INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

Linear Programming
Simplex and Interior Point Methods

Author:
Naveen Balaji
roll no: 170420

Assignment of Algorithmic studies and analyses
Optimization Methods in Engineering Design
Course [ME752]

June 9, 2020

Introduction

Linear Programming (LP) occupies a significant area in many areas of applied science today. LP is used extensively in engineering, operations, research, economics, and even more abstract mathematical areas such as combinatorics. Owing to their vast applicability, there has been much interest in finding efficient algorithms that find the best solutions to linear programs. The very first algorithm used for LP was the simplex algorithm. After 20 years, problems arose around the ability of simplex over large problems. This challenge led to the introduction to the interior point methods for LP, which gave an efficient performance guarantee.

In this article, We focus on the two main algorithms of LP, which is the simplex and interior point. The implementation of the algorithms are guided by the *Linear Programming using MATLAB* book [8]. The aim of report is to compare and analyze the algorithms based on their time complexity. In the next section, we introduce simplex implementation and its computational study. Then interior-point implementation and its computational study. After the implementation of the algorithm, a comparison of benchmark problems between both methods is shown. By the end, we illustrate the graphical visualization of the two algorithms to demonstrate when we should prefer simplex or interior-point methods depending on the problem's nature.

Simplex Method

The most well-known method for solving LPs is the simplex algorithm developed by George B. Dantzig [2] [1]. The simplex algorithm begins with a primal feasible basis and uses algebraic operations until an optimum solution is computed. It searches for an optimal solution by moving from one feasible solution to another, along the edges of the feasible region. The simplex method tends to run in linear time according to the number of constraints in the problem. The worst case time complexity of the algorithm is exponential behaviour $O(2^n - 1)$ [5].

This report implements the primal simplex method to solve LP. Presolve methods are used in order to eliminate redundant constraints and variables, transform bounds of single structural variables, and improve sparsity. This technique is able to spot possible infeasibiltiy and unboundedness of the problem. Then the algorithm performs successive pivoting operations in order to reach the optimal solution. The pivoting rule used here is Dantzig rule as mentioned in [2].

Simplex: Computational Study

Benchmark problems are publicly available LPs that are hard to solve. Most of them are degenerate LPs. Many libraries containing LPs exist. The most widely used benchmark library is the Netlib LP test set [7]. All computational studies have been performed on a quad-processor Intel Core i5-8250U (3.4 GHz) running under Linux Ubuntu 16.04 . The MATLAB version used is R2018a.

The aim of the computational study is to analyse the efficiency of the simplex algorithm over various problems. The test set used in the computational study is 19 LPs from the Netlib(optimal, Kennington, and infeasible LPs)[7]. The Table present the total execution time and iterations of the benchmark problems

Table 1: Result of the implemented Primal Simplex algorithm over the benchmark set

Problem Name	Constraints	Variables	Non-zeros	Time (s)	Iterations
beaconfd	173	262	3,375	0.466	26
cari	400	1,200	152,800	12.240	771
farm	7	12	36	0.248	7
itest6	11	8	20	0.303	2
klein2	477	54	4,585	0.460	1
nsic1	451	463	2,853	0.520	369
nsic2	465	463	3,015	0.627	524
osa-07	1,118	23,949	143,694	7.793	1,056
osa-14	2,337	52,460	314,760	52.323	2,478
osa-30	4,350	100,024	600,138	295.284	4,681
rosen1	520	1,024	23,274	8.142	3,445
rosen2	1,032	2,048	46,504	39.267	6,878
rosen7	264	512	7,770	0.668	659
rosen8	520	1,024	15,538	2.060	1,300
rosen10	2,056	4,096	62,136	81.33	8,311
sc205	205	203	551	0.537	148
scfxm1	330	457	2,589	1.295	384
sctap2	1,090	1,024	23,274	2.704	1,501
sctap3	1,480	2,480	8,874	6.729	2,488

Interior Point Method

Initially the proposed Interior Point Methods (IPMs) that traverse across the interior of feasible region did not compete with the simplex algorithm in practice due to the expensive computation and numerical instability [3]. The first IPM that outperformed simplex algorithm was proposed by Kamarkar [4] with a lower bound on the computational complexity of O(n L), requiring a total of O($n^{3.5}$ L) bit operations, where L is the length of input data. We focused on primal-dual IPMs, due to their wide popularity and performance. In this report, we present the IPM based on Mehrotra's Predictor-Corrector (MPC) method [6]. The main aspects of the IPM noted for the implementation in this report is

- **Central path:** The central path is a trajectory of feasible solution in the interior of the feasible set which leads to an optimal point at the boundary. As μ decreases to 0, the central path C converges to the primal-dual solution through newton method with approximated step length.

