Interior Point Methods

Kripa Tharakan (20171159)

I. Introduction

Interior point methods are a class of algorithm that are used in solving both linear and nonlinear convex optimization problems that contain inequalities as constraints. The LP Interior-Point method relies on having a linear programming model with the objective function and all constraints being continuous and twice continuously differentiable. In general, a problem is assumed to be strictly feasible and will have a dual optimal that will satisfy Karush-Kuhn-Tucker (KKT) constraints described below. The problem is solved (assuming there IS a solution) either by iteratively solving for KKT conditions or to the original problem with equality instead of inequality constraints, and then applying Newton's method to these conditions.[1]

II. Background

Interior point methods came about from a desire for algorithms with better theoretical bases than the simplex method. While the two strategies are similar in a few ways, the interior point methods involve relatively expensive (in terms of computing) iterations that quickly close in on a solution, while the simplex method involves usually requires many more inexpensive iterations. From a geometric standpoint, interior point methods approach a solution from the interior or exterior of the feasible region but are never on the boundary.

An interior point method, was discovered by Soviet mathematician I. I. Dikin in 1967 and reinvented in the U.S. in the mid-1980s. In 1984, Narendra Karmarkar developed a method for linear programming called Karmarkar's algorithm, which runs in provably polynomial time and is also very efficient in practice. It enabled solutions of linear programming problems that were beyond the capabilities of the simplex method. Contrary to the simplex method, it reaches a best solution by traversing the interior of the feasible region. The method can be generalized to convex programming based on a self-concordant barrier function used to encode the convex set. Karmarkar's breakthrough revitalized the study of interior-point methods and barrier problems, showing that it was possible to create an algorithm for linear programming characterized by polynomial complexity and, moreover, that was competitive with the simplex method. The class of primal-dual path-following interior-point methods is considered the most successful.

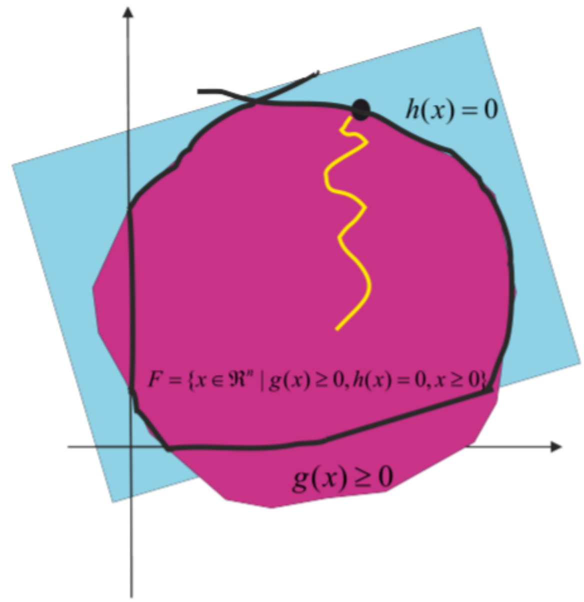

Fig. 1. Feasible set F

There are two important interior point algorithms: the barrier method and the primal-dual IP method. The primal-dual method is usually preferred due to its efficiency and accuracy. Major differences between the two methods are as follows. There is only one loop/iteration in primal-dual because there is no distinction between outer and inner iterations as with the barrier method. In primal-dual, the primal and dual iterates do not have to be feasible.

III. Details

Basics of the Interior Point Method

Consider a non-linear program min_x f(x) such that:

$$\begin{aligned} g_i(x) &>= 0, i = 1, 2, ..m_1; \\ h_j(x) &>= 0, j = 1, 2, ..m_2; \\ x &>= 0 \end{aligned} \tag{1}$$

where f, g_i, h_j: $\Re^n \rightarrow \Re$ are at least once differentiable functions and $x_{min}, x_{max} \in R^n$ are given vectors.

We define the feasible set of the NLP as:

$$F := x \in \Re^n | g_i(x) >= 0, i = 1, ..m_1; \quad h_j(x) = 0, j = 1, 2, ..m_2; \quad x >= 0 \tag{2}$$

The idea of the interior point method is to iteratively approach the optimal solution from the interior of the feasible set rather than from the boundary. You can refer to Figure 1 using this method. Therefore the requirements for the Interior Point Method is that:

- the interior of the feasible set should not be empty.
- almost all iterates should remain in (the interior of the) feasible set.

