall_i \in I \tag{12}$$

$$\sum_{j\in I} a_{ij} x_j \leq b_i \quad i \in Q$$

$$d_{j(l)} \leq x_j \leq d_{j(u)} \quad \forall_j \in I \,,$$

$$0 < x_{n+1} \leq \frac{1}{M}$$

where

$$d_{ij} = \begin{cases} -\rho_i c_j^i & \forall_i \in I_1 \\ \rho_i c_j^i & \forall_i \in I_2 \end{cases} , \; j \in J$$

$$d_{n+1} = \begin{cases} -\left(\sum\limits_{i\in J} c_j^i x_j^0 - \sum\limits_{i\in J} c_j^i x_{j(\min)}^{(j)} \right) & \forall_i \in I_1 \\ \sum\limits_{i\in J} c_j^i x_{j(\max)}^{(j)} - \sum\limits_{j\in J} c_j^i x_j^{0(i)} & \forall_i \in I_2 \end{cases}$$

$$d_i = \begin{cases} -\sum\limits_{i\in I} c_j^i x_j^{0(i)} \\ \sum\limits_{i\in J} c_j^i x_j^{0(i)} \end{cases}$$

The problem (12) can be solved by using the simplex technique.

Consider MILP and MBLP problems. For these problems we will use interactive procedure of the Method of Constraints successively reducing the value of parameter k_0 and then checking to see whether the inequalities comprising the constraint set a still consistent. If so, the feasible set defined by these constraints is checked to see if it is small enough to allow the solution to be found by an exhaustive search procedure. If so, the search is performed and the method stops. If not, k_0 is once again reduced and procedure is repeated until the feasible set is sufficiently small to permit the use of exhaustive search procedure. If after u_0 has been reduced the inequalities in the constraint set are found to be inconsistent, k_0 is increased and the constraints rechecked for consistency. For checking constraint consistency we will use a procedure known as sequential analysis and sifting decision, which Mikhalevich developed earlier for solving integer programming problems in general and the sequential analysis scheme proposed in [4-7] for the solution of the discrete optimization problems. We will not go into details with all aspects of this approach and refer readings to reset source [8] explaining some important matters. The general view does not concern inequalities (11).

Denote $\prod^{(0)} = \prod \, [d_{j(l)}^{(0)} \, , \, d_{j(u)}^{(0)}]$ a parallelepiped within which the variables x_j, $j \in J$ vary at the original statement of problem. Consider arbitrary linear constraint

$$\sum_{i \in I} [\cdot]_{ij} \, x_j \leq [\cdot]_i \tag{13}$$

There is $[\cdot]_{ij}$ and $[\cdot]_i$ we will understand arbitrary coefficients right or left part of constraints (11).

Definition. The value $\triangle \, x_{j(e)}^{(j)}$ or $\triangle \, x_{j(u)}^{(0)}$ will name the level and upper correspondent by tolerance of variables $x_j \in [d_{j(e)}^{(0)} \, , \, d_{j(u)}^{(0)}]$ by constraint view (13) if from that value $x_j < \triangle x_{j(e)}^{i}$ or $x_j > \triangle \, x_{j(u)}^{(j)}$ follow that this value x_j can not form admissible solution for inequality (13).

Theorem 2. Value

$$[\frac{1}{[\cdot]_{ij}} \, ([\cdot]_i - \sum_{p \in J_i^+} [\cdot]_{ip} \, d_{p(l)}^{(0)} - \sum_{p \in J_i^-} [\cdot]_{ip} \, d_{p(u)}^{(0)}) \,] \tag{14}$$

is the level tolerance of variable x_j if $[\cdot]_{ij} > 0$ and is the upper tolerance of variable x_{ji} if $[\cdot]_{ij} < 0$, we [] denote the integer part from expression standing in brackets.

In the same way one can determine the set $J_i^+ = \{j : j \in J \, , [\cdot]_{ij} > 0\}$, $J_i^- = \{j : j \in J \, , [\cdot]_{ij} < 0\}$.

Proof. We will prove only one part of this theorem for accident when $[\cdot]_{ij} > 0$. Let value $\bar{x}_j$ for which

$$\bar{x}_j > [\frac{1}{[\cdot]_{ij}} ([\cdot]_i - \sum_{p \in J_i^+} [\cdot]_{ip} d_{p(l)} - \sum_{p \in J_i^-} [\cdot]_{ip} d^0_{p(u)})]$$

can form decision $x' = x_1{}', \dots, x_{j-1}{}', \bar{x}_j, x_{j+1}{}', \dots, x_n{}'$ satisfying inequality (14). Then $\sum_{p \in J/j} [\cdot]_{ip} x_p{}' + [\cdot]_{ij} \bar{x}_j \leq [\cdot]_i$. Rewrite this expression resetting manner

$$[\cdot]_{ij} \bar{x}_j \leq [\cdot]_i - \sum_{p \in J/j} [\cdot]_{ip} x_p{}' \leq [\cdot]_i - \min_{x \in \prod^{(0)}} \sum_{p \in J/p} [\cdot]_{ip} x_p$$

where $\prod^{(0)} = \prod_{j \in J/j} [d^{(0)}_{j(l)}\ , d^0_{j(u)}]$ we receive contradiction.

