AN APPLICATION OF GENETIC ALGORITHM AND L-CLASS ENUMERATION TO THE INTEGER PROGRAMMING PROBLEM

Anton V. Eremeev, Alexander A. Kolokolov

Mathematical Department, Omsk State University, Russia
E-Mail: eremeev@iitam.omsk.su , kolo@iitam.omsk.su

1 Introduction

This paper introduces a new hybrid algorithm for solving an integer linear programming (ILP) problem, which incorporates a genetic algorithm (GA) and an L-class enumeration (LCE) method. Here the LCE method is also a hybrid algorithm which includes Gomory cutting planes. In addition, this paper describes a pure GA which may be used for the search of near-optimal solutions of linear and nonlinear integer problems. The combination of GA with the exact LCE method guarantees that the hybrid algorithm will deliver an exact solution for the ILP problem (if this solution exists). The scheme of this algorithm would be particularly effective if implemented on a multiprocessor computer.

[3-5] and other papers developed an approach to investigation and solving of integer programming problems, which is based on the application of some special (regular) partitions of space R^n, in particular, L-partition.

L-partition is determined as follows: points $x, y \in R^n (x \succ y)$ belong to the same class of the partition if no $z \in Z^n$ exists where $x \succeq z \succeq y$. Here $\succ$, $\succeq$ are the symbols of the lexicographical order. We denote as X/L a factor-set, generated by the L-partition for an arbitrary set $X \subseteq R^n$. The elements from X/L are called L-classes. This partition has some important properties. For example:

1) if X is bounded, then X/L is finite;

2) for any $Y, Y' \in X/L$ we write $Y \succ Y'$ if $y \succ y'$ for any $y \in Y$ and $y' \in Y'$. This relation of the elements of X/L is a linear order.

We are considering the following ILP problem:

$$f(x) = (c, x) \to \max \tag{1}$$

subject to

$$Ax \leq b, x \geq 0, \tag{2}$$

$$x \in Z^n. \tag{3}$$

Any point which meets the set of conditions (2) will be called an admissible point. Assume that matrix A and vectors b, c are integer and the set M determined by (2) is bounded. Let M_0 be a set of all optimal solutions for (1)–(3), $z^* = \text{lexmax } M_0$ (if it exists), $M/L = \{V_1, \ldots, V_p\}$, $V_i \succ V_{i+1}$, $i = 1, \ldots, p-1$.

2 L-class Enumeration

The LCE algorithm generates a sequence of current admissible points $S = \{x(t)\}$, $t = 1, \dots, T$ with the following properties:

a) $x(t) \succ x(t+1)$, $t = 1, \dots, T-1$,

b) all $x(t)$ belong to different L-classes;

c) if $M_0 \neq \emptyset$, then S contains an integer subsequence $Q = \{x(t_k)\}$, $k = 1, \dots, q$, for which $z^* \in Q$ and if $t > t_k$, then $f(x(t)) > f(x(t_k))$, $t = 1, \dots, T$; $k = 1, \dots, q$.

It is not difficult to see, that $|S| \leq |M/L|$. The current points $x(t)$ are constructed by solving special linear programming problems.

The aim of the LCE process is to find the integer optimum, moving along the L-class 'tape' $\{V_1, \dots, V_p\}$. However it is not necessary to look through the whole 'tape'. An additional condition (record inequality) such as $f(x) \geq f(x(t_k)) + 1$, where $x(t_k)$ is the best integer solution found before the current step $t > t_k$, would exclude some unnecessary L-classes. The value of the goal function $f(x(t_k))$ will be called a record on the current step. Before the first step the record is assumed to be $-\infty$. Every time when a new integer point is found, the LCE excludes it with the help of record inequality and constructs a few Gomory cutting planes.

3 The Pure Genetic Algorithm

The pure GA scheme may be applied to nonlinear integer programming problems but so far it has been tested on ILP problems only. The polyhedron of admissible solutions is surrounded by an n-dimensional cube. The phenotypes of the first population are the points with integer coordinates distributed inside the cube with the expectation close to the continuous optimum (i.e. the solution of problem (1),(2)). The chromosome consists of concatenated binary representations of the phenotype coordinates. The crossover sites are allowed only between the binary representations of coordinates.

