An Improved Simplex-Genetic Method to Solve Hard Linear Programming Problems

Juan Frausto-Solís and Alma Nieto-Yáñez

ITESM Campus Cuernavaca
Reforma 182-A, Col Lomas De Cuernavaca,
62589, Temixco Morelos, México
{juan.frausto, delia.nieto}@itesm.mx

Abstract. Linear programming (LP) is an important field of optimization. Even though, interior point methods are polynomial algorithms, many LP practical problems are solved more efficiently by the primal and dual revised simplex methods (RSM); however, RSM has a poor performance in hard LP problems (HLPP) as in the Klee-Minty Cubes problem. Among LP methods, the hybrid method known as Simplex-Genetic (SG) is very robust to solve HLPP. The objective of SG is to obtain the optimal solution of a HLPP, taking advantages from each one of the combined methods -a genetic algorithm (GA) and the classical primal RSM-. In this paper a new SG method named Improved Simplex Genetic Method (ISG) is presented. ISG combines a GA (with special genetic operators) with both primal and dual RSM. Numerical experimentation using some instances of the Klee-Minty cubes problem shows that ISG has a better performance than both RSM and SG.

Keywords: Genetic algorithm, Simplex method, Hybrid methods, Simplex-Genetic, Linear Programming.

1 Introduction

Linear programming problems (LPPs) have a wide range of real applications on diverse research areas [1], for example production planning, market analysis, routing, among others [2]. The development of the Simplex Method (SM) to solve LPPs in 1947 marks the start of the modern era in optimization; since then many algorithms, mainly interior point algorithms [3][4] have been developed. SM is a deterministic method and guarantees to find the optimal solution; however, for some LPPs, SM is not efficient and has poor worst-case behavior. Even though SM is an exponential algorithm, it is still the most popular and efficient method for many practical problems [5][6]. The most popular SM implementations are the primal and dual revised simplex methods (RSM). On other hand, genetic algorithms (GAs) are global search methods based on natural evolution and genetics. GAs have the property of maintaining a population of potential solutions using a selection process based on the fitness of each individual, and its evolution is made using genetic operators. GAs do not have optimality conditions, therefore they are not able to know when an optimal solution is found.

The hybrid approach Simplex-Genetic (SG) presented in [7], takes advantages of both methods and combines them with the goal of solving an LPP more efficiently. In [7] a GA is combined with only the primal version of RSM. SG was tested with hard linear programming problems (HLPP) known as Klee-Minty Cubes.

In this paper a new method named Improved Simplex Genetic (ISG) is presented. The principal characteristics of ISG are: i) the combination of a GA with both the primal and dual RSM, ii) the LPP representation is made for both bounded and unbounded variables and iii) new genetic operators are used in order to promote feasibility and optimality searching features. Numerical experimentation using some instances of Klee-Minty Cubes problems (the number of instances is limited by the IEEE infinitum definition [8]) shows that ISG has better performance than both RSM and SG.

This paper is organized as follows: In section 2, LPPs and RSM are briefly described, while the principal characteristics of a classical GA are presented in section 3; in section 4 SG is reviewed; in section 5 the new hybrid algorithm ISG is presented; finally in section 6, numerical experimentation with Klee-Minty Cubes is presented; the conclusion and future works are then given in section 7.

2 Linear Programming Problem and the Simplex Method

Linear Programming deals with linear mathematical models with the goal of optimizing a linear objective function subject to linear constraints. A LPP can be modeled as:

$$\begin{aligned} & \begin{bmatrix} \text{Max} \\ \text{Min} \end{bmatrix} \quad \sum_{j=1}^n c_{ij}x_j \\ & \text{subject to} \quad \sum_{j=1}^n a_{ij}x_j = b_i \quad (i=1,2,\dots,m) \quad . \\ & \quad \quad \quad l_j \leq x_j \leq u_j \quad (j=1,2,\dots,n) \end{aligned} \quad (1)$$

In (1), n is the number of variables, m is the number of constraints, x is the vector of variables to be solved, l_j and u_j represent the lower and the upper bound of the variable x_j , c is the cost vector, each equation $a_{ij}x_j=b$ is a constraint and the expression " cx " is the objective function.

