БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ . BULGARIAN ACADEMY OF SCIENCES

КИБЕРНЕТИКА И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ · Том **4**, № 2
CYBERNETICS AND INFORMATION TECHNOLOGIES · Volume **4**, No 2

София . 2004 . Sofia

A Partition-Based Interactive Algorithm for Solving Multicriteria Linear Integer Programming Problems*

Vassil Vassilev[1], *Subhash Narula*[2], *Mariyana Vassileva*[1], *Krassimira Genova*[1]

[1] *Institute of Information Technologies, 1113 Sofia*
[2] *School of Business, Virginia Commonwealth University, Richmond, USA*

Abstract: *We propose a learning-oriented interactive algorithm for solving multicriteria linear integer programming (MCLIP) problems, considered as multicriteria decision making problems. At each iteration, the DM may partition the criteria set into at most seven classes, namely: improvement, improvement by a desired amount, deterioration, deterioration by at most a certain amount, non-deterioration, changes allowed within limits, and free changes. Based on the partition of the criteria set, two types of scalarizing problems are formulated – linear and mixed integer programming problems. One or more (weak) nondominated solutions of the continuous relaxation of MCLIP problem are computed at most of the iterations. A mixed-integer scalarizing problem is solved, only at some iterations, in order to find one or more (weak) nondominated or near (weak) nondominated solutions (close to the nondominated surface of the MCLIP problem). At some iteration, when the DM wants to see more than one solution, he/she may select the preferred solution based on nonformalized information about his/her preferences or may use a ranking procedure based on additional formal information. Based on the proposed algorithm, we have developed a research decision support system.*

Keywords: *decision support system; multicriteria decision making; ranking procedure; scalarizing problem.*

* This paper is partially supported by NSF of MES – contract “Interactive Algorithms and Software Systems Supporting Multicriteria Decision Making”.

1. Introduction

Interactive algorithms are widely used for solving multicriteria linear programming (MCLP) problems (considered as multicriteria ddecision making problems), see Benayoun et al. [2], Steuer [31], Wierzbicki [39], Korhonen, Laakso [20], and Kaliszewski, Michalowski [17]. The quality of an interactive algorithm depends, to a large extent, on the quality of the dialogue with the decision-maker (DM), namely:

- the type of information given by the DM to improve the current preferred nondominated solution;
- the time needed to solve the scalarizing problem;
- the type and the number of new solutions computed at an iteration;
- the ability to change search strategies for computing new solutions;
- the possibility for the DM to learn about the multicriteria problem.

When solving multicriteria linear programming (MCLP) problems, linear programming problems are used as scalarizing problems. These problems are comparatively easy to solve. For this reason, the time needed to solve the scalarizing problems in interactive algorithms for solving MCLP problems plays a minor role.

Interactive algorithms are also used (Climaco et al. [5]) to solve multicriteria linear integer programming (MCLIP) problems. These algorithms may be divided into two groups. The interactive algorithms in the first group, e.g., Karwan et al. [19], Marcotte, Soland [21], Ramesh et al. [28], Durso [7], and Alves, Climaco [1], are modifications of single criterion integer algorithms, in which the DM is involved in the iterative computational process to obtain efficient integer solutions of the MCLIP problems. The main purpose in the development of the first group of algorithms is to reduce the number of computational interruptions and the number of comparisons the DM has to make.

The interactive algorithms that form the second group, e.g., Teghem, Kunsh [33], Gabbani, Magazine [9], Hajela, Shih [15], Narula, Vassilev [24], Karaivanova et al. [18], Vassileva [34], are modifications of interactive approaches for solving MCLP problems. These interactive algorithms use linear integer programming problems as scalarizing problems, which are NP-hard, Garey, Johnson [11]. Therefore, the time to solve the scalarizing problems in these interactive algorithms plays a major role. That is why some efforts are made in the design of these algorithms to reduce the number of the integer problems solved, to use heuristic algorithms that solve integer problems, to solve continuous instead of integer problems at most of the iterations and to present continuous solutions to the DM for evaluation, especially during the learning phase.

We propose a learning-oriented (Gardiner, Vanderpooten [10]), interactive algorithm that belongs to the second group of algorithms. In the proposed algorithm we attempt to improve the dialogue with the DM to describe his/her preferences, to reduce the time to find new solutions and to help the DM evaluate more than one solution.

The proposed algorithm uses a scalarizing problem based on the partition of the criteria, implicitly done by the DM. In this classification, the DM specifies what changes he/she would like to see in the criteria values of the current preferred solution. Depending on the desired or acceptable changes, each criterion may belong to one and only one of the seven classes – improvement, improvement by a desired amount (if possible), deterioration, deterioration by at most a certain amount, non-deterioration, changes allowed within limits and free changes. This offers the DM more possibilities to express his/her preferences with respect to current preferred solution compared to the well known algorithms for solving linear and nonlinear multicriteria programming problems such as STEP algorithm, Benayoun et al. [2], the reference point algorithms, Wierzbicki [39], reference direction algorithm, Korhonen, Laakso [20], the NIMBUS algorithm, Miettinen, Makela [22], and algorithms used for solving multicriteria integer linear and nonlinear programming problems, e.g. Narula, Vassilev [24] and Vassilev et al. [36].

In the proposed algorithm, the DM is given a choice, at each iteration, to either solve a continuous or an integer scalarizing problem. The DM is encouraged especially during the learning phase or when solving large problems to solve continuous scalarizing problems or to solve integer scalarizing problems approximately at many iterations. This considerably reduces the computational time at each iteration.

At an iteration, when the DM wants to see more than one solution, the DM may select the current preferred solution based on nonformalized information about his/her preferences or may use a ranking procedure based on additional formal information.

The rest of the paper is organized as follows: Some notation and definitions are introduced in the next section. The description and properties of the scalarizing problems are given in Section 3, and the formal ranking procedure is presented in Section 4. A brief description of the proposed interactive algorithm is given in Section 5, and the research DSS and some experimental results are described in Section 6. A few concluding remarks are given in Section 7.

2. Problem formulation

Many practical problems, e.g., location-allocation, transportation, scheduling, assignment, planning problems, etc., can be formulated as MCLIP problems. For a survey of multicriteria programming formulation and the methodology applied to solve some of these problems, the reader may refer to Osleeb, Ratick [26], Current et al. [6], Blazewich et al. [3], Weber, Current [37], Ritzel et al. [30] and Ferreira et al. [8].

The MCLIP problem may be formulated as:

(1) $$\text{“max”}\{f_k(x)\}, \quad k \in K,$$

subject to:

(2) $$\sum_{j \in N} a_{ij} x_j \le b_i, \; i \in M$$

(3) $$0 \le x_j \le d_j, \quad j \in N$$

(4) $$x_j \text{ – integer}, \quad j \in N' \subset N,$$

where $f_k(x)$, $k \in K$, are linear criteria (objective functions); $f_k(x) = \sum_{j \in N} c_j^k x_j$, and “max” means that all the objective functions have to be maximized simultaneously; $x = (x_1, x_2, \ldots, x_j, \ldots, x_n)^{\mathrm{T}}$ is the vector of the decision variables; and $K = \{1, 2, \ldots, p\}$, $M = \{1, 2, \ldots, m\}$, $N = \{1, 2, \ldots, n\}$, and$N' = \{1, 2, \ldots, n' / n' \le n\}$ denote the index sets of the criteria (objective functions), the linear constraints, the decision variables and the integer decision variables, respectively.