The interior of the feasible set of a non-linear program (NLP) is said to be non-empty:

- if there is $\bar{x} \in \Re^n$ such that

$$g_i(\bar{x}) > 0, i = 1, ..m_1; \quad h_j(\bar{x}) = 0, j = 1, 2, ..m_2; \quad \bar{x} > 0 \tag{3}$$

- if the Mangasarian-Fromovitz Constraint Qualification (MFCQ) is satisfied at a feasible point $\bar{x}$.

Mangasarian–Fromovitz Constraint Qualification

We say that a constraint qualification is Mangasarian–Fromovitz-like (MFCQ-like) at a point $y^0 \in C$ if by removing some constraints and transforming some inequalities into equalities one can obtain a set, which locally does not differ from C, but for which the MFCQ holds at the point y^0.[2]
Let $\bar{x}$ be a feasible point of NLP i.e.: $\bar{x} \in F$. Then MFCQ is said to be satisfied at $\bar{x}$ if there is a vector $d \in \Re^n$, $d \neq 0$, such that:

1)
$$d^T \nabla g_i(\bar{x}) > 0, i \in A(\bar{x})$$

2)
$$d^T \nabla h_1(\bar{x}) = 0, d^T \nabla h_2(\bar{x}) = ..., d^T \nabla h_{m_2}(\bar{x}) = 0$$

We examine the active constraints for MFCQ. An inequality constraint $g_i(x)$ is said to be active at $x \in F$ if $g_i(\bar{x}) = 0$. The set $A(\bar{x})$ = {i $\in$ $\{1, ..., m_1$ } | $g_i(\bar{x}) = 0$ } acts as the index set of active inequality constraints at $\bar{x}$.

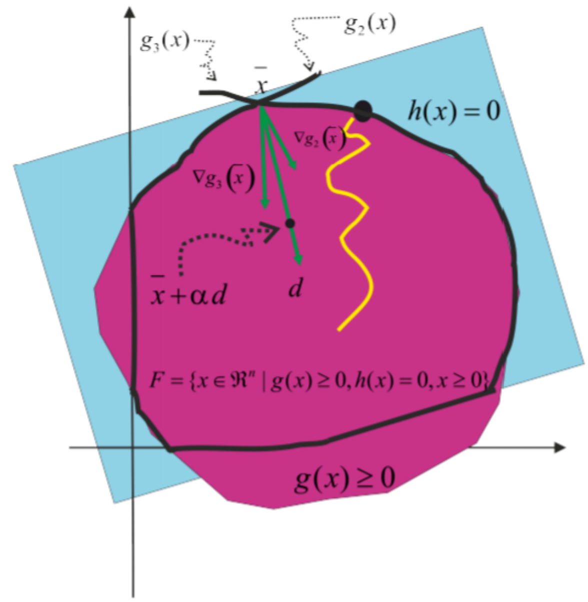

Fig. 2. A Mangasarian-Fromowitz Vector d. Here, d forms an acute angle ($< 90°$) with each $\nabla g_i(\bar{x}), i \in A(\bar{x})$.

The implications of the MFCQ are that there exists α such that:

- $\bar{x} + \alpha d > 0$
- $g(\bar{x} + \alpha d) \approx g(\bar{x}) + \alpha d^T \nabla g_i(\bar{x}) > 0, i = 1, ..., m1$
- $h_j(\bar{x} + \alpha d) \approx h_j(\bar{x}) + \alpha d^T \nabla h_j(\bar{x}) > 0, j = 1, ..., m2$

$\implies$ $x + \alpha d$ is in the interior of the feasible set F.
$\implies$ The interior of the feasible set is not empty.