Many methods have been proposed to solve LPPs. Even though SM was proposed by Dantzig since 1947, it is the most efficient for many practical problems [6]. As it is well known, the fundamental theorem of LP establishes that the optimal solution can only occur at a boundary point of the feasible region; therefore, SM is based on a search of solutions over the extreme points of a LPP. The SM first identifies an initial feasible solution and with this solution the initial tableau is built. The SM constructs a sequence of tableaux until the optimal solution is achieved. In every iteration, a basic variable is selected to leave the basis and one non-basic variable is selected to enter.

Revised Simplex Method (RSM) is the more popular SM implementation. All RSM variants [9] for solving model (1) are similar and the principal difference is the criterion of selecting the entering and leaving variables. Dual Simplex Method (DSM)

is another SM variant. DSM is based on the duality concept and was designed by Lemke in 1954 [10]. It also starts with a simplex tableau that defines an optimal but infeasible solution, and their objective is to move toward feasibility without loss of optimality [5].

SM behavior can be improved by changing: i) the pivoting form, that is, the way of selecting the entering and leaving variables [11][12]; and ii) the initial solution. A good initial solution is the key for efficient algorithms and so several alternatives are available: i) SM classical strategies, where the two-phase method and the big M method are the most popular and ii) Cosine Simplex Method (CSM) [13]; the latter uses a heuristic named Theta Heuristic [14] to determine the initial vertex based on the analysis of the angles between the gradient of the objective function and the gradients of the constraints. The initial solution can be classified in four cases [14]: Case I, the initial solution is feasible and optimal; case II, the initial solution is feasible but not optimal; case III, the solution is optimal but infeasible; and case IV, the solution is neither feasible nor optimal.

3 Genetic Algorithms

Genetic algorithms (GAs) are global search algorithms that are based on genetic principles and on the evolution [15]. The GAs were proposed by Holland in 1975. Since then, new applications have emerged in a wide variety of fields like scheduling, data fitting, and many combinatorial optimization problems [16]. A GA can be defined through the following components [15]: a genetic representation of the problem, a manner of generating an initial population, an evaluation function, genetic operators and parameters values (population size, probabilities of applying genetic operators, etc.). The performance of the GAs can be improved by an adequate selection of these components, parallel implementations and hybridization with other heuristic or exact algorithms.

4 Simplex-Genetic Method

In order to solve an LPP efficiently, hybrid methods can be designed; in [7] a Simplex-Genetic is presented; it combines a GA with the primal RSM and proves their efficiency using Klee-Minty Cubes; the hybridization is sequential. In the first phase (or genetic phase) a GA is used, while RSM is used in the second phase (or

Fig. 1. Hybrid approach used to solve a LPP

simplex phase). RSM starts with the best feasible solution obtained in the first phase. Notice that in this hybrid algorithm the last solution of the genetic phase must be a feasible one, and then a primal problem should be solved by the simplex phase; so we rename this method as Primal Simplex-Genetic (PSG). The general scheme is shown in figure 1. Numerical results show that SG has better behavior than primal RSM [7].

The genetic phase of SG is characterized by [7]:

- Genetic representation of the solutions. Each string represents a basic solution for LPPs, that is, if an LPP has n variables, m constraints and h slack variables, the length of the string is $n+h$ and the string must have m basic variables. Any basic variable in a string is represented by a one and all the $(n+h-m)$ non-basic variables are represented by a zero.
- A tabu list, the purpose of which is to measure the feasibility of the variables during the process is used (see figure 2). The tabu list is built by the average of infeasibility of each variable in the population; in this case one variable is infeasible if the non-negative constraint is violated. Each position in the list represents one variable. If the variable was feasible in the previous population then the list has a zero, if not, the list has a positive number that measures how far from the feasibility region is the variable. In each generation this list is updated.

Fig. 2. Tabu approach for crossover operator

- Two genetic operators: To maintain a population of basic solutions, special crossover and mutation operators are used. The crossover operator works with two parent solutions producing a child solution as is common in GA, but in this case two tasks are done: 1) identical bits in the parents are copied in the child and the others are empty sites; 2) the empty sites are filled using the tabu list, zeroes are set in the sites where the tabu list has a high value and the child must be a basic solution. In the figure 2 an example of this operator is shown. The mutation operator interchanges two different values selected at random and avoids non-basic solutions (i.e. only corners are permitted).