Constraints (2)-(4) define the feasible region X_1 for the integer variables. Problem (1)-(3) is a MCLP problem, which is a relaxation of MCLIP problem. The feasible region for MCLP problem variables is denoted by X_2.

For clarity of exposition, we introduce a few definitions:

Definition 1. The feasible solution x is called an efficient solution of MCLP or MCLIP problem, if there does not exist any other feasible solution $\bar{x}$, such that the following inequalities are satisfied:

$f_k(\bar{x}) \geq f_k(x)$, for every $k \in K$ and $f_k(\bar{x}) > f_k(x)$, for at least one index $k \in K$.

Definition 2. The feasible solution x is called a weak efficient solution of MCLP or MCLIP problem if there does not exist another feasible solution $\bar{x}$ such that the following inequalities hold:

$f_k(\bar{x}) > f_k(x)$, for every $k \in K$.

Definition 3. The feasible solution x is called a (weak) efficient solution of MCLP or MCLIP problem if x is an efficient or weak efficient solution of the corresponding problem.

Definition 4. The vector $f(x) = (f_1(x),..., f_p(x))'$ is called a (weak) nondominated solution in the criteria space, if x is a (weak) efficient solution in the variable space.

Definition 5. A near (weak) nondominated solution of MCLIP problem is a feasible solution in the criteria space obtained by solving an integer scalarizing problem using an heuristic algorithm.

Definition 6. A current preferred solution is a (weak) nondominated solution of MCLIP or MCLP problem or a near (weak) nondominated solution of MCLIP problem, chosen by the DM at the current iteration. The most preferred solution of MCLIP problem is a current preferred solution of MCLIP problem that satisfies the DM to the greatest degree.

Problems MCLIP and MCLP do not possess a mathematically well-defined optimal solution. Therefore, it is necessary to select one of the (weak) nondominated solutions that satisfies the DM to the greatest degree. This choice is subjective and depends entirely on the DM.

3. Scalarizing problems

The DM evaluates the current (weak) nondominated solution of MCLIP or MCLP problem, or the near (weak) nondominated solution of MCLIP problem at each iteration. If the DM wants to look for a “better” solution, he/she sets the preferences for desired or acceptable changes in the values of some or all of the criteria. Depending on these preferences, the criteria set is partitioned into seven or fewer criteria classes $K^{>}, K^{\geq}, K^{=}, K^{<}, K^{\leq}, K^{><}$and K^{0}. Criterion $f_k(x)$, $k \in K$, may belong to one and only one of the following classes:

$k \in K^{>}$, if the DM wishes the value of criterion $f_k(x)$ to be improved;

$k \in K^{\geq}$, if the DM wishes the value of criterion $f_k(x)$ to be improved by a desired (aspiration) amount Δ_k, $\Delta_k > 0$;

$k \in K^{<}$, if the DM agrees to worsen the value of criterion $f_k(x)$;

$k \in K^{\leq}$, if the DM agrees to worsen the value of criterion $f_k(x)$ by no more than δ_k, $\delta_k > 0$;

$k \in K^{><}$, if the DM wishes the value of criterion $f_k(x)$ to lie within limits of the current value f_k, $(f_k - t_k^- \leq f_k(x) \leq f_k + t_k^+)$, $t_k^-, t_k^+ > 0$;

$k \in K^{=}$ if the DM does not want to worsen the value of criterion $f_k(x)$;

$k \in K^{0}$ if the DM agrees that the criterion $f_k(x)$ may be changed freely.

On the basis of the partition of the criteria set, we propose the following scalarizing problem to compute a (weak) nondominated solution of the MCLIP problem.

Minimize

(5) $$S(x) = \max\left[\max_{k \in K^{\geq}} \left(\bar{f}_k - f_k(x)\right) / \left|f_k'\right|, \max_{k \in K^{<} \cup K^{\leq}} \left(f_k - f_k(x)\right) / \left|f_k'\right|\right] + \max_{k \in K^{>}} \left(f_k - f_k(x)\right) / \left|f_k'\right|,$$

subject to:

(6) $$f_k(x) \geq f_k, k \in K^{>} \cup K^{=} \cup K^{\geq},$$

(7) $$f_k(x) \geq f_k - \delta_k, k \in K^{\leq},$$

(8) $$f_k(x) \geq f_k - t_k^-, k \in K^{><},$$

(9) $$f_k(x) \leq f_k + t_k^+, k \in K^{><},$$

(10) $$x \in X_1,$$

(11) $$f_k' = \begin{cases} \varepsilon, & \text{if } |f_k| \leq \varepsilon, \\ f_k, & \text{if } |f_k| > \varepsilon, \end{cases}$$

where f_k is the value of the criterion with an index $k \in K$ in the current preferred solution,

$\bar{f}_k = f_k + \Delta_k$ – the aspiration level of the criterion with an index $k \in K^{\geq}$,

f_k' – a scaling coefficient, and ε is a small positive number.

The objective function of the scalarizing problem E_1 shows that it is a sum of two values. For each $x \in X_1$, the first value is the maximum of two sets of numbers. Each number in the first set represents the difference between the aspiration level of the criterion with an index $k \in K^{\geq}$ and $f_k(x)$. Each number from the second set is the difference between the value of the criterion with an index $k \in K^{<} \cup K^{\leq}$ in the current preferred solution and $f_k(x)$. For each $x \in X_1$, the second value is the maximum of one set of numbers. Each number in this set is the difference between the value of the criterion with an index $k \in K^{>}$ in the current preferred solution and $f_k(x)$. The optimal solution of problem E_1 may assure not great improvement of the criteria from the sets $k \in K^{>} \cup K^{\geq}$ and a small deterioration of the criteria from the sets $k \in K^{<} \cup K^{\leq}$.

Theorem 1. The optimal solution of the scalarizing problem E_1 is a weak efficient solution of the MCLIP problem.

For a proof, please see the Appendix.

To obtain a weak nondominated solution for MCLP problem, we may use the scalarizing problem E_2, which is obtained from E_1 by replacing constraint (10) with constraint (12):

(12) $$x \in X_2 .$$

The optimal solution of the scalarizing problem E_2 is a weak efficient solution of MCLP problem. This follows from Theorem 1 because the nature of the variables x_i, $i = 1,...,n$, is not explicitly used to prove it.

