

Research Article

A New Technique for Determining Approximate Center of a Polytope

Syed Inayatullah , Maria Aman, Asma Rani, Hina Zaheer, and Tanveer Ahmed Siddiqi

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

Correspondence should be addressed to Syed Inayatullah; inayat@uok.edu.pk

Received 5 March 2019; Revised 19 June 2019; Accepted 23 June 2019; Published 15 November 2019

Academic Editor: Imed Kacem

Copyright © 2019 Syed Inayatullah et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this article, we have presented a method for finding the approximate center of a linear programming polytope. This method provides a point near the center of a polytope in few simple and easy steps. Geometrical interpretation and some numerical examples have also been presented to demonstrate the proposed approach and comparison of quality of the center obtained by using the new method with existing methods of finding exact and approximate centers. At the end, we also presented computational results on the randomly generated polytopes to compare the quality of the center obtained by using the new method.

1. Introduction

Linear programming (LP) is a mathematical technique for optimizing a linear function subject to a set of linear constraints and nonnegativity restrictions. Linear programs frequently show up in various areas of applied sciences today. The prime reason for this is their manageable, enormous impact in various disciplines; it has become a core research area of many mathematicians, economists, decision scientists, etc. Linear programming was developed during World War II, when a system with which to maximize the efficiency of resources was of utmost importance. Since then, many researchers have strived to advance their ideas and made centering of the polytope as a core step in the major optimization techniques (named as interior point methods) in science and industry.

2. Definitions of the Center of a Polytope

There are several ways to define the center of a polytope, it may be the center of gravity i.e., *centroid*, mean position of all vertices i.e., *vertex centroid*, point at the location where product of distances from all boundary lines is maximized i.e., *analytic center*, center of the least volume ellipsoid that contains the polytope, or the center of the biggest ball

inside the polytope. Therefore, the center of a polytope depends on the definition we are using. But, fortunately, all those definitions are equivalent in the sense that, as shown in [1], if we get a polynomial time algorithm for a center of a polytope, then that algorithm could also be used to construct a polynomial time algorithm for solving linear program.

Many techniques [2–4] are used for finding the center of a polytope, but they are taking so many iterations, slow in convergence, and require mostly complex computations.

Most of the interior point methods for solving LPs depend on the computations of a center finding method, either explicitly or implicitly [3, 5].

The analytic center [6–12] is no doubt the most used notion of center of a polytope in linear optimization because of its easy computation, but its disadvantage is that it can be pushed near the boundary of the polytope by using redundant constraints because its position depends on the spatial positions of the half-spaces that define the polytope. So in that case, analytic center may not look like located at a good central position (*see also* Section 3). The P-center [3], also to be discussed in Section 4, provides a much better center than the analytic center, but it is found in practice that it takes much longer time to obtain a good approximation of the P-center.

3. Analytic Center

Let S be a polytope described by a (normalized) system of linear inequalities:

$$\sum_{j=1}^n a_{ij}x_j \leq b_i, \quad i = 1, 2, \dots, m. \quad (1)$$

The analytic center of S is the point $\xi = (\xi_1, \xi_2, \dots, \xi_n) \in \mathfrak{R}^n$ which satisfies the following maximization problem:

$$\begin{aligned} & \text{Maximize} && \prod_{i=1}^m \left(b_i - \sum_{j=1}^n a_{ij}x_j \right) \\ & \text{subject to} && x \in S. \end{aligned} \quad (2)$$

When S is bounded, this maximization problem always has a solution.

In general, the analytic center depends on how the set of particular inequalities is defined. The addition of redundant inequalities could push analytic center towards the boundary. In the Sections 4 and 5, we discuss two new efficient and alternative methods, which could be used to find a good approximation of the central location of a polytope.

4. P-Center [3]

Consider a linear programming polytope described by a set $S = \{x \in \mathfrak{R}^n : Ax \leq b\}$, where $A \in \mathfrak{R}^{m \times n}$. Let $H_i = \{x : a_i^T x \leq b_i\}$ be the half-space corresponding to the i^{th} row of A and $G_i = \{x : a_i^T x = b_i\}$ be the corresponding hyperplane. The method assumes that the polytope is full dimensional. The overall technique is based on the following definitions of projections.

4.1. Orthogonal Feasible Projections. Let x^k be some feasible interior point; for each hyperplane G_i , we can easily generate two distinct points (projections) $P_i^+(x^k)$ and $P_i^-(x^k)$ on the boundary, defined by $P_i^+(x^k) = x^k + \theta_i a_i$ and $P_i^-(x^k) = x^k - \lambda_i a_i$, where $\theta_i = \max\{t : x^k + t a_i \in S\}$ and $\lambda_i = \max\{t : x^k - t a_i \in S\}$.