Consider an example NLP where $min_x f(x) = x_1^2 - x_2^2$ such that: [3]

$$\begin{aligned} g_1(x) &= x_1^2 + x_2^2 + x_3^2 + 3 >= 0, \\ g_2(x) &= 2x_1 - 4x_2 + x_3^2 + 1 >= 0, \\ g_3(x) &= -5x_1 + 3x_2 + 2 >= 0, \\ x_1 &>= 0, x_2 >= 0, x_3 >= 0 \end{aligned} \tag{4}$$

The vector $\bar{x}^T$ = (1, 1, 1) is feasible to the NLP and $g_2(\bar{x}) = 0$ and $g_3(\bar{x}) = 0$, the active index set is $A(\bar{x})$ = {2, 3 }.

$$\nabla g_2(\bar{x}) = (2, -4, 2)$$
$$\nabla g_3(\bar{x}) = (-5, 3, 0)$$

For d^T = (-1, 0, 2), we have

$$d^T \nabla g_2(\bar{x}) > 0$$

and

$$d^T \nabla g_3(\bar{x}) > 0;$$

x = (1, 1, 1) + $\frac{1}{10=\alpha}(-1, 0, 2) > 0$
MFCQ guarantees that the interior of F is not empty. You can refer to Figure 2 for an example to illustrate this MFCQ.

Barrier Functions

A barrier function G(x) is a continuous function with the property that is approaches ∞ as one of $g_j(x)$ approaches 0 from negative values. We use barrier functions to force almost all iterates to remain in the interior of the feasible set F. Given a problem in the form of:-
Minimize $f_0(x)$ subject to $f_i(x) <= 0$, Ax = b
We must reformulate it to implicitly include the inequalities in the objective function. We can do this by creating a function that greatly increases the objective if a constraint is not met. Our conditions then changes to minimize $f_0(x) + \sum_i^m I - (f_i(x))$ such that $Ax = b$, where:

$$I - (x) = \begin{vmatrix} 0 & x \leq 0 \\ \infty & x > 0 \end{vmatrix}$$

This problem, however, is not continuous. A modification can be made by approximating $I - (x)$ as a logarithm log (-x), which approaches infinity when x approaches 0 as we want, and makes all functions twice differentiable. We then put the logarithm over a variable that sets a level of accuracy for the approximation we make. Here we will call that variable t. We define $\phi(x) = -\sum_i^m log - (f_i(x))$ which blows up if any of our constraints are violated.
Our LP problem now becomes :- minimize $f_0(x) + \frac{1}{t}\phi(x)$ such that $Ax = b$. This allows us to use Newton's method to follow what is called a Central Path, which is a series of points we iterate through that all satisfy the equality constraints Ax=b from the original problem, but give increasingly more optimized values for the objective function, with the inequality constraints $f_i(x)$ not necessarily equal to 0.

Algorithm

Given strictly feasible $x, t := t^0 > 0, \mu > 1, \in < 0$:
Repeat

1) Compute $x^*(t)$ by minimizing $tf_0 + \phi$ subject to Ax = b, starting at x.
2) Update $x := x^*(t)$
3) Quit if $\frac{m}{t} \leq \epsilon$, else
4) Increase $t := \mu t$

We look at a numerical example to better understand the algorithm:
Minimize f(x) = $x_1 - 2x_2$
such that 1 + $x_1 - (x_2)^2 \geq 0$ and $x_2 \geq 0$.
Use the Frisch barrier function to yield the unconstrained problem:

$$Minr(c, x) = x_1 - 2x_2 - clog(1 + x_1 - (x_2)^2) - clog(x_2)$$

For a specific parameter c, the first order necessary conditions for optimality are:

$$1 - \frac{c}{1 + x_1 - (x_2)^2} = 0$$

$$-2 + \frac{2cx_2}{1 + x_1 - (x_2)^2} - \frac{c}{x_2} = 0$$

Hence, $c = 1 + x_1 - (x_2)^2$
Substituting c in $-2 + \frac{2cx_2}{1+x_1-(x_2)^2} - \frac{c}{x_2} = 0$, we obtain $(x_2)^2 - x_2 - \frac{1}{2}c = 0$.
Hence, $x_2 = \frac{1 \pm \sqrt{1+2c}}{2}$ of which the positive is the only feasible solution.

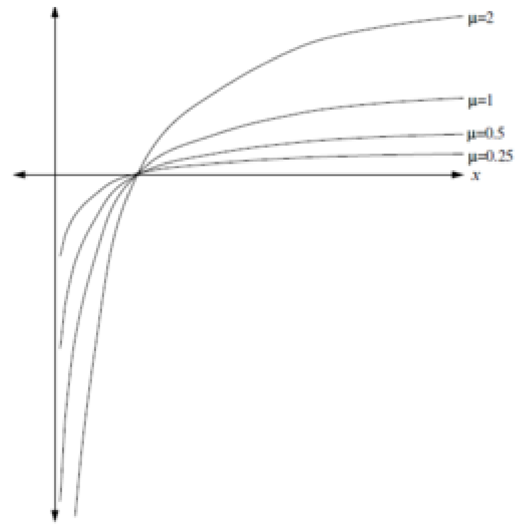

Fig. 3. Examples of logarithmic barrier functions.

