Chapter 1

Exterior and Interior Point Methods

1.1. Penalty(Exterior) Function Methods

This write includes a survey of a class of algorithms called the sequential unconstrained minimization techniques (SUMT) to solve the optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_j(x) \leq 0,\ j = 1, 2, ..., l \\ & h_i(x) = 0,\ i = 1, 2, ..., m \qquad (\mathcal{P}_1) \\ & x \in \mathbb{X} \subseteq \mathbb{R}^n \end{array}$$

The goal of SUMT is to come up with an equivalent unconstrained optimization problem so that the large body of algorithms used to solve unconstrained optimization problems can be used. The class of algorithms which form the focus of this survey convert the constrained problem P_1 to a sequence of unconstrained problems parametrized by some parameter. In the limiting value of the parameter, as $\mu \to \infty$ or as $\mu \to 0^+$, it is expected that the sequence of optimal solutions to the corresponding unconstrained problems approaches the solution of the original constrained problem $\mathcal{P}_1$. The term SUMT was coined by Fiacco and McCormick in [3]. The SUMT can be divided into important categories, the exterior penalty function methods and the interior point methods. The exterior penalty function methods involve putting a high penalty for violating the constraints and the iterates proceed from the exterior of the feasible region converging to the optimal solution. On the other hand the interior point methods involve the use of barrier function which forces the iterates to stay within the feasible region.

This write up is a survey of some crucial aspects of the external penalty function methods as well as the barrier function method. The discussion on the penalty function method is fairly general in nature, that is, we will not consider special structure on the objective or constraint functions, like convexity etc. The discussion on the barrier method in a general setting but focus mainly on the applications of the barrier method to solve the convex optimization problem, in particular, the linear programming problem. The material was studied primarily from some classical references in the field including Bazaraa, Shetty and Sherali [2], Fiacco and McCormick [3], Convex Optimization by Boyd and Vanderberghe[1] and Primal Dual Interior Point Methods by Stephen Wright []. The structure of this write up is as follows.

Structure of the write up. We begin with a description of the penalty function method in brief and motivate the method with two examples. This is followed by a discussion of the construction of the penalty function based on certain desired properties of the objective function which . We state some lemmas and theorems that specify the properties of penalty functions and justify the use of the exterior penalty function method. This is followed by a discussion of some computational difficulties associated with the use of certain exterior penalty functions. Section 1.3 involves a survey of barrier and interior point methods. We begin with

a discussion of the barrier function method in a general setting primarily based on the comprehensive survey papers by M Wright et el [4]. Following the general discussion we look at the barrier method to solve a convex optimization problem with convex inequality and affine equality constraints. This is followed by a survey of linear programming problems and some salient features of the associated theory. We will then look at the application of the barrier method to the linear programming problem. We then motivate the primal dual interior point methods and look at a specific primal dual interior point method namely the short step algorithm. A detailed analysis of this method including a proof of polynomial complexity has been included. We will look at some computational experiments that confirm the theory.

1.2. Penalty Method

General Description. These methods involve constructing a new objective function which contains the function $f(x)$ and the constraint functions $g(x)$ and $h(x)$. For the problem $\mathcal{P}_1$, define an auxiliary function $Q(x, \mu)$ as

$$Q(x, \mu) = f(x) + q(\mu)P(x) \tag{1.1}$$

where $q(\mu)$ is a scalar valued function of the single variable μ with the properties that if $0 < \mu_1 < \mu_2$, then $0 < q(\mu_1) < q(\mu_2)$, and if μ_k is a monotonically increasing sequence of positive values where $\lim_{k\to\infty} \mu_k = +\infty$, then $\lim_{k\to\infty} q(\mu_k) = +\infty$. The function $P(x)$ is the penalty function and is defined as

$$P(x) = \sum_{j=1}^{m} \phi(g_j(x)) + \sum_{i=1}^{l} \psi(h_i(x)) \tag{1.2}$$

where the functions $\phi(x)$ and $\psi(x)$ are continuous and satisfy the following

$$\phi(y) = 0 \ \text{ if } y \leq 0 \ \text{ and } \ \phi(y) > 0 \ \text{ if } \ y > 0 \tag{1.3}$$

$$\psi(y) = 0 \ \text{ if } y = 0 \ \text{ and } \ \psi(y) > 0 \ \text{ if } \ y \neq 0 \tag{1.4}$$

Then the exterior penalty unconstrained minimization technique is as follows

1. Define a sequence μ_k such that $\mu_{k+1} < \mu_k$ for all k.
2. Compute an unconstrained local minimum for the auxiliary function $Q(x, \mu_k)$. So we have

$$x_k = argmin_x \ Q(x, \mu_k). \tag{1.5}$$

3. Presumably as $\mu_k \to \infty$, the sequence of unconstrained local minimum x_k will converge to the optimal solution of the constrained problem, $x_k \to \bar{x}$

Typically the functions p, ϕ and ψ have the following forms

$$p(y) = y$$
$$\phi(y) = [max\{0, y\}]^p$$
$$\psi(y) = |y|^p$$

where p is a positive integer. Thus the penalty function is of the form

$$P(x) = \sum_{j=1}^{m} [max\{0, g_j(x)\}]^p + \sum_{i=1}^{l} |h_i(x)|^p \tag{1.6}$$

It should be noted that the penalty function **??** is not differentiable. However for $p = 2$ we get the so called quadratic penalty function

$$f(x) + t \sum_{j=1}^{m} g_j^2(x) + t \sum_{i=1}^{l} h_i^2(x). \tag{1.7}$$

Let us now consider the following examples from [2], to heauristically describe the penalty function method using the quadratic penalty function.

Example 1. Consider the problem

$$\begin{aligned} &\text{minimize} \quad x \\ &\text{subject to} \quad -x+2 \leq 0, \end{aligned} \tag{1.8}$$

Since there is only one constraint which is simply $x \geq 2$, it is clear that the minimizer for problem is $\bar{x} = 2$. Employ the penalty function $P(x) = [max\{0, g(x)\}]^2$, then we have

$$P(x) = \begin{cases} 0, & x \geq 0 \\ (-x+2)^2, & x < 0 \end{cases}. \tag{1.9}$$

The auxillary function $Q(x, \mu)$ is then given as

$$Q(x, \mu) = x + \mu(-x+2)^2. \tag{1.10}$$

As a one dimensional unconstrained minimization problem, for a fixed value of μ the critical point for $Q(x, \mu)$ is given by the value of $x = \bar{x}$ for which $Q_x(x, \mu) = 0$. This gives, $1 - 2\mu(-\bar{x}+2) = 0$ and thus $\bar{x}(\mu) = 2 - \frac{1}{2\mu}$. Clearly, as $\mu \to \infty$, $\bar{x}(\mu) \to 2 = \bar{x}$.

Now consider the constrained problem of example 1.1.1 from the previous write on constrained optimization and KKT conditions.

Example 2 Consider the problem

$$\begin{aligned} &\text{minimize} \quad f(x) = x_1^2 + x_2^2 \\ &\text{subject to} \quad h(x) = x_1 + x_2 - 1 = 0 \end{aligned} \tag{1.11}$$

The solution to the problem is $\bar{x} = (\frac{1}{2}, \frac{1}{2})$. Now consider the penalty problem with auxiliary function $Q(x, \mu) = x_1^2 + x_2^2 + \mu(x_1 + x_1 - 1)$. For a fixed μ, equate the gradient of the auxiliary function to zero,

$$\begin{aligned} x_1 + \mu(x_1 + x_2 - 1) = 0 \\ x_2 + \mu(x_1 + x_2 - 1) = 0. \end{aligned} \tag{1.12}$$

Solving them simultaneously we get $x_1(\mu) = x_2(\mu) = \frac{\mu}{2\mu+1}$. Clearly as $\mu \to \infty$, $x_1(t) \to \frac{1}{2}$ and $x_2(t) \to \frac{1}{2}$ which is the optimal solution to the given constrained problem.

We now state some important theorems and results which ascertain the use of penalty function method to solve constrained optimization problem. Lemma states some properties of auxillary and penalty functions as $\mu \to \infty$. Theorem states the conditions under which the sequence of unconstrained minimizers converges to the solution of the original constrained optimization problem. These results and proof have been taken from Bazaraa, Shetty and Sherali [2].

Lemma 1.1. If $x_k = argmin_x \; Q(x, t_k)$ and $t_{k+1} > t_k$, then

1. $Q(x_k, t_k) \leq Q(x_{k+1}, t_{k+1})$
2. $P(x_k) \geq P(x_{k+1})$
3. $f(x_k) \leq f(x_{k+1})$

Proof. (1) By definition $Q(x_k, t_k) = f(x_k) + q(t_k)P(x_k)$. As $x_k = argmin_x\ Q(x, t_k)$, so for fixed t_k we have

$$Q(x_k, t_k) = f(x_k) + q(t_k)P(x_k) < f(x_{k+1}) + q(t_k)P(x_{k+1})$$

As $t_k < t_{k+1}$, we have $q(t_k) < q(t_{k+1})$ which gives

$$Q(x_k, t_k) = f(x_k) + q(t_k)P(x_k) < f(x_{k+1}) + q(t_k)P(x_{k+1}) < f(x_{k+1}) + q(t_{k+1})P(x_{k+1}) = Q(x_{k+1}, t_{k+1})$$

Thus $Q(x_k, t_k) < Q(x_{k+1}, t_{k+1})$.