Because the objective function of the scalarizing problem E_1 is non-differentiable, we may solve the following equivalent mixed integer programming problem E_1':

(13) $$\min(\alpha + \beta)$$

subject to:

(14) $$\alpha \geq \left(\bar{f}_k - f_k(x)\right) / \left| f'_k \right|, k \in K^{\geq},$$

(15) $$\alpha \geq \left(f_k - f_k(x)\right) / \left| f'_k \right|, k \in K^{<} \cup K^{\leq},$$

(16) $$\beta \geq \left(f_k - f_k(x)\right) / \left| f'_k \right|, k \in K^{>},$$

(17) $$f_k(x) \geq f_k , k \in K^{>} \cup K^{=} \cup K^{\geq},$$

(18) $$f_k(x) \geq f_k - \delta_k , k \in K^{\leq},$$

(19) $$f_k(x) \geq f_k - t_k^-, \ k \in K^{><},$$

(20) $$f_k(x) \leq f_k + t_k^+, \ k \in K^{><},$$

(21) $$x \in X_1 ,$$

(22) $$\alpha, \beta \text{ – arbitrary.}$$

Theorem 2. The optimal values of the objective functions of problems E_1 and E_1' are equal, i.e.,

$$\min_{x \in X_1} (\alpha + \beta) = \min_{x \in X_1} \{ \max[\max_{k \in K^{\geq}} (\bar{f}_k - f_k(x)) / | f'_k |, \max_{k \in K^{<} \cup K^{\leq}} (f_k - f_k(x)) / | f'_k |] + \max_{k \in K^{>}} (f_k - f_k(x)) / | f'_k | \}.$$

For a proof, please refer to the Appendix.

The scalarizing problem E_1' is formulated on the basis of the partition of the criteria set, implicitly done by the DM. It is a generalization of the scalarizing problems suggested in Narula, Vassilev [24], Vassileva [34], Vassilev et al. [36], and has the following properties: The first property is related to the information given by the DM. To improve the current (weak) nondominated solution, the DM may present his/her preferences not only as desired and acceptable levels (as it is in the different scalarizing problems of the reference point, Wierzbicki [39]), but also as desired and acceptable directions and intervals of change in the values of the criteria. The second property is that the improvement in the value of one criterion may not result in big loss in the value of another criterion. Thus the DM can realize the search strategy "no great benefit – little loss". The third property is that the current (weak) efficient preferred solution can be used as an initial feasible solution for the next integer programming problem E_1'. This facilitates the single criterion algorithms, especially the heuristic algorithms, because they can start with a feasible integer solution. The fourth property is related to the fact that the feasible region of problem E_1' is a part of the feasible region of MCLIP problem and depending on the values of parameters $f_k, \Delta_k, \delta_k, t_k^+, t_k^-$, this region can be relatively narrow. The solutions obtained using heuristic algorithms to solve integer programming problem E_1' may lie near the non-dominated surface of MCLIP problem. The use of near (weak) nondominated solutions may considerably reduce DM's waiting time to obtain new solutions. When applying an heuristic integer algorithm to solve scalarizing problem E_1', a set of near (weak) nondomonated solutions is obtained, ranked according to their "proximity" to the "desired" (weak) nondominated solution. Because the search strategy "no great benefit - little loss", not only the first-ranked solution, but all the remaining solutions found are comparatively close to the "desired" (weak) nondominated solution. If the DM wishes to re-rank the near (weak) nondominated solutions obtained on the basis of additional local preference information (pairwise comparison of the criteria or inter- and intra-criteria information) he/she may use a formal ranking procedure and choose the next preferred solution on the basis of the two ranked sets.

The problem E_2' is the linear programming problem obtained from E_1' by replacing constraint (21) with constraint (12). One (weak) nondominated solution of MCLP can be obtained by solving problem E_2', which is easy to solve, see Garey and Johnson (1979).

According to the reference direction approach, Korhonen and Laakso (1986), the presence of more (weak) nondominated solutions may speed-up DM's understanding of the problem solved. More than one (weak) nondominated solutions of MCLP problem can be obtained by solving parametric extension of E_2'. A parametric extension of problem E_2' denoted by $\overline{E_2'}$ may have the form, Murtagh [23].

(23) $$\min(\alpha + \beta)$$

subject to:

(24) $$f_k(x) + \left|f_k'\right|\alpha \ge \bar{f}_k + \Delta f_k t, \ k \in K^{\ge},$$

(25) $$f_k(x) + \left|f_k'\right|\alpha \ge f_k - \Delta f_k t, \ k \in K^{<} \cup K^{\le},$$

(26) $$f_k(x)+\left|f_k^{'}\right|\beta \geq f_k + \Delta f_k t\,,\ k \in K^{>}\,,$$

(27) $$f_k(x) \geq f_k\,,\ k \in K^{>} \cup K^{=} \cup K^{\geq}\,,$$

(28) $$f_k(x) \geq f_k - \delta_k\,,\ k \in K^{\leq}\,,$$

(29) $$f_k(x) \geq f_k - t_k^{-}\,,\ k \in K^{><}\,,$$

(30) $$f_k(x) \leq f_k + t_k^{+}\,,\ k \in K^{><}\,,$$

(31) $$x \in X_2\,,$$

(32) $$t \geq 0\,,$$

(33) $$\alpha, \beta \text{ – arbitrary,}$$

where Δf_k is a parameter.

The parametric problem $\overline{E_2^{'}}$ is also easy to solve, see Murtagh [23]. The first solution is supposed to best satisfy the DM's preferences. In the remaining solutions, the desired improvements and the acceptable deteriorations are increased. Depending on parameter Δf_k, $k \in K$, the parametric (weak) nondominated solutions obtained may be comparatively close, but may also differ significantly.

Let us assume that we have computed a (weak) nondominated solution of MCLP problem using scalarizing problem E_2' or $\overline{E_2^{'}}$ and wish to find a (weak) nondominated solution of MCLIP problem, which is close to the (weak) nondominated solution of MCLP problem. Let us denote a (weak) nondominated solution of MCLP problem by $\overset{?}{f} = (\overset{?}{f}_1,\ldots,\overset{?}{f}_p)^{\mathrm{T}}$.

To find a (weak) nondominated solution of MCLIP problem close to the (weak) nondominated solution $\overset{?}{f}_k$ of MCLP problem, we may solve the following Chebychev problem E_3, Wierzbicki [39]:

minimize

(34) $$S(x) = \max_{k \in K}(\overset{?}{f}_k - f_k(x)) / |\ \overset{?}{f}{}_k^{'}\ |,$$

subject to

(35) $$x \in X_1,$$

where

(36) $$\overset{?}{f}{}_k^{'} = \begin{cases} \overset{?}{f}_k, & \text{if } |\ \overset{?}{f}_k\ | > \varepsilon, \\ \varepsilon, & \text{if } |\ \overset{?}{f}_k\ | \leq \varepsilon \end{cases}$$

and ε is a small positive number.