A well-known barrier function is the logarithmic barrier function.

$$B(x, \mu) = f(x) - \mu(\sum_{i=1}^{m_1} log(g_i(x)) + \sum_{l=1}^{n} log(x_l)$$

where μ is known as the barrier parameter, and the logarithmic terms $log(g_i(x))$ and $log(x_l)$ are defined at points x for which $g_i(x) > 0$ and $x_l > 0$, l-1,..,n. Figure 3 shows some examples of logarithmic barrier functions.

So now, moving on back to the interior point method, instead of the problem NLP, we consider the parametric problem:

$$\begin{aligned} (NLP)_\mu \quad & min_x B(x, \mu) \\ s.t. \quad & h_j(x) = 0, j = 1, .., m_2 \end{aligned} \tag{5}$$

We need to find an optimal solution x_μ of $(NLP)_\mu$ for a fixed value of the barrier parameter μ.
Consider the Lagrange function of $(NLP)_\mu$:

$$L_\mu(x,\lambda) = f(x) - \mu(\sum_{i=1}^{m_1} log(g_i(x)) + \sum_{l=1}^{n} log(x_l)) - \sum_{j=1}^{m_2} \lambda_j h_j(x) \quad (6)$$

Karush-Kuhn-Tucker condition
In mathematical optimization, the Karush–Kuhn–Tucker (KKT) conditions, also known as the Kuhn–Tucker conditions, are first derivative tests (sometimes called first-order necessary conditions) for a solution in nonlinear programming to be optimal, provided that some regularity conditions are satisfied. Allowing inequality constraints, the KKT approach to nonlinear programming generalizes the method of Lagrange multipliers, which allows only equality constraints. Similar to the Lagrange approach, the constrained maximization (minimization) problem is rewritten as a Lagrange function whose optimal point is a saddle point, i.e. a global maximum (minimum) over the domain of the choice variables and a global minimum (maximum) over the multipliers, which is why the Karush–Kuhn–Tucker theorem is sometimes referred to as the saddle-point theorem.

The KKT is a necessary optimality function for a given μ, a vector x_μ is a minimum point of $(NLP)_\mu$ if there is a Lagrange parameter λ_μ such that, the pair (x_μ, λ_μ) satisfies:

$$\nabla_\lambda L_\mu(x,\lambda) = 0$$
$$\nabla_x L_\mu(x,\lambda) = 0$$

Thus we need to solve the system:

$$-h(x) = 0$$
$$\nabla f(x) - \mu(\sum_{i=1}^{m_1} \frac{1}{(g_i(x))}\nabla(g_i(x)) + \sum_{l=1}^{m_1}\frac{1}{x_l}e_l) + \sum_{j=1}^{m_2}\lambda_j h_j(x) = 0 \quad (7)$$

Commonly, this system is solved iteratively using the Newton Method.

According to the algorithm, for each given μ, the above algorithm can provide a minimal point x_μ of the problem $(NLP)_\mu$. A general strategy for choosing $\mu's$ is to choose a sequence μ_k of decreasing, sufficiently small non-negative barrier parameter values and to obtain associated sequence x_{μ_k} optimal solutions of $(NLP)_{\mu_k}$.
The properties of the optimal solution are that the optimal solutions x_μ lie in the interior of the feasible set of NLP. The solutions x_{μ_k} converge to a solution x^* of NLP; i.e.

$$\lim_{\mu \searrow 0^+} x_\mu = x^*$$

One of the drawbacks of the primal barrier interior is that as the values of μ get closer to 0 the matrix D can become ill-conditioned. Note that the expression $\frac{1}{g(x)}$ gets larger as g(x) gets smaller, usually near to the boundary of the feasible region. Also the matrix $\nabla g(x)[\nabla g(x)]^T$ is of rank 1, so is not invertible and has a large condition number. Instead, we introduce slack variables $s(s_1, s_2, ..., s_{m1})$ for inequality constraints so that:

$$g_i(x) - s_i = 0, s_i \geq 0, i = 1, ..., m_1$$

This leads us to the Primal-Dual Interior Point Method. Figure 4 for PD-IP iterations graph.

