On the Ray-based procedure for determining initial bound in branch and bound method

Bajalinov Erik, Rácz Anett
Debrecen University, Hungary

VOCAL - 2008, Veszprém

2008.12.15

Main methods of integer programming:
1. Branch and Bound
2. Gomory cutting plane
3. Branch and Cut

Alternative methods:
1. Lagrangian relaxation method
2. **Ray-method**

Ray-method related investigations:
Computer Center of Russian Academy of Science.
1. Khachaturov, V.R., Mirzoyan, N.A.: *Solving problems of integer programming with ray-method*. Notes on applied mathematics, Computer Center of Soviet Academy of Science, 1987.
2. Khachaturov, V.R.: *Combinatorial methods and algorithms for solving large-scale discrete optimization problems*. Moscow, Nauka, 2000.

The structure of the our talk

1. Briefly on the original Ray-method (how does it work)

2. Main idea of the procedure proposed
   Ray definition, Search algorithm

3. Implementation issue

4. Computational results

5. Difficulties and further investigations

Original Ray-method

Given the following (non-) linear programming problem in a common form

$$f(x) \to \text{min} \tag{1}$$

subject to

$$x \in S \subset R^n \tag{2}$$

$$x_j - \text{ integer}, \quad j = 1, 2, \ldots, n, \tag{3}$$

where $x = (x_1, x_2, \ldots, x_n)$, and S denotes bounded convex feasible set of relaxation problem. Here and in what follows we will say that point x is integer if all its elements x_j are integer.

Briefly on the original Ray-method

Definition. *Let point $x^0 \in R^n$ be an integer feasible solutions of problem (1)-(3). We will say that integer point $x' \in R^n$ is an element of set $O(x^0)$, i.e. $x' \in O(x^0)$ if values*

$$|x_1^0 - x_1'|, \;\; |x_2^0 - x_2'|, \;\; \dots\;, \;\; |x_n^0 - x_n'|$$

are relative primes.

Graphical illustration: f1.bmp, f2.bmp, f3.bmp, f4.bmp, f5.bmp, f5a.bmp, f5b.bmp

Main properties of the set $O(x^0)$

Proposition. *Point x^0 is not element of set $O(x^0)$, i.e.*

$$x^0 \notin O(x^0) \; .$$

Proposition. *If point $x' \in O(x^0)$, then on the straight line segment (x^0, x') there is no integer point, i.e. there is no point such that*

$$x \; = \; x^0 + \lambda(x' - x^0), \;\; 0 < \lambda < 1 \; .$$

Main theorems

Theorem. *Any integer point belongs to the ray which begins at point x^0 and passes through one of the points from set $O(x^0)$.*

Let $S(x^0)$ denote such subset of feasible set S where $f(x)$ is strictly less than $f(x^0)$, i.e.

$$S(x^0) \;=\; \{x \in S \mid f(x) < f(x^0)\}\;.$$

Using this notation the following criteria of optimality can be formulated.

Theorem. *Let point x^0 be a feasible integer solution for problem (1)-(3). Then point x^0 is optimal solution for problem (1)-(3) if and only if*

$$O(x^0) \cap S(x^0) = \emptyset\;. \tag{4}$$

General scheme of the original Ray-method

Let x^0 be an integer feasible solution for problem (1)-(3). In accordance with the main idea of the method we have to find such an integer feasible point $x' \in O(x^0)$ that $f(x') < f(x^0)$, i.e. $x' \in O(x^0) \cap S(x^0)$. Obviously, if $O(x^0) \cap S(x^0) = \emptyset$, it means that point x^0 is an optimal solution for problem (1)-(3). Stop. The problem is solved.