4.2. Central Location Using Orthogonal Projections. A vertex centroid of a polytope could be defined as the average of all points on the boundary and [3] defined the P-center as an approximation of vertex centroid, which could be obtained by taking average of some finite number of points on the boundary of a polytope. The main task of the method is to generate points on the boundary as much as possible. For this purpose, an initial feasible interior point is needed and then the method generates the iterates by taking a convex combination of the orthogonal projections into the hyperplanes associated with the inequalities that define the polytope.

Each G_i could generate two points $P_i^+(x^k)$ and $P_i^-(x^k)$ on the boundary of S and a midpoint $x_i^k = (P_i^+(x^k) + P_i^-(x^k))/2$ on the chord joining them. The new iterate x^{k+1} is obtained

by taking the average of all m midpoints $\{x_i^k : i = 1, \dots, m\}$. That is $x^{k+1} = \sum_{i=1}^m x_i^k / m$.

If $\max\{\text{abs}(x^{k+1} - x^k)\} < \varepsilon$ (where ε is the tolerance value), then stop with the result that “P-center is obtained up to tolerance level of ε .” Otherwise, perform the $(k + 1)$ th iteration taking x^{k+1} as the initial interior point.

Carlos defined it as an approximation to the vertex centroid because in every iteration, he has taken 2- m distinct points on the boundary in distinct directions. Method looks to be very effective and he has also shown that quality of the center is also very good, even most of the times the centrality of P-center is way better than analytic center, but practically, it is found that it become very slow to converge and to attain a particular tolerance. Sometimes, it requires a huge number of iterations just to reach near about centroid. Computational results are shown in the Section 6.

5. A New Approximation of Central Location: CN-Center

In this section, we describe a recursive version of the method described in Section 4. Because of the recursive nature of this method, it uses the most updated value of center for next computation so it has a quick movement towards the central location. We would call the central location obtained by this method as CN-center.

Overall, for a problem of m constraints, each iteration of this procedure holds m steps. Here onward, x_i^k denotes the coordinates of center obtained in i th intermediate step of k th iteration, x^k denotes the center obtained after k th iteration, and x^0 denotes the coordinates of the initial feasible point.

Here, \hat{x}^k would be used to represent the midpoint of $P_i^+(x^k)$ and $P_i^-(x^k)$, where $P_i^+(x^k)$ and $P_i^-(x^k)$ on the boundary defined by $P_i^+(x^k) = x^k + \theta_i a_i$ and $P_i^-(x^k) = x^k - \lambda_i a_i$, where $\theta_i = \max\{t : x^k + t a_i \in S\}$ and $\lambda_i = \max\{t : x^k - t a_i \in S\}$.

For any k th iteration, the method needs an interior feasible point x^{k-1} :

$$\begin{aligned} & \text{Step } k_1: \text{ compute } \hat{x}_1^k := 1/2(P_1^+(x^{k-1}) + P_1^-(x^{k-1})). \text{ Set} \\ & x_1^k := \hat{x}_1^k. \\ & \text{Step } k_i: \text{ compute } \hat{x}_i^k := 1/2(P_i^+(x_{i-1}^k) + P_i^-(x_{i-1}^k)). \text{ Set} \\ & x_i^k := 1/i(\sum_{j=1}^{i-1} x_j^k + \hat{x}_i^k), \quad \forall i = 2, \dots, m. \end{aligned}$$

Now, since all hyperplanes have contributed, we can set $x^k := x_m^k$. We can terminate the procedure when $\max\{\text{abs}(x^k - x^{k-1})\} < \varepsilon$, where ε is the required tolerance. If the tolerance level is not achieved, then the process could proceed to $(k + 1)$ th iteration, taking x^k as an initial point.

The main difference between computation of P-center and CN-center could simply be illustrated by performing initial two steps of the first iteration.

Iteration 1. Step 1: both the methods identically take an interior feasible point, say x^0 , as starting point and find two points $P_1^+(x^k)$ and $P_1^-(x^k)$ on the boundary of the feasible region using the direction of normal of first constraint. Then, both methods take the average of boundary points to get a new point, say x_1^0 .