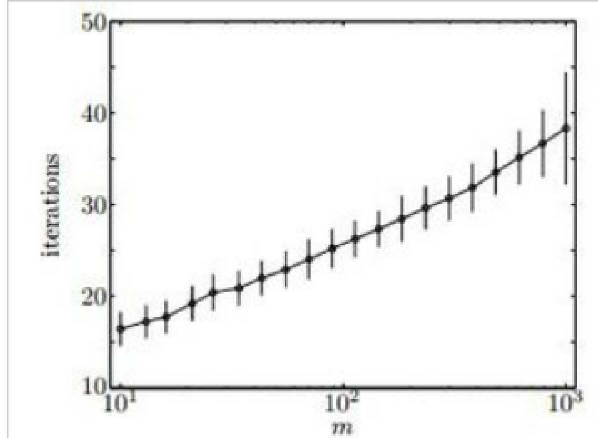

Fig. 4. Number of iterations required for Primal-Dual Interior Point method to solve randomly generated standard LPs of different dimensions, with n = 2m. Error bars show standard deviation.

Primal-Dual Interior Point Algorithm
The primal-dual interior-point method can easily be understood by using the simplest NLP problem; one with only inequality constraints. Consider the following:

Minimize f(x) such that $g_i(x) \geq 0$ with i = 1,..,m.

We now introduce slack variables to turn all inequalities into non-negativities:

Minimize f(x) such that $g(x) - s = 0$ with $s \geq 0$

The logarithmic barrier function is now introduced:

Minimize $f(x) - \mu\sum_{i=1}^{m} log(s_i)$ such that h(x) - s = 0.

Now incorporate the equality constraint(s) into the objective function using Lagrange multipliers:
Minimize $f(x) - \mu\sum_{i=1}^{m} log(s_i) - y^T(g(x) - s)$

Next, set all of the derivatives equal to 0:

$$\begin{aligned} \nabla f(x) - \nabla g(x)^T y = 0 \\ -\mu W^{-1}e + y = 0 \\ g(x) - s = 0 \end{aligned} \tag{8}$$

Rearranging the above equations to get:

$$\begin{aligned} \nabla f(x) - \nabla g(x)^T y = 0 \\ WYe = \mu e \\ g(x) - s = 0 \end{aligned} \tag{9}$$

Utilize Newton's Method to determine the search directions, $\Delta x, \Delta s, \Delta y$:

$$\begin{bmatrix} G(x,y) & 0 & -A(x)^T \\ 0 & Y & W \\ A(x) & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x) + A(x)^T y \\ \mu e - WYe \\ -g(x) + s \end{bmatrix} \tag{10}$$

where

$$G(x,y) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 g_i(x)$$

and $A(x) = \nabla g(x)$.
Using the 2nd equation, we solve for Δ's, the result of which is the reduced KKT system:

$$\begin{bmatrix} -G(x,y) & A^T(x) \\ A(x) & WY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \nabla f(x) - A^T(x)y \\ -g(x) + \mu Y^{-1}e \end{bmatrix} \tag{11}$$

From here, perform iterations:

$$\begin{aligned} x^{k+1} = x^k + \alpha^k \Delta x^k \\ s^{k+1} = s^k + \alpha^k \Delta s^k \\ y^{k+1} = y^k + \alpha^k \Delta y^k \end{aligned} \tag{12}$$

Algorithm
So we can write the algorithm as, given an initial point (x_0, λ_0, s_0) with $(x_0, s_0) > 0$:
Set $k \leftarrow 0$ and $\mu_0 = \frac{x_0^T s_0}{n}$
Repeat:

1) Choose $\sigma_k \in$ (0,1].