Suppose that there exists such point $x' \in O(x^0) \cap S(x^0)$. In this case we have to solve the following one-variable optimization problem

$$f(\lambda) = f(x^0 + \lambda(x' - x^0)) \to \min \tag{5}$$

subject to

$$x^0 + \lambda(x' - x^0) \in S \, , \tag{6}$$

$$\lambda \geq 0 \ . \tag{7}$$

Obviously, problem (5)-(7) is solvable. Let λ_{min} be optimal solution for problem (5)-(7) and $[\lambda_{min}]$ denote its integer part. Here it may happen that

$$x^0 + ([\lambda_{min}] + 1)(x' - x^0) \notin S \ , \tag{8}$$

or

$$f(x^0 + ([\lambda_{min}] + 1)(x' - x^0)) \geq f(x^0 + [\lambda_{min}](x' - x^0)) \ . \tag{9}$$

Then we construct a new integer feasible point x'' using the following rule:

$$x'' = \begin{cases} x^0 + [\lambda_{min}](x' - x^0), & \text{if takes place (8) or (9)} \\ x^0 + ([\lambda_{min}] + 1)(x' - x^0), & \text{otherwise} \end{cases}$$

In other words, using minimization problem (5)-(7) we search in feasible set S a minimal value of function $f(x)$ on the ray which begins from point x^0 and passes through point x'. Using the optimal solution obtained then on the same ray we have to find such integer point x'' that provides objective value most close to $f(\lambda_{min})$. Next, we denote point x'' by x^0 and repeat the process. The new set $S(x^0)$ differs from the previous one with only one constraint

$$f(x) \leq f(x^0) = f(x'')$$

Since number of integer points in feasible set S is bounded, the process will terminate in finite number of iterations.

Implementation issue

In contrast to transparency of the main idea of the method and its mathematical background there are serious difficulties with its implementation and computational efficiency. The main and the most hard of them is checking optimality criteria for a given feasible integer point x^0 since set $O(x^0)$ may contain too many integer points to be checked. As it was mentioned above, the corresponding numbers (the elements of the points to be checked) are relative primes, so the determining of these points generally speaking is a very hard and computationally very expensive operation. This is why the developers of the method for numerical experiments used not the process described above but its different approximate variants (mostly with different rules for determining the ray) and tested it using problems of relatively small size (up to 100×120).

Proposed algorithm

Integer linear programming problem to solve:

$$P(x) = \sum_{j=1}^{n} p_j x_j \to \max \tag{10}$$

st.

$$\sum_{j=1}^{n} a_{ij} x_j \leq (=, \geq) b_i, \quad i = 1, 2, \ldots, m \tag{11}$$

$$x_j \geq 0, \quad \text{integer}, \quad j = 1, 2, \ldots, n \tag{12}$$

Assume **maximization** relaxation problem is solvable and vector

$$x^{max} = (x_1^{max},\ x_2^{max},\ \dots,\ x_n^{max})$$

is its optimal solution.

Assume **minimization** relaxation problem is solvable and vector

$$x^{min} = (x_1^{min},\ x_2^{min},\ \dots,\ x_n^{min})$$

is its optimal solution.

Also, we suppose that vector x^{max} is not integer.

Ray:

$$R = x^{max} + \lambda(x^{min} - x^{max}); \quad 0 \le \lambda \le 1$$

Graphical illustration:Pictures/ray1.bmp

Obviously, for any point $x \in R$ we have the following coordinates:

$$x_j = x_j^{max} + \lambda(x_j^{min} - x_j^{max}), \quad j = 1, 2, \ldots, n$$

Moreover, since the feasible set S is convex,

$$\forall x \in R \quad \Rightarrow x \in S$$

Consider the unit cube U^{max}, which contains the point x^{max}

$$\left.\begin{array}{c} [\,x_1^{max}\,] \leq x_1 \leq [\,x_1^{max}\,] + 1 \\ [\,x_2^{max}\,] \leq x_2 \leq [\,x_2^{max}\,] + 1 \\ \vdots \\ [\,x_n^{max}\,] \leq x_n \leq [\,x_n^{max}\,] + 1 \end{array}\right\} \tag{13}$$

Graphical illustration:Pictures/ray2.bmp

Now we formulate and solve the following maximization LP problem:

$$\lambda \longrightarrow \text{max} \tag{14}$$

$$\left.\begin{array}{c} x_1 = x_1^{max} + \lambda(x_1^{min} - x_1^{max}) \\ x_2 = x_2^{max} + \lambda(x_2^{min} - x_2^{max}) \\ \vdots \\ x_n = x_n^{max} + \lambda(x_n^{min} - x_n^{max}) \end{array}\right\} \quad : x \in R \tag{15}$$

$$\left.\begin{array}{c} [\,x_1^{max}\,] \leq x_1 \leq [\,x_1^{max}\,] + 1 \\ [\,x_2^{max}\,] \leq x_2 \leq [\,x_2^{max}\,] + 1 \\ \vdots \\ [\,x_n^{max}\,] \leq x_n \leq [\,x_n^{max}\,] + 1 \end{array}\right\} \quad : x \in U^{max} \tag{16}$$

Obviously, problem (14)-(16) is solvable.