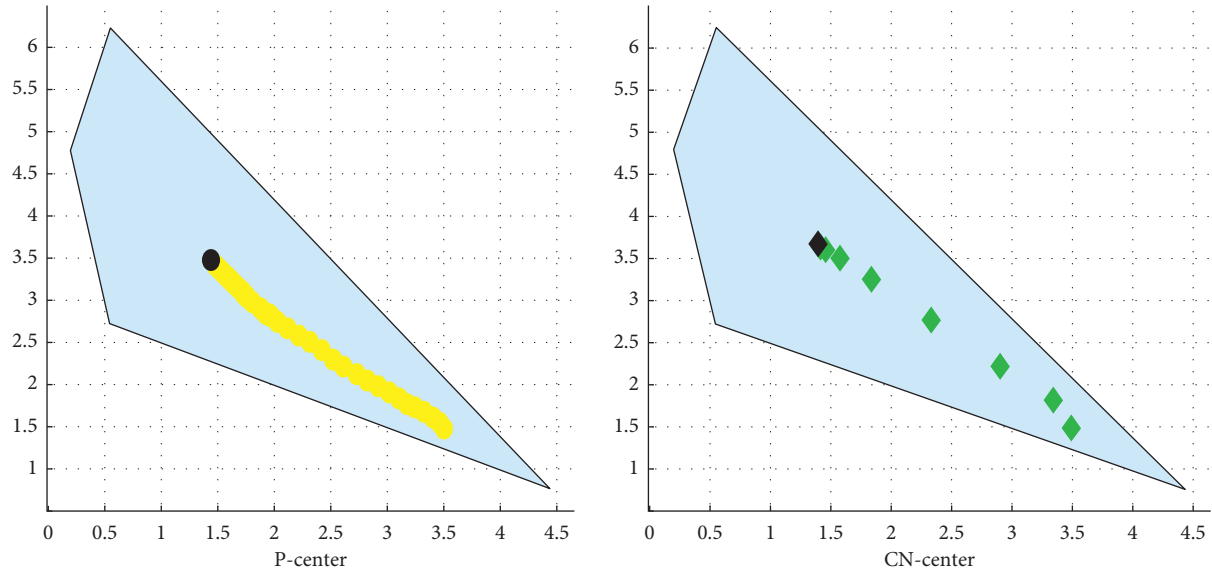


FIGURE 1: Polytope 1, P-center is obtained in 56 iterations but CN-center is obtained in just 11 iterations. Centrality wise both looked equivalent.

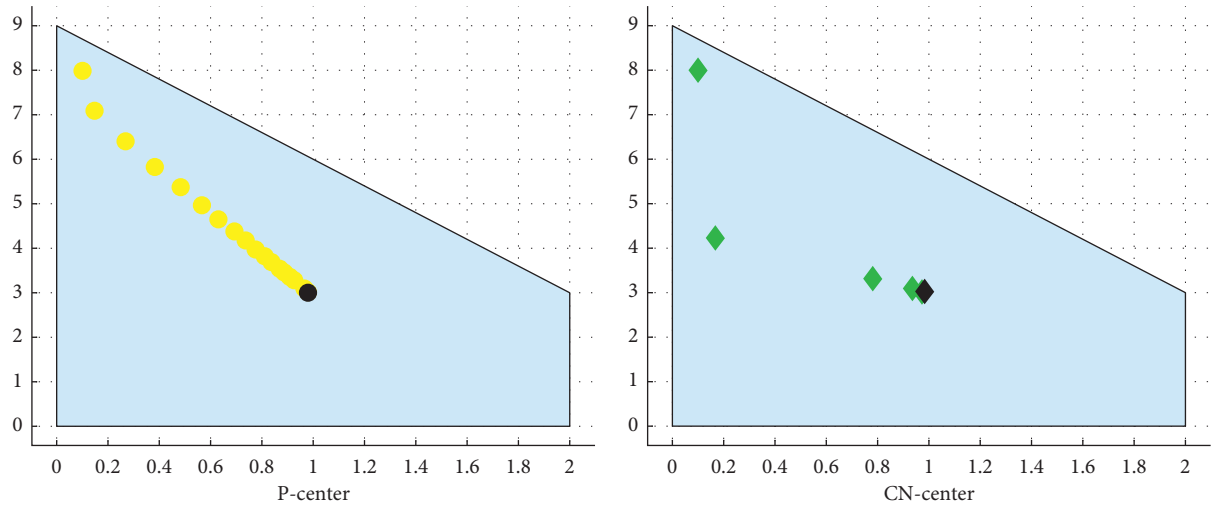


FIGURE 2: Polytope 2, convergence towards center.