Basing this theorem on each step denoting value parameter k_0 for inequality (11) we apply a procedure which builds the intervals of variation of each variables according elimination principle which is applied for every constraints (11) on set $I_1 U I_2 U Q$.

If after applying this elimination principle the remaining set of values for any term of decision x is empty, the Method of Sequential Analysis has revealed the constraint set to be inconsistent and consequently the value assigned to k_0 was too small and so must be increased and the procedure repeated.

On the other hand, if none of the remaining sets of vector component values turns out to be empty, then the constraint set is still consistent. In this case, there are a number of possibilities:

(1) If the set of remaining vector component values is still too large to allow the selection of the preferred decision by exhaustive analysis, then a lower value must be assigned to k_0 and the procedure repeated.

(2) If the number of remaining vector component values is not too great for exhaustive analysis, then the decision(s) are found which minimize

$$W_{(x)} = \left\{ \rho_i \frac{f_i^0 - f_i(x)}{f_i^0 - f_{i,\min}\ ,} i \in I_1 \ \ \rho_i \frac{f_i(x) - f_i^0}{f_{i,\max} - f_i^0\ ,} i \in I_2 \right\} \qquad (15)$$

(a) If a unique decision is found which satisfies the problem's constraints, then it is the desired compromise solution.

(b) If there is not a unique decision then select the one that minimizes the global criteria (4). If this decision x^* satisfies the constraint set shown in (11) then it is the desired compromise solution. Decisions that meet the conditions stipulated in either (a) or (b) are said to have met Criterion I for best compromise solutions.

(c) If the alternative found in (b) does not satisfy the constraints in (11) then the initial problem's objective functions are required to satisfy the following additional new constraints:

$$\sum_{j\in I} c_j^{(i)} x_j \geq f_i(x^*) \quad i \in I_1$$

$$\sum_{j\in I} c_j^{(i)} x_j \geq f_i(x^*) \quad i \in I_2$$

With these constraints added to the original problem (9) the Method of Sequential Analysis is again employed to eliminate decision from the initial set V, producing either a single decision or a set of decisions which will now satisfy the conditions described in (a) or (b). Alternatives that meet the conditions stipulated in (c) are said to have met Criterion II for best compromise solutions [1].

Consider a discrete separable programming problem with a set separable objective function. Let $X = \prod_{j\in J} X_j$ - a set of possible solutions where $|X| = N = \prod_{j\in J} J_j$. In other words, x is a vector consisting of n terms $x = \{x_{1(l_1)}, \ldots, x_{j(l_j)}, \ldots, x_{n(l_n)}\}$. The j-th component of vector x is denoted $x_{j(l_j)}$ and takes on J_j possible values, an arbitrary one of which is the l_j-th. Let

$$f = \{f_i(x) = \sum_{j\in J} f_i(x_{j(l_j)})\} \; i \in I \tag{16}$$

a set of separable functions which we must find a compromise solution. The compromise solution must belong to the possible solution system of constraints

$$\begin{cases} g_p(x) = \sum_{j\in J} g_p(x_{j(l_j)}) \leq g_p^* \; p = \overline{1,q} \\ g_p(x) = \sum_{j\in J} g_p(x_{j(l_j)}) \leq g_p^* \; p = \overline{q+1,Q} \end{cases} \tag{17}$$

The problem is then transformed into the Method of Constraints formulation. Substituting for view of objective functions (16) or constraints (17) the system inequalities (11) is then rewritten.

$$\begin{cases} \sum\limits_{j\in J} f_j(x_{j(l_j)}) \geq f_i^*(k_0) \text{ , } i \in I_1 \text{ ,} \\ \sum\limits_{j\in J} f_j(x_{j(l_j)}) \leq f_i^*(k_0) \text{ , } \forall i \in I_2 \text{ ,} \\ \sum\limits_{j\in J} g_p(x_{j(l_j)}) \leq g_p^* \text{ , } p = \overline{1,q} \text{ ,} \\ \sum\limits_{j\in J} g_p(x_{j(l_j)}) \geq g_p^* \text{ , } p = \overline{q+1,Q} \text{ ,} \\ x \in X \text{ ,} \end{cases} \tag{18}$$

where

$$f_i^*(k_0) = \begin{cases} f_i(x_i^0) - \dfrac{k_0}{P_i}(f_i(x_i^0) - f_i(x_{i(min)}) \\ f_i(x_i^0) + \dfrac{k_0}{P_i}(f_i(x_{i(max)}) - f_i(x_i^0)) \text{ .} \end{cases} \tag{19}$$