5 Improved Simplex-Genetic Method

In [17] different strategies to combine heuristics are presented. In this case we proposed a similar scheme used in [7] (and presented in figure 1). In this scheme the hybridization is sequential: GA in the first phase and RSM in the second one. Now in the genetic phase of ISG, a hybrid elitist GA is used; then, in the second phase, the

primal or dual RSM are used. Additionally, the genetic phase can be stopped with a feasible solution or with a non-feasible but optimal solution therefore the genetic phase stops until the improvement between one generation and the next is not significant (tends to zero).

The elitism included in the genetic phase consists only in keeping the best solution (feasible or non-feasible but optimal). Other improvements proposed in the genetic phase are as follows:

- Genetic representation of the solutions. It is important that the solution can represent basic solutions to LPPs. The majority of real LPPs have explicit bounds. Therefore an alternative representation is proposed. The length string is $(n+h)$ where n is the number of variables and h is the number of slack variables. The number of basic variables is equal to m , it is the number of constraints. In the string a 1 represents a basic variable; a 0 represents a non-basic variable in its lower bound while a non-basic variable in its upper bound is represented by -1.
- Generation of the initial population. This generation is random but the basic variables are added one by one verifying that they do not have dependent columns in the basis (matrix B) using LU factorization. Another approach used to generate the initial population is using the cosines calculated with the equations presented in [13][14], with the purpose of minimizing the number of individuals with dependent columns. Similar cosines form a cluster, and then basic variables are randomly selected, taken care that two basic variables do not belong to the same cluster. If there are not enough clusters to select the m variables required, new variables are randomly generated.
- Evaluation Function. The evaluation function is the objective function of the LPPs and the infeasibility is penalized adding this to the original objective function.
- Genetic Operators. New operators were proposed with the objective to promote feasibility and optimality.
 - Mutation promoting feasibility (Mut Feasible). Using a tabu list of infeasibility, selecting the basic variable more infeasible and interchange it by one non-basic variable randomly selected.
 - Mutation using the cosines (Mut Cosine). The cosines of the angles between the gradient of constraints and the gradient of objective function are calculated with the equations presented in [13][14] and according to the KKT in the optimal solution the cosine tends to be infinite, therefore the mutation pretends that the basic variables are those associated with the greater cosines. The mutation operator works as follows: first it selects a basic variable (randomly) and then interchanges it with the non-basic variable associated with the greater cosine.
 - These two operators are jointly used (Mut Fea-Cos), additionally the mutation operator presented in [7] is added. The results are shown in the next section.

6 Numerical Experimentation

In order to prove the performance of ISG the Klee-Minty Cubes problem [18] is used; as is generally known, RSM has an exponential behavior and the number of iterations required to solve this problem is $2^n - 1$ [5]. The model of this problem is:

Max $\sum_{j=1}^n 10^{n-j} x_j$

subject to $\left(2 \sum_{j=1}^{i-1} 10^{i-j} \right) + x_i \leq 100^{i-1} \quad i = 1, 2, \dots, n$

$x_j \geq 0 \quad j = 1, 2, \dots, n$

(2)

The instances used in this paper have a dimension equal to (n) , the value of which is less than nineteen because for bigger values, the right hand side of some constraints is bigger than the IEEE definition of infinite [8]. The population size used here was determined by experimentation and its resulting value was two times the chromosome size; in this paper we use this value in both SG and ISG. The ISG is compared with SG and with RSM. SG combines the operators described in section 4 (with 80% of mutations and 20% of crossovers), the GA phase stops when the improvement between one generation and the next one is not significant and the best solution is feasible. The mutation operators (described in section 5 for ISG) are used first separately and then combined in a second experiment (50% each). The experimentation was made in a computer with an Intel® Pentium® M processor 1.60GHz 590MHz and 503MB of RAM. In figure 3, the comparison between the SG versus ISG with the proposed genetic operators is presented (using 30 executions in average); we notice that the operators using cosines have better performance and this improvement increases as the problem size increases. In the figure 4 the comparison between the cosines operator and RSM starting with the slacks as basic variables is shown.