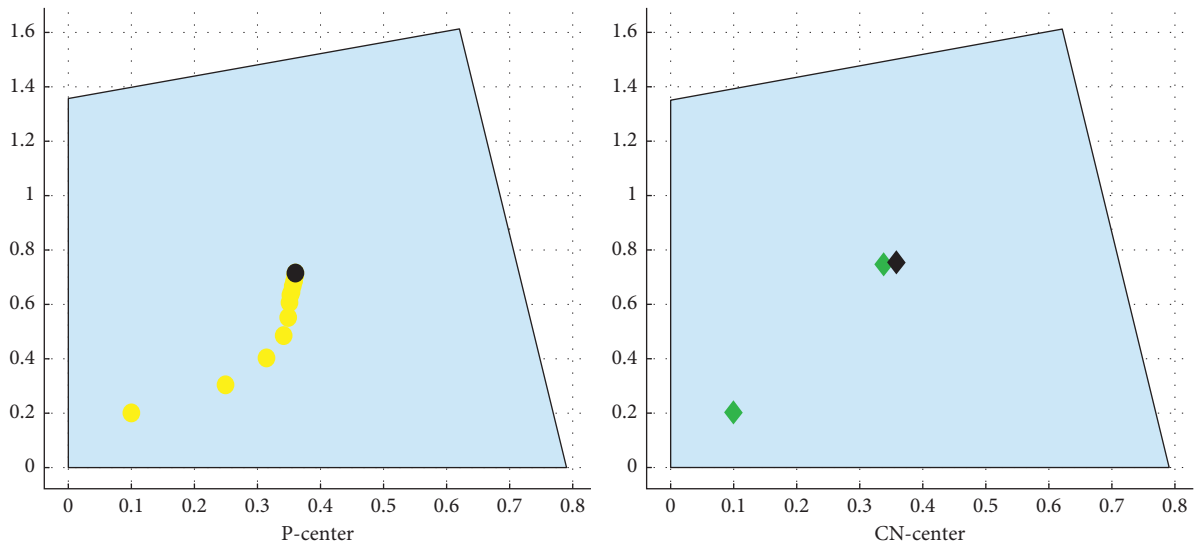


FIGURE 3: Polytope 3, convergence towards center.

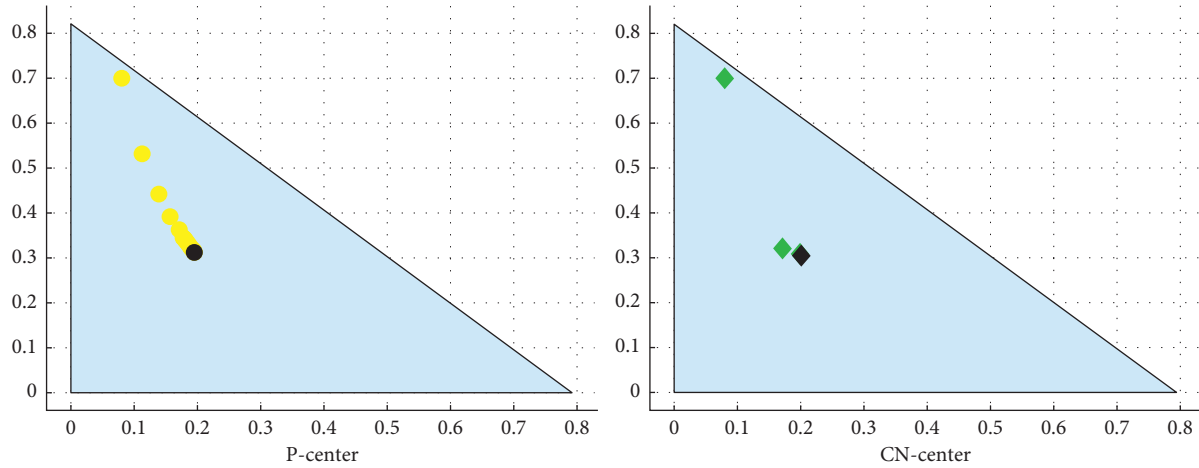


FIGURE 4: Polytope 4, convergence towards center.

TABLE 1: A comparison of number of iterations and quality of P-center and CN-center.

Polytope number	P-center			CN-center		
	$C(x)$	X	Iterations	$C(x)$	X	Iterations
1	0.7597	(1.4453, 3.4772)	56	0.6903	(1.3980, 3.6719)	11
2	0.9272	(0.9790, 3.0236)	38	0.9262	(0.9843, 3.0236)	7
3	0.3332	(0.3602, 0.7120)	16	0.3311	(0.3585, 0.7503)	2
4	0.1594	(0.1947, 0.3126)	12	0.1656	(0.2014, 0.3064)	3

TABLE 2: A numerical comparison between coordinates of analytic center, P-center, CN-center, and centroid.