- **Initial Point:** Mehrotra proposed the following heuristics to obtain the starting point of the problem by below method. If it is infeasible then the point is moved by step length along the averaged direction.

$$x_0 = A^T(A\,A^T)^{-1}\,b$$
$$w_0 = (A\,A^T)^{-1}\,Ac$$
$$s_0 = c - A^T\,w_0$$

Interior Point: Computational Study

The aim of the computational study is to analyse the efficiency of the Interior point method over various problems. The test set used in the computational study is the same 19 LPs from the Netlib(optimal, Kennington, and infeasible LPs)[7] as in previous section. The Table present the total execution time and iterations of the benchmark problems

Table 2: Result of the implemented Interior Point method over the benchmark-set

Problem Name	Constraints	Variables	Non-zeros	Time (s)	Iterations
beaconfd	173	262	3,375	0.344	11
cari	400	1,200	152,800	7.674	18
farm	7	12	36	0.204	8
itest6	11	8	20	0.253	2
klein2	477	54	4,585	0.185	1
nsic1	451	463	2,853	0.278	9
nsic2	465	463	3,015	0.575	74
osa-07	1,118	23,949	143,694	4.234	19
osa-14	2,337	52,460	314,760	19.456	16
osa-30	4,350	100,024	600,138	82.405	18
rosen1	520	1,024	23,274	0.419	13
rosen2	1,032	2,048	46,504	1.274	15
rosen7	264	512	7,770	0.183	11
rosen8	520	1,024	15,538	0.426	14
rosen10	2,056	4,096	62,136	4.607	16
sc205	205	203	551	0.406	10
scfxm1	330	457	2,589	0.74	22
sctap2	1,090	1,024	23,274	0.706	13
sctap3	1,480	2,480	8,874	1.069	13

In assessing the value of LP techniques, such as interior point methods, there are many issues to be considered. Among them are stability, ease of use, restart efficiency, compatibility with every cases and the processing speed.