Graphical illustration:Pictures/ray3.bmp

Let vector $x' = (x'_1, x'_2, \ldots, x'_n)$ be the optimal solution of problem (14)-(16) and the optimal value of the objective function (14) is λ'.

Consider the following two possible cases:
Case 1: $\lambda' = 1$: Obviously, in this case $x' = x^{min}$.
1.1: If point x' is integer (Graphical illustration: Pictures/ray4.bmp), then x' is feasible integer solution for original problem (10)-(12) and value $P(x')$ may be used as initial bound for branch and bound method. **Stop**.
1.2: If point x' is not integer (Graphical illustration: Pictures/ray5.bmp) it may occur that feasible set contains some integer points but the ray does not point it. In this case the only possibility to complete the search is to check all vertices of the unit cube U^{max} for feasibility (up to 2^n points to check !!!).

Case 2: $\lambda' < 1$: In this case point x' "*perforates*" the surface (facet or edge) of unit cube U^{max}. (Graphical illustration: Pictures/ray6.bmp).

In this stage we search in the unit cube U^{max} feasible vertices. Here we propose the following two possible procedures.

Procedure 1. Since point x' belongs to one of facets of unit cube U^{max} it means that at least one coordinate of perforation point x' is integer. Let J' denote the set of indices for integer elements of vector x', i.e.

$$J' = \{j : x'_j = [x'_j]\}$$

Now we construct set $E(x')$ of such vertices of unit cube U^{max} which belong to the perforated face:

$$E(x') = \{x \in U^{max} : x_j = \left\{ \begin{array}{ll} [x'_j] \text{ or } [x'_i] + 1, & \text{if } j \notin J' \\ x'_j, & \text{if } j \in J' \end{array} \right\}\} \quad (17)$$

Obviously, the total number of such vertices is 2^{n-k}, where k is number of elements in set J'. (Graphical illustration: Pictures/ray7.bmp)

If set $E(x')$ contains feasible points, i.e.

$$\exists x' \in E(x') \text{ that } x' \in S,$$

that is $E(x') \bigcap S \neq \emptyset$, we choose such a feasible vertex $x'' \in E(x')$ which provides maximal value for objective function (10), i.e.

$$P(x'') \geq P(x), \;\; \forall x \in E(x')$$

This value $P(x'')$ may be used as initial bound for branch and bound method. **Stop**.

Note: Vertex search limited by the set $E(x')$ may not be complete since unit cube may contain feasible vertices out of $E(x')$ and, hence, initial bound may be not the best for current unit cube. See graphical illustration. (Graphical illustration: Pictures/ray9.bmp)

Procedure 2. Let us introduce the following new variables:

$$y_j = x_j - [x_j^{max}], \ \ j = 1, 2, \ldots, n \tag{18}$$

and then replace original variables $x_j, \ j = 1, 2, \ldots, n,$ in the original LP problem (10)-(12) with the new variables y_j. We obtain the following LP problem:

$$P(y) = \sum_{j=1}^{n} p_j y_j + p_0 \to \max \tag{19}$$

st.

$$\sum_{j=1}^{n} a_{ij} y_j \leq (=, \geq) b'_i, \ \ i = 1, 2, \ldots, m \tag{20}$$

$$y_j = 0/1, \quad j = 1, 2, \ldots, n \tag{21}$$

where

$$p_0 = \sum_{j=1}^{n} p_j [x_j^{max}]; \quad b'_i \ = \ b_i - \sum_{j=1}^{n} a_{ij} [x_j^{max}], \ \ i = 1, 2, \ldots, m$$

then solve this problem using **Balas**-type procedure (implicit enumeration).