Polytope number	Analytic center	P-center	CN-center	Centroid
1	(2.3932, 2.8696)	(1.4453, 3.4772)	(1.3980, 3.6719)	(1.4364, 0.6318)
2	(0.6340, 3.5490)	(0.9790, 3.0236)	(0.984, 3.0236)	(1.0000, 3.0000)
3	(0.3732, 0.8454)	(0.3602, 0.7120)	(0.358, 0.7503)	(0.3524, 0.7403)
4	(0.2152, 0.3704)	(0.1947, 0.3126)	(0.2014, 0.3064)	(0.2644, 0.2738)

Iteration 1. Step 2: difference of strategy starts from here; the method of P-center now again finds the average of two other boundary points using normal of 2^{nd} constraint and the same starting point x^0 , and calls it x_2^0 .

In contrast, method of CN-center takes normal of 2^{nd} constraint and a new point x_1^0 to obtain the average of boundary points, denoting it by \hat{x}_2^0 . Now, x_2^0 is obtained by taking the average of x_1^0 and \hat{x}_2^0 .

6. Computational Experiences

We performed numerical experiences to compare the CN-center against P-center, analytic center, and centroid in several polytopes in MATLAB and convergence with a suitable tolerance level. First, we show pictures of polytopes in 2D space to illustrate the convergence of the CN-center, P-center, analytic center, and centroid. Second, we present the numerical results in tables for randomly generated polytope.

To visualize the convergence of P-center and CN-center, we have taken some examples from [13]. Figures 1–4

represent four polytopes with convergences to their P-center and CN-center, respectively.

Table 1 shows the measure of coordinates of center x , number of iterations, and measure of centrality $C(x)$ for P-center and CN-center for each polytope illustrated in Figures 1–4

As we see in Table 1 as well as in Figures 1–4, centrality $C(x)$ of CN-center and P-center is almost equal but convergence towards CN-center is multiplex faster than P-center.

Now, we are taking other four different polytopes and presenting the comparison between observable quality of analytic center, P-center, centroid, and CN-center. Table 2 and Figure 5 represent numerical and graphical results. Based on Figures 1–5, it is easy to see that location-wise CN-center is almost equal to P-center but with a less number of iterations.

Our calculation shows that if we take the initial point near a narrow corner of the region, then there is a huge difference in the number of iterations for P-center and CN-center as shown in Figure 6, and if we take the initial point near a wide corner of the region (see Figure 7), difference in