(2) As $x_k = argmin_x\ Q(x, t_k)$ and $x_{k+1} = argmin_x\ Q(x, t_{k+1})$, we have for fixed t_k and t_{k+1}

$$f(x_k) + q(t_k)P(x_k) \leq f(x_{k+1}) + q(t_k)P(x_{k+1}) \tag{1.13}$$

$$f(x_{k+1}) + q(t_{k+1})P(x_{k+1}) \leq f(x_k) + q(t_{k+1})P(x_k) \tag{1.14}$$

Adding (1.13) and (1.14) we get

$$q(t_k)P(x_k) + q(t_{k+1})P(x_{k+1}) \leq q(t_k)P(x_{k+1}) + q(t_{k+1})P(x_k) \tag{1.15}$$

Since $q(t_k) - q(t_{k+1}) < 0$ we have $P(x_k) \geq P(x_{k+1})$.

(3) Rearrange (1.13) to obtain

$$f(x_k) - f(x_{k+1}) \leq q(t_k)(P(x_{k+1}) - P(x_k)) \tag{1.16}$$

But $(P(x_{k+1}) - P(x_k)) \leq 0$ from (2) and result follows. □

Theorem 1.2.1. For the constrained optimization problem $\mathcal{P}_1$ where $f, g_1, \ldots, g_l$ and $h_1, \ldots, h_m$ are continuous functions on and X is non empty set in $\mathbb{R}^n$. Suppose that the problem has a feasible solution and let $P(x)$ be continuous as given by **??**. Now suppose that for each μ there exists $x_\mu \in X$ as the solution to the unconstrained problem to minimize $f(x) + \mu P(x)$ subject to $x \in X$, and that x_μ is contained in a compact subset of X. Then

$$inf\{f(x) : g(x) \leq 0,\ h(x) = 0,\ x \in X\} = sup_{t \geq 0}\, \theta(\mu) = \lim_{t \to \infty} \theta(\mu) \tag{1.17}$$

where $\theta(\mu) = inf\{f(x) + \mu P(x) :\ x \in X\} = f(x_\mu) + \mu P(x_\mu)$. Furthermore the limit $\bar{x}$ of any convergent sub-sequence of x_μ is an optimal solution to the original problem and $\mu P(x_\mu) \to 0$ as $\mu \to \infty$.

Proof. From lemma **??** we have that $\theta(\mu)$ is monotone so we have $sup_{\mu \geq 0}\theta(\mu) = \lim_{\mu \to \infty} \theta(\mu)$. The first step is now to show that $P(x_\mu) \to 0$ as $\mu \to \infty$. Let y be a feasible point and $\epsilon > 0$. Let x_1 be an optimal solution to the problem to minimize $f(x) + \mu P(x)$ subject to $x \in X$ for $\mu = 1$. If $\mu \geq (\frac{1}{\epsilon})|f(y) - f(x_1)| + 2$. Note that $\mu > 1$ and since $f(x_\mu)$ is a non-decreasing function of μ from the previous lemma, we have $f(x_\mu) \geq f(x_1)$. The strategy to show $P(x_\mu) \to 0$ is to show that $P(x_\mu) \leq \epsilon$ for every $x_\mu > 0$. Suppose not, then $P(x_\mu) > \epsilon$. From lemma () we have

$$\begin{aligned} inf\{f(x) : g(x) \leq 0,\ h(x) = 0,\ x \in X\} \geq 0\theta(\mu) = f(x_\mu) + \mu P(x_\mu) \geq f(x_1) + \mu P(x_\mu) \\ >\ f(x_1) + |f(y) - f(x_1)| + 2\epsilon > f(y). \end{aligned} \tag{1.18}$$

This inequality is not possible as y is a feasible point and hence $f(y)$ cannot be less than the infimum (greatest lower bound) of $f(x) : x \in$ feasible set. Thus the assumption is incorrect and we have $P(x_\mu) \leq \epsilon$ for all ϵ such that $\mu \geq (\frac{1}{\epsilon})|f(y) - f(x_1)| + 2$. Thus . Now let x_{μ_k} be any convergent sub-sequence of x_μ and let $\bar{x}$ be its limit. It should be noted that such a convergent sub-sequence exists as we have assumed x_μ lies in a compact subset of X. We then have

$$sup_{\mu \geq 0}\theta(\mu) \geq \theta(\mu_k) = f(x_{\mu_k}) + \mu_k P(x_{\mu_k}) \geq f(x_{\mu_k}). \tag{1.19}$$

Since $x_{\mu_k} \to \bar{x}$ and since f is continuous we have from the above inequality

$$sup_{\mu \geq 0} \theta(\mu) \geq f(\bar{x}) \tag{1.20}$$

. Since $P(x_{\mu_k}) \to 0$ as $\mu \to \infty$, $P(\bar{x}) = 0$; that is, $\bar{x}$ is a feasible solution to the original problem. □

KKT Lagrange multipliers at Optimality. This is a discussion on how the solutions to the sequence of unconstrained minimization problems can be used to recover the KKT multipliers at associated with the constraints at optimality. Consider the problem $\mathcal{P}_1$ with $X = \mathbb{R}^n$. Also suppose the penalty function $P(x)$ is as given in **??**. However further assumptions are made on ϕ and ψ as being in C^1 and $\phi' \geq 0$ for all $y > 0$ and $\phi' = 0$ for $y \leq 0$. Assuming the conditions of theorem **??** to be true, since the gradient of objective function must vanish at the optimal x_μ, we have at x_μ,

$$\nabla f(x_\mu) + \sum_{j=1}^{m} \mu \phi'[g_j(x_\mu)] \nabla g_j(x_\mu) + \sum_{i=1}^{l} \mu \nabla \psi'[h_i(x_\mu)] \nabla h_i(x_\mu) = 0 \tag{1.21}$$

Let the limit point of x_μ be $\bar{x}$. Let $\mathcal{A}$ be the set of active constraints at $\bar{x}$. Since $g_j(\bar{x}) < 0$ for $j \in \mathcal{A}$, we have by the continuity of g, $g_j(x_\mu) < 0$ for sufficiently large μ and thus $\mu \phi' g_j(x_\mu) = 0$. Thus for sufficiently large μ, **??** can be written as

$$\nabla f(x_\mu) + \sum_{j \in \mathcal{A}} (u_\mu)_j \nabla g_j(x_\mu) + \sum_{i \in \mathcal{E}} (v_\mu)_i \nabla h_i(x_\mu) = 0 \tag{1.22}$$

where u_μ and v_μ are vectors with components

$$(u_\mu)_j = \phi'[g_j(x_\mu)] \geq 0 \; j \in \mathcal{A} \tag{1.23}$$

$$\text{and} (v_\mu)_i = \psi'[h_i(x_\mu)] \geq 0 \; i \in \mathcal{E} \tag{1.24}$$

Assume that LICQ holds at $\bar{x}$, then we know there exist unique KKT multipliers $\bar{u}_j \geq 0$, $j \in \mathcal{A}$ and $\bar{v}_i = 0$ $i = 1 \ldots l$ such that

$$\nabla f(\bar{x}) + \sum_{j \in \mathcal{A}} \bar{u}_j \nabla g_j(\bar{x}) + \sum_{i=1}^{l} \bar{v}_i \nabla h_i(\bar{x}) = 0 \tag{1.25}$$

Since g, h, ϕ and ψ are all continuously differentiable and since $x_\mu \to \bar{x}$ where LICQ is satisfied, the KKT multipliers are unique and hence $(u_\mu)_j \to \bar{u}_j$, $j \in \mathcal{A}$ and $(v_\mu)_i \to \bar{v}_i$, $i = 1 \ldots l$. Hence for sufficiently large values of μ, the multipliers in **??** can be used to estimate KKT multipliers at optimality. For the quadratic penalty function $P(x) = \sum_{1}^{m} [max\{0, g_j(x)\}]^2 + \sum_{i=1}^{l} h_i^2(x)$, then we have $\phi(y) = [max\{0, y\}]^2$, $\phi'(y) = 2max\{0, y\}$, $\psi(y) = y^2$ and $\psi'(y) = 2y$, we obtain the following expressions for estimates of KKT multiplers at optimality,

$$(u_\mu)_j = 2\mu max\{0, g_j(x_\mu)\} \tag{1.26}$$

$$(v_\mu)_j = 2\mu h_i(x_\mu) \tag{1.27}$$

It should be observed that if for some , then for sufficiently large values of μ. This is due to continuity of ϕ' and g and as a result $g_j(x_\mu) > 0$. This implies that throughout the iterations the constraint is violated and in the limit $g_j(\bar{x}) = 0$. Hence if and then all the constraints are violated along the trajectory leading to $\bar{x}$ This motivates the name exterior penalty function method.

Computational difficulties associated with Penalty function method. Following is a discussion of some computational challenges posed by the use of exterior penalty function method.

1. The solution to the constrained optimization problem is obtained by solving the unconstrained penalty problem for increasing values of parameter t. However solving an unconstrained problem for large values

of parameter creates difficulties related to ill- conditioning of the problem. Consider the problem with only equality constraints. For this problem use the penalty function $f(x) + \sum_{i=1}^{l} \psi(h(x))$ where ψ is assumed to be twice differentiable. We have then the gradient and hessian of the penalty function as

$$\nabla f(x) + \sum_{i=1}^{l} \mu \psi^{'}[h_i(x)] \nabla h_i(x) \tag{1.28}$$

$$\left[\nabla^2 f(x) + \sum_{i=1}^{l} \mu \psi^{'}[h_i(x)] \nabla^2 h_i(x)\right] + \mu \sum_{i=1}^{l} \psi^{''}[h_i(x)] \nabla h_i(x) (\nabla h_i(x))^T \tag{1.29}$$

It was shown in []that as $t \to \infty$, and the hessian $\nabla^2 Q$ has l eigen values that approach ∞ while $n - l$ eigen values approach a finite limit. This eigen structure causes the hessian matrix to be severely ill conditioned.