The transformation (18) shifts simultaneously the unit cube U^{max} and feasible set S to the zero-point. (Graphical illustration: Pictures/ray8.bmp)

If the problem (19)-(21) is solved and vector y^* is its optimal solution, then using (18) we determine vector $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$, where

$$x_j^* = y_j + [x_j^{max}], \quad j = 1, 2, \ldots, n$$

and use value $P(x^*)$ as initial bound for branch and bound method.
Stop.

Let us suppose that unit cube U^{max} does not contain any feasible vertex. In this case we determine the next unit cube U^1 along the ray R and find next perforation point using the following maximization problem.

Determining the next unit cube along the ray

Now we formulate and solve the following maximization LP problem:

$$\lambda \longrightarrow \text{max} \tag{22}$$

$$\left.\begin{array}{c} x_1 = x_1^{max} + \lambda(x_1^{min} - x_1^{max}) \\ x_2 = x_2^{max} + \lambda(x_2^{min} - x_2^{max}) \\ \vdots \\ x_n = x_n^{max} + \lambda(x_n^{min} - x_n^{max}) \end{array}\right\} \quad : x \in R \tag{23}$$

$$\left.\begin{array}{lcccll} [\,x_j^{max}\,] & \leq & x_j & \leq & [\,x_j^{max}\,]+1, & j \notin J' \\ x'_j - 1 & \leq & x_j & \leq & x'_j, & j \in J',\ R_j < 0 \\ x'_j & \leq & x_j & \leq & x'_j + 1, & j \in J',\ R_j > 0 \\ & & x_j & = & x'_j, & j \in J',\ R_j = 0 \end{array}\right\} \ : x \in U^1 \tag{24}$$

Then in the new unit cube we repeat steps described above. (Graphical illustration: Pictures/ray10.bmp).

Since the number of unit cubes the ray R crosses is finite the procedure will terminate after finite number of iterations (optimal value $\lambda' \geq 1$).

Note: If this procedure does not result any feasible unit cube vertex it does not mean that original ILP does not have any integer solution.
(Graphical illustration: Pictures/ray11.bmp)

Implementation issue and test results

The method proposed was implemented in the frame of linear and linear-fractional package WinGulf. There are several rules implemented for searching feasible vertices in the current unit cube:

1. *Rounding along the ray*, i.e. if there is a point x' and J' denotes index set of integer elements x'_j then we construct set $T(x')$ of unit cube vertices using the following rule:

$$x_j = \begin{cases} [x'_j], & j \notin J',\ R_j < 0 \\ [x'_j] + 1, & j \notin J',\ R_j > 0 \\ [x'_j] \text{ or } [x'_i] + 1, & j \notin J',\ R_j = 0 \\ x'_j - 1, & j \in J',\ R_j < 0 \\ x'_j + 1, & j \in J',\ R_j > 0 \\ x'_j, & j \in J',\ R_j = 0 \end{cases}$$

For example, if

$$\begin{array}{rcl} x' & = & (3.2, \quad 4.0, \quad 2.0, \quad 5.2, \quad 8.1, \quad 7.0) \quad \text{and} \\ R & = & (0.0, \ -4.0, \quad 0.0, \quad 1.3, \ -4.1, \quad 1.0) \end{array}$$

we obtain the following points in $T(x')$:

$$x^1 = (3, 3, 2, 6, 8, 8)$$
$$x^2 = (4, 3, 2, 6, 8, 8)$$

2. λ *maximization* on the ray in unit cube

Note: *Procedure 2*, i.e. "shifting + Balas 0/1 optimization" is not implemented yet.

Difficulties and further investigations

For relatively small problems the method proposed gives quite attractive results and may increase performance of branch and bound method for such problems where most of elements a_{ij} of matrix A are fractional.

For problems where most of elements a_{ij} and RHS vector b_i are integer the method usually does not lead to any advantages.

It is obvious that the larger size of problem to solve the more expensive the procedure becomes since in each unit cube along the ray we have to check up to 2^n vertices of the unit cube.

This is why on the first place in our further investigations we intend to implement and test the "shift + Balas" approach.