In these sectors x_i^0 , $x_{i(max)}$, $x_{i(min)}$ it is found which optimize the objective functions for $x \in X$.

$$x_i^0 = \begin{cases} \arg\max\limits_{x\in X} f_i(x) = \sum\limits_{j\in J} \arg\max\limits_{x_{j(l_j)}\in X_j} f_j(x_{j(l_j)}) \text{ ,} i \in I_1 \text{ ,} \\ \arg\min\limits_{x\in X} f_i(x) = \sum\limits_{j\in J} \arg\min\limits_{x_{j(l_j)}\in X_j} f_i(x_{j(l_j)}) \text{ ,} i \in I_2 \text{ ,} \end{cases}$$

$$x_{i(max)} = \arg \min f_i(x) = \sum_{j\in J} \arg\max_{x_{j(l_j)}\in x_j} f_i(x_{j(l_j)}) \text{ , } \forall i \in I_1 \text{ ,}$$

$$x_{i(max)} = \arg\max_{x\in x} f_i(x) = \sum_{j\in J} \arg\max_{x_{j(l_i)}\in x_j} f_i(x_{j(l_j)}) \text{ , } \forall i \in I_2 \text{ .}$$

We then define x/x_j as any vector $\tilde{v}$ without its j-th component, i.e.

$$x/x_j = x_{1(l_1)} \text{ , } \cdots \text{ , } x_{j-1(l_{j-1})} \text{ , } x_{j+1(l_{j+1})} \text{ , } \cdots \text{ , } x_{n(l_n)}$$

The condition used by the Method of Sequential Analysis to eliminate vector component values is

Eliminate vector component value $x_{j(l_j)}$ from further consideration if the following is true for any $i = 1,\dots, M$.

$$f_i(x_{j(l_j)}) < f_i^{(j)*}(k_0^{(h)}), \quad i \in I_1 \tag{20}$$

$$f_i(x_{j(l_j)}) > f_i^{(j)*}(k_0^{(h)}), \quad i \in I_2$$

where

$$f_i^{(j)\,*}(k_0^{(h)}) = f_i^*(k_0^{(h)}) - \sum_{\substack{p=1\\p\neq j}}^{n} f_i(x_{p(l_p)}^{(i)})$$

$$x_{p(l_p)}^{(i)} = \begin{cases} \operatorname*{argmax}_{x_p} f_i(x_{j(l_j)}) \,, & i = 1 \,, \cdots \,, m \\ \operatorname*{argmax}_{x_p} f_i(x_{j(l_j)}) \,, & i = M - m \,, \cdots \,, M \end{cases}$$

In effect these conditions state: Assuming all components of vector x, except the j-th component, are contributing the optimal possible value to objective function f_i, then eliminating any values for the j-th term of vector x that violate the constraint placed on the i-th objective function by setting $k_0 = k_0^{(h)}$, namely the condition that

$$f_i(x) \geq f_i^*(k_0) \quad i \in I_1 \,, \tag{21}$$

$$f_i(x) \leq f_i^*(k_0) \quad i \in I_2 \,.$$

The above elimination principle is applied for every component x_j, $j = 1 \,, \cdots n$ of vector x and for every objective function $f_i(x) < i = 1 \,, \cdots \,, M$.

In this elimination principle the remaining set of values for any term of vector x applies by the scheme method of Sequential Analysis using the above analogy.

The results of computational experience and the application of the Method of Constraints to integer and separable multi-objective programming problems will become clearer by considering the following.

3. Method of Constraints Applied to Integer Problem Without Side Constraints

An illustrative example of the application of the Method of Constraints to an integer programming problem without side constraints. In the problem three objective functions [1]

$$\min_{x \in X} f_1(x), \ \min_{x \in X} f_2(x), \ \min_{x \in X} f_3(x) \,,$$

are minimized over the set $X = \{X_1 \times X_2 \times \cdots \times X_8\}$

$$\{x\} = \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \\ X_8 \end{Bmatrix} = \begin{Bmatrix} x_{11} & x_{12} & x_{13} & \bullet & \bullet \\ x_{21} & x_{22} & x_{23} & x_{24} & \bullet \\ x_{31} & x_{32} & \bullet & \bullet & \bullet \\ x_{41} & x_{42} & x_{43} & \bullet & \bullet \\ x_{51} & x_{52} & x_{53} & x_{54} & x_{55} \\ x_{61} & x_{62} & x_{63} & x_{64} & \bullet \\ x_{71} & x_{72} & \bullet & \bullet & \bullet \\ x_{81} & x_{82} & x_{83} & \bullet & \bullet \end{Bmatrix}$$

Thus, x is a vector with eight terms all of which feasible values are represented above in set x. From set x we see that there are 8640 possible combination values of vector x.