2. The ill conditioning of the hessian matrix causes convergence of unconstrained algorithms used to solve these. is the problem of pre mature termination of the algorithm to a non optimal solution. As the value of μ is increased, more emphasis is placed on feasibility and . Suppose at a given stage in the process, a feasible point is reached with . Now in the presence of equality constraints a movement along any direction results in infeasible points or feasible points having large objective function values. Feasible points with larger objective function values can be easily seen, as $P(x) = P(x + \lambda d) = 0$, it is possible for a non optimal point x to have larger objective function value. If $x + \lambda d$ is not feasible, then $P(x + \lambda d) > 0$ and since μ is very large any reduction in the value of $f(x + \lambda d)$ over $f(x)$ is compensated by an increase in the term $\mu P(x + \lambda d)$. Thus improvement (reduction) in objective function value is possible only when λ, the step length is very small so that the term $P(x + \lambda d)$ is small in spite of μ being very large. These small step lengths can cause pre mature termination of the algorithm.

As an example, consider the optimization problem . The penalized objective function is given by $Q(x) =$. The hessian of $Q(x)$ is given by

$$\begin{bmatrix} 2(1+\mu) & 2\mu \\ 2\mu & 2(1+\mu) \end{bmatrix} \tag{1.30}$$

The eigen values of the hessian can be calculated by solving the equations $det(\nabla^2 Q(x) - \lambda I) = 0$. The eigen values are $\lambda_1 = 2$ and $\lambda_2 = 2(1 + 2\mu)$. Clearly as $\mu \to \infty$, the eigen value $\lambda_2 \to \infty$.

The penalty function can be summarized as follows.

Penalty Function Method
Given an initial point x_0, penalty parameter μ_0 and scaler $\beta > 1$
Set $k = 1$
for k=1,2,3,...
1. With x_k as initial point solve minimize $f(x) + \mu_k \alpha(x)$, $x \in X$
2. x_{k+1} optimal solution in (1)
3. If $\mu_k \alpha(x_{k+1}) < \epsilon$ then stop
4. $\mu_{k+1} = \beta \mu_k$
5. $k = k + 1$, Back to step 1
end

We now look at a concrete computational example. This is example 9.2.4 from Bazaraa, Shetty and Sherali[]. Consider the equality constrained problem

$$\begin{aligned} \text{minimize} \quad & (x_1 - 2)^4 + (x_1 - 2x_2)^2 \\ \text{subject to} \quad & x_1^2 - x_2 = 0, \end{aligned} \tag{1.31}$$

The optimal solution to the problem is with . At every iteration k, for a given penalty parameter μ_k, we minimize the quadratic penalty function as given below

$$\text{minimize } (x_1 - 2)^4 + (x_1 - 2x_2)^2 + \mu_k(x_1^2 - x_2)^2 \tag{1.32}$$

The unconstrained minimization problem 1.32 was solved using modified Newton's method with line search (Armijo-Goldstein's condition). Since the goal is to create a set of notes that are largely self contained the Newton's method to solve an unconstrained optimization problem has been summarized in the appendix. The code was written in Matlab and all the subroutines have been uploaded on github.

The results obtained from the code have been summarized in the table below. Starting with a value of $\mu_0 = 0.1$ and $x^0 = (2.0, 1.0)$, we solve the unconstrained problem 1.32 for values of μ_k increasing by a factor of $\beta = 10$ at every iteration. The previous optimal solution is fed as the initial point for the new unconstrained problem with an increased value of μ_k. The second column shows the progress of iterations from the starting point $x^0 = (2.0, 1.0)$ to the optimal solution $(0.94611, 0.8934)$. The iterations are characterized by a consistent reduction in the value of the penalty function $h^2(x)$ which is shown in column 3. The progress of $\mu\alpha(x)$ is shown in column 4. The contours for this problem, the constraint set and the

μ_k	optimal x_μ	$\alpha(x) = h^2(x)$	$\mu_k \alpha_x$
0.1	(1.45387,0.76076)	1.8306	0.18306
1	(1.16872, 0.74067)	0.39093	0.3908
10	(0.99062, 0.84246)	0.01928	0.19282
100	(0.95076, 0.88747)	0.000271	0.0271
1000	(0.94611, 0.8934)	0.0000028	0.00292

Table 1.1. **Solution to problem 1.31**

progression of iterates are shown in the figure.

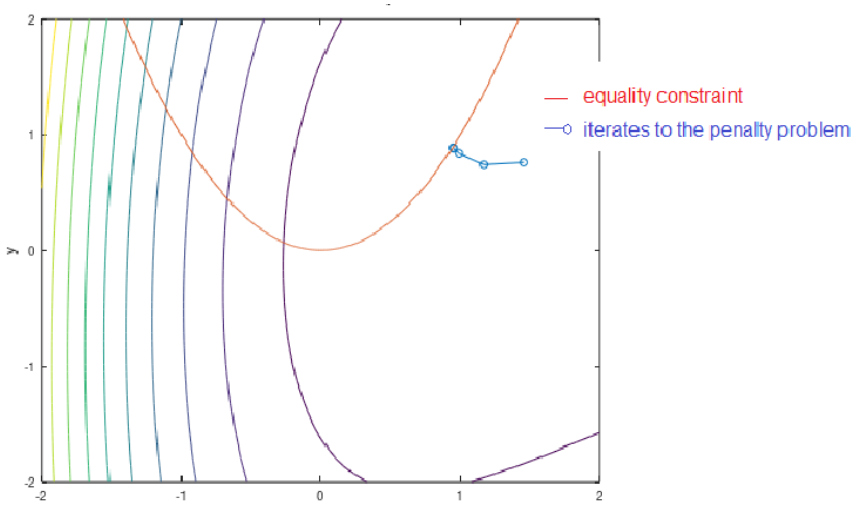


Figure 1.1. **Progression of iterates for the problem 1.31**

1.3. Interior Point Methods

1.3.1. Introduction.

The second important class of methods that are categorized under the SUMT involve the use of barrier functions as opposed to the use of penalty functions to consume the constraints into the objective function. As mentioned earlier, SUMT is a way to convert a constrained minimization problem to a sequence of unconstrained minimization problems. The solutions to these unconstrained problems are expected to converge to the minimizer of the original constrained problem. The penalty function methods can be interpreted as minimizing a function f and a positive penalty if evaluated at any infeasible point. A different way of achieving the goal of converting the constrained problem to a family of unconstrained problems is to minimize a function combining f and a positively weighted barrier that prevents the iterates from leaving the feasible region. This forms the basis for the so called barrier function methods also called as the interior point methods.

This section is a short survey of some important notions related to the barrier function methods or the interior point methods. The discussion in this section and subsequent sections is primarily based on Margaret Wright [4], Steve Wright [5] and Boyd Vanderberghe [6]. The field of interior point methods is vast and this section can be termed as being a very basic and a first introduction to these ideas. In this section we shall first discuss the barrier methods in general, look at an example which motivates the general ideas and state a convergence theorem without proof. We will then focus primarily on the interior point methods applied to the linear programming problem. Linear programming problem is a very special convex optimization problem with linear objective as well as constraint functions. A major thrust to the field of interior point methods was due to the successful development and application of these methods to linear programming problems.

1.3.2. Barrier Methods - A General Discussion.

In this sub section we consider the following optimization problem with only inequality constraints. The primary source of the current discussion is an exhaustive survey on interior point methods by M. Wright [4].

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && g(x) \leq 0 \end{aligned}$$

where $g : \mathbb{R}^n \to \mathbb{R}^m$ and is assumed to be twice continuously differentiable. The feasible region is given by $C = \{x \in \mathbb{R}^n;\ g(x) \leq 0\}$. The barrier methods are based on the following idea. That is, a way to introduce the influence of constraints in the objective function is by defining a composite function that is defined only at feasible points. One extreme way of doing this is by defining it as $f(x)$ when x is feasible and $+\infty$ when x is infeasible. This can also be viewed as the minimization of $f(x) + I_C(x)$ where $I_C(x)$ is the indicator function as defined in the write up on convex analysis, that is

$$\begin{aligned} I_C(x) &= \begin{cases} 0, & x \in C \\ +\infty, & x \notin C \end{cases} \\ \Rightarrow f(x) + I_C(x) &= \begin{cases} f(x), & x \in C \\ +\infty, & x \notin C \end{cases} \end{aligned} \tag{1.33}$$

However, it is obvious that $f(x) + I_C(x)$ is discontinuous and is not of much use in practice. A way to get around this problem is to define an interior barrier function that retains certain desirable properties of the constraint functions such as smoothness while at the same time not not allowing to leave the interior of the feasible region. By appropriately combining $f(x)$ with $g(x)$, as well as reducing the effect of the constraints, it is possible to construct a composite function whose unconstrained minimizers will converge to a local constrained minimizer of the original problem.

This suggests the following desireable properties for $I_C(x)$,

1. $I(x)$ depends only on the constraint functions.

2. $I(x)$ preserves continuity properties of $g(x)$ at all points in $int(C)$.

3. For any sequence of points in $int(C)$ converging to a point on the boundary of the feasible region, $I(x) \to \infty$.

Examples One of the earliest known use of an interior functions $I(x)$ is the inverse interior function by Carroll []

$$I_{inv} = -\sum_{j=1}^{l} \frac{1}{g_j(x)} \tag{1.34}$$

Clearly, I_{inv} satisfies the three properties.