Fig. 3. Comparison of SG and the new operators in ISG

Fig. 4. Comparison of SM ant the new operators in ISG

7 Conclusions and Future Works

In this paper an Improved Simplex Genetic algorithm (ISG) is presented, which includes new genetic operators. These genetic operators use the cosine of the angles between the constraints and the objective function and use a list of infeasibility of each variable updated from previous solutions with the objective to promote optimality (using cosine values) and to avoid infeasibility (using a tabu list of infeasibility). The representation proposed here allows solving LPPs with bounded variables. The best solution obtained in the genetic phase of ISG is a basic solution for both, RSM or DSM. ISG, SG and RSM always finish with the optimal solution (if there is any); therefore, the comparison is made using only execution time. Experimentation presented in this paper for Klee-Minty Cubes problems showed that ISG has better performance than both RSM and the classical Simplex Genetic algorithms.

References

1. McMillan, C. Jr.: Mathematical Programming. John Wiley and Sons, Inc., New York (1970)
2. ILOG Optimization: <http://www.ilog.com/products/optimization/industry/index.cfm>. Last access: (2007)
3. Adler, I., Karmarkar, N., Resende, M., Veiga, G.: An Implementation of Karmarkar's Algorithm for Linear Programming. *Mathematical Programming* Vol. 44 (1989) 297-335
4. Mitra, G., Tamiz, M.: Experimental Investigation of an Interior Search Method within a Simplex Framework. *Communications of the ACM*, Vol. 31, Issue 12 (1988) 1474-1482
5. Chvátal, V.: Linear Programming. W. H. Freeman and Company, New York (1983)
6. Garey, M., Johnson, D.: Computers and intractability. A guide to the theory of NP-Completeness. Nineteenth printing, W.H. Freeman and Company, New York (1997)

7. Frausto-Solís, J., Rivera, R., Ramos, F.: A Simplex-Genetic method for solving the Klee-Minty cube. *WSEAS Transactions on Systems* Vol. 2, No. 1 (2002) 232-237
8. IEEE: Standard 754 for Binary Floating-Point Arithmetic. *Lecture Notes on the Status of the IEEE 754* (1997)
9. Morgan, S.: A Comparison of Simplex Method Algorithms. Master Thesis, U. Florida (1997)
10. Lemke, C.: The dual method of solving the lineal programming problem. *Naval Research Logistic Quarterly* (1954)
11. Terlaky, T., Zhang, S.: A survey on pivot rules for linear programming. Delft University of Technology, Report No. 91-99, ISSN 0922-5641 (1991)
12. Bland, R.: New finite pivoting rules for the simplex method. *Mathematics on Operations Research* Vol. 2 (1977) 103-107
13. Trigos, F., Frausto-Solís, J., Rivera, R.: A Simplex Cosine Method for Solving the Klee-Minty Cube, *Advances in Simulation, System Theory and Systems Engineering*, WSEAS Press, ISBN 960852 70X (2002) 27-32
14. Trigos, F., Frausto-Solís, J.: Experimental Evaluation of the Theta Heuristic for Starting the Cosine Simplex Method. *International Conference on Computational Science and its Applications*, Singapore, ISBN 981-05-3498-1 (2005)
15. Michalewicz, Z.: *Genetic Algorithms + Data Structures= Evolution programs*. Third Edition, Springer-Verlag, Berlin Heidelberg New York (1996)
16. Illinois Genetic Algorithms Laboratory: <http://www-illgal.ge.uiuc.edu/index.php3>, Director David Goldberg, last access: April (2006)
17. Puchinger, J., Günther R.: Combining Metaheuristics and Exact Algorithms in Combinatorial Optimization: A Survey and Classification. In *Proceedings of the First International Work-Conference on the Interplay Between Natural and Artificial Computation*, Part II, Vol. 3562 of LNCS, Springer (2005) 41-53
18. Klee, V., Minty, G.: How good is the Simplex Algorithm? In *Inequalities III*, Shisha, O., Ed., Academic Press, New York (1972) 159-179