Let the fitness function in points $x \in M$ be defined as $F(x) = f(x) + C$, where constant C is big enough to guarantee that $F(x) > 1$ for all $x \in M \cap Z^n$. For the points which don't meet the conditions (2), the fitness function $F(x)$ is assumed to be inversely proportional to the sum of constraints violations. In this GA we are using the roulette wheel scheme of selection with elite genotypes.

4 The Hybrid Algorithm

During the first stage of the hybrid algorithm the described GA is searching for near-optimal solutions for a fixed number of generations. During the second stage, several LCE processes are started from the points in Z^n corresponding to the GA's elite genotypes. These processes are exploring the 'tape' of L-classes in different places simultaneously. Throughout the second stage the GA continues to work, but the elite genotypes are fixed as long as the corresponding LCE processes work. An elite member of the second stage is a pair which consists of an admissible integer point related to the elite genotype and an LCE process for which this point is the last integer solution found.

In the case of implementation of this algorithm on a multiprocessor computer, the number of LCE processes may be set equal to the number of processors. In this case, all of the LCE processes may take a step simultaneously. If the algorithm is programmed for a one processor computer, in order to take a step, the LCE process must be selected. For example, with the help of the roulette wheel selection.

A step from one L-class to the next one changes the coordinates of the current point of LCE process. If the new current point is integer, the corresponding elite genotype is to be changed so that this new point would be it's phenotype.

The number of the working LCE processes is not fixed. It may decrease when an LCE process P' makes a step and the new current point of this process appears in the area of the L-class 'tape', which is already explored by some other LCE process P''. In order to avoid the investigation of the same area for the second time, the process P' will be deleted. The vacant place among the LCE processes may be filled when a new elite genotype is recruited from the population. The hybrid algorithm stops when the last LCE process ends its work.

If among the LCE processes there is one with the current point situated at $w = \text{lexmax } M$, then the best integer solution found by the hybrid algorithm is the exact solution of ILP problem. To guarantee this we suggest adding an extra elite member with phenotype $(]w_1[,]w_2[, \dots,]w_n[)$ at the end of the first stage. Here][indicates rounding up: $]w_j[= \min\{z : z \geq w_j, z \in Z\}, j = 1, \dots, n$.

On one hand, GA in this hybrid algorithm quickly finds the integer admissible solutions and consequently improves the record for the LCE processes, making them work faster. On the other hand, the LCE processes enable GA to avoid premature convergence to the local optima.

The suggested hybrid algorithm was programmed using Borland Pascal (see [1]). For most of the well known ILP problems described in [6] this algorithm found the optimal solutions faster then the pure LCE method.

References

[1] A.V.Eremeev, A.A.Kolokolov, *On Some Genetic and L-class Enumeration Algorithms in Integer Programming* In: Proceedings of First International Conference on Evolutionary Computation and Its Applications. Moskow, Russia, 1996, pp. 297-303.

[2] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[3] A.A. Kolokolov *On the L -structure of integer linear programming problems.* In: J.Henry and J.-P.Yvon (Eds.). *System Modeling and Optimization. Lecture Notes in Control and Information Sciences 197.* Proceedings of the 16th IFIP-TC7 Conference. Compiegne, France. Springer Verlag, 1994, pp. 756-760.

[4] A.A. Kolokolov *Some L-class enumeration algorithms for Integer Programming Problems.* Proceedings of 3rd IFIP WG-7.6 Working Conference on Optimization-Based Computer-Aided Modeling and Design. Prague, Czech Republic, 1994, pp. 256-260.

[5] A.A. Kolokolov A.A.*Regular partitions and cuts in integer programming.* In: A.D.Korshunov (ed.), *Discrete Analysis and Operations Research*, Kluwer Academic Publishers, 1996, pp. 59-79.

[6] C.A. Trauth, R.E.D. Woolsey, *Integer programming: a study in computational efficiency.* Manag. Science, 1969, vol.15, N9, pp.481-493.