The most well known and practically used interior function is the Frisch's logarithmic barrier function. It is given as

$$I_{log}(x) = -\sum_{j=1}^{l} log(-g_j(x)). \tag{1.35}$$

Clearly, if g_j is assumed to be twice continuously differentiable, then $I_{log}(x)$ is twice continuously differentiable. Further as $g_j \to 0^-$, $I_{log}(x) \to +\infty$.

The Barrier Function $B(x, \mu)$. The use of logarithmic interior function gives rise to the logarithmic barrier function given as

$$B(x, \mu) = f(x) - \mu \sum_{j=1}^{l} log(-g_j(x)) \tag{1.36}$$

where μ is the barrier parameter. An important feature of barrier function $B(x, \mu)$ is that it retains the smoothness of $f(x)$ and $g(x)$ as long as $g(x) < 0$. For very small values of $\mu > 0$, $B(x, \mu)$ acts like $f(x)$ except close to points where any constraint is zero. The idea then is that minimizing $B(x, \mu)$ for a sequence of positive values converging to zero will cause the unconstrained minimizers of $B(x, \mu)$ to converge to a local constrained minimizer of the original problem.

Observation. Since $B(x, \mu)$ is twice continuously differentiable, at the local minimizer x_μ of we have

$$\nabla B(x_\mu, \mu) = \nabla f(x) + \sum_{j=1}^{l} \frac{\mu}{g_j(x_\mu)} \nabla g_j(x_\mu) = 0 \tag{1.37}$$

Now let $u_\mu = \frac{\mu.}{g(x_\mu)}$ where u_μ and $g(x_\mu)$ are both vectors and the notation used is inspired from MATLAB to denote component wise multiplication. Also, $u_\mu > 0$ as it can be shown that $g(x_\mu) > 0$ [4]. Thus the unconstrained optimal x_μ must satisfy

$$\begin{aligned} \nabla f(x) + \sum_{j=1}^{l} \frac{\mu}{g_j(x_\mu)} \nabla g_j(x_\mu) = 0 \quad (a) \\ g(x_\mu).u_\mu = \mu \text{ or } g_j(x_\mu).(u_\mu)_j = \mu, \ j = 1 \dots l \quad (b) \end{aligned} \tag{1.38}$$

Note that 1.38 (a) and $u_\mu \geq 0$ are precisely the stationarity and non negativity conditions that hold at a KKT point. Equation 1.38 (b) is called as the perturbed complimentarity and is analogous as $\mu \to 0$ to the

complimentary slackness condition that holds at a KKT point.

We now state a general convergence theorem about the barrier function method. This is theorem 3.12 from M. Wright [4] and is stated without proof. The general convergence theory of SUMT using logarithmic barrier functions is extremely technical and involves rather abstract notions form topology and analysis. The theorem summarizes the conditions under which the sequence of barrier minimizers $x(\mu)$ not only converges to the local minimizer $\bar{x}$ of the original constrained problem but also defines a differentiable path to $\bar{x}$. This path in literature is called as the central path or barrier trajectory. The central path shall be studied in more detail while discussing interior point methods in the context of linear programming problems. The theorem involves certain terminology which has not yet been discussed and will be discussed briefly after the theorem is stated.

Theorem 1.3.1. Given the optimization problem $\mathcal{P}_1$. Let C denote the feasible region and assume that the set $strict(C)$ is of strictly feasible points is non empty. Let $\bar{x}$ be a local constrained minimizer with $\bar{g}$ denoting $g(\bar{x})$, $\bar{J}$ denoting $J(\bar{x})$, the Jacobian of $g : \mathbb{R}^n \to \mathbb{R}^m$ and let $\bar{\mathcal{A}}$ denote $\mathcal{A}(\bar{x})$, the set of active constraint indices. Assume that the following sufficient optimality conditions hold at $\bar{x}$.

(a). $\bar{x}$ is a KKT point, that is, there exists a nonempty set M_u of Lagrange multipliers satisfying

$$\begin{aligned} M_u &= \{u : \ \nabla f(\bar{x}) + \sum u_j g_j(\bar{x}) = 0, \ u \geq 0 \text{ and } u_j g_j(\bar{x}) = 0 \, \forall \, j\} \\ &= \{u : \ \nabla f(\bar{x}) + \bar{J}^T u = 0, \ u \geq 0 \text{ and } g(\bar{x}).u = 0\} \end{aligned} \tag{1.39}$$

(b). The Mangasarian-Fromovitz Constraint Qualification(MFCQ) holds at $\bar{x}$, that is, there exists d such that $\bar{J}^T d < 0$

(c). There exists $\omega > 0$ such that for all and all non zero and all non zero satisfying and where is the hessian of the Lagrangian.

Assume that a logarithmic barrier method is applied in which μ_k converges monotonically to 0 as $k \to \infty$. Then,

1. there is at least one subsequence of unconstrained minimizers of the barrier functions $B(x, \mu)$ converging to $\bar{x}$.

2. Let $\{x^k\}$ denote such a convergent subsequence, with the notation that g_j^k denotes $g_j(x^k)$ and so on. Then the sequence of barrier multipliers u^k whose $j - th$ component is $\frac{\mu_k}{g_j^k}$, is bounded.

3. $\lim_{k \to \infty} u^k = \bar{u} \in M_u$

If in addition strict complimentarity holds at $\bar{x}$, that is, there is a vector $u \in M_u$ such that $u_j > 0$ for all $j \in \mathcal{A}$, then

4. $\bar{\lambda}_{\mathcal{A}} > 0$

5. for sufficiently large k, the Hessian matrix $\nabla^2 B(x^k, \mu_k)$ is positive definite.

6. A unique, continuously differentiable function $x(\mu)$ of unconstrained minimizers of $B(x, \mu)$ exists for postive μ in a neighborhood of $\mu = 0$.

7. $\lim_{\mu \to 0^+} x(\mu) = \bar{x}$

We now discuss some of the terms used in this theorem like $strict(C)$, strict complimentarity and conditions a, b and c which have not yet been discussed. We shall briefly discuss the significance of these notions in the theorem.

Strictly Feasible Set $(strict(C))$**.** We first define the strictly feasible set $strict(C)$. It should be noted that properties 2 and 3 listed under the three desirable properties of the interior function depend on the interior and boundary of the feasible set C. However there are some subtle issues that need to be taken into consideration. To this extent we first define the set of strictly feasible points $strict(C)$.

Definition 1.3.1. Strictly feasible points. The subset of points in C for which the constraint functions are strictly positive is denoted by $strict(C)$ and defined as

$$strict(C) = \{x : \ g_j(x) < 0, \text{ for all } j = 1 \dots l\} \tag{1.40}$$

The important issue to be noted here is that the interior and boundary of the feasible region are purely topological notions and may not be captured by the algebraic representation of the feasible set C. For example, it may happen that a point may lie in the interior of the feasible region but one of the constraints may prove to be active that is equal to zero at that point. This would mean that even in the interior of the feasible region the interior function would shoot up to infinity. And it is such inconsistencies known as topological inconsistencies that must be taken care of.

The topological interior of the feasible set C may not same as the set of strictly feasible points $strict(C)$ and this can be seen by considering the following example with two constraints,

$$g_1(x) = x^2 \geq 0 \text{ and } g_2(x) = x + \gamma \geq 0. \tag{1.41}$$

where $\gamma > 0$, with the associated feasible set

$$C = \{x : \ x^2 \geq 0 \text{ and } x \geq -\gamma\} \tag{1.42}$$

Since all reals satisfy the constraint $x^2 \geq 0$, the first constraint is essentially redundant. Moreover since $\gamma > 0$, $x = 0$ is an interior point of C while the first constraint at this point is equal to zero. Thus $int(C) = \{x : x > -\gamma\}$, whereas origin is excluded from $strict(C) = \{x : \ (\gamma, 0) \cup (0, \infty)\}$ and thus $strict(C) \neq int(C)$.

Definition 1.3.2. Topological inconsistency. A constraint $g_j(x) \leq 0$ is said to be topologically inconsistent at $\tilde{x}$, if $\tilde{x} \in int(C)$ but $g_j(\tilde{x}) = 0$. The constraint C is topologically consistent if for all $x \in C$, $g_j(x) = 0$ only if $x \notin int(C)$. A further discussion in [4] gives criteria for topologically inconsistent constraints to be locally redundant.

Due to such inconsistencies, the desirable properties 2 and 3 must be restated in terms of $strict(C)$ as opposed to $int(C)$.

Sufficiency Conditions a, b and c. We now discuss the importance of condition c in the theorem. This condition together with conditions a and b is a second order sufficient condition for a local minimizer satisfying the first order KKT connditions to be an isolated minimizer. Local minimizers can be classified as isolated local minimizers and strict local minimizers defined as follows [1].

Definition 1.3.3. Strict local minimum. Let C be a feasible set, a point $\bar{x}$ is a strict local minimum if $\bar{x} \in C$ and there is a neighborhood $\mathcal{N}$ of $\bar{x}$ such that $f(x) > f(\bar{x})$ for all $x \in \mathcal{N} \cap C$ with $x \neq \bar{x}$.

Definition 1.3.4. Isolated local minimizer. A point $\bar{x}$ is an isolated local minimum if $\bar{x} \in C$ and there is a neighborhood $\mathcal{N}$ of $\bar{x}$ such that $\bar{x}$ is the only local solution in $\mathcal{N} \cap C$. A simple example is the local minimizer for a quadratic function.

An isolated minimizer has the property that it lies in a neighbourhood containing no other minimizers. On the other hand a strict local minimizer is a point whose function value is strictly greater than at all points in a neighborhood. This difference in the two definitions may seem a bit counterintuitive. However the definition of strict local minimizer allows for the existence of other local minimizers in a neighbourhood containing the point termed as strict local minimizer. It may so happen that in a neighborhood of the strict local minimizer there may exist other local minimizers with a greater function value but nevertheless they are local minimizers. Consider the following examples.

