A Note on Branch and Bound Algorithm for Integer Linear Programming

Syed Inayatullah[1], Wajeeha Riaz[2], Hafsa Ather Jafree[3], Tanveer Ahmed Siddiqi[4], Muhammad Imtiaz[5] , Saba Naz[6], Syed Ahmad Hassan[7]

Department of Mathematics, University of Karachi, Karachi, Pakistan. 75270

[1]inayat@uok.edu.pk, [2]wajihariaz@gmail.com, [3]hafsa_jafree@hotmail.com, [4]tanveer@uok.edu.pk, [5]imtiaz@uok.edu.pk, [6]sanaz@uok.edu.pk,[7]ahmedhassan@uok.edu.pk

Abstract

In branch and bound algorithm for integer linear programming the usual approach is incorporating dual simplex method to achieve feasibility for each sub-problem. Although one can also employ the phase 1 simplex method but the simplicity and easy implementation of the dual simplex method bounds the users to use it. In this paper a new technique for handling sub problems in branch and bound method has been presented, which is an efficient alternative of dual simplex method.

Keywords: Integer programming, Branch and bound method, dual simplex algorithm, two-phase simplex method

1. Introduction:

One of the most recognized fields amongst the decision makers today is linear programming which was initiated by Dantzig during the World War II (Dantzig G. B., 1951). Researchers belonging to diverse fields are now working on techniques for optimization of linear programming problems. Applying linear programming to real world problems gave rise to situations where the required solution could not assume fractional values. One instant thought that may occur to one's mind is rounding off the solution to about an integer solution. But it was discovered that this resulted in infeasibility or often compromised optimality. Assignment problem is one of the examples that fall into the category where the required solution must assume integer values.

Optimization of such problems belongs to a broader category termed as 'Discrete optimization' which involve techniques that furnish integer solutions to a given problem. Because of its vast applicability linear programming problem with integer constraints termed as 'Integer linear programming' (ILP) has been in the spotlight of researchers and is being incorporated by decision analysts to obtain solution of complex systems (Galati, 2010). Although the applications of ILPs started to emerge in the late 1940's, however it became a major interest of researchers following from the work of Dantzig, Fulkerson, and Johnson (1954) on the solution of famous travelling salesman problem TSP.

For solving ILPs, first finite integer programming technique was developed by Gomory (1958). Little J. , Murty, Sweeney, and Karel (1963) solved TSP using another technique, branch and bound, that widely spurred among the researchers and considered to be most effective approach in practical computations. Motivated by the work of Lin (1965) many heuristic algorithm were proposed. Numbers of other specialized algorithms were made and are still being developed, see (Junger, et al., 2010) and (Bixby, 2012) for detailed history of progresses made in the field of ILP.

Amongst many approaches used for handling ILPs, branch and bound methods are enumerative in nature. They handle the problem by applying bounds to eliminate the solutions that cannot be optimal for a given ILP. Feasible space of ILP is a subset of feasible space of the linear program obtained by ignoring the integrality condition on variables, known as relaxed linear program RLP, hence if the RLP is infeasible then ILP is also **infeasible** (Mitten, 1970). If the optimal solution of RLP is already integer, then that would also be the solution of associated ILP, otherwise feasible space of RLP can be divided into sub-problems by adding further constraints designed to preserve integer solutions. The optimal integer solution, if exist must belong to any one the resulted Sub-RLPs. The process of creating Sub-RLPs would be repeated till all Sub-RLPs provide an integer solution or fathomed. Optimal ILP solution would be the best among all Sub-RLP's optimal solutions that are integer, One sufficient condition of infeasibility of ILP is the infeasibility of RLP, but in contrast, it may be possible that the feasible space of RLP is unbounded but the associated ILP is still infeasible.

Branch and bound algorithms firstly proposed by Land and Doig (1960), they gained popularity after the work of Little J. D., Murty, Sweeney, & Karel, 1963 (1963) as it showed that by controlled enumeration comparatively large problems can be solved. Dakin (1965) presented improved branching rule, then Beale and small (1965) modified the method and suggested computation rules of upper bounds. Tomlin (1971) proposed extension of the algorithm proposed by Beale and small. Taha (1971) introduced computation rule for lower bound. From 70's to 80,s more sophisticated variable and node selection procedures were developed (Land & Powell, 1979). Although these refinements made branch and bound a powerful tool to solve ILPs, there were no major fundamental development till late 1990's. Around 1990's advancements in solving LP overall improved the ILP codes. Dual simplex algorithm emerged as a general purpose solver, Linear algebra vastly improved for large sparse models in the application of simplex algorithms. Consequently progress in LP solvers made ILP solvers more efficient.