This problem is equivalent to the following mixed integer linear programming problem E_3':

(37) $$\min \alpha$$

subject to

(38) $$\alpha \geq (\overset{?}{f}_k - f_k(x)) / | \overset{?}{f}_k^{'} |,$$

(39) $$x \in X_1,$$

$$\alpha - \text{arbitrary.}$$

4. Ranking procedure

Heuristic integer algorithms, namely, Ibaraki et al. [16], Goldberg [13], Werra, Hertz [38], Vassilev, Genova [35], Reeves [29], Pirlot [27], and Glover, Laguna [12], find solutions of the scalarizing problem E_1' that lie comparatively close to the efficient (nondominated) surface of the MCLIP problem. These near (weak) nondominated solutions are computed relatively quickly and presented to the DM for evaluation. The interruption of an exact integer algorithm, Nemhauser, Woolsey [25] (if the waiting time is too long) is also appropriate and the near (weak) nondominated solutions obtained so far can be presented to the DM for evaluation. The solution of scalarizing problem $\overline{E_2^{'}}$ using a linear parametric programming algorithm will lead to more than one (weak) nondominated continuous solutions of MCLP problem. To select the current preferred solution, the DM has three possibilities:

– to select the first-ranked near (weak) nondominated integer solution or to choose the first-ranked (weak) nondominated continuous solution;

– on the basis of additional nonformalized information, the DM may choose one solution, other than the first-ranked, from the set of ranked near (weak) nondominated integer solutions or from the set of ranked (weak) nondominated continuous solutions;

– on the basis of additional intra- and inter-crirteria information or information about the pairwise comparison of the criteria provided by the DM, the set of near (weak) nondominated integer solutions or the set of (weak) nondominated continuous solutions can be re-ranked by the RP ranking procedure. The DM may choose the first-ranked or another solution from one of these ranked sets.

Let us denote the set of solutions (alternatives) by $M_1 = (i_1, i_2, ..., i_s)$, where s is the number of alternatives computed at the current iteration and i_1 is the first feasible solution obtained when solving scalarizing problem E_1' or $\overline{E_2^{'}}$. The RP procedure includes two modules for complete ranking of the alternatives. The first module uses Promethee II outranking method, Brans, Mareschal [4]. To rank the alternatives with the help of the outranking procedure, the DM provides two types of local information. The first (intra-criteria) type of information consists of two thresholds. For each criterion $k \in K$, the DM defines an indifference threshold q_k and a preference threshold p_k. The indifference threshold q_k is equal to the difference between the values of the two criteria that have no practical significance for the DM. The preference threshold p_k is equal to the difference between the values of the two criteria that indicates that one of them is preferred over the other. The second type of local information provided by the DM is the inter-criteria information that refers to the relative importance of the criteria for the DM. This importance is expressed in weights defined by the DM.

The second module uses the AHP method, Saaty [32]. To rank the alternatives using this method, the DM provides local pairwise comparison of the criteria.

The algorithmic scheme of RP procedure is in 5 steps.

Step 1. Record near (weak) nondominated solutions of MCLIP problem or (weak) nondominated solutions of MCLP problem obtained from set M_1 as alternatives in matrix *A*.

Step 2. Ask the DM to choose which type of additional local information he/she is able or willing to provide. If he/she prefers to provide local inter- and intra-criteria information, go to Step 3; otherwise, go to Step 4.

Step 3. Ask the DM to provide the weights w_k, $k\in K$, for the criteria. The DM may use the weights defined at a previous iteration or offer new weights.

Ask the DM to provide the thresholds q_k and p_k, $k\in K$. The DM may wish to use the thresholds from at a previous iteration or offer new thresholds.

Rank the alternatives of matrix *A* by Promethee II method and go to Step 5.

Step 4. The DM provides the matrix of pairwise comparisons of the criteria. The DM may wish to use the matrix from a previous iteration, or offer a new matrix of pairwise comparisons.

Rank the alternatives of matrix *A* by AHP method.

Step 5. Present the ranked set of alternatives to the DM for evaluation and selection of the preferred solution for MCLIP or MCLP problem.

5. A partition-based interactive algorithm

A partition-based interactive algorithm to solve MCLIP problems can be developed on the basis of scalarizing problems E_1', E_2', $\overline{E_2'}$, E_3' and the ranking procedure RP. The scalarizing problems E_1' and E_3' are mixed integer programming problems. The problems of mixed integer programming are NP problems, i.e., the time for their exact solution is an exponential function of their dimensions. When solving integer problems, especially problems of large dimension (above 100 variables and constraints), heuristic algorithms, e.g., Ibaraki et al. [16], Vassilev, Genova [35], Reeves [29], Pirlot [27], Glover, Laguna [12]. Because finding an initial solution of the integer problems is as difficult as finding an optimal solution, the heuristic algorithms do not guarantee finding even an initial feasible integer solution in the general case. But, if an initial feasible integer solution is known and the feasible region is comparatively "narrow", then using heuristic algorithms, especially those that include meta-heuristics such as "tabu search", Glover, Laguna [12], some good, and in many cases, optimal integer solutions can be obtained.

The proposed interactive algorithm is learning-oriented. That is, the DM may search freely for the most preferred solution from the sets of (weak) nondominated or near (weak) nondominated solutions. For this reason during the learning phase, the DM is encourage to explore these sets, get some idea about the feasible ranges of criteria values, and some relations among the criteria. To achieve this and to overcome some computational difficulties (especially when solving a large problems), three different strategies for computing new solutions are used in the interactive algorithm. The first strategy, called integer strategy, searches for a (weak) nondominated integer solution at each iteration by solving the integer scalarizing problems exactly. The second strategy, called approximate integer strategy, searches for near (weak) nondominated integer solutions at some iterations by computing approximate solution of integer scalarizing problems. During the learning phase, and for large problems until the very end, only near (weak) nondominated solutions may be searched. The third strategy, called the mixed strategy, searches for continuous (weak) nondominated solutions by solving continuous scalarizing problems at most of the iterations, and only sometimes searches for a (weak) nondominated integer or near (weak) nondominated integer solution that is close to the continuous (weak) nondominated solution.

The integer strategy is appropriate when solving small multicriteria integer problems. The approximate integer and mixed strategies are appropriate for solving medium and large multicriteria integer problems.

The basic steps of the algorithm are 11.

Step 1. Find an initial (weak) nondominated solution of MCLP problem by setting f_k=1, $k \in K$, $\bar{f}_k = 2$, $k \in K$, and solving problem E_2'. Let it be the current preferred solution.

Step 2. Ask the DM to specify the desired or acceptable levels, directions or intervals of changes in the values of some or all of the criteria in relation to the current preferred solution.

Step 3. If the DM chooses to search for (weak) nondominated solutions of MCLP problem, execute Step 4; if the DM chooses to find (weak) nondominated solutions of MCLIP problem, go to Step 9.

Step 4. Ask the DM, if he/she wants to see more than one (weak) nondominated solutions of MCLP problem, go to Step 6. Otherwise, go to Step 5.