Example[15] Consider the function

$$f(x) = \begin{cases} 1, & x \in (-1, 1), \ x \neq 0 \\ 0, & x = 0 \end{cases} \tag{1.43}$$

For this function with a discontinuity at $x = 0$ we have that $x = 0$ is a strict local minimizer but every point in a neighborhood of $x = 0$ is a local minimizer.

Example. Consider the function $f(x) = x^4(2 + cos(\frac{1}{x}))$ with $f(0) = 0$. Then since $2 + cos(\frac{1}{x}) > 0$ and $x^4 > 0$ when $x \neq 0$, thus $f(x) > 0$ when $x \neq 0$. Yet the infinite oscillations in $cos(\frac{1}{x})$ mean that every neighborhood of the origin contains infinitely many local minimizers. Thus we have $x = 0$ as a strict local minimizer.

Now there are several sufficiency conditions available in literature that classify a KKT point as a strict or isolated local minimizer [4]. The condition c together with conditions a and b stated in the theorem are second order sufficient conditions for a KKT point to be an isolated local minimizer. So the theorem really ensures that the local minimizer being approached is an isolated local minimizer.

Strict Complimentarity. Finally we briefly discuss the notion of strict complimentarity used in theorem (1.3.1). The complimentary slackness condition stated as part of the first order KKT necessary conditions states that the product $u_j g_j(x)$ should be zero for all $j = 1, \ldots, l$. Now as discussed in the derivation of the KKT necessary conditions in the write up on KKT conditions and constraint qualifications, for all the constraints that are inactive at the optimal solution $\bar{x}$ the corresponding multipliers are equal to 0. However it does not bar the multipliers from being equal to 0 when the corresponding constraint is active that is $g_j(\bar{x}) = 0$ for some $j \in \mathcal{A}$. However if it so happens that for all $j \in \mathcal{A}$, the corresponding multiplier is non zero and hence strictly positive ,this is termed as strict complimentarity.

Definition 1.3.5. Strict complimentarity. Strict complimentarity is said to exist at a KKT point $\bar{x}$ if there is a multiplier $\bar{u} \in M_u$ such that $\bar{u}_j > 0$ for all $j \in \mathcal{A}$.

There are several implications of strict complimentarity and have been discussed in [4]. The imposition of strict complimentarity allows for obtaining certain bounds on the difference between (x_μ) and the optimal solution $\bar{x}$. These bounds eventually play a crucial role in the proof of the theorem 1.3.1.

This concludes the section. What follows next is a discussion of the barrier method to the convex optimization problem with a particular focus on the linear programming problems. But before that, it is important to look at certain important aspects of linear programming problems which forms the next section.

1.3.3. Linear Programming Problems- Some Basics Ideas

A linear programming problem is a special type of convex optimization problem which involves minimization or maximization of a linear function over a special convex constraint set called polyhedral set.

$$\begin{aligned} &\text{minimize} \quad c^T x \\ &\text{subject to} \quad x \in S \,(\text{polyhedral set}) \end{aligned}$$

Now as seen in one of the convex separation theorem in the write up on convex analysis, any closed convex set is the intersection of all the closed half spaces containing it. A polyhedral set is a special closed convex set which requires only a finite number of half spaces to represent it.

Definition 1.3.6. Polyhedral Set A set S in $\mathbb{R}^n$ is called a polyhedral set if it is the intersection of a finite number of closed half spaces. That is

$$S = \{x : \ p_j^T x \leq \alpha_j, \ i = \ldots m\} \tag{1.44}$$

where p_i is a non zero vector and α_j is a scaler. Note that since an equality can be represented by two inequalities, a polyhedral set can be represented by a finite number of inequalities and/or equations. Some typical polyhedral sets are

$$S = \{x : \ Ax \leq b\} \tag{1.45}$$

$$S = \{x : \ Ax = b, \ x \geq 0\} \tag{1.46}$$

$$S = \{x : Ax \geq b, \ x \geq 0\} \tag{1.47}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

Some examples of polyhedral sets are as follows. These examples have been taken from **??**.

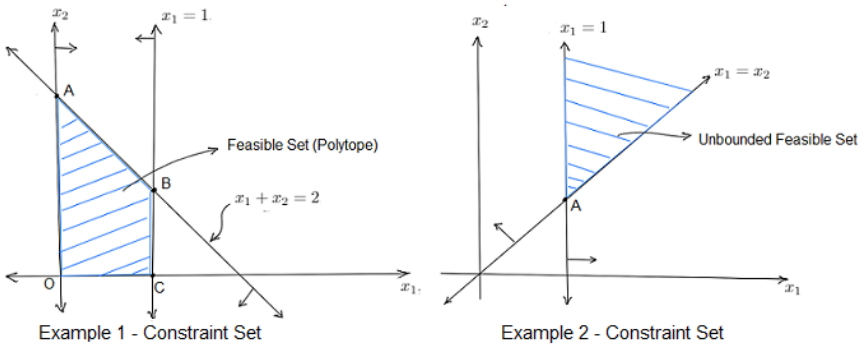


Figure 1.2. **GCQ vs ACQ**

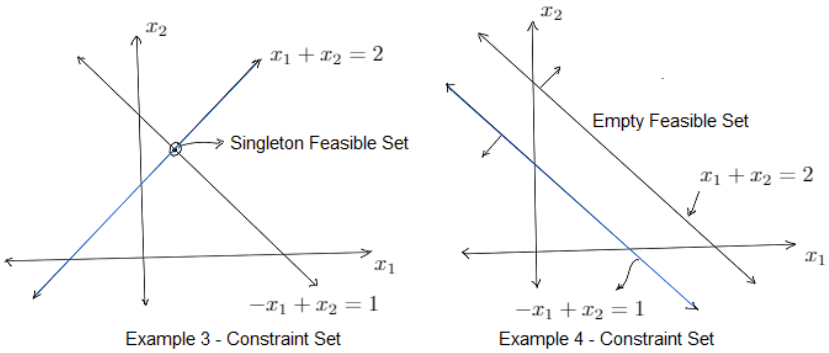


Figure 1.3. **GCQ vs ACQ**

Examples Consider the constraint sets in $\mathbb{R}^2$ given by

1. $S_1 = \{x \in \mathbb{R}^2 : \ x_1 + x_2 \leq 2, \ x_1 \leq 1, \ x_1 \geq 0, \ x_2 \geq 0\}$
2. $S_2 = \{x \in \mathbb{R}^2 : \ x_1 \geq 1, \ x_2 \geq x_1\}$
3. $S_3 = \{x \in \mathbb{R}^2 : \ x_1 + x_2 = 2, \ -x_1 + x_2 = 1\}$
4. $S_4 = \{x \in \mathbb{R}^2 : \ x_1 + x_2 \geq 2, \ -x_1 + x_2 \leq 1\}$

As shown in the figure, set S_1 is a bounded polyhedral set, set S_2 is an unbounded polyhedral set, S_3 is a singleton set whereas S_4 is empty.

Polytope A bounded polyhedral set is called a polytope.

We now state a linear programming problem in its standard form.

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \qquad\qquad (\text{LP- Standard Form}) \end{aligned}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. We denote the set feasible set by $S = \{x : \mathbb{R}^n : \ Ax = b, \ x \geq 0\}$.

Coming back to the polyhedral sets, notice in the figure (a) the presence of corner points A, B, C and D in the polyhedral set. These corner points or extreme points of a polyhedral set play an important role in the solution of a LP.

Extreme Point For a convex set S, a point $x \in S$ is said to be an extreme point of S, if x cannot be written as a strict linear combination ($\lambda \in (0,1)$) of two distinct points in S. Consider the polyhedral set S_1 in the figure. The point E can be represented as a strict convex combination of two points in S_1. This can be done by taking any line segment passing through E. However the same does not hold true for points A, B, C and D.

Definition 1.3.7. Extreme Point More rigorously, a point $x \in S$ is said to be an extreme point if $x = \lambda x_1 + (1-\lambda)x_2$ with $x_1, x_2 \in S$ and $\lambda \in (0,1)$ implies that $x = x_1 = x_2$.

In figures (b,c and d), are extreme points.

Algebraic Characterization of Extreme Points. Consider the LP in standard form with $rank(A) = m$. Let B be formed using m linearly independent columns of A. This can be done since $rank(A) = m$. Also, wlog we can assume there are the first m columns of A. So we have $A = \begin{pmatrix} B & N \end{pmatrix}$ with $B \in \mathbb{R}^{m \times m}$ and $N \in \mathbb{R}^{m \times (n-m)}$ and the system can be written as

$$\begin{pmatrix} B & N \end{pmatrix} \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b \tag{1.48}$$

The variables x_B are called basic variables and x_N are called non basic variables. Letting x_B we get $Bx_B = b$ and since B is full rank we have $x_B = B^{-1}b$. We then call $(x_B, 0)^T$ as the basic solution wrt the basis matrix B. Now if is feasible it is called as the basic feasible solution.

Basic Feasible Solution (BFS). If $x_B \geq 0$, then $(x_B, 0)^T$ is called as the Basic Feasible Solution of

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned} \tag{1.49}$$

wrt to the basis matrix B. The extreme points are characterized in terms of basic feasible solutions. In fact the next theorem shows that extreme points for a polyhedral set S are nothing but the basic feasible solutions for the constraint set S.

Theorem 1.3.2. Let $S = \{x : \mathbb{R}^n : \ Ax = b, \ x \geq 0\}$, then x is an extreme point of S if and only if x is a basic feasible solution of $Ax = b, \ x \geq 0$.