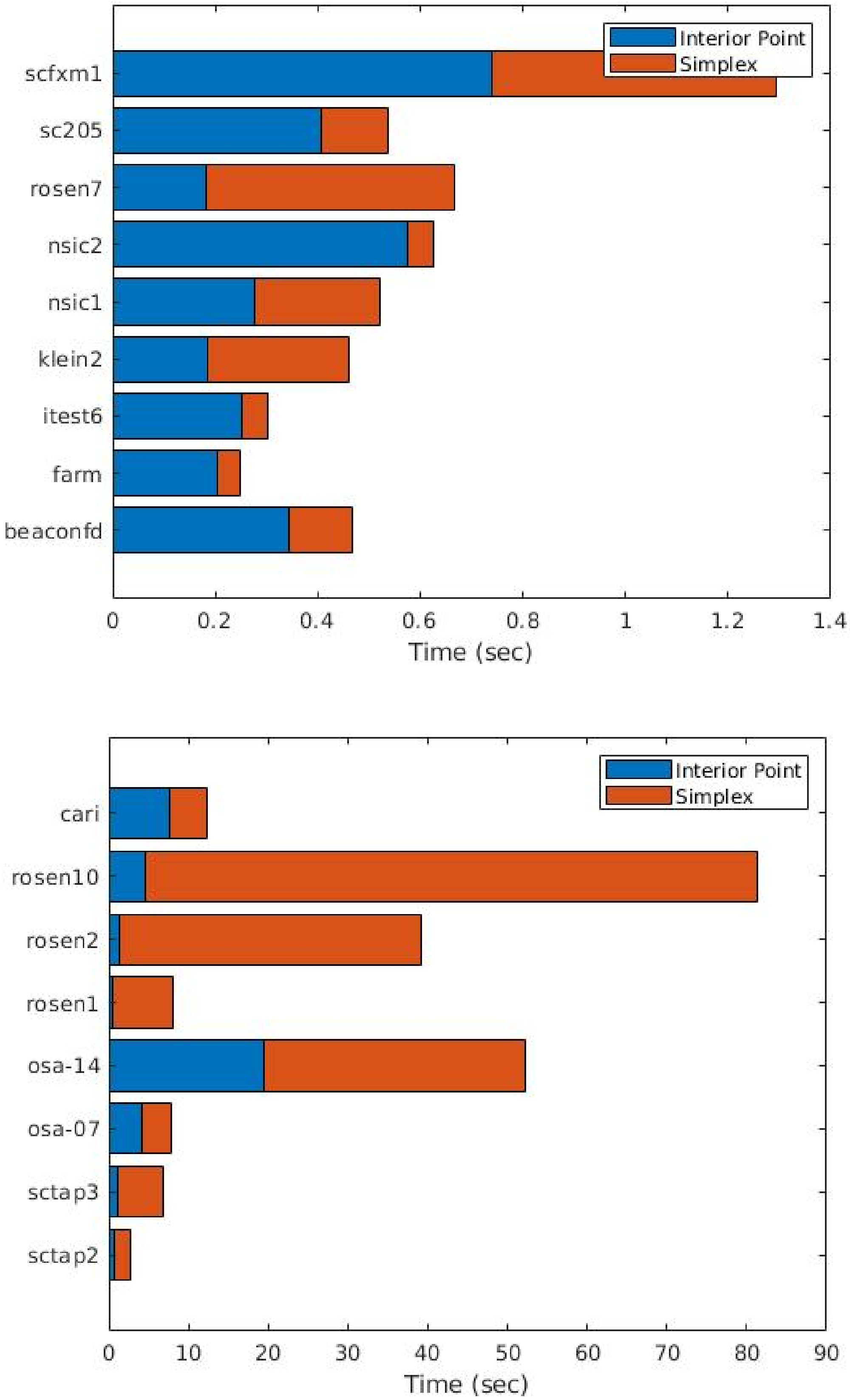


Figure 1: Benchmark Problems: Comparison between the Simplex and Interior Point

Benchmark Comparison

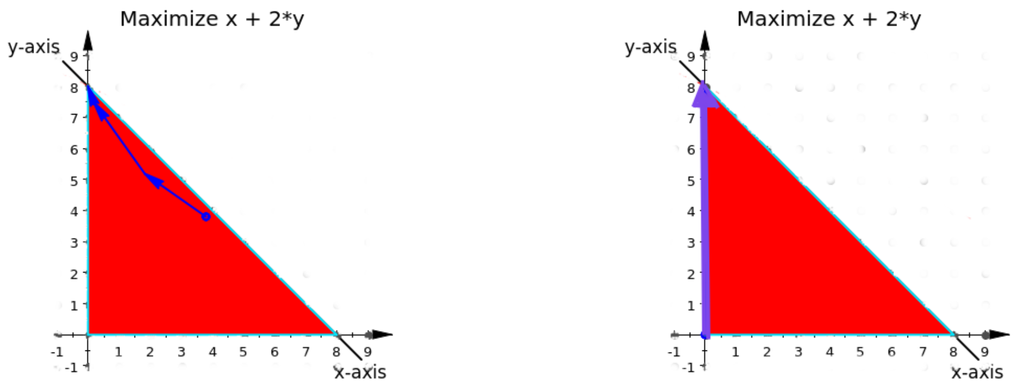
We have tested the benchmark problems on both the simplex and interior point method. Figure 1 is the comparison bar graph showing the time complexity of the algorithms. The first figure 1(a) is for the problems with relatively less complexity (average is 175 constraints and 250 variables) and the second figure 1(b) is for complex problems (2,300 constraints and 30,000 variables).

From the result, we can see that the interior point method is faster in all of the benchmark problems. The time complexity comparison we can observe from the both the figures. In first figure 1(a) the time taken by simplex method is comparable (nearly twice) to interior point. While in the second figure 1(b), where large problems presented the time taken by simplex is approximately ten times the interior point method. This verifies that for a complex linear programming problem with large numbers of decision variables and constraints Interior point method is more suitable.