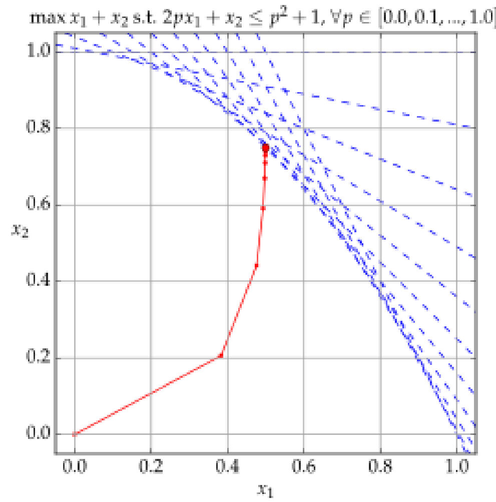

Fig. 5. Purple lines show constraints, red points show iterated solutions.

2) Solve the linear system with $\mu = \mu_k$ and $\sigma = \sigma_k$ to obtain $(\delta x_k, \delta\lambda_k, \delta s_k)$
3) Choose step-length $\alpha_k \in$ (0, 1]
4) And set

$$\begin{aligned} x^{k+1} = x^k + \alpha^k \Delta x^k \\ s^{k+1} = s^k + \alpha^k \Delta s^k \\ y^{k+1} = y^k + \alpha^k \Delta y^k \end{aligned} \tag{13}$$

Until: Some termination criteria is satisfied.

The algorithm can be terminated at iteration step k if the duality gap

$$\mu_k = \frac{x_k^T s_k}{n}$$

is sufficiently small, say $\mu_k < \epsilon$.

The efficiency of the primal-dual interior point methods is highly dependent on the algorithm used to solve this 2n + m linear system. The choice of an algorithm depends on the structure and properties of the coefficient matrix: $\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ X & 0 & S \end{bmatrix}$

IV. Summary

The Interior Point method approximates the constraints of a linear programming model as a set of boundaries surrounding a region. These approximations are used when the problem has constraints that are discontinuous or otherwise troublesome, but can me modified so that a linear solver can handle them. Once

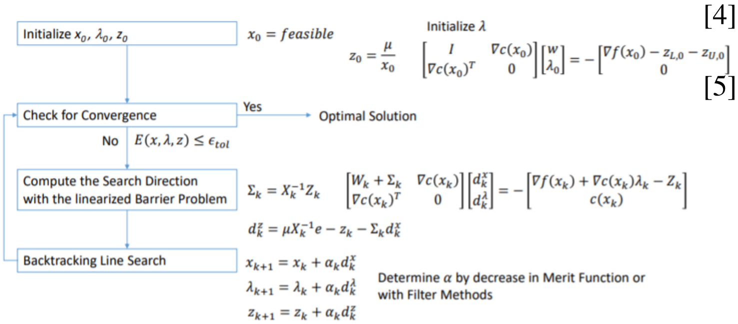

Fig. 6. Interior Point Method Overview.

the problem is formulated in the correct way, Newton's method is used to iteratively approach more and more optimal solutions within the feasible region. Figure 5 shows an example search for a solution using interior point methods. Two practical algorithms exist in IP method: barrier and primal-dual. Primal-dual method is a more promising way to solve larger problems with more efficiency and accuracy. As shown in the figure above, the number of iterations needed for the primal-dual method to solve a problem increases logarithmically with the number of variables, and standard error only increases rapidly when a very large number of dimensions exist. Interior point methods are best suited for very large-scale problems with many degrees of freedom (design variables). Refer figure 6 for a summary on how to solve using interior point methods.[4]

Appendix

Current efficient implementations are mostly based on a predictor-corrector technique (Mehrotra 1992), where the Cholesky decomposition of the normal equation or LDL^T factorization of the symmetric indefinite system augmented system is used to perform Newton's method (together with some heuristics to estimate the penalty parameter). All current interior point methods implementations rely heavily on very efficient code for factoring sparse symmetric matrices.[5]

References

[1] W. S. John Plaxco Alex Valdes, *Interior-point method for LP*. McCormick School of Engineering and Applied Science, Northwestern University, ChE 345 Spring 2014.

[2] L. Minchenko, *Note on Mangasarian–Fromovitz-Like Constraint Qualifications*. Journal of Optimization Theory and Applications, 2019.

[3] A. Geletu, *Introduction to Interior Point Methods*. Ilmenau University of Technology Department of Process Optimization.

[4] J. Hedengren, *Interior Point Methods*. ME575 – Optimization Methods, APMonitor.

[5] E. W. Weisstein, *Interior Point Method*. MathWorld–A Wolfram Web Resource.