Proof. ($\Leftarrow$) Let x be a BFS of $Ax = b, \ x \geq 0$. To show that x is an extreme point of S, we must show that it cannot be expressed as a strictly convex combination of any two points in S. This is shown by contradiction. Since x is a BFS, x can be written as $x = (x_1, \ldots, x_m, 0, \ldots, 0)$. Let $B = \big(a_1, \ldots, a_m\big)$, where $a_1, \ldots, a_m$ are linearly independent columns of A. So

$$x_1 a_1 + \ldots + x_m a_m = b \tag{1.50}$$

Now suppose x is not an extreme point, then x can be expressed as a strictly convex combination of say $y, z \in S$, $y \neq z$.

$$x = \lambda y + (1 - \lambda)z;\ 0 < \alpha < 1. \tag{1.51}$$

Since $y, z \geq 0$ we have

$$y_{m+1} = \ldots = y_n = 0\ ,\ z_{m+1} = \ldots = z_n = 0 \tag{1.52}$$

$$\text{and } y_1 a_1 + \ldots + y_m a_m = b\ ,\ z_{m+1} a_{m+1} + \ldots + z_n a_n = b \tag{1.53}$$

Since $a_1, \ldots, a_m$ are linearly independent, we must have $x = y = z$ and thus x is an extreme point.

$\Rightarrow$ Let x be an extreme point of S. This means that there are n linearly independent constraints active at x. Note that m of these n linearly independent constraints correspond to $Ax = b$. The remaining $n - m$ active constraints correspond to $x \geq 0$. Thus x can be written as $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$ where $x_N = 0$. Now since $Ax = b$ we have $A\begin{pmatrix} x_B \\ x_N \end{pmatrix} = b$. Now A can be split as (B, N) where $B \in \mathbb{R}^{m \times m}$ and is such that $Bx_B = b$ □

Remarks. Note that there are $\begin{pmatrix} n \\ m \end{pmatrix}$ ways in which m columns can be chosen from n columns. Therefore $\begin{pmatrix} n \\ m \end{pmatrix}$ is an upper bound on the number of basic feasible solutions. Since there is a correspondence between an extreme point and basic feasible solution, we can say there are finite number of extreme points for a linear program.

The following result states that a polyhedral set must have at least one extreme point.

Theorem 1.3.3. Let $S = \{x :\ Ax = b\, x \geq 0, \}$ be non empty. Then S has at least one extreme point. Since an extreme point is a basic feasible solution BFS, we can interpret this theorem as stating that if there exists a feasible solution to a linear program in standard form, there exists a basic feasible solution.

The next important notion that is of importance in the study of linear programming is the notion of Extreme Directions.

Recession Direction and Extreme Direction Let S be a non empty closed convex set in $\mathbb{R}^n$ given by $S = \{x :\ Ax = b\, x \geq 0, \}$. A vector $d \in \mathbb{R}^n$ is called a recession direction of S if for each $x \in S$, $x + \lambda d \in S$ for all $\lambda \geq 0$. Two directions d_1 and d_2 of S are called distinct if $d_1 \neq \alpha d_2$ for any $\alpha \neq 0$. A direction d of S is called an extreme direction if it cannot be expressed as a linear combination of two distinct directions, that is, if $d = \lambda_1 d_1 + \lambda_2 d_2$ for $\lambda_1,\ \lambda_2 > 0$, then $d_1 = \alpha d_2$ for some $\alpha > 0$. As an example consider the constraint set $S = \{(x_1, x_2);\ x_2 \geq |x_1|\}$. The recession directions of S are given by non zero vectors that form an angle of $\frac{\pi}{4}$ with the vector $(0, 1)^T$. The extreme directions are $d_1 = (1, 1)^t$ and $d_2 = (-1, 1)^t$. It can be seen easily that a direction d is a direction of recession if $Ad = 0$ and $d \geq 0$. The converse holds true for $\lambda > 0$. An important result as stated in [] says that the number of extreme directions of S is finite with an upper bound given by .

Now by definition, a polyhedral set is the intersection of a finite number of half spaces. There is another representation of the polyhedral set in terms of the extreme points and extreme directions. We state an extremely important result often called as the Representation theorem in optimization literature. It states that any point in a non empty polyhedral set $S = \{x :\ Ax = b\, x \geq 0, \}$ can be represented as a convex combination of its extreme points plus a non negative linear combination of its extreme directions. Now if S is bounded, then it contains no extreme directions and hence any point in the set S can be represented as a convex combination of its extreme points (or the basic feasible solutions).

Theorem 1.3.4. Let S be a non empty polyhedral set in $\mathbb{R}^n$ of the form $S = \{x : \ Ax = b\, x \geq 0, \}$. Let $x_1, \ldots, x_k$ be the extreme points of S and $d_1, \ldots, d_l$ be the extreme directions of S. Then $x \in S$ if and only if

$$x = \sum_{j=1}^{k} \lambda_j x_j + \sum_{i=} l\mu_i d_i, \ \lambda_j, \mu_i \geq 0 \, \forall \, j, \, i \tag{1.54}$$

$$\sum_{j=1}^{k} \lambda_j = 1 \tag{1.55}$$

Theorem 1.3.5. For the linear programming problem in the standard form with a non empty feasible set S as in . Let be the extreme points or the basic feasible solutions and be the extreme directions of the feasible region. A necessary and sufficient condition for a finite optimal solution is that for all . If this condition holds true, there exists an extreme point that solves the problem.

Proof. Now given feasibility, if $c^T d_j < 0$ for some j, then μ_j can be chosen arbitrarily large leading to an unbounded objective function value. Thus a necessary and sufficient condition for a finite optimum objective function value is that for all . Now if this condition holds true the minimum objective function value is obtained by choosing for all . This reduces the problem to subject to . Clearly the optimal objective function value for the new problem is finite and can be found by letting $\lambda_i = 1$ and $\lambda_j = 0$ for $j \neq i$, where the index i is given by $c^t x_i = min_{1 \leq j \leq k} c^t x_j$. Thus there exists an optimal extreme point. □

We can combine the results of theorems and from the write ups on convex analysis and constrained optimization and obtain the following.

Duality in LP. Given a convex programming problem (say the LP), if $\bar{x}, \bar{u}, \bar{v}$ is a KKT point implies is a Lagrangian saddle which implies $\bar{x}$ and $\bar{u}, \bar{v}$ are primal and dual optimals with zero duality gap.

Conversely $\bar{x}$ and $\bar{u}, \bar{v}$ are primal and dual optimals with zero duality gap implies $\bar{x}, \bar{u}, \bar{v}$ is a Lagrangian saddle implying $\bar{x}, \bar{u}, \bar{v}$ satisfies the KKT conditions. Thus in particular for a LP we have the following result.

Given the following LP in standard form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \qquad \text{(LP- Standard Form)} \end{array}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. The matrix A is assumed to be full row rank in all our discussions. The vector $\bar{x} \in \mathbb{R}^n$ is a solution of LP if and only if there exist vectors $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$ such that the following KKT conditions hold for $(x, u, v) = (\bar{x}, \bar{u}, \bar{v})$

$$A^T v + u = c \tag{1.56}$$

$$Ax = b \tag{1.57}$$

$$x_j u_j = 0 \ j = 1, \ldots, n \tag{1.58}$$

$$(x, u) \geq 0 \tag{1.59}$$

Notice that the KKT conditions have been modified appropriately for the greater than equal to type inequality constraints $x \geq 0$. The condition 1.84 is the complimentary slackness condition.

1.3.4. Barrier Method for convex optimization problem

Consider the convex optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_j(x) \leq 0,\ j = 1, 2, ..., l \\ & Ax = b \qquad (\mathcal{CP}_1) \\ & x \in \mathbb{R}^n \end{array}$$

The functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^l$ are convex functions and twice continuously differentiable. The matrix $A \in \mathbb{R}^{p \times n}$ with full row rank p. The constraint set is $C = \{x \in \mathbb{R}^n,\ g(x) \leq 0,\ Ax = 0\}$. We also assume the Slater's constraint qualification which as discussed in the write up on convex analysis, implies that strong duality holds, the KKT point is the global minimizer and the KKT multipliers are solutions to the dual of the convex optimization problem. In subsection we discussed the barrier function method in general and stated a general convergence theorem. The goal now is to use the barrier function method as discussed in subsection to the convex optimization problem $\mathcal{C}P_1$. In particular we will discuss the application of the barrier method to the linear programming problem. Due to the special structure provided by linearity and convexity of the problem being a we can derive some valuable insights and use them to come up with algorithms. The discussion in this section is primarily based on chapter 11 of Boyd and Vanderberghe [6].

The essential idea is to use the logarithmic barrier function to absorb the inequality constraints into the objective function while leaving the equality constraints given by $Ax = b$ as it is. Since the problem is a convex optimization problem, the solution to the barrier problem is simply the solution to the associated KKT conditions. The barrier function for the problem can be written as $B(x, \mu) = -\frac{1}{\mu} \sum_j log(g_j(x))$. We thus have the following barrier problem.

$$\begin{array}{ll} \text{minimize} & f(x) + \dfrac{1}{\mu} B(x, \mu) \\ \text{subject to} & Ax = b \qquad (\mathcal{BP}_1) \end{array}$$

The only difference between the barrier function as written in **??** in subsection and this barrier function is that the parameter is present as $\frac{1}{\mu}$. As a result, instead of reducing μ to zero as discussed in previous sections, here μ will be increased to infinity.

Notation As opposed to earlier write ups, we shall denote the optimal solutions by x^* and not $\bar{x}$.

