Step Algorithm to solve positive-integer type-linear programming problem with two variables and n linear constraints

Monal Vaishnani
Department of Industrial Engineering
Pandit Deendayal Pertoleum University, Gandhinagar
Gujarat, India
monalsvaishnani2100@gmail.com

Poonam Savsani
Department of Industrial Engineering
Pandit Deendayal Pertoleum University, Gandhinagar
Gujarat, India
poonam.savsani@gmail.com

Abstract

In this paper, Step algorithm to solve positive integer type linear programming problems with two variables and n linear constraints is proposed. Classical methods to solve integer programming problems are branch and bound method and Gomory's cutting plane algorithm. Manual solution by both these methods is tedious and tiring. Moreover, to get multiple optima through branch and bond method, it is required to select different branch and need to solve for that particular branch. While in Gomory's cutting plane algorithm, it is required to solve the problem again with different sets of iterations to have multi optima. Disadvantages of branch and bond method and Gomory's cutting plane method give motivation to develop Step algorithm. Proposed method i.e. step algorithm works on the principle that in linear programming problem optimum solution always lies on the boundary of feasible region in maximization as well as in minimization problem. Furthermore, step algorithm is less complex and can give multi-optima solution in one go. To test the effectiveness of step algorithm few instances of integer programming problem with two variables are experimented. Obtained results prove that step algorithm is more efficient, easy and less time consuming.

Keywords

- Linear programming
- Integer type
- Gomory's cutting plane algorithm
- Branch and bound method
- Multi-optima

1. Introduction

The algorithm is constructed to solve positive integer type linear programming problems with two variables and n linear constraints. By word positive integer type, it is meant that the solution will focus on first quadrant of graph and will have integer values for both the variables. It is only applicable for two variables as the algorithm involves plotting one variable on x-axis and other on y-axis on graph. It can accommodate any number of linear constraints, as the number of constraints combine to give a feasible region in the graph. It is applicable on both types of problems, whether it be maximisation, or it be minimisation.

The principle behind this algorithm is that for linear programming type problems, the optimum solution always lies on the border of feasible region. Following the above principle, the algorithm tries to follow the boundary of feasible region. This happens such that it resembles the steps in case of solving for integers. For maximisation type of problem, the step-down algorithm is followed as it starts from y-axis and tries to reach x-axis. For minimization type of problem, the step-up algorithm is followed as it starts from x-axis and tries to reach y-axis.

2. Algorithm

The algorithm has three sub algorithms. The First one is named as general part. The Second and Third are part A and part B respectively. The general part is to be followed by all the problems. At the end of general part there are three outcomes: (a) either follow Part A algorithm (b) either follow Part B algorithm (c)either stop with general part and solution is obtained.

In algorithm, to begin with, the linear objective function with two variables is required. This objective function should also have number of linear constraints including the constraint of positivity. The proceeding step is to plot x-axis and y-axis on a graph paper such that one unit on graph is equal to one integer value. Then by plotting all the constraints, a feasible region is found. Feasible regions are of two types; bounded and unbounded. The algorithm then divides path into two depending on type of feasible region.

In case of unbounded region, check whether problem is of maximisation of objective function or it is of minimisation of objective function. For unbounded region and case of maximisation of objective function, the solution lies at infinity and the maximum answer is also infinite. Now for case of minimisation, there arises the need to follow Part B algorithm, and end general algorithm.

In case of Bounded region, there lies two possibilities, either the case is of maximisation of objective function or it is the case of minimisation of objective function. For maximisation of objective function in case of bounded region, there arises the need to exit general algorithm and follow Part A algorithm to obtain the solution.

In case of minimisation in bounded region, there are three cases possible; either (a)both the co-efficient in objective function are positive (b) Co-efficient of x is positive and y is negative (c) Co-efficient of y is positive and x is negative. The case of both the negative co-efficient is not considered as this type of problem becomes the problem of maximisation with a sign change. If both the co-efficient are positive, then the solution lies at Origin and the answer is 0. If co-efficient of x is negative and y is positive, the answer lies at the greatest integer point on x-axis in the feasible region. If co-efficient of y is negative and x is positive, the answer lies at the greatest integer point on y-axis in the feasible region. With this there comes the end of general part algorithm.