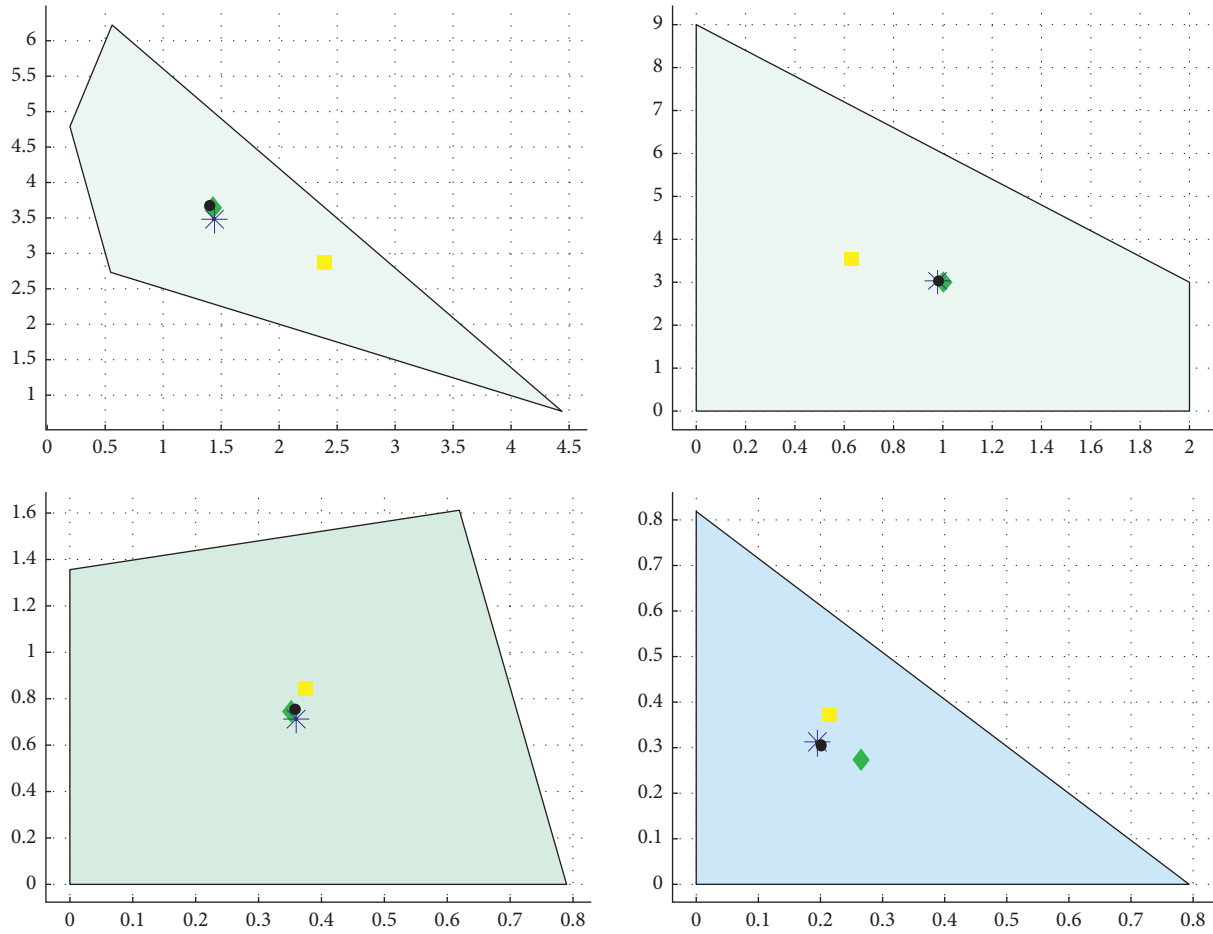


FIGURE 5: Graphical comparison of quality between analytic center (denoted by yellow square), P-center (denoted by blue asterisk), CN-center (denoted by black dot), and centroid (denoted by green diamond).

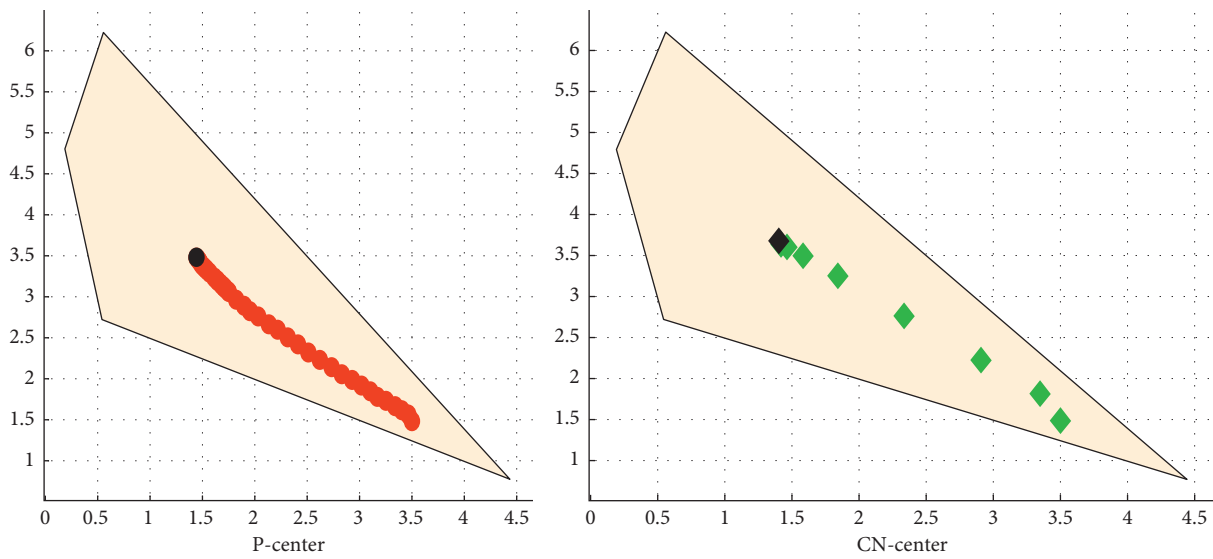


FIGURE 6: When initial point lies near a narrow corner, convergence of CN-center is much quicker than P-center.