Step 5. Solve the scalarizing problem E_2' using linear programming algorithm to find a solution of MCLP problem. If the DM wants to see one (weak) nondominated solution of MCLIP problem close to the current solution of MCLP problem, go to Step 7. If the DM wants to see one or more near (weak) nondominated solutions of MCLIP problem close to the current solution of MCLP problem, go to Step 8. Otherwise go to Step 2.

Step 6. Ask the DM to specify parameter *s* – the number of (weak) nondominated solutions of MCLP problem, that should be saved in set M_1. Solve the scalarizing problem $\overline{E_2^{'}}$ by a linear parametric programming algorithm. Present the set M_1 to the DM for evaluation and selection of a current preferred solution. If the DM wishes, he/she may use the ranking procedure RP to aid him/her to select the current preferred solution of the MCLP problem. If the DM wants to see a (weak) nondominated solution of MCLIP problem close to the current preferred solution, execute Step 7. If the DM wants to see a near (weak) nondominated solution of MCLIP problem close to the current preferred solution, go to Step 8. Otherwise go to Step 2.

Step 7. Solve problem E_3'. Show to the DM the (weak) nondominated solution of MCLIP problem. If the DM selects this solution as a current preferred solution of MCLIP problem, go to Step 2. If this is the most preferred solution of MCLIP problem, Stop.

Step 8. Solve problem E_3'. Show to the DM near (weak) nondominated solution of MCLIP problem. If the DM selects this solution as a current preferred solution of MCLIP problem, go to Step 2. If this is the most preferred solution of MCLIP problem, **Stop**.

Step 9. If the DM choose to see one (weak) nondominated solution of MCLIP problem go to Step 11. If the DM wants to see one or more near (weak) nondominated solutions of MCLIP problem, go to Step 10.

Step 10. Ask the DM to specify *s*–the number of near (weak) nondominated solutions of MCLIP problem, which should be stored in set M_1. Solve the scalarizing problem E_1' by an heuristic integer programming algorithm and present the set M_1 to the DM for evaluation and selection. If the DM wishes, he/she may use the ranking procedure RP to help him/her to select the current preferred solution of MCLIP problem. If the current preferred solution is the most preferred solution of MCLIP problem, Stop; otherwise go to Step 2.

Step 11. Solve problem E_1'. Show the (weak) nondominated solution or near (weak) nondominated solution (if the computational process is interrupted because the computation time is too long) to the DM. If the DM approves this solution as a current preferred solution of MCLIP problem, go to Step 2. If it is the most preferred solution of MCLIP problem, Stop.

The proposed interactive algorithm for solving multicriteria linear integer problems is learning-oriented in which the DM controls the dialogue, the computations, and the stopping rules. Linear parametric programming problems (scalarizing problems $\overline{E_2'}$) are often used when solving medium or large MCLIP problems. A few mixed integer linear programming (scalarizing problems E_1' and E_3') are also solved when the DM either feels uncomfortable to work with continuous variables, or when he/she is looking for an integer solution close to the current preferred continuous solution. When solving a medium or a large MCLIP problems, it is appropriate (especially, in the learning phase) to solve the integer scalarizing problems by heuristic algorithms. Many heuristic algorithms operate well in a “narrow feasible region” and a known initial feasible integer solution helps them to find good and, in many cases, optimal solutions of the integer scalarizing problems. The evaluation of more than one solution, even near (weak) nondominated solutions, enables the DM to learn quickly about the problem being solved. The use of RP procedure may help the DM to evaluate these solutions.

6. Implementation and experimental results

We have developed a research decision support system, called MOLIP, for solving multicriteria linear integer programming problems. It consists of three main parts: a control program, interface modules and optimization modules.

The basic functions of the control program can be divided in three groups. The first group includes the possibility of using resources of MS Windows operating system in the program environment. The second group of control program includes creating, modifying, and saving files containing the description and the interactive process of solving MCLIP problems, and the localization of different errors. The third group of control program includes the visualization of important information for the DM and about the system operation as a whole.

The interface modules realize the dialogue between the DM and MOLIP system during the entry and correction of the input data, the interactive process of problem solution, and the dynamic visualization of the main parameters of the process.

The optimization modules implement the proposed partition-based interactive algorithm. These modules include one exact branch-and-bound algorithm (based on one-sided branching, including heuristics, Nemhauser, Woolsey [25], and three heuristic algorithms, namely, algorithm of Ibaraki et al. [16], algorithm of internal feasible directions, Vassilev, Genova [35], and a "tabu search" type algorithm, Gouljashki, Vassilev [14], for solving single criterion linear integer programming problems. Linear and linear parametric programming algorithms, AHP, and Promethee II procedure are also included in the optimization modules.

The software for all single criterion algorithms were developed at the Institute of Information Technologies – BAS. Particular attention has been paid to the heuristic algorithms. Extensive tests of the algorithm of Ibaraki et al. [16] and the algorithm of internal feasible directions were conducted in Vassilev, Genova [35] with respect to the speed of the algorithms and the quality of the solutions obtained. The algorithm of Ibaraki et al. [16] is one of the first good heuristic algorithms for solving linear of integer programming problems. It formulates and solves linear sub-problems on the basis of several heuristics and rounds off the values of the variables to integer values. It is especially useful for finding initial feasible integer solutions. The algorithm of the internal feasible integer directions belongs to the class of component algorithms and improves the feasible integer solutions comparatively quickly. The quality of the solutions generated by these two algorithms is generally good. For example, in 66% of the cases the algorithm of Ibaraki et al. (1974) generated a solution (that is, the criteria values) within 3% of the optimal criteria values, and in 85% of the cases, the solution was within 10% of the optimal criteria values, see Vassilev, Genova [35]. The performance of the internal feasible integer directions algorithm was similar. Better results were obtained by "tabu search" type heuristic algorithm in some cases when an initial feasible integer solution existed and feasible region was "narrow", see Gouljashki, Vassilev [14].

The speed and the quality of the solutions obtained by the three heuristic algorithms are comparatively good. Nevertheless, the quality of the solution obtained by each algorithm depends, to a great extent, on the type and the structure of the problem. To expand the class of the problems that can be solved with high probability quickly and efficiently, these three algorithms in DSS MOLIP are used together as one generalized algorithm, first executing the algorithm of Ibaraki et al., then the algorithm of internal feasible directions, and at the end – a "tabu search" type algorithm. On the basis of the properties described above, we decided to use this order for operation of the three algorithms.

Recently a number of new heuristic algorithms have been proposed to solve single criterion integer problems, which use some of the meta-heuristics, namely "evolutionary search", Reeves [29], "simulated annealing", Pirlot [27], and "tabu search", Glover, Laguna [12]. In DSS MOLIP, we have included the heuristic algorithms described earlier because we have their software and have gained significant experience in relation to their computational behaviour.

Two types of tests were conducted on the research DSS MOLIP. The first test was directed toward computational analysis of the single criterion integer algorithms included in the system, where as the second test was performed to learn some aspects of DM's behavior in the operation of the proposed partition-based interactive algorithm.