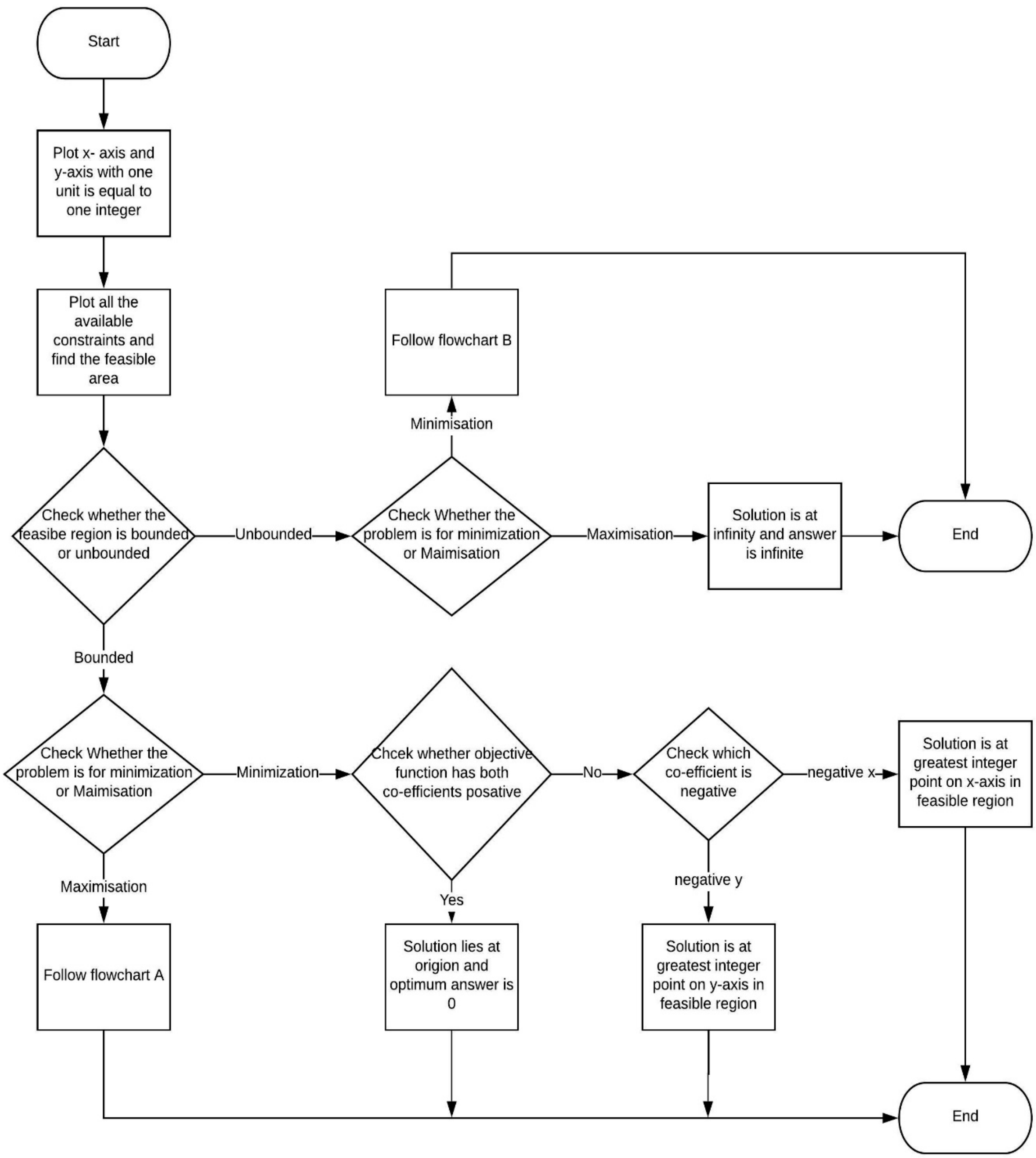

Figure 1. General part algorithm

To begin with Part A, select the greatest integer point on y-axis in the feasible region. Evaluate the value of objective function at that point and list down that point as well as the value of objective function at that point. Now move one unit right along x-axis at same level of y-axis. After every movement on the graph, always check whether the y coordinate of new point is negative or positive. If it is negative, then the algorithm ends at that point; and if it is positive then the algorithm goes ahead. Going ahead, for new point, check whether the point is in the feasible region or not. If the newly obtained point is in the region, then evaluate objective function at that point and list down the point as well as the value of objective function at that point. If the new obtained point is not in the region, then there is need to do step down process. Step down process is to move one step down along y-axis from the same point. Now another point is obtained, such that the shape formed in the graph will resemble a single step of stairs. Now the process of checking in the region needs to be repeated. If the newly obtained point is in the region, then again evaluate objective function at that point and list down the point as well as the value of objective function at that point. If not in the region, perform step down process again. This loop goes on until, the Y coordinate of the newly obtained point is negative.Now from the list of all the points and their evaluations; find the maximum value. The point at which maximum value lies is the solution for our integer type linear programming problem, and the maximum value corresponding to it is the optimum solution. There may be the case where in the list multiple maximum values are found. Such case is the case of multiple optima. All the points where optimum solution is available, are obtained in one go. This is the advantage of this method over other traditional methods. Also, in manual manner this algorithm is easy to perform.