Each feasible vector component value $x_{j(l_j)}$ contributes $f_j^i(x_{j(l_j)})$ to the i-th objective function, where i is the objective function index ($i = 1,2,3$) and $x_{j(l_j)}$ is the l-th feasible value for the j-th term of vector x. This can be depicted by the function f_i shown below which explicitly associates with every $x_{j(l_j)}$ a corresponding objective function value $f_j^i(x_{j(l_j)})$:

$$f_i = \left\{\begin{array}{ccc|ccc} x_{1(1_1)} & \dots & x_{1(m_1)} & f_i(x_{1(1_1)}) & \dots & f_i(x_{1(m_1)}) \\ \bullet & \dots & \bullet & \bullet & \dots & \bullet \\ \bullet & \dots & \bullet & \bullet & \dots & \bullet \\ x_{j(1_j)} & \dots & x_{j(m_j)} & f_i(x_{j(1_j)}) & \dots & f_i(x_{j(m_j)}) \\ \bullet & \dots & \bullet & \bullet & \dots & \bullet \\ \bullet & \dots & \bullet & \bullet & \dots & \bullet \\ x_{8(1_8)} & \dots & x_{8(m_8)} & f_i(x_{8(1_8)} & \dots & f_i(x_{8(m_8)}) \end{array}\right\}$$

(For the sake of clarity all subsequent representations of function f_i will only indicate the l_j subscripts of the vector component values $x_{j(l_j)}$ on the lefthand side of the function.)

In the specific problem under consideration, the objective function values corresponding to the elements of set x are:

$$f_1 = \left\{\begin{array}{ccccc|ccccc} 1 & 2 & 3 & \cdot & \cdot & 76 & 180 & 125 & \cdot & \cdot \\ 1 & 2 & 3 & 4 & \cdot & 40 & 260 & 520 & 35 & \cdot \\ 1 & 2 & \cdot & \cdot & \cdot & 30 & 80 & \cdot & \cdot & \cdot \\ 1 & 2 & 3 & \cdot & \cdot & 510 & 520 & 120 & \cdot & \cdot \\ 1 & 2 & 3 & 4 & 5 & 65 & 80 & 90 & 100 & 35 \\ 1 & 2 & 3 & 4 & \cdot & 210 & 220 & 500 & 120 & \cdot \\ 1 & 2 & \cdot & \cdot & \cdot & 70 & 50 & \cdot & \cdot & \cdot \\ 1 & 2 & 3 & \cdot & \cdot & 290 & 310 & 400 & \cdot & \cdot \end{array}\right\}$$

$$f_2 = \left\{\begin{array}{ccccc|ccccc} 1 & 2 & 3 & \cdot & \cdot & 10 & 15 & 5 & \cdot & \cdot \\ 1 & 2 & 3 & 4 & \cdot & 7 & 11 & 8 & 13 & \cdot \\ 1 & 2 & \cdot & \cdot & \cdot & 4 & 12 & \cdot & \cdot & \cdot \\ 1 & 2 & 3 & \cdot & \cdot & 11 & 7 & 5 & \cdot & \cdot \\ 1 & 2 & 3 & 4 & 5 & 6 & 9 & 8 & 4 & 10 \\ 1 & 2 & 3 & 4 & \cdot & 5 & 11 & 14 & 4 & \cdot \\ 1 & 2 & \cdot & \cdot & \cdot & 8 & 9 & \cdot & \cdot & \cdot \\ 1 & 2 & 3 & \cdot & \cdot & 4 & 5 & 7 & \cdot & \cdot \end{array}\right\}$$

$$f_3 = \left\{\begin{array}{ccccc||ccccc}
1 & 2 & 3 & \cdot & \cdot & 70 & 80 & 30 & \cdot & \cdot \\
1 & 2 & 3 & 4 & \cdot & 65 & 90 & 70 & 80 & \cdot \\
1 & 2 & \cdot & \cdot & \cdot & 95 & 90 & \cdot & \cdot & \cdot \\
1 & 2 & 3 & \cdot & \cdot & 65 & 75 & 90 & \cdot & \cdot \\
1 & 2 & 3 & 4 & 5 & 90 & 70 & 95 & 80 & 55 \\
1 & 2 & 3 & 4 & \cdot & 65 & 95 & 50 & 85 & \cdot \\
1 & 2 & \cdot & \cdot & \cdot & 60 & 75 & \cdot & \cdot & \cdot \\
1 & 2 & 3 & \cdot & \cdot & 65 & 95 & 60 & \cdot & \cdot
\end{array}\right\}$$

We now apply the Method of Constraints and the Method of Sequential Analysis to the problem.