In the first test, we solved several multicriteria problems. The purpose of this test was to observe the difference between the computing times for obtaining exact and approximate integer solutions, and the quality of the obtained approximate integer solutions. These problems were taken from Vassilev, Genova [35] and by adding two new criteria to each problem, we modified them as three criteria problems.

The number of constraints and variables for only eight problems are given in Table 1. In the table, we also give percent of the non-zero elements for all the problems. It may be noted that the first five problems are binary where as the last three are all integer. For the first three problems, the coefficients of the variables were zero or one; for the fourth problem, they ranged over zero to 100; and for the fifth problem over zero to 200. The coefficients for problems 6, 7 and 8, ranged over zero to 600, zero to 1000, and zero to 100, respectively. For each of these eight problems, three iterations of the interactive algorithm were executed using the exact or generalized heuristic single criterion integer algorithms. Three approaches were used, namely: 1) an exact algorithm at each iteration, denoted by EEE ; 2) a generalized heuristic algorithm at the first two iterations and an exact algorithm at the third iteration, denoted by HHE and 3) a generalized heuristic algorithm at each iteration, denoted by HHH. For these approaches we report the average CPU time (in seconds) for performing these three iterations in Table 1. The computations were performed on a PC Pentium II, 400 MHZ, 128 MB RAM. In the last two columns, we present the average deviation and average percent deviation of the approximate solutions from the exact solutions for the second and third approach, respectively.

Table 1. Computational Results

Problem number	Number of		Percent non-zero elements	Computational Time (CPU seconds)			Δf	Percent Δf
	Constraints	Variables		EEE	HHE	HHH		
1	15	15	50	1.2	0.5	0.30	0	0.0
2	31	31	50	24.3	8.7	0.60	1	6.1
3	50	32	10	64.5	24.8	0.85	0	0.0
4	50	50	30	4860.0	1632.0	11.70	3	2.6
5	87	48	10	9360.0	3348.0	116.00	0	0.0
6	12	50	10	31.2	11.4	1.30	10	4.3
7	30	60	100	3720.0	1210.0	10.10	0	0.0
8	82	108	50	16831.0	4852.0	64.00	16	3.9

From Table 1, we observe that the average CPU time needed for the three iterations to compute an exact (weak) nondominated or near (weak) nondominated solutions differs considerably. The larger the dimension of the problem, the greater is this difference. Clearly, this difference reflects the lack of guarantee in the quality of the near (weak) nondominated solution. During the learning phase, this compromise in the quality of the solutions may be acceptable, while this may be the only practical option for solving large problems.

A second test was performed to learn how DMs familiar with different interactive algorithms may use the possibilities offered by the proposed algorithm with DSS

MOLIP to solve the MCLIP problems. Ten students of economics and mathematics at Sofia University and the New Bulgarian University, who had learned the theory and practice of multicriteria optimization and multicriteria decision making were used as DMs. Each student was given a different set of three problems, from the problems used in the first test, to solve. The students were instructed to find a solution such that the relative difference between the single objective optimum value of a criterion and its value in the most preferred solution should be in a non decreasing order of the criterion number. Furthermore, the values of the criteria in the most preferred solution should be relatively close to the ideal point. Since the proposed interactive algorithm is learning-oriented, they had complete freedom to express their preferences, to use continuous or integer single criterion algorithms, to apply the ranking procedure RP in the selection of the current and the most preferred solution. The decision makers worked independently and their experience in solving different problems may be summarized as follows:

1. The DMs used all the choices offered by the DSS MOLIP in stating their local preferences. However, at different phases of the solution process, some options were used more often than others. For example, in the initial solution phase, the DMs preferred to set more desired or acceptable directions of the criteria, where as in the final phase they mainly preferred to set desired or acceptable amount of changes in terms of levels and intervals of the criteria values. This behavior is understandable because in the initial phase every DM desires to know a rough estimate of the ranges for changes of the separate criteria, while in the final phase, he/she wants to maked more precise search for the most preferred solution.

2. In the initial phase, the DMs showed tendency to mainly solve continuous scalarizing problems or to solve integer scalarizing problems approximately. In the final phase, they chose exact solutions of the integer scalarizing problems. For large problems (after the DMs realized how long it takes to solve an integer scalarizing problem), many preferred not to use the exact single criterion algorithm.

3. At a given iteration, when the DMs desired more solutions to choose the current preferred solution, he/she used the RP ranking procedure to rank these solutions. They realized early on that it was easier to set additional preference information than to compare many solutions directly.

4. For each problem, most DMs found the same or very close most preferred solutions.

Based on these observations, we believe that the proposed interactive algorithm possesses some relatively good characteristics. Still better results could be obtained if DSS MOLIP were further developed, with improved interface with the DM, and included more recent single criterion algorithms.

7. Concluding remarks

We have proposed a learning-oriented interactive algorithm to solve multicriteria linear integer programming problems. The algorithm offers the DM flexibility to express his/her preferences with respect to the current preferred solution. At each iteration, the DM has the choice to compute one or more (weak) nondominated (continuous or integer) solutions or near (weak) nondominated integer solutions. The DM is encouraged, especially in the learning phase or when solving large problems, to solve continuous scalarizing problems or the integer scalarizing problems approximately at many iterations. This considerably reduces the computational time at each iteration. When the

DM wants to see more than one continuous or integer solutions at an iteration, the DM may select the current preferred solution based on nonformalized information about his/her preferences or may use a ranking procedure based on additional formal information.

We have developed a research decision support system based on the proposed algorithm. Our experimental results confirm that the DMs use all the choices in stating their preferences, some choices more often than others, and different choices at different stages of solving the problem. Furthermore, the computational effort and time are reduced considerably by using continuous and heuristic integer algorithms in the learning phase and for solving large MCLIP problems.

Appendix

Theorem 1. The optimal solution of the scalarizing problem E_1 is a weak efficient solution of the MCLIP problem.

Proof.

Let $K^{\geq}$ or $K^{>} \neq \varnothing$.

Let $x^{*} \in X_1$ be an optimal solution of problem E_1. Then the following conditions are satisfied:

(40) $$S(x^{*}) \leq S(x), \quad \text{for every} \quad x \in X_1$$

and

$$f_k(x^{*}) \geq f_k, k \in K^{>} \cup K^{=} \cup K^{\geq},$$

$$f_k(x^{*}) \geq f_k - \delta_k, k \in K^{\leq},$$

$$f_k(x^{*}) \geq f_k - t_k^{-}, k \in K^{><},$$

$$f_k(x^{*}) \leq f_k + t_k^{+}, k \in K^{><}.$$

Let us assume that $x^{*} \in X_1$ is not a weak efficient solution of the MCLIP problem. There must exist $x' \in X_1$ for which:

(41) $$f_k(x^{*}) < f_k(x'), \; k \in K,$$

and

$$f_k(x') \geq f_k, k \in K^{>} \cup K^{=} \cup K^{\geq},$$

$$f_k(x') \geq f_k - \delta_k, k \in K^{\leq},$$

$$f_k(x') \geq f_k - t_k^{-}, \; k \in K^{><},$$

$$f_k(x') \leq f_k + t_k^{+}, \; k \in K^{><}.$$

The condition (41) follows from the definition of weak efficient solution and the remaining conditions follow from the fact that x' must satisfy the constraints of the scalarizing problem E_1.