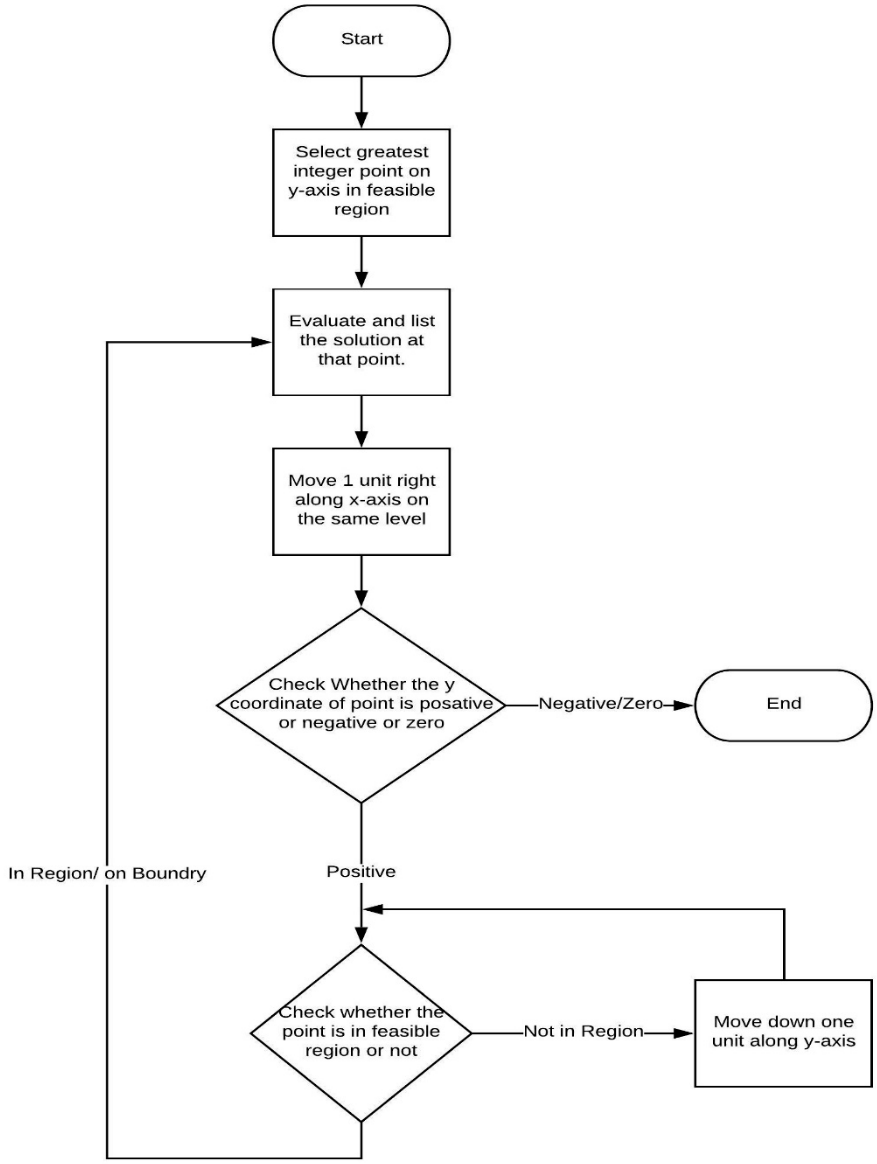

Figure 2. Part A algorithm

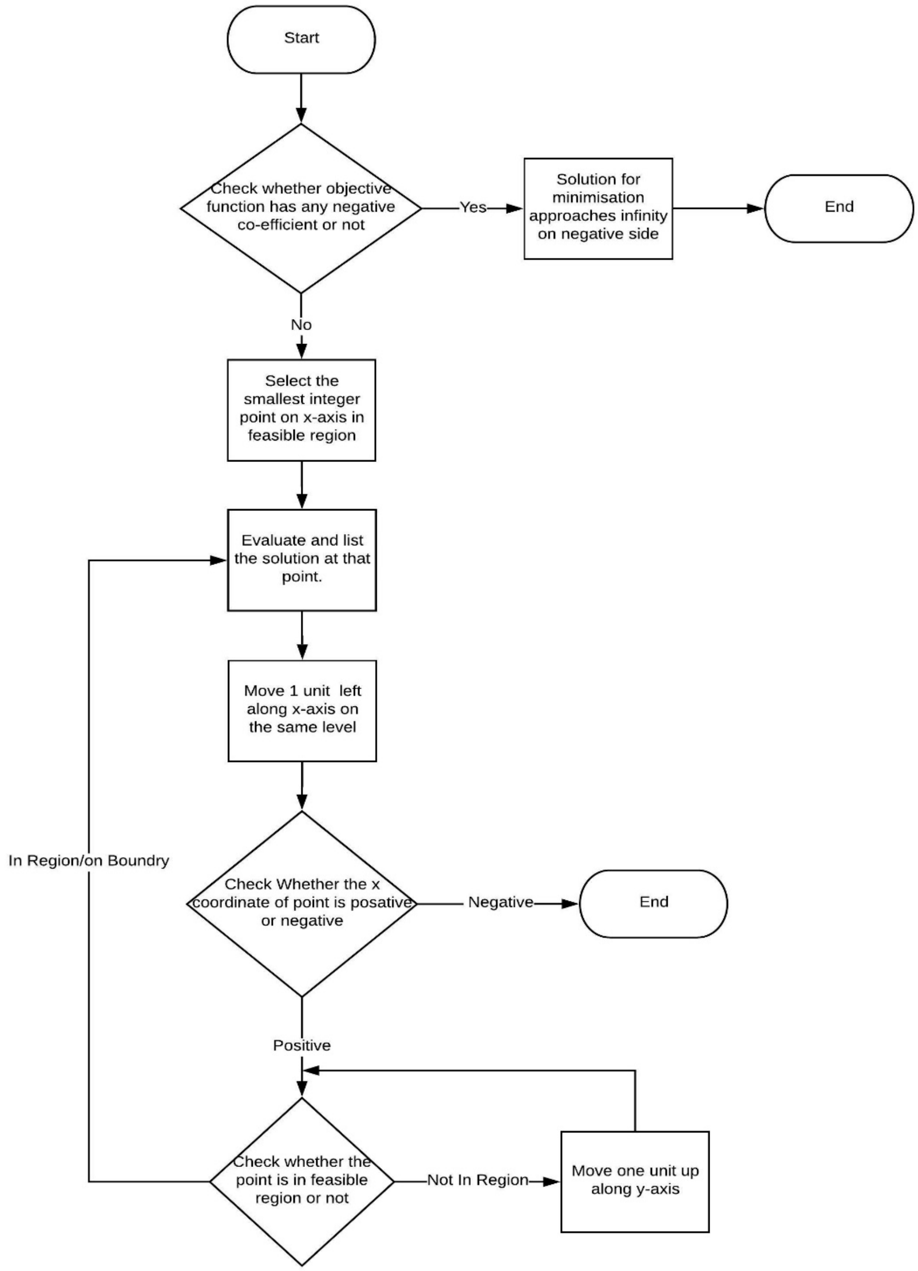

Figure 3. Part B algorithm

Now for Part B algorithm, the process remains similar to Part A algorithm, just the difference is that in Part A, the step-down rule was followed but now in Part B step-up will be followed. The critical difference is that in Part A, the movement was from y-axis toward x-axis and in case of Part B, the movement will be from x-axis to y-axis. And also, in Part A algorithm, the solution was made for bounded region, but in Part B algorithm, the solution is for unbounded region