Step 1: We rearrange the objective function values in each row of f_1, f_2, and f_3 in order of increasing values (i.e., from best to worst)

$$f_1^{ord} = \left\{\begin{array}{ccccc||ccccc}
1 & 3 & 2 & \cdot & \cdot & 75 & 125 & 180 & \cdot & \cdot \\
4 & 1 & 2 & 3 & \cdot & 35 & 40 & 260 & 520 & \cdot \\
1 & 2 & \cdot & \cdot & \cdot & 30 & 80 & \cdot & \cdot & \cdot \\
3 & 1 & 2 & \cdot & \cdot & 120 & 510 & 520 & \cdot & \cdot \\
5 & 1 & 2 & 3 & 4 & 35 & 65 & 80 & 90 & 100 \\
4 & 1 & 2 & 3 & \cdot & 120 & 210 & 220 & 500 & \cdot \\
2 & 1 & \cdot & \cdot & \cdot & 50 & 70 & \cdot & \cdot & \cdot \\
1 & 2 & 3 & \cdot & \cdot & 290 & 310 & 400 & \cdot & \cdot
\end{array}\right\}$$

$$f_2^{ord} = \left\{\begin{array}{ccccc||ccccc}
3 & 1 & 2 & \cdot & \cdot & 5 & 10 & 15 & \cdot & \cdot \\
1 & 3 & 2 & 4 & \cdot & 7 & 8 & 11 & 13 & \cdot \\
1 & 2 & \cdot & \cdot & \cdot & 4 & 12 & \cdot & \cdot & \cdot \\
3 & 2 & 1 & \cdot & \cdot & 5 & 7 & 11 & \cdot & \cdot \\
4 & 1 & 3 & 2 & 5 & 4 & 6 & 8 & 9 & 10 \\
4 & 1 & 2 & 3 & \cdot & 4 & 5 & 11 & 14 & \cdot \\
1 & 2 & \cdot & \cdot & \cdot & 8 & 9 & \cdot & \cdot & \cdot \\
1 & 2 & 3 & 3 & \cdot & 4 & 5 & 7 & \cdot & \cdot
\end{array}\right\}$$

$$f_3^{ord} = \left\{\begin{array}{ccccc||ccccc}
3 & 1 & 2 & \cdot & \cdot & 30 & 70 & 80 & \cdot & \cdot \\
1 & 3 & 4 & 2 & \cdot & 65 & 70 & 80 & 90 & \cdot \\
2 & 1 & \cdot & \cdot & \cdot & 90 & 95 & \cdot & \cdot & \cdot \\
1 & 2 & 3 & \cdot & \cdot & 65 & 75 & 90 & \cdot & \cdot \\
5 & 2 & 4 & 1 & 3 & 55 & 70 & 80 & 90 & 95 \\
3 & 1 & 4 & 2 & \cdot & 50 & 65 & 85 & 95 & \cdot \\
1 & 2 & \cdot & \cdot & \cdot & 60 & 75 & \cdot & \cdot & \cdot \\
3 & 1 & 2 & \cdot & \cdot & 60 & 65 & 95 & \cdot & \cdot
\end{array}\right\}$$

Step 2: We are now ready to apply the Method of Constraints formulation to the original problem. (Note that in the case at hand all enumerated alternatives are feasible; there are no constraints of the form $g_p(x) \leq g_p^*$ or $g_p(x) \leq g_p^*$. Thus, in the Method of Constraints reformulation the only constraints are those imposed on the objective functions. Since all objective functions are being minimized these constraints will take the form $f_i(x) \leq f_i^*(k_0)$. Thus, we must set k_0 and calculate $f_i^*(k_0)$.

From the ordered sets of objective function component values found in Step 1 we find the optimal and worst values for each objective function simply by summing the first and last columns on the right-hand side of each table. Thus,

$$f_1^0 = 755, \quad f_2^0 = 41, \quad f_3^0 = 475,$$

$$f_{1,max} = 2370, \quad f_{2,max} = 91, \quad f_{3,max} = 715,$$

In applying the Method of Constraints to this problem it is assumed that each objective function is equally weighted, i.e., $\rho_1 = \rho_2 = \rho_3$. Then for computational convenience we can set $\rho_i = 1$ for every i. Arbitrarily, we set $k_0^{(1)} = .4$ for the first iteration. Then we calculate

$$f_i^*(k_0^{(1)}) = f_i^0 + \frac{k_0^{(1)}}{\rho_i} (f_{i,max} - f_i^0) \quad \text{for } i = 1,2,3$$

yielding $f_1^*(k_0^{(1)}) = 1401$, $f_2^*(k_0^{(1)}) = 61$, $f_3^*(k_0^{(1)}) = 570$.