After transformation of the objective function $S(x)$ of the scalarizing problem E_1, using inequalities (41), the following relation is obtained:

$$(42)\quad S(x') = \max\Big[\max_{k\in K^{\geq}}(\bar{f}_k - f_k(x'))/|f_k'|, \max_{k\in K^{<}\cup K^{\leq}}(f_k - f_k(x'))/|f_k'|\Big] + \max_{k\in K^{>}}(f_k - f_k(x'))/\left|f_k'\right| = \max\Big[\max_{k\in K^{\geq}}((\bar{f}_k - f_k(x^*)) + (f_k(x^*) - f_k(x')))/|f_k'|,$$

$$\max_{k\in K^{<}\cup K^{\leq}}((f_k - f_k(x^*)) + (f_k(x^*) - f_k(x')))/|f_k'|\Big] +$$
$$+\max_{k\in K^{>}}((f_k - f_k(x^*)) + (f_k(x^*) - f_k(x')))/\left|f_k'\right|$$
$$< \max\Big[\max_{k\in K^{\geq}}(\bar{f}_k - f_k(x^*))/|f_k'|, \max_{k\in K^{<}\cup K^{\leq}}(f_k - f_k(x^*))/|f_k'|\Big] + \max_{k\in K^{>}}(f_k - f_k(x^*))/\left|f_k'\right|$$
$$\equiv S(x^*).$$

From (42), it follows that $S(x') < S(x^*)$, which contradicts (40). Hence, $x^* \in X_1$ is a weak efficient solution of the MCLIP problem. Therefore, the corresponding solution $f(x^*)$ is a weak nondominated solution.

Theorem 2. The optimal values of the objective functions of problems E_1 and E_1' are equal, i.e.

$$\min_{x\in X_1}(\alpha+\beta) = \min_{x\in X_1}\{\max[\max_{k\in K^{\geq}}(\bar{f}_k - f_k(x))/|f_k'|, \max_{k\in K^{<}\cup K^{\leq}}(f_k - f_k(x))/|f_k'|]$$
$$+\max_{k\in K^{>}}(f_k - f_k(x))/|f_k'|\}.$$

Proof.

From (14), $\alpha \geq (\bar{f}_k - f_k(x))/|f_k'|,\ k\in K^{\geq}$.

Since this inequality is true for every $k \in K^{\geq}$, it is also true that

$$(43)\qquad \alpha \geq \max_{k\in K^{\geq}}(\bar{f}_k - f_k(x))/|f_k'|.$$

From (15), $\alpha \geq (f_k - f_k(x))/|f_k'|,\quad k\in K^{<}\cup K^{\leq}$.

Because this inequality is true for every $k\in K^{<}\cup K^{\leq}$, it is also true that

$$(44)\qquad \alpha \geq \max_{k\in K^{<}\cup K^{\leq}}(f_k - f_k(x))/|f_k'|.$$

From (43) and (44), we can write

$$(45)\qquad \alpha \geq \max[\max_{k\in K^{\geq}}(\bar{f}_k - f_k(x))/|f_k'|, \max_{k\in K^{<}\cup K^{\leq}}(f_k - f_k(x))/|f_k'|].$$

From (16) follows

$$(46)\qquad \beta \geq \max_{k\in K^{>}}(f_k - f_k(x))/|f_k'|.$$

If the left and right sides of inequalities (45) and (46) are summed, we obtained:

$$(\alpha+\beta) \geq \max[\max_{k\in K^{\geq}}(\bar{f}_k - f_k(x))/|f_k'|, \max_{k\in K^{<}\cup K^{\leq}}(f_k - f_k(x))/|f_k'|]$$
$$+\max_{k\in K^{>}}(f_k - f_k(x))/|f_k'|.$$

Let $x^* \in X_1$ be an optimal solution of problem E_1'. Then

$$\begin{aligned}\min_{x\in X_1}(\alpha+\beta) = \max[\max_{k\in K^{\geq}}(\bar{f}_k - f_k(x^*))/|f_k^{'}|, \max_{k\in K^{<}\cup K^{\leq}}(f_k - f_k(x^*))/|f_k^{'}|]\\ + \max_{k\in K^{>}}(f_k - f_k(x^*))/|f_k^{'}|.\end{aligned} \tag{47}$$

The right side of (47) can be rewritten as:

$$\begin{aligned}\min_{x\in X_1}\{\max[\max_{k\in K^{\geq}}(\bar{f}_k - f_k(x))/|f_k^{'}|, \max_{k\in K^{<}\cup K^{\leq}}(f_k - f_k(x))/|f_k^{'}|]\\ + \max_{k\in K^{>}}(f_k - f_k(x))/|f_k^{'}|\},\end{aligned}$$

which proves the theorem.

References

1. Alves, M. J., J. Climaco. An interactive reference point approach for multiobjective mixed-integer programming using branch-and-bound. – European Journal of Operational Research, **124**, 2000, 478-494.
2. Benayoun, R., J. de Montgolfier, J. Tergny, O. Larichev. Linear programming with multiple objective functions: Step method (STEM). – Mathematical Programming, **1**, 1971, 366-375.
3. Blazewich, J., M. Drior, J. Weglarz. Mathematical programming formulations for machine scheduling: A survey. – European Journal of Operational Research, **51**, 1991, 283-300.
4. Brans, J., B. Mareschal. The PROMETHEE methods for MCDM: The PROMCALC, GAIA and BANKADVISER software. – In: Readings in Multiple Criteria Decision Aid. A. Carlos, C. Bana e Costa (Eds.). Berlin, Springer Verlag, 1990, 216-252.
5. Climaco, J., G. Ferreira, M. E. Captivo. Multicriteria integer programming: An overview of the different algorithmic approaches. – In: Multicriteria Analysis. J. Climaco (Ed.). Berlin, Springer Verlag, 1997, 248-258.
6. Current, J., H. Min, D. Schilling. Multiobjective analysis of facility location decisions. – European Journal of Operational Research, **49**, 1990, 295-307.
7. Durso, A. An interactive combined branch-and-bound / Tchebycheff algorithm for multiple criteria. – In: Multiple Criteria Decision Making. A. Goicoechea, L. Duckstein, S. Zionts (Eds.). Berlin, Springer Verlag, 1992, 107-122.
8. Ferreira, C., B. S. Santos, M. E. Captivo, J. Climaco, C. C. Silva. Multiobjective location of unwelcome or central facilities involving environmental aspects: A prototype of a decision support system. – Belgian Journal of Operations Research, Statistics and Computer Science, **36**, 1996, 159-172.
9. Gabbani, D., M. Magazine. An interactive heuristic approach for multiobjective integer programming. – Journal of Operational Research Society, **37**, 1986, 285-291.
10. Gardiner, L. R., D. Vanderpooten. Interactive multiple criteria procedures: Some reflections. – In: Multicriteria Analysis. J. Climaco (Ed.). Berlin, Springer Verlag, 290-301.
11. Garey, M., D. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco, W. H. Freeman, 1997.
12. Glover, F., M. Laguna. Tabu Search. Boston, Kluwer Academic Press, 1997.
13. Goldberg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley Publishing Company, Inc., 1989.
14. Gouljashki, V., V. Vassilev. Two nonlinear integer optimization algorithms – comperative analysis of test results. – Problems of Engineering Cybernetics and Robotics, **42**, 1995, 15-23.
15. Hajela, P., C. J. Shih. Multiobjective optimum design in mixed integer and discrete design variable problems. – AIAA Journal, **28**, 1990, 670-675.