As Part B algorithm is for unbounded region, there are chances that solution for minimisation may approach infinity on negative side. If in objective function, any one co-efficient has negative sign, then the solution will approach to infinity on negative side. Now select the smallest integer point on x-axis in the feasible region. Evaluate the value of objective function at that point and list down the point as well as the solution of objective function at that point. Next is to move one unit left on the x-axis, keeping level of y constant. Check whether the x-coordinate of newly obtained point is positive or negative. If it is negative, then the algorithm ends, and if it is positive, proceed further. Proceeding further, check whether the newly obtained point is in the region or not. If the point is in the region, then evaluate the value of objective function at that point and list down that point as well as the value of objective function at that point. If the point is not in the region, move one unit up along y-axis, keeping x-coordinate constant. If still the new point is not in the region, repeat the same process again. This process is called step-up, as it will resemble step of staircase. When the new point is in the region, stop and evaluate the value of objective function, as well as list down the point and value of objective function at that point. Now again run in the loop, move left one unit along x-axis, keeping y-coordinate constant. If the point is in the region, evaluate; if not in the region, move up one unit, following the algorithm.

Run in the loop, till the algorithm ends. After the end of algorithm, from the list of points and their evaluations, find the most minimum value. It can be said that the solution of minimisation lies at that point and the corresponding solution is the answer. There may be the possibility that there is multiple optimum solution. This case is evaluated easily by this algorithm. Select all the points where the minimum value is obtained.

3. Illustrations

3.1 Illustration 1-

Maximise Z= x + y; With the constraints $x \geq 0$; $y \geq 0$; $2x + 5y \leq 16$; $6x + 5y \leq 30$

Solution – After plotting all the constraints, it was found that problem is of bounded type. It will follow General part algorithm and then will enter Part A algorithm.

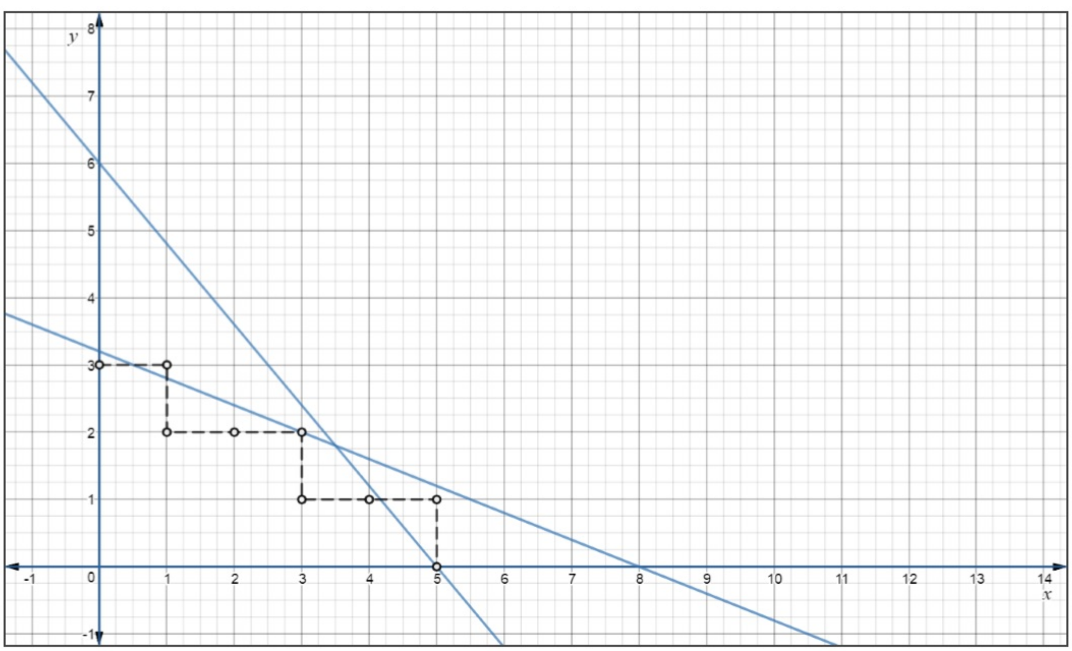

Figure 4. Solution graph for Illustration 1

- List of the points and their evaluations
  - (0,3) → 3
  - (1,2) → 3
  - (2,2) → 4
  - (3,2) → 5
  - (3,1) → 4
  - (4,1) → 5
  - (5,0) → 5

This is the case of multiple optima, the maximum solution lies at (3,2), (3,1) and (5,0) and the maximum answer is 5.