Step 3: Now the Method of Sequential Analysis is used to eliminate components of set X which violate the conditions imposed by the Method of Constraints, i.e.,

$$f_1(x) \leq 1401$$

$$f_2(x) \leq 61$$

$$f_3(x) \leq 570$$

For each term in each objective function we calculate $f_i^{(j)^*}(k_0^{(h)})$, the cut-off value for the j-th term of the i-th objective function. In effect this cut-off value says, "Suppose all other terms of the i-th objective function were at their minimal value, what is the highest value that the j-th term of the i-th objective could assume before the objective violates conditions". In the problem at hand, calculations are shown below for

$$f_i^{(j)^*}(k_0^{(1)}) = f_i^*(.4) - \sum_{\substack{p \neq i \\ p \neq j}}^{n} f_i(x_{p(i_p)}^{(i)}) \quad \text{for every } j = 1, \ldots, 8,\ i = 1,2,3$$

These are the cut-off values for each term of each objective function. Comparing these cut-off values to the objective function component values themselves are shown in

Table: Cut-Off Levels for the Terms of the Objective Function

	$f_1^{(j)*}(0,4)$	$f_2^{(j)*}(0,4)$	$f_3^{(j)*}(0,4)$
j = 1	721	25	125
2	631	27	160
3	676	24	188
4	766	25	160
5	681	24	150
6	766	24	145
7	669	28	155
8	936	24	155

f_i^{ord}, $i = 1,2,3$, we see that no objective term exceeds its cut-off.

Step 4: We reduce k_0, recalculate $f_i^*(k_0)$ as in Step 2, and then repeat the procedure in Step 3. Letting $k_0^{(2)} = 1/2(.4) = .2$, we find that $f_1^*(k_0^{(2)}) = 1078$, f_2^* $(k_0^{(2)}) = 51$, $f_3^*(k_0^{(2)}) = 523$. Our new table of objective function component cut-off values is

	$f_1^{(j)*}(0,2)$	$f_2^{(j)*}(0,2)$	$f_3^{(j)*}(0,2)$
j = 1	398	15	78
2	358	17	113
3	353	14	138
4	443	15	113
5	358	14	103
6	443	14	98
7	373	18	108
8	613	14	108

Using these cut-off values the following vector component values turn out to violate the given elimination principles:

For $f_1^{(j)^*}(0,2)$-

$$x_{23} : f_1(x_{23}) = 520 \leq f_1^{(2)^*}(0,2) = 358$$

$$x_{41} : f_1(x_{41}) = 520 \leq f_1^{(4)^*}(0,2) = 443$$

$$x_{42} : f_1(x_{42}) = 510 \leq f_1^{(4)^*}(0,2) = 443$$

$$x_{63} : f_1(x_{63}) = 500 \leq f_1^{(6)^*}(0,2) = 443$$

For $f_2^{(j)^*}(0,2)$-

none

For $f_3^{(j)^*}(0,2)$-

$$x_{12} : f_3(x_{12}) = 80 \leq f_3^{(1)^*}(0,2) = 78$$

Eliminating these vector term values from set X we find

$$\{X^{(2)}\} = \left\{\begin{array}{|ccccc|} x_{11} & \bullet & x_{13} & \bullet & \bullet \\ x_{21} & x_{22} & \bullet & x_{24} & \bullet \\ x_{31} & x_{32} & \bullet & \bullet & \bullet \\ \bullet & \bullet & x_{43} & \bullet & \bullet \\ x_{51} & x_{52} & x_{53} & x_{54} & x_{55} \\ x_{61} & x_{62} & \bullet & x_{64} & \bullet \\ x_{71} & x_{72} & \bullet & \bullet & \bullet \\ x_{81} & x_{82} & x_{83} & \bullet & \bullet \end{array}\right\}$$

Step 5: As a result of the elimination process performed in the previous step the table of criteria values for f_3 is now

$$f_3^{(2)} = \left\{\begin{array}{|ccccc||ccccc|} 3 & 1 & \cdot & \cdot & \cdot & 30 & 70 & \cdot & \cdot & \cdot \\ 1 & 4 & 2 & \cdot & \cdot & 65 & 80 & 90 & \cdot & \cdot \\ 2 & 1 & \cdot & \cdot & \cdot & 90 & 95 & \cdot & \cdot & \cdot \\ 3 & \cdot & \cdot & \cdot & \cdot & 90 & \cdot & \cdot & \cdot & \cdot \\ 5 & 2 & 4 & 1 & 3 & 55 & 70 & 80 & 90 & 95 \\ 1 & 4 & 2 & \cdot & \cdot & 65 & 85 & 95 & \cdot & \cdot \\ 1 & 2 & \cdot & \cdot & \cdot & 60 & 75 & \cdot & \cdot & \cdot \\ 3 & 1 & 2 & \cdot & \cdot & 60 & 65 & 95 & \cdot & \cdot \end{array}\right\}$$

Notice that in the fourth and sixth rows the lowest attainable objective function component values have increased due to the elimination of vector value components x_{41},x_{42}, and x_{63}. As a result now $f_3^{0(2)} = 515$. Therefore, without changing $f_3^*(k_0^{(2)}) = 523$ we can recalculate the table of cut-off values $f_3^{(j)^*}(0,2)$ for the third objective function.