Graphical Comparison

In the previous section, we have seen the benchmark comparison of both the methods. In this section we will visualize the implementation of the methods, how they iterate over time and converge to the solution. We have carried out test problems with linear constraint over two dimensions for simplicity of visualization.

$$\textbf{Problem 1:} \quad \textit{Maximize} \quad z = x_1 + 2x_2$$
$$\textit{subjected to} \quad x_1 + x_2 <= 8$$
$$\textit{and} \quad x_1\ , x_2 \ >= 0$$



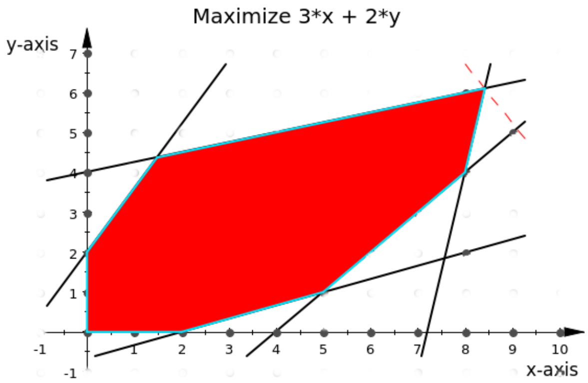
(a) Interior Point (iteration=3)

(b) Simplex (iteration=1)

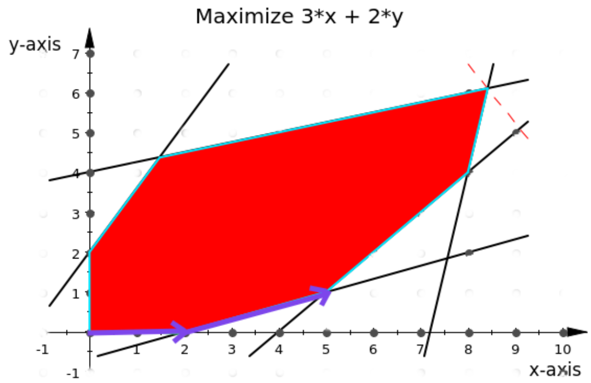
Figure 3: Visualization of Problem-1 Interior Point method and Simplex

The red color represent the feasible region of the linear constraints. The interior point method gets initialized from the $(4, 4)$ point and after three iterations it reaches the optimal point $(0, 8)$. The simplex method gets started from the origin and in a single iteration reaches the optimal point which is adjacent vertex. This infers that the simplex is a better algorithm, when constraints are low. The interior point needs five iterations to converge with tolerance error 10^{-5}.

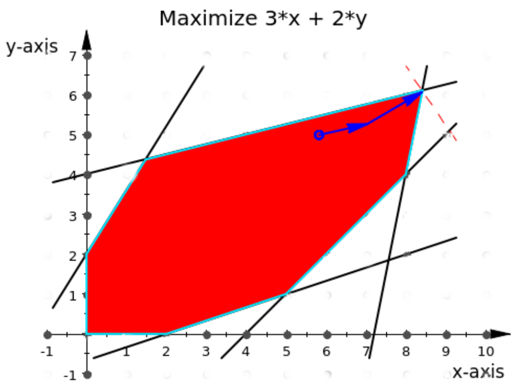
$$
\begin{array}{rrl}
\textbf{Problem 2:} & \textit{Maximize} & z = 3x_1 + 2x_2 \\
& \textit{subjected to} & x_1 - 3x_2 <= 2 \\
& & x_1 - x_2 <= 4 \\
& & 5x_1 - x_2 <= 36 \\
& & -4x_1 + 2.5x_2 <= 5 \\
& & -x_1 + 4x_2 <= 16 \\
& \textit{and} & x_1\ , x_2 \ >= 0
\end{array}
$$



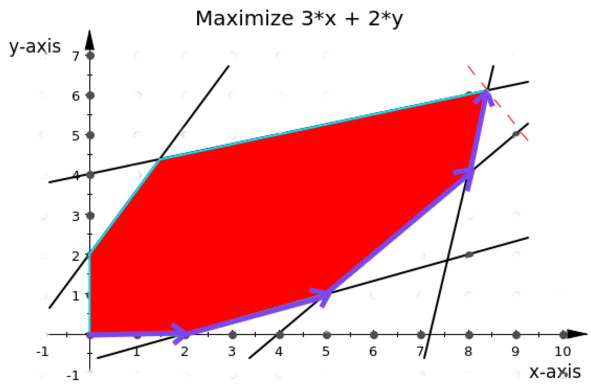
(a) Constrained Problem



(b) Simplex (iteration=3)