3.2 Illustration 2 –

Minimise Z = 5x + 4y; With constraints x ≥ 0; y ≥ 0; 3x + 2y ≥ 5; 2x + 3y ≥ 7

Solution – After plotting all the constraints, it was found that this is the case of unbounded region. Th problem will follow General part algorithm and then it will enter Part B algorithm.

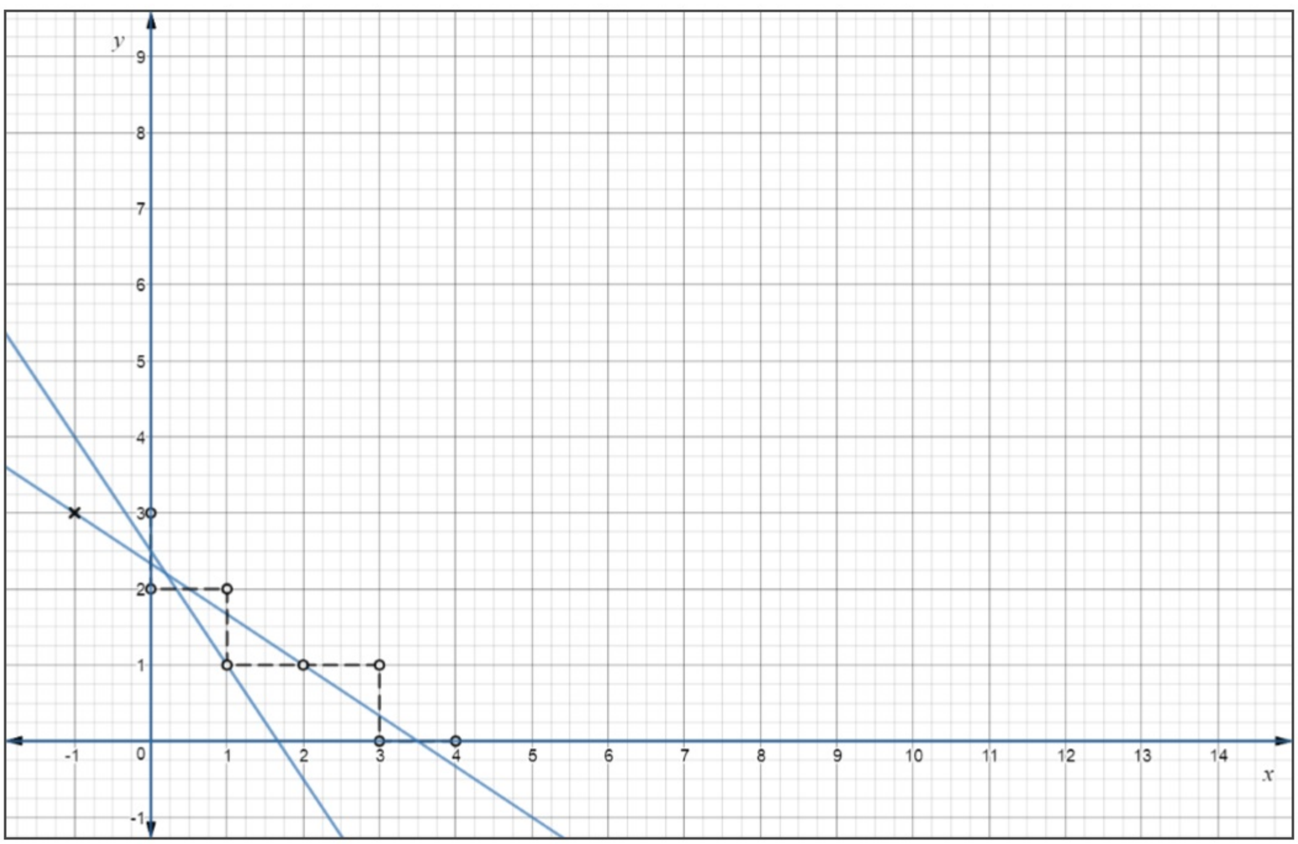

Figure 5. Solution graph for Illustration 2

- List of Points and their evaluations
  - (4,0) → 20
  - (3,1) → 19
  - (2,1) → 14
  - (1,2) → 13
  - (0,3) → 12

The Minimum solution lies at (0,3) and the minimum solution is 12.