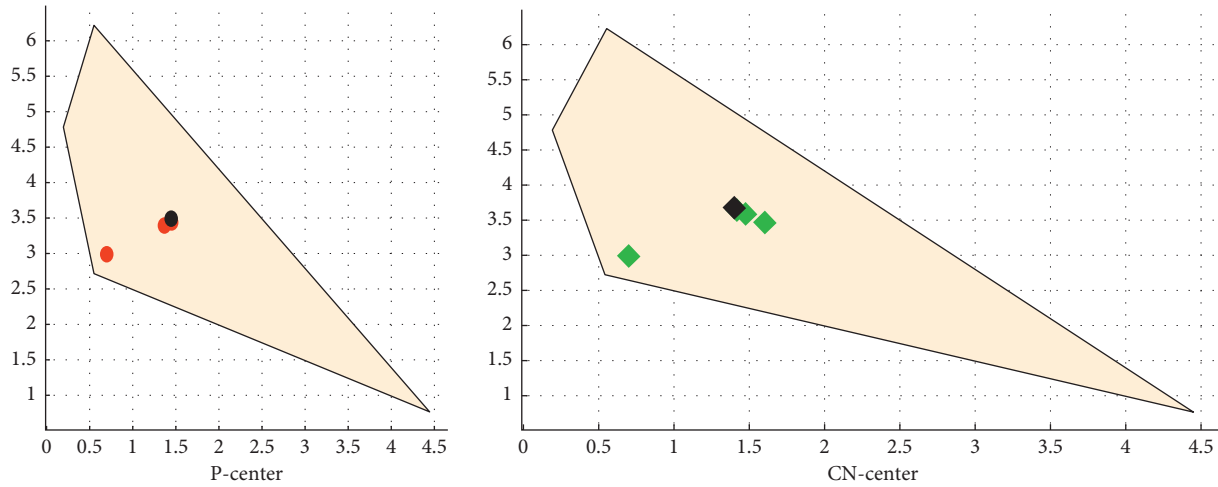


FIGURE 7: When initial point lies near a wide corner, both P-center and CN-center bear almost equal number of iterations.

TABLE 3: Computational results of randomly generated LPs using MATLAB.

Number of constraint	P-center		CN-center	
	Coordinates of center	Number of iteration	Coordinates of center	Number of iteration
25	(0.0965, 0.0752)	6	(0.1122, 0.0793)	1
	(0.0465, 0.0746)	8	(0.0469, 0.1064)	1
	(0.3132, 0.7748)	12	(0.3187, 0.8938)	1
	(0.4704, 0.0892)	24	(0.4837, 0.0898)	2
	(0.5839, 0.1610)	21	(0.5442, 0.1541)	2
	(0.0686, 0.2428)	32	(0.0731, 0.3027)	5
	(0.1244, 0.3280)	12	(0.1294, 0.3847)	1
	(0.3476, 0.0421)	24	(0.3151, 0.0403)	4
	(0.2793, 0.1585)	12	(0.2714, 0.1569)	2
	(0.2134, 0.0223)	2	(0.2255, 0.0221)	3
	(0.1856, 0.0341)	23	(0.2183, 0.03257)	3
50	(0.0346, 0.0922)	13	(0.0328, 0.1205)	1
	(0.0761, 0.0101)	19	(0.0943, 0.0098)	2
	(0.1607, 0.2225)	6	(0.1714, 0.2511)	1
	(0.1092, 0.0408)	17	(0.1538, 0.0368)	2
	(0.1669, 0.0229)	19	(0.1592, 0.0229)	1
	(0.2277, 0.0482)	28	(0.2159, 0.0479)	2
	(0.4047, 0.03670)	31	(0.3385, 0.0342)	3
	(0.1925, 0.3076)	14	(0.1881, 0.3085)	1
	(0.1546, 0.1301)	12	(0.1678, 0.1517)	1
	(0.2142, 0.0126)	89	(0.3065, 0.0111)	10
	(0.3657, 0.2198)	16	(0.370951, 0.2122)	1

TABLE 4: Comparison of average number of iterations for higher dimensional random LPs.

Order	P-center	CN-center
3 × 5	62.71	16.57
5 × 3	35.85	8.23
5 × 5	85.5	55.33
10 × 5	136.8	34.4
10 × 10	476.44	77.22
15 × 15	650.71	249.85
15 × 10	348.6	96.34
20 × 20	331.44	258.88
30 × 20	213.2	162.7
20 × 30	508.2	236
30 × 30	206.75	193.125

Note: here, $\epsilon = 0.001$ is more than enough. If we observe the convergence pattern of CN-center, we can see that convergence is initially fast and gets slower in later iterations. Generally, we do not need exact central point; for practical purposes, any good central location is enough for working. So a good central point is obtainable by CN-center within just a few iterations for a 100×100 or even a very high dimensional problem.

the number of iterations for P-center and CN-center would not be so significant.