In branch and bound method branching creates sub-problems that can be optimized by using some pivotal method. Today there are two well-known efficient methods for this purpose. First is two-phase method and the second is dual simplex method. Unfortunately the introduction of artificial variables makes the implementation of two-phase simplex difficult and tedious; so practically the users have no choice other than to implement the dual simplex method. In this paper the proposed algorithm incorporates the recently developed method, called as DP1 by Khan, Inayatullah, Imtiaz, and Khan (2009) (Pan, 2014) as an easy to implement alternative to two-phase simplex method. In future the proposed approach of DP1 for the solution of ILPs in place of dual simplex may work as a stepping stone towards new variations in class of branch and bound algorithms.

2. Some basic terminologies and notations:

An integer linear programming problem copes with maximization/minimization of a linear objective function subject to a system of linear equations/inequalities where variables are constrained to assume an integer value. A general ILP problem is

$$\begin{aligned}
&\text{maximize} \quad \mathbf{c}^T\mathbf{x} \\
&\text{subject to} \\
&\qquad A\mathbf{x} = \mathbf{b} \\
&\qquad \mathbf{x} \geq 0, \quad \mathbf{x} \in Z^{\mathrm{m}} \\
&\qquad \mathbf{b} \in \Re^m, \mathbf{c} \in \Re^n, A \in \Re^{m\times n}
\end{aligned}$$

The associated relaxed linear program (RLP) obtained by omitting the integral constraint of the given ILP is

$$\begin{aligned}&\text{maximize} \quad \mathbf{c}^T\mathbf{x}\\&\text{subject to}\\&\qquad A\mathbf{x}=\mathbf{b}\\&\qquad \mathbf{x}\geq 0, \mathbf{x}\in\Re^n\\&\qquad \mathbf{b}\in\Re^m, \mathbf{c}\in\Re^n, A\in\Re^{m\times n}\end{aligned}$$

Where $A\in\Re^{m\times n}, b\in\Re^m, c\in\Re^n$, $\mathbf{x}$ is the vector of decision variables and $m\leq n$.

Let $\{\mathbf{a}_i \mid i=1,2,\ldots,n\}$ be the set of column vectors of matrix A. For any index set I, we define A_I to be the matrix formed by $\mathbf{a}_i$ such that $i\in I$. Consider an index set $B\subseteq\{1,2,\ldots,n\}$ as a basis set provided A_B where is non-singular and $|B|=m$. Also $N=\{1,2,\ldots,n\}\setminus B$ is termed as the set on non-basic variables. In other words x_i is called a **basic variable** if $i\in B$ and it is a **Non basic variable** if $i\in N$. A solution $\mathbf{x}$ such that $x_i=0, \forall i\in N$ is called a **basic solution**. $\mathbf{x}$ is a **basic feasible solution** if it is also feasible and corresponding basis is called **feasible basis**.

2.1 Dictionary

A dictionary of any RLP for a basis *B*, may be element-wise represented by the following collection of equations, denoted by *D*(*B*), which is slightly modified form of (Chvatal, 1983) (Kaluzny, 2001).

$$D(B)=\left\{\begin{array}{c} x_i+\sum_{j\in N}\alpha_{ij}x_j=\beta_i,\ i\in B\\ \hdashline Maximize\ z=\sum_{j\in N}\gamma_j x_j+\hat{z}\end{array}\right\}$$

Where β_i is the component of vector $A_B^{-1}\mathbf{b}\in\Re^B$ representing value of the basic variable x_i, α_{ij} is the element of $A_B^{-1}A_N\in\Re^{B\times N}$ denoting the coefficient of the non-basic variable x_j in the equation containing basic variable x_i, γ_j is the component of $(\mathbf{c}_N^T-\mathbf{c}_B^T A_B^{-1}A_N)^T\in\Re^N$ representing the coefficient of non-basic variable x_j in the objective function of the current dictionary, and $\hat{z}=\mathbf{c}_B^T A_B^{-1}\mathbf{b}\in\Re$ is the objective scalar value associated with current basis *B*. A basis *B* (or a dictionary *D*(*B*)) is said to be *feasible* if $\beta_i\geq 0$ for all $i\in B$.

The branch and bound approach yields an integer feasible solution by successively adding valid pair of cuts to the feasible region of RLP. If the optimal solution of RLP has fractional components $\bar{x}_j$ then integer cuts, $x_j \le \lfloor \bar{x}_j \rfloor$ and $x_j \ge \lceil \bar{x}_j \rceil$, are valid pair of cuts associated with x_j.These cuts prune away the area $\lfloor \bar{x}_j \rfloor < x_j < \lceil \bar{x}_j \rceil$ from the feasible region of RLP and creating pairs of sub problems known to be branches. Each branch is again treated the same way as an RLP. A branch is **fathomed** if it doesn't require further branching. A branch is fathomed in following cases:

i. Associated RLP solution is infeasible
ii. The optimal value is less than the incumbent (current best)
iii. If the solution is an integer feasible solution.