The barrier method. The method involves solving a sequence of barrier problems for increasing values of μ to obtain a solution $x^*(\mu)$ at each step. Starting with an initial value of $\mu = \mu_0 > 0$ and a starting point say x_0, we solve the barrier problem using the Newton's method discussed earlier. The solution $x^*(\mu)$ at the current iteration is fed as the initial starting point for the next iteration of the barrier problem with an increased value of barrier parameter μ. Note that as discussed in section , for increasing values of μ the barrier function approximates the indicator function and the hope is that as $\mu \to \infty$, $x^*(\mu) \to x^*$ where x^* is the solution to the original problem $\mathcal{CP}_1$. We have seen a general convergence theorem, however for the convex optimization problem, some simple observations can be made by analyzing the KKT conditions and dual variables for the problem . These observations would eventually justify the use of barrier method and confirm the intuition that indeed as . However before we discuss that, it is important to address an important question.

An important question that arises is to why not set μ to a very large value and then solve the problem $\mathcal{BP}_1$ and seek results directly at the end of the central path. There are several issues that may arise while

doing this. Firstly solving the problem $\mathcal{BP}_1$ for a very large value of μ may not be computationally feasible due to severe instabilities that may arise. And secondly the Newton's method would take quite a while before it gets to the quadratic convergence space. As a result it is better to traverse the whole central path for increasing values of μ and at each intermediate step, the solution is fed as the starting point for the next iteration of the Newton's method to solve the barrier problem for increasing values of μ. This essentially allows us to keep Newton's method in the quadratically convergent space.

Now as a convex optimization problem with equality constraints, the solution to the barrier problem $\mathcal{BP}_1$ can be simply found by solving the KKT conditions. We now analyze the KKT conditions for the problem $\mathcal{BP}_1$ and derive some simple observations with regards to the dual variables of the original problem $\mathcal{CP}_1$. For the barrier problem, the KKT conditions can be written as follows

$$\mu \nabla f(\bar{x}^*(\mu)) + \sum_{j=1}^{l} \frac{1}{g_j(x^*(\mu))} \nabla g_j(x^*(\mu)) + A^T w = 0, \tag{1.60}$$

$$Ax^*(\mu) = b, \tag{1.61}$$

$$g_j(x^*(\mu)) < 0,\ j = 1, \ldots, l, \tag{1.62}$$

Since $\mu > 0$, the stationarity condition 1.60 can be equivalently written as

$$\nabla f(\bar{x}^*(\mu)) + \sum_{j=1}^{l} \frac{1}{\mu g_j(x^*(\mu))} \nabla g_j(x^*(\mu)) + A^T \frac{w}{\mu} = 0 \tag{1.63}$$

Since the barrier problem is a convex optimization problem, these condition are necessary and sufficient for $x^*(\mu)$ to be a global minimizer for the barrier or the central path problem. Also note that there is no complimentary slackness condition as there are no inequality constraints in the barrier problem. Now the solution to the system of equations would solve the central path problem completely and $\bar{x}(\mu)$ so obtained will be primal feasible for the original problem $\mathcal{P}_1$. This is because $x^*(\mu)$ satisfies the primal feasibility conditions 1.61 and 1.62. Now from the central path problem, we can derive the dual feasible points for the original problem $\mathcal{P}_1$. This can be seen as follows. We define

$$u_j^* = -\frac{1}{\mu g_j(x^*(\mu))};\ j = 1, \ldots, l \tag{1.64}$$

$$v^* = \frac{w}{\mu} \tag{1.65}$$

The claim now is that u^* and v^* are dual feasible points for the original problem. The Lagrangian for the original problem is

$$L(x, u, v) = f(x) + \sum_{j=1}^{l} u_j g_j(x) + v^T(Ax - b). \tag{1.66}$$

For u^* and v^* to be dual feasible for the original problem $\mathcal{P}_1$, we should prove that $u_j^* > 0$ and that u_j^* and v^* lie in the domain of the Lagrangian dual function of the original problem. We first notice that $u_j^* > 0$. This is because $g_j(x^*(\mu)) < 0$ and $\mu > 0$. Now notice that the Lagrangian $L(x, u, v)$ is a convex function since it is a linear combination of convex functions. Now for the particular choice of $u = u^*$ and $v = v^*$ we have

$$L(x, u^*, v^*) = f(x) + \sum_{j=1}^{l} u_j^* g_j(x) + v^{*T}(Ax - b) \tag{1.67}$$

The stationary point for $L(x, u^*, v^*)$ is the global minimizer for $L()$ since $L(x, u, v)$ is a convex function. Notice from 1.63, $x^*(\mu)$ is the minimizer for $L(x, u^*, v^*)$ since $\nabla L(x^*(\mu), u^*, v^*) = 0$. Since the dual objective function for the original problem is

$$g(u, v) = inf_x\, L(x, u, v),\ u \geq 0 \tag{1.68}$$

we have

$$g(u^*, v^*) = inf_x\, L(x, u^*, v^*) \tag{1.69}$$

Since $x^*(\mu)$ minimizes $L(x, u^*, v^*)$, we have $g(u^*, v^*) > -\infty$, (u^*, v^*) lies in the domain of the Lagrangian dual objective function for the original problem $\mathcal{P}_1$. Thus u^* and v^* as given by 1.64 and 1.65 are dual feasible points for the original problem. This allows us to find the duality gap between at each step along the central path. Since $x^*(\mu)$ solves **??** we can write

$$g((u^*, v^*)) = L(x^*(\mu), u^*, v^*) = f(x^*(\mu)) + \sum_{j=1}^{l} u_j^* g_j(x) + {v^*}^T (Ax^*(\mu) - b) \tag{1.70}$$

Substituting the value for u^* from 1.64 and since $Ax^*(\mu) - b$ we obtain

$$g((u^*, v^*)) = f(x^*(\mu)) - \frac{l}{\mu} \tag{1.71}$$

So we have the duality gap associated with the primal feasible and dual feasible pair and

Now the KKT system of equations 1.60 and 1.61 can be solved using the Newton's method. We can write the system as

$$r(x, v) = \begin{pmatrix} \nabla f(\bar{x}) + \sum\limits_{j=1}^{l} \frac{1}{\mu g_j(x)} \nabla g_j(x) + A^T v \\ Ax = b \end{pmatrix} \tag{1.72}$$

The Newton's step on this system of equations is given as

$$\begin{pmatrix} H_r(x) & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta v \end{pmatrix} = -r(x, v) \tag{1.73}$$

where $H_r(x) = \nabla^2 f(x) + \sum\limits_{j=1}^{l} \frac{1}{\mu g_j(x)^2} \nabla g_j(x) \nabla g_j(x)^T + \sum\limits_{j=1}^{l} \frac{-1}{\mu g_j(x)} \nabla^2 g_j(x)$ and the matrix on the left side is just the Jacobian of the non linear function $r(x, v)$. The system 1.73 is a linear approximation of the KKT system 1.60 and 1.61. The solution to the equation $r(x, v) = 0$ for a particular value of μ is the solution to the barrier optimization problem for this particular value of μ. Once we get the optimal solution, the value of μ is increased and the system $r(x, v) = 0$ is solved for this new value of μ using the previous optimal solution (x^*, v^*) as the initial guess to solve the new system. If we denote the value of μ by μ_{k+1}, where $\mu_{k+1} > \mu_k$, the system $r(x(\mu_{k+1}), v(\mu_{k+1})) = 0$ is solved iteratively using $\big(x^*(\mu_k), v^*(\mu_k)\big)$ as the initial guess and 1.73 as the linear approximation for $r\big(x(\mu_{k+1}), v(\mu_{k+1})\big) = 0$ at every iteration. The solution to the system $r(x, v) = 0$ is obtained to the required accuracy. Thus the barrier method involves inner and outer iterations. The outer iterations keep a track of increasing values of μ. And for every such value of μ, the inner iterations solve the system of equations $r(x, v) = 0$ obtaining the optimal solution to the barrier problem for the particular value of μ.

Primal Dual Central Path. Also note that, from the solution $\big(x^*(\mu), v^*(\mu)\big)$ for the barrier problem corresponding to the parameter value μ, we can find $u^*(\mu)$ from 1.64. The locus of $\big(x^*(\mu), u^*(\mu), v^*(\mu)\big)$ for increasing or decreasing values of barrier parameter μ (depending on how the barrier problem is set up), is called the primal-dual central path. The barrier method now can be stated as follows.

Let us now look at the barrier method as applied to the linear programming problem in the standard form.

1.3.5. Barrier Method on the Linear Programming Problem

Consider the linear programming problem in the standard form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \qquad \text{(LP- Standard Form)} \end{array}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. The discussion in this subsection and the subsequent section on primal dual interior point methods to solve the linear programming problem is based on Wright[]. To be consistent with the notations in [] we denote the constraint set by $\mathcal{F}$ and the strictly feasible set by $\mathcal{F}^0$. They are given as,

$$\mathcal{F} = \{(x, v, u) \mid Ax = b,\ A^T v + u = c,\ (x, u) \geq 0\} \tag{1.74}$$

$$\mathcal{F}^\wr = \{(x, v, u) \mid Ax = b,\ A^T v + u = c,\ (x, u) > 0\} \tag{1.75}$$

The strictly feasible condition can be written as

$$(x^k, v^k, u^k) \in \mathcal{F}^\wr \tag{1.76}$$

Now as discussed in previous sections, x^* and (u^*, v^*) are primal and dual solutions for the given linear programming problem if and only if

$$A^T v + u = c \tag{1.77}$$

$$Ax = b \tag{1.78}$$

$$x_j u_j = 0 \ j = 1, \ldots, n \tag{1.79}$$

$$(x, u) \geq 0 \tag{1.80}$$

Now the barrier method suggests that, instead of solving the KKT conditions directly we solve the perturbed KKT conditions given below

$$A^T v + u = c \tag{1.81}$$

$$Ax = b \tag{1.82}$$

$$x_j u_j = \mu \ j = 1, \ldots, n \tag{1.83}$$

$$(x, u) \geq 0 \tag{1.84}$$

The solution to the perturbed KKT condition is same as the solution to the barrier problem with corresponding barrier parameter μ. In the barrier method we solve the perturbed KKT system till optimality as follows.