16. Ibaraki, T., T. Ohashi, H. Mine. A heuristic algorithm for mixed integer programming problems. – Mathematical Programming Study, **2**, 1974, 115-136.
17. Kaliszewski, I., W. Michalowski. Searching for psychologically stable solutions of multiple criteria decision problems. – European Journal of Operational Research, **118**, 1999, 549-562.
18. Karaivanova, J., P. Korhonen, S. Narula, J. Wallenius, V. Vassilev. A reference direction approach to multiple objective integer linear programming. – European Journal of Operational Research, **24**, 1995, 176-187.
19. Karwan, M. H., S. Zionts, B. Villareal, R. Ramesh. An improved interactive multicriteria integer programming algorithm. – In: Decision Making with Multiple Objectives. Y. Haimes, V. Chankong (Eds.). Springer Verlag, 1985, 261-271.
20. Korhonen, P., J. Laakso. A visual interactive method for solving the multiple criteria problem. – European Journal of Operational Research, **24**, 1986, 277-287.
21. Marcotte, O., R. Soland. 1986. An interactive branch and bound algorithm for multicriteria optimization. – Management Science, **32**, 1986, 61-75.
22. Miettinen, K., M. M. Makela. Interactive multiobjective optimization system WWW – NIMBUS on the Internet. – Computer and Operations Research, **27**, 2000, 709-723.
23. Murtagh, B. A. Advanced Linear Programming: Computation and Practice. New York, McGraw-Hill, 1981.
24. Narula, S. C., V. Vassilev. An interactive algorithm for solving multiple objective integer linear programming problems. – European Journal of Operational Research, **79**, 1994, 443-450.
25. Nemhauser, G. L., L. A. Woolsey. 1988. Integer and Combinatorial Optimization. New York, John Wiley and Sons, 1988.
26. Osleeb, J. P., S. J. Ratick. 1983. A mixed ineger multiple objective programming model to analyze coal handling in New England. – European Journal of Operational Research, **12**, 1983, 302-313.
27. Pirlot, M. General local search methods. – European Journal of Operational Research, **92**, 1996, 493-511.
28. Ramesh, R., S. Zionts, H. Karwan. A class of practical interactive branch and bound algorithms for multicriteria integer programming. – European Journal of Operational Research, **26**, 1986, 161-172.
29. Reeves, C. R. Genetic algorithms and neighbornhood search. – In: Evolutionary Computing. T. Fogarty (Ed.). Berlin, Springer Verlag, 1994, 115-130.
30. Ritzel, B. J.,J. W. Eheart, S. Ranjithan. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. – Water Resources Research, **30**, 1994, 1589-1603.
31. Steuer, R. An interactive multiple objective linear programming procedure. – In: TIMS Studies in the Management Sciences. M. K. Star, M. Zeleny (Eds.). Vol 6. North Holland, 1977, 225-339.
32. Saaty, T. The Analytical Hierarchy Process. New York, McGraw-Hill, 1980.
33. Teghem, J., P. L. Kunsh. Interactive methods for multiobjective integer linear programming. – In: Large Scale Modelling and Interactive Decision Analysis. G. Fandel, M. Grauer, A. Karzanski, A. P. Wierzbicki (Eds.). Berllin, Springer Verlag, 1985, 75-87.
34. Vassileva, M. Scalarizing problems of multiobjective linear programming problems. – Problems of Engineering Cybernetics and Robotics, **50**, 2000, 54-64.
35. Vassilev, V., K. Genova. An algorithm of integer feasible directions for linear integer programming. – European Journal of Operational Research, **52**, 1991, 203-214.
36. Vassilev, V., S. Narula, V. Gouljaski. An interactive reference directions algorithm for solving multiobjective convex nonlinear integer programming problems. – International Transactions in Operational Research, **8**, 2001, 367-380.
37. Weber, C. A. J. Current. 1993. A multiobjective approach to vendor selection. – European Journal of Operational Research, **68**, 1993, 173-184.
38. Werra, D., A. Hert. Tabu search techniques: Tutorial and an application to neural networks. – OR Spectrum, 1989, 131-141.
39. Wierzbicki, A. P. 1980. The use of reference objectives in multiobjective optimization. – In: Multiple Criteria Decision Making: Theory and Application. Lecture Notes in Economics and Mathematical Systems. G. Fandel, T. Gal (Eds.). Vol. 177. Berlin, Springer Verlag, 1980, 468-486.

Интерактивен алгоритъм за решаване на задачи на многокритериалното линейно целочислено програмиране

Васил Василев[1], *Субхаш Нарула*[2], *Марияна Василева*[1], *Красимира Генова*[1]

[1] *Институт по информационни технологии, 1113 София*
[2] *School of Business, Virginia Commonwealth University, Richmond, USA*

(Резюме)

Предлага се интерактивен алгоритъм, ориентиран към обучение, за решаване на задачи на многокритериалното линейно целочислено програмиране (МКЛЦП), които се разглеждат като многокритериални задачи за вземане на решение. На всяка итерация лицето, вземащо решение (ЛВР), може да раздели множеството на критериите най-много в седем класа, а именно: подобрение, подобрение с желана стойност, влошаване, влошаване до определена степен, невлошаване, промени, позволени в определен интервал, и свободни промени. Въз основа на разделянето на множеството на критериите, се формулират два типа скаларизиращи задачи – на линейното и на смесеното целочислено програмиране. На повечето итерации се намират едно или повече (слаби) недоминирани решения на непрекъснатата релаксация на задачата на МКЛЦП. Само на някои итерации се решава смесена целочислена скаларизираща задача, за да се намерят едно или повече (слаби) недоминирани или близки (слаби) недоминирани решения (близо до недоминирата повърхност на задачата на МКЛЦП). При определена итерация, ако ЛВР желае да види повече от едно решение, той/тя може да избере предпочитано решение на основата на неформализирана информация за неговите предпочитания или да използва процедурата за подреждане, основана на допълнителна формална информация. На основата на предложения алгоритъм е разработена изследователска система за подпомагане вземането на решения.