(a) Interior Point (iteration=3)



(b) Simplex (iteration=5)

Figure 4: Visualization of Problem-2 Interior Point method and Simplex

Problem-2 consists of seven linear constraints. The interior point method gets initialized from the $(5.826, 4.986)$ point and after two iterations it reaches the optimal point $(8.421, 6.105)$ as shown in Figure 4. The simplex method gets started from the origin, then searches its adjacent vertex for optimal point. Finally, simplex reaches the optimal point $(8.421, 6.105)$ after five iterations. This infers that when the constraints are higher the simplex is ineffective and interior point method is efficient.

Conclusion

We first discussed on the introduction and history of Linear Programming. We then discussed on the Simplex method and the Interior Point method to solve problems. We then evaluated both of them on the Libnet benchmark problems. We then finally compared geometrically the path of the iterations carried by the two methods. The simplex method was easier to solve involved less complexity compared to the interior method. The graphical visualization gave us basic working principle of the algorithm. We conclude that for a huge linear programming problem with large numbers of decision variables and constraints Interior point method is more suitable and for less complicated problem Simplex method is suit to give better results.

Problems Evaluated

In this report, We have implemented the benchmark problems individually for Simplex method and Interior point method. The benchmark problems are available in Netlib LP test set [7]. The major of the articles and books in linear programming uses this benchmark algorithms to compare the results. The table 1 and table 2 shows our implementation results. Our source book "Linear Programming using MATLAB" has used this dataset. It can be noted that these benchmark problems are complex high dimensional problems with thousands of variables and constraints. So we have mentioned only the name of the problems with the link in this section. In visualization of the algorithm section we have taken a simple problem to analyze it in a graphical way. We presented how both the algorithms iterate to their optimal solution. In this section, we have mentioned our 42 evaluation set as 21 problems on each of the method as follows

- beaconfd
- cari
- farm
- itest6
- klein2
- sctap2
- sctap3
- osa-07
- osa-14
- rosen1
- osa-30
- rosen2
- rosen7
- rosen8
- rosen10
- sc205
- scfxm1
- nsic1
- nsic2

- $$\begin{aligned} Maximize \quad & z = x_1 + 2x_2 \\ subjected\ to \quad & x_1 + x_2 <= 8 \\ and \quad & x_1\ , x_2\ >= 0 \end{aligned}$$

$$
\begin{aligned}
\bullet \qquad \mathit{Maximize} \quad & z = 3x_1 + 2x_2 \\
\mathit{subjected\ to} \quad & x_1 - 3x_2 <= 2 \\
& x_1 - x_2 <= 4 \\
& 5x_1 - x_2 <= 36 \\
& -4x_1 + 2.5x_2 <= 5 \\
& -x_1 + 4x_2 <= 16 \\
\mathit{and} \quad & x_1 \, , x_2 \; >= 0
\end{aligned}
$$

The results of the implementations were described in detail at Section2 and Section3.

Bibliography

[1] George B Dantzig. "Programming in a linear structure". In: *Bulletin of the American Mathematical Society*. Vol. 54. 11. AMER MATHEMATICAL SOC 201 CHARLES ST, PROVIDENCE, RI 02940-2213. 1948, pp. 1074–1074.

[2] George Bernard Dantzig. *Linear programming and extensions*. Vol. 48. Princeton university press, 1998.

[3] KR Frisch. "The logarithmic potential method of convex programming". In: *Memorandum, University Institute of Economics, Oslo* 5.6 (1955).

[4] Narendra Karmarkar. "A new polynomial-time algorithm for linear programming". In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. 1984, pp. 302–311.

[5] Victor Klee and George J Minty. "How good is the simplex algorithm". In: *Inequalities* 3.3 (1972), pp. 159–175.

[6] Sanjay Mehrotra. "On the implementation of a primal-dual interior point method". In: *SIAM Journal on optimization* 2.4 (1992), pp. 575–601.

[7] *Netlib Repository*. URL: http://www.netlib.org/lp/data/.

[8] Nikolaos Ploskas and Nikolaos Samaras. *Linear Programming Using MATLAB®*. Vol. 127. Springer, 2017.