Now, Table 3 presents some computational results on number of iterations needed for obtaining P-center and CN-center of randomly generated 2D LPs with higher number of constraints.

Finally, results for higher dimensional random LPs are presented in Table 4. Here, we have taken the average of number of iterations of 20 random LPs of each order with $\varepsilon = 0.001$. Results showed that the new approach is still superior in efficiency even in higher dimensional problems.

7. Applications

There are a lot of areas where approximate center finding methods for LPs could be used, for example, solving both linear and general convex programming [13], the support vector machine (SVM) solution that corresponds to the center of the largest sphere inscribed in version space [9, 14], computing cubature formulae [15], and sphere method for linear programming [16].

8. Conclusion

In this paper, we have presented a modified form of P-center [3] and called it as CN-center. Our experimental results show that quality of centrality of P-center and CN-center is almost same, but in terms of number of iterations, CN-center is much faster in computation of a good central location in the feasible region in lower and as well as in higher dimensional problems.

Generally, finding the central location of an LP is the main crucial step for most of the interior point methods. Usually, we do not need the exact center instead a good central location would be enough if it is obtained in less number of computations. So, in this sense, CN-center is a better option to use instead of P-center or analytic center.

Data Availability

Data were randomly generated using MATLAB software. The seed of the random numbers and associated MATLAB files could be provided on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] V. Klee, "Convexity," in *Proceedings of the Seventh Symposium in Pure Mathematics of American Mathematical Society*, Seattle, WA, USA, June 1961.

[2] N. D. Botkin and V. L. Turvova, "An algorithm for finding the Chebyshev center of a convex polyhedron," *Applied Mathematics & Optimization*, vol. 29, no. 2, 1995.

[3] A. Carlos Moretti, "A weighted projection centering method," *Computational and Applied Mathematics*, vol. 22, no. 1, pp. 19–36, 2003.

[4] D. P. Bertsekas, *Convex Optimization Theory*, Athena Scientific, Belmont, MA, USA, 1st edition, 2009.

[5] E. R. Barnes and A. C. Moretti, "Some results on centers of polytopes," *Optimization Methods and Software*, vol. 20, no. 1, pp. 9–24, 2005.

[6] J. Renegar, "A polynomial-time algorithm, based on Newton's method," *For Linear Programming*, vol. 40, no. 1–3, pp. 59–93, 1988.

[7] P. T. Boggs, P. D. Domich, J. R. Donaldson, and C. Witzgall, "Algorithmic enhancements to the method of centers for linear programming problems," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 159–171, 1989.

[8] R. Freund, "Projective transformation for interior point method and superlinear convergent algorithm for the w-center problem," *Mathematical Programming*, vol. 58, no. 1–3, pp. 385–414, 1993.

[9] P. Huard, "Resolution of mathematical programming with nonlinear constraints by the method of centers," in *Nonlinear Programming*, J. Abadie, Ed., pp. 207–219, North-Holland Publishing Company, Amsterdam, Netherlands, 1967.

[10] F. Jarre, G. Sonnevend, and J. Stoer, "An implementation of the method of analytic center," in *Lecture Notes in Control and Information Sciences*, pp. 297–307, Springer-Verlag, Berlin, Germany, 1998.

[11] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[12] D. G. Luenberger and Y. Yinyu, *Linear and Non Linear Programming*, Springer, New York, NY, USA, third edition, 2008.

[13] A. Carlos Moretti, *A Technique for Finding the Center of a Polytope*, Georgia Institute of Technology, Atlanta, GA, USA, 1992.

[14] F. Maire, "An algorithm for the exact computation of the centroid of higher dimensional polyhedra and its application to kernel machines," in *Proceedings of the Third IEEE International Conference on Data Mining*, Melbourne, FL, USA, November 2003.

[15] G. Sonnevend, "Applications of the notion of analytic center in approximation (estimation) problems," *Journal of Computational and Applied Mathematics*, vol. 28, pp. 349–358, 1989.

[16] K. G. Murty, *Ball Center of Special Polytopes*, Department of Industrial and Operation Engineering, University of Michigan, Ann Arbor, MI, USA, 2009.