	$f_3^{(j)^*}(0,2)$
j = 1	38
2	73
3	98
4	98
5	63
6	73
7	68
8	68

Using these cut-off values the following vector component values are eliminated: x_{11}, x_{22}, x_{24}, x_{51}, x_{52}, x_{54}, x_{62}, x_{64}, x_{72}, x_{82}. Eliminating these values from set $X^{(2)}$, we are left with the following set of vector component values

$$\{X^{(3)}\} = \left\{\begin{array}{ccccc} \bullet & \bullet & x_{13} & \bullet & \bullet \\ x_{21} & \bullet & \bullet & \bullet & \bullet \\ x_{31} & x_{32} & \bullet & \bullet & \bullet \\ \bullet & \bullet & x_{43} & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & x_{55} \\ x_{61} & \bullet & \bullet & \bullet & \bullet \\ x_{71} & \bullet & \bullet & \bullet & \bullet \\ x_{81} & \bullet & x_{83} & \bullet & \bullet \end{array}\right\}$$

Step 6: The ordered tables of criteria values for all three objective functions are now:

$$f_1^{(3)} = \left\{\begin{array}{cc||cc} 3 & \cdot & 125 & \cdot \\ 1 & \cdot & 40 & \cdot \\ 1 & 2 & 30 & 80 \\ 3 & \cdot & 120 & \cdot \\ 5 & \cdot & 35 & \cdot \\ 1 & \cdot & 210 & \cdot \\ 1 & \cdot & 70 & \cdot \\ 1 & 3 & 290 & 400 \end{array}\right\} \qquad f_2^{(3)} = \left\{\begin{array}{cc||cc} 3 & \cdot & 5 & \cdot \\ 1 & \cdot & 7 & \cdot \\ 1 & 2 & 4 & 12 \\ 3 & \cdot & 5 & \cdot \\ 5 & \cdot & 10 & \cdot \\ 1 & \cdot & 5 & \cdot \\ 1 & \cdot & 8 & \cdot \\ 1 & 3 & 4 & 7 \end{array}\right\} \qquad f_3^{(3)} = \left\{\begin{array}{cc||cc} 3 & \cdot & 30 & \cdot \\ 1 & \cdot & 65 & \cdot \\ 2 & 1 & 90 & 95 \\ 3 & \cdot & 90 & \cdot \\ 5 & \cdot & 55 & \cdot \\ 1 & \cdot & 65 & \cdot \\ 1 & \cdot & 60 & \cdot \\ 3 & 1 & 60 & 65 \end{array}\right\}$$

Without changing $f_1^*(k_0^{(2)}) = 1078$ or $f_2^*(k_0(2)) = 51$, we can now recalculate the cut-off values for the first and second objective functions as we did in the previous step for the third objective function. The new optimal objective function component values are $f_1^{0(3)} = 920$ and $f_2^{0(3)} = 48$ and objective function component cut-off values are:

	$f_1^{(j)*}\ (0, 2)$	$f_2^{(j)*}\ (0, 2)$
j = 1	283	8
2	198	10
3	188	7
4	178	8
5	193	13
6	368	8
7	228	11
8	448	7

These cut-off values eliminate one vector component value x_{32} as a result of the violation of the elimination principle for $f_2^{(j)^*}(0,2)$-

$$x_{32}\ ;\ f_2(x_{32}) = 12 \leq f_2^{(3)^*}(0,2) = 7$$

Thus, the set X of remaining vector component values is

$$\{X^{(4)}\} = \left\{ \begin{array}{ccccc} \bullet & \bullet & x_{13} & \bullet & \bullet \\ x_{21} & \bullet & \bullet & \bullet & \bullet \\ x_{31} & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & x_{43} & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & x_{55} \\ x_{61} & \bullet & \bullet & \bullet & \bullet \\ x_{71} & \bullet & \bullet & \bullet & \bullet \\ x_{81} & \bullet & x_{83} & \bullet & \bullet \end{array} \right\}$$

Step 7: With the elimination of vector component value u_{32} we can once again recalculate cut-off values for the third objective function since now $f_3^*(k_0^{(2)}) = 520$ and the table of objective function cut-off values $f_3^{(j)^*}(0,2)$ is:

	$f_3^{(j)*}\,(0,2)$
j = 1	33
2	68
3	98
4	93
5	58
6	68
7	63
8	63

Vector component value x_{81} is eliminated for violating condition $f_3(x_{81}) = 65 \leq f_3^{(8)*}(0,2) = 63$.