Branching creates sub-problems, and each sub-problem should be solved for optimality.

2.2 Pivot Operations

Let (r,k) be the position of the pivot element $d_{rk} (\neq 0)$ of D where $r \in B, k \in N$, then an updated equivalent dictionary $D(B)$ with a new basis $B := (B \cup \{k\}) \setminus \{r\}$ and the new non-basis $N := (N \cup \{r\}) \setminus \{k\}$ can be obtained by performing the subsequent operations on $D(B)$

$$d_{ik} := d_{ij} - d_{rj} \times \frac{d_{ik}}{d_{rk}}, \quad i \in B \setminus \{r\}, j \in N \setminus \{k\}$$

$$d_{rj} := \frac{d_{rj}}{d_{rk}}, \quad j \in N \setminus \{k\}$$

$$d_{ik} := -\frac{d_{ik}}{d_{rk}}, \quad i \in B \setminus \{r\}$$

$$d_{kr} := \frac{1}{d_{rk}}$$

The above replacement is known as pivot operation on (r,k).

2.3 Dynamic Phase 1 method (Khan, Inayatullah, Imtiaz, & Khan, 2009)

Algorithm 2.1: Dynamic Phase 1 (DP1)

Step 1: Let S be a maximal subset of B such that $S = \{s \mid \beta_s < 0, s \in B\}$. If $S = \phi$ then basis B is primal feasible. **Exit.**

Step 2: Construct a row-vector $\mathbf{w} \in \Re^N$ such that $w_j = \sum_{s \in S} \alpha_{sj}$.

Step 3: Let $K \subseteq N$ such that $K = \{j : w_j > 0, j \in N\}$. If $K = \phi$, basis B is primal inconsistent. **Exit.**

Step 4: Choose $k \in K$ such that $w_k \geq w_h \; \forall h \in K$

(Ties could be broken on minimum index)

Step 5: Choose $r \in B$ such that

$$r = \arg\min \left\{ \left\{ \frac{\beta_i}{\alpha_{ik}} : \beta_i < 0, \alpha_{ik} < 0, i \in B \right\} \cup \left\{ \frac{\beta_i}{\alpha_{ik}} : \beta_i \geq 0, \alpha_{ik} > 0, i \in B \right\} \right\}$$

(Ties could be broken on minimum index)

Step 6: Make a pivot on (r,k) ($\Rightarrow$ Set $B := (B \cup \{k\}) \setminus \{r\}$, $N := (N \cup \{r\}) \setminus \{k\}$ and update $D(B)$).

Step 7: Go to Step 1.

3. Branch and Bound Algorithm Using DP1 Method

The algorithm is described as follows.

The Algorithm 3.1

Step 1: Drop the integral conditions from ILP and get the RLP, say $L_{i,k}$.

Initialize: Set $i = 0$, $k = 1$, incumbent $I_Z = -\infty$, Active set $L_{0,1}$.

Step 2: Pick any member of active set say $L_{i,k}$ and set active set:=active set– $\{L_{i,k}\}$

Obtain $D(B)$ for $L_{i,k}$

Step 3 If RLP is inconsistent, then ILP is also inconsistent **Go to** step 7**.**

Otherwise, find the optimal solution using DP1 method.

Step 4: If the solution is integer and optimal value >= I_Z then set

I_Z :=optimal value and **Go to** step 7

If optimal value < I_Z then **Go to** Step 7

Step 5: Branch on the variable with largest fraction, say x_j . (break the ties randomly)

Step 6: $\mathrm{i} := \mathrm{i} + 1$

Form $L_{i,k+1}$ and $L_{i,k+2}$ as follow

$L_{i,k+1} = L_{i,k} \cap \{x \mid x_j \le \lfloor \bar{x}_j \rfloor\}$

$L_{i,k+2} = L_{i,k} \cap \{x \mid x_j \ge \lceil \bar{x}_j \rceil\}$

Active set:=Active set $\cup \{L_{i,k+1}, L_{i,k+2}\}$

Step 7: If Active set $= \phi$, **Go to** step 8

Otherwise, **Go to** step 2

Step 8: I_Z is the optimal value of ILP and the corresponding solution is the optimal solution.