4. Conclusion

For solving positive-integer type-linear programming problem (ILPP) with two variables and n linear constraints; the new algorithm is developed in this paper. The algorithm is developed on the principle that, solution for ILPP type of problems always lie in the proximity of boundary of feasible region. Comparing this algorithm with existing methods, this algorithm is easy to solve manually. And also this algorithm gives multiple optimum solution in one go, where as in existing methods, multiple iterations are required in case of multiple optimum solutions.

References

- Anthony Przybylski, Xavier Gandibleux, Multi-Objective Branch and Bound, European Journal of Operational Research (2017), doi: 10.1016/j.ejor.2017.01.032.
- M. K¨oksalan, B. Lokman, Finding nadir points in multi-objective integer programs, Journal of Global Optimization 62 (2015) 55–77.
- O. Ozpeynirci, M. K¨oksalan, An Exact Algorithm for Finding Extreme Supported Nondom- ¨ inated Points of Multiobjective Mixed Integer Programs, Management Science 56 (2010) 2302–2315.
- B. Villarreal, M. H. Karwan, Multicriteria integer programming: A (hybrid) dynamic programming recursive approach, Mathematical Programming 21 (1) (1981) 204–223.
- M. Ehrgott, X. Gandibleux, Bound sets for biobjective combinatorial optimization problems, Computers & Operations Research 34 (2007) 2674–2694.
- C. Bazgan, H. Hugot, D. Vanderpooten, Solving efficiently the 0-1 multi-objective knapsack problem, Computers & Operations Research 36 (2009) 260–279.
- P. Belotti, B. Soylu, M. M. Wiecek, A branch-and-bound algorithm for biobjective Mixedinteger Programs. Technical Report, URL www.optimization-online.org/DB_FILE/2013/ 01/3719.pdf, 2013.
- T. Stidsen, K. A. Andersen, B. Dammann, A Branch and Bound Algorithm for a Class of Biobjective Mixed Integer Programs, Management Science 60 (4) (2014) 1009-1032.
- F. Sourd, O. Spanjaard, A Multiobjective Branch-and-Bound Framework: Application to the Biobjective Spanning Tree Problem, INFORMS Journal on Computing 20 (2008) 472–484.
- G. Kiziltan, E. Yucaoglu, An algorithm for multiobjective zero-one linear programming, Management Science 29 (1983) 1444–1453.
- N. Boland, H. Charkhgard, M. Savelsbergh, A Criterion Space Search Algorithm for Biobjective Mixed Integer Programming: The Triangle Search Method, INFORMS Journal on Computing 27 (2015) 597–618.
- G. Mavrotas, D. Diakoulaki, Multi-criteria branch and bound: A vector maximization algorithm for Mixed 0-1 Multiple Objective Linear Programming, Applied Mathematics and Computation 171 (1) (2005) 53–71.
- J. Hooker, Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction., John Wiley & Sons, New York, 2000.

Biographies

Monal Vaishnani is fourth year bachelor student of Industrial engineering department of Pandit Deendayal Petroleum University, Gandhinagar, Gujarat, India. His areas of interest are operations research, supply chain management, Materials management and safety.

Poonam Savsani is assistant professor, in Industrial department in Pandit Deendayal Petroleum University, Gandhinagar, Gujarat, India. She has pursued Ph.D. in Robot trajectory optimization from Pacific University in year 2014. She has pursued her M.E. in machine design from B.V.M Engineering college, V.V.Nagar, India. She completed her B.E. in mechanical engineering from V.V.P Engineering college, Rajkot, India. Her areas of research are Robotics simulation, Advanced optimization techniques and motion tracking.