We are left with a single variant $x^* = (x_{13}, x_{21}, x_{31}, x_{43}, x_{55}, x_{61}\ \ x_{71}, x_{83})$ which is the best compromise solution with $f_1(x^*) = 1030$, $f_2(x^*) = 51$, $f_3(x^*) = 520$, and

$$f_1(x) = \left\{\begin{array}{c|c} 3 & 125 \\ 1 & 40 \\ 1 & 30 \\ 3 & 120 \\ 5 & 35 \\ 1 & 210 \\ 1 & 70 \\ 3 & 400 \end{array}\right\} \qquad f_2(x) = \left\{\begin{array}{c|c} 3 & 5 \\ 1 & 7 \\ 1 & 4 \\ 3 & 5 \\ 5 & 10 \\ 1 & 5 \\ 1 & 8 \\ 3 & 7 \end{array}\right\} \qquad f_3(x) = \left\{\begin{array}{c|c} 3 & 30 \\ 1 & 65 \\ 1 & 95 \\ 3 & 90 \\ 5 & 55 \\ 1 & 65 \\ 1 & 60 \\ 3 & 60 \end{array}\right\}$$

4. Computational Experience in Application of Method of Constraints to Integer Multi-Objective Problems [1]

Using FORTRAN Dargeiko wrote a standard Method of Constraints algorithm for the Soviet BESM-6 computer running under the 'Dubna' operating system. The program was capable of solving in operating memory a problem of dimension $n \times l \times M \leq 3000$, where

n = number of variables

l = local number of elements in sets X_j, $j = l, \ldots, n$ (X_j is the set of alternative values for the j-th term of vector x)

M = number of criteria

The initial data for the experimental runs was produced by a pseudo-random number generator whose output was uniformly distributed on an interval (A,B).By varying the bounds of interval (A,B), criteria values were generated for all $n \times l$ elements. Table 1. shows how to determine the type and number of computational operations performed at each iteration. Table 2. reproduces the results of Dargeiko's experiments for equally weighted criteria.

Dargeiko also performed experiments to analyze the impact of varying criteria weights on the speed with which a solution was found. Thus, problem 2 in Table 1.4 was solved for three different combinations of criteria weights. For the three sets of weights, the size of the set of candidate variants was reduced from 50^{10} to 810, 530, 120 respectively after 9, 15, and 12 iterations on k_0. In all three cases Dargeiko reports that the computational time did not exceed a minute.

Table 1.: Computational Operations Performed by Method of Constraints Algorithm for Integer Multi-Objective Programming Problems

Number of Operations in a Single Iteration			
Addition:	$n \times M$	Division:	1
Subtraction:	$n \times M$	Multiplication:	M
Permutations:	$n \times l \times M$	Comparisons:	$n \times l \times M$
Volume of Memory Used Each Iteration:		On the order of $n \times l \times M$	

Table 2.: Computational Experience with Method of Constraints Integer Multi-Objective Programming Problems

	Problem Number				
	1	2	3	4	5
n = Number of Variables	7	10	10	50	100
l = Elements in Set U_j,j=1,...,n	100	50	50	10	5
M = Number of Criteria	100	50	50	10	5
General Number of Variants	10^{14}	50^{10}	50^{10}	10^{50}	5^{100}
Number of Variants after Elimination Procedure	6	80	130	2500	1580
Number of Iterations on k_0	11	14	6	6	7
Solution Time (in seconds)	23	46	68	86	71

REFERENCES

[1] Mikhalevich, V.S., and Volkovich, V.L. (1982) *Vychislitel'nyi metod issledovaniya i proektirovaniya slozhnykh sistem.* (Computational method for research and design of complex systems.) Moscow: Nauka.

[2] Mikhalevich, V.S. (1965a) Sequential Optimization Algorithms and Their Application. Part 1. *Cybernetics* 8, no. 5: 851-855.

[3] Mikhalevich, V.S. (1965b) Sequential Optimization Algorithms and Their Application. Part 2. *Cybernetics* 1, no. 2: 87-92.

[4] Mikhalevich, V.S., and Shor, N.Z. (1962a) Chislennoe reshenie mnogovariantnykh zadach po metodu posledovatel'nogo analiza variantov. (Numerical solution of multivariant problems by the method of sequential analysis of variants.) *Nauchno-metododicheskie materialy ekonomiko-matematicheskogo seminara. Vypuck 1.* Mos-

cow: LEMM Akademii nauk SSSR. n.p.

[5] Mikhalevich, V.S., and Shor, N.Z. (1962b) O chislennykhmetodakh resheniya mnogovariantnykh planovykh i tekhniko-ekonomicheskikh zadach. (Numerical methods for solving multivariant planning and technico-economical problems). *Nauchno-metododicheskie materialy ekonomiko-matematicheskogo seminara VTs Akademii nauk USSR* no. 1: 15-42.

[6] Volkovich, V.L., and Dargeiko, V.F. (1976) A Method of Constraints in Vector Optimization Problems. *Soviet Automatic Control* no. 3: 10-13.

[7] Volkovich, V.L., and Dargeiko, V.F. (1972) On an Algorithm of Choosing a Compromise Solution for Linear Criteria. *Cybernetics* 8, no. 5: 851-855.

[8] Volkovich, V.L., and Voloshin, V.L. (1978) A Scheme of the Method of Sequential Analysis and Sifting of Variables. *Cybernetics* 14, no. 4: 585-593.