Explanation:

If the RLP is infeasible then so is the ILP. Set optimal value of the parent problem as incumbent (current best) solution. At the end of each sub-problem check if the incumbent can be replaced by the current optimal if yes then update incumbent. If optimal solution of RLP is also feasible for ILP then terminate the problem this process is called **fathoming.** Else divide the problem in further sub-problems and add to the active set of problems.

At each level find the optimal dictionary $D(B)$ of RLP using the Algorithm 2.1. Where B contains the optimal basis. Let $B' = B \cup \{m+1\}$, then $D(B')$ can be formed as follows. Consider we need to branch on x_j where $j \in B$. Form the augmented dictionary by appending a $1 \times N + 1$ matrix $[b_{m+1} \ \bar{A}_{m+1}]$.

For $L_{i,k+1}$

Where,

$b_{m+1} = -\ (\bar{x}_j - \lfloor \bar{x}_j \rfloor)$

$\bar{A}_{m+1} = -(j^{th} \ row\ vector\ of\ \bar{A}\ )$

For $L_{i,k+2}$

Where,

$b_{m+1} = (\bar{x}_j - \lceil \bar{x}_j \rceil)$

$\bar{A}_{m+1} = \ (j^{th} \ row\ vector\ of\ \bar{A}\ )$

4. Conclusion

In this paper we have used a new technique DP1 for solving sub-problems in branch and bound method. The proposed variant serves as an alternative for dual simplex method. As compared to 2-phase simplex method it avoids use of artificial variables making it easy to implement. Computation-wise DP1 is much efficient than phase 1 simplex method and almost equivalent to dual simplex method, for details see (Khan, Inayatullah, Imtiaz, & Khan, 2009).

REFERENCES

Beale, E. M., & Small, R. E. (1965). Mixed integer programming by a branch-and-bound technique. *proceedings of IFIP Congress Vol 2*, 450-451.

Bixby, R. E. (2012). A brief history of linear and mixed-integer programming computation. *Documenta Math*, 107-121.

Chvatal, V. (1983). *Linear Programming.* United States of America: W.H. Freeman and Company.

Dakin, R. J. (1965). A tree-search algorithm for mixed integer programming problems. *The computer journal*, 250-255.

Dantzig, G. B. (1951). Maximization of a linear function of variables subject to linear inequalities. *In Koopmans TC (Ed.)*, 339–347.

Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a Large-Scale Traveling-Salesman problem. *Journal of the Operations Research Society of America*.

Galati, M. (2010). PHD thesis on Decomposition methods for integer linear programming.

Gomory, R. E. (1958). Outline of an algorithm for integer solutions to linear programs . *Bulletin of the American Mathematical Society* , 275-278.

Junger, M., Liebling, T., Naddef, D., Nemhauser, G., Pulleyblank, W., Reinelt, G., et al. (2010). *50 Years of Integer Programming 1958–2008.* Newyork: Springer-Verlag Berlin Heidelberg.

Kaluzny, B. (2001). Finite Pivot algoirthms and Feasibility. *MS thesis, Faculty of Graduate Studies and Reseach, School of Computer Science, McGill University, Montreal, Quebec, Canada* .

Khan, N., Inayatullah, S., Imtiaz, M., & Khan, F. H. (2009). New Artificial-Free Phase 1 Simplex Method. *International Journal of Basic & Applied Sciences IJBAS-IJENS*, 69-75.

Land, A. H., & Doig, A. G. (1960). An Automatic Method of Solving Discrete Programming Problems. *Econometrica*, 497-520.

Land, A., & Powell, S. (1979). Computer Codes for Problems of Integer Programming. *Annals of Discrete Mathematics*, 221-269.

Lee, M. S. (1972). Goal Programming for Decision Analysis. *Auerbach, Philadelphia., 4*.

Lin, S. (1965). Computer Solutions of the Traveling Salesman Problem. *Bell Labs Technical Journal*, 2245-2269.

Little, J. D., Murty, K. G., Sweeney, D. W., & Karel, C. (1963). An Algorithm for the Traveling Salesman Problem. *Operations Research*, 972-989 .

Little, J., Murty, K., Sweeney, D., & Karel, C. (1963). An Algorithm for the Traveling Salesman Problem. *Operations research*, 972-989.

Mitten, L. G. (1970). Branch and Bound methods: General formulation and properties. *Operations Research*, 24-34.

Pan, P.-Q. (2014). *Linear Programming Computation.* Springer Heidelberg New York Dordrecht London.

Taha, H. A. (1971). On the solution of zero-one linear programs by ranking the extreme points. *Tech report no 71-5 dept of ind* .

Tomlin, J. A. (1971). An improved branch-and-bound method for integer programming. *Operations Research*, 1070-1075.