Concluding Remarks. Note the barrier method involves the solving the barrier problem for a particular value of the parameter μ till we reach its optimal solution. As a result we always stay on the central path. Now staying on the central path is really not necessary if we can stay close to the central path while reaching out for the optimal solution at the boundary of the feasible set. So, instead of solving the barrier problem up to optimality by solving the KKT system using several iterations of the Newton's method, we take just one Newton's step, use this as the solution and move on. This solution is certainly not the solution to the barrier problem and hence does not lie on the central path. However it may still be good enough if it can be controlled to stay within acceptable neighborhood of the primal dual central path. This is the motivation behind the topic for the next section. The figure is a depiction of these ideas.

1.3.6. Primal Dual Central Path Algorithms for the Linear Programming Problem

As seen at the end of the previous section, the solution to the equations 1.81-1.83 while ensuring 1.84, is the solution to the barrier problem for linear program in standard form, for the barrier parameter μ. Let us now investigate a few subtle issues. We observe the equation 1.83 more carefully. The solution $(x(\mu), u(\mu), v(\mu))$ to the equations 1.81-1.83 is such that $x_j u_j = \mu$ for every j. Now if we calculate the average of $x_j u_j$ for $j = 1 \dots n$, we get $\frac{1}{n} \sum_{1}^{l} x_j u_j = \mu$. So what equation 1.83 suggests is that the solution to 1.81-1.83 should be such that the average $\frac{1}{n} x^T u$ equals μ, and each individual term $x_j u_j$ is equal to the average. This is like, one possible way to have the average of 10 numbers to be equal to 2 is by taking all 10 numbers equal to 2. Now we would like to solve the KKT equations 1.81-1.83 while ensuring 1.84, for values of μ reducing to zero. This can be done by reducing μ by a factor say $\sigma \in (0, 1)$. Then the KKT equations can be written as

$$A^T v + u = c \tag{1.85}$$

$$Ax = b \tag{1.86}$$

$$x_j u_j = \sigma\mu, \ j = 1, \dots, n \tag{1.87}$$

$$(x, u) \geq 0 \tag{1.88}$$

Now, if we were to solve the KKT equations 1.85-1.87 ensuring $(x, u) \geq 0$, till we obtained a solution (within certain accuracy) we would be solving the corresponding barrier problem till optimality. The solution $x_{\sigma\mu}, u_{\sigma\mu}, v_{\sigma\mu}$ to the KKT equations corresponding to the parameter $\sigma\mu$ would be such that the average $\frac{x^T u}{n}$ equals $\sigma\mu$ and each individual term $x_j u_j$ equals the average. Now in the motivation for the primal dual central path algorithms, we said that the key idea behind the primal dual central path algorithms is that they do not solve the KKT equations up to optimality, but they take only one Newton iteration and move on. It should be noted again that solving KKT equations to optimality would require several Newton iterations where the Newton direction at each step would now be given by the following linearized system corresponding to the system 1.85-1.87.

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ U & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta v \\ \Delta u \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -XUe + \sigma\mu e \end{pmatrix} \tag{1.89}$$

We obtain 1.89 as follows. The system 1.85-1.88 can be written as

$$r(x, u, v) = \begin{pmatrix} A^T v + u - c \\ Ax - b \\ x_j u_j \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \sigma\mu \end{pmatrix} \tag{1.90}$$

Then, the Newtons method suggests solving following system iteratively

$$JF(x, u, v) \begin{pmatrix} \Delta x \\ \Delta u \\ \Delta v \end{pmatrix} = -F(x, u, v) \tag{1.91}$$

where JF is the Jacobian of F and this is precisely the left hand side of 1.89. Note that X and U are diagonal matrices as defined in **??**. Now since we are going to take only one Newton iteration, it is clear that this is not enough to reach to the solution of 1.85-1.87. As a result, subsequent iterates do not stay on the central path as shown in figure . Then how do we control the iterates from swaying too far away from the central path and ensuring that one Newton iteration for each value of $\sigma\mu$ is enough to guide them to the solution of the original problem while staying close to the central path. And it is at this step we define the two neighborhoods $N_2(\theta)$ and $N_{-\infty}(\theta)$ of the central path. These primal dual central path algorithms that we shall see, ensure that the iterates $(x_{\sigma_k\mu_k}, u_{\sigma_k\mu_k}, v_{\sigma_k\mu_k})$ obtained by taking just one step for the KKT

system corresponding to $\sigma_k\mu_k$ remain confined to these neighborhoods for each value of k as $k \to 0$. This ensures that the iterates do not sway too far away from the central path as well as the duality gap achieves substantial reduction.

We now first define the neighborhood $N_2(\theta)$ as

$$N_2(\theta) = \{(x,u,v) \in \mathcal{F}^0;\ ||XU - \mu e||_2 \leq \theta\mu\} \tag{1.92}$$

$$N_{-\infty}(\theta) = \{(x,u,v) \in \mathcal{F}^0;\ x_j u_j \geq \gamma\mu,\ j = 1 \ldots n\} \tag{1.93}$$

for $\theta \in (0,1)$ and $\gamma \in (0,1)$, where X and U are as defined in **??** and **??**. We shall first discuss in detail the neighborhood $N_2(\theta)$ which will lead us to the short step algorithm. The neighborhood $N_{-\infty}(\gamma)$ will be discussed in detail later which will lead us to the long step algorithm. Let us first understand what the neighborhood $N_2(\theta)$ really means.

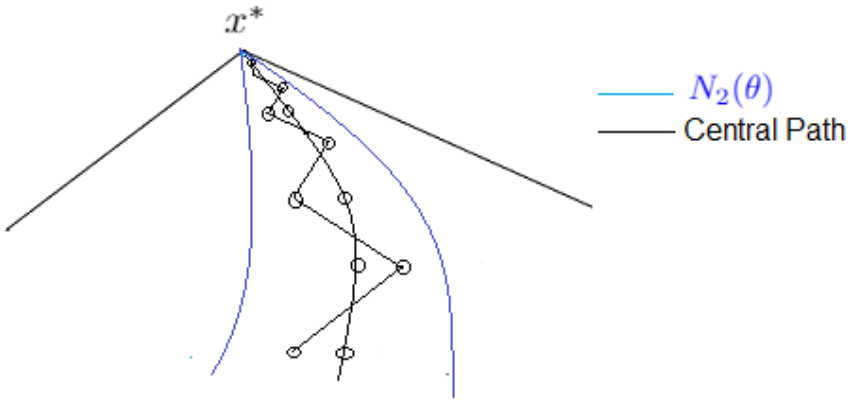


Figure 1.4. **The $N_2(\theta)$ neighborhood**

The $N_2(\theta)$ neighborhood. The neighborhood $N_2(\theta)$ is a collection of points which satisfy 1.81, 1.82 and 1.83 since $(x,v,u) \in \mathcal{F}^0$. What they do not satisfy is the equation 1.84 which ensures that the solution stays on the central path. Instead the neighborhood is a collection of all points $(x,u,v) \in \mathcal{F}^0$ whose average values of $x_j u_j$, $j = 1 \ldots n$ is at the most equal to μ and that individual terms $x_j u_j$ may not be equal to μ. This is because if all the terms $x_j u_j$ are equal to μ, then the point is on the central path. Figure gives a geometric picture of how the neighborhood looks like. Now since the optimal corner of the constraint polytope is characterized by $\mu = 0$, for decreasing values of μ the neighborhood $N_2(\theta)$ has a tapering shape. Now we solve the KKT system 1.81-1.83 for decreasing values of μ by some factor σ. This sequence of different values of μ reducing by a factor of σ is denoted as μ_k. Then for decreasing values of μ_k the neighbourhood shrinks in size and becomes narrower as $\mu_k \to 0^+$.

Short Step Algorithm. The short step algorithm now works as follows. Suppose the central path is characterized by solutions to the KKT system 1.85-1.87 parametrized by $\sigma_k\mu_k$. Then the central path is as shown in the figure . Now if we were to actually solve the system 1.85-1.87 using Newtons update given by 1.89, then the solution $(x^{k+1}, v^{k+1}, u^{k+1})$ would lie on the central path with a reduction in the duality gap by the factor σ_k and each term $x_j^{k+1}u_j^{k+1}$ being equal to $\sigma_k\mu_k$. But we do not solve the system 1.85-1.87 till we actually obtain a solution, instead we take only one Newton step by solving the system 1.89 only once. Thus, our solution does not lie on the central path as expected. However with clever choices of $\theta \in (0,1)$ and the parameter $\sigma_k \in (0,1)$ we can ensure that $(x^{k+1}, v^{k+1}, u^{k+1}) = (x^k, v^k, u^k) + (\Delta x, \Delta v, \Delta u)$ where $(\Delta x, \Delta v, \Delta u)$ is obtained by solving 1.89 for $\sigma_k\mu_k$, lies in $N_2(\theta)$. This means that the $(x^{k+1}, v^{k+1}, u^{k+1}) \in \mathcal{F}^0$ and the average values of $x_j^{k+1}u_j^{k+1}$ are at the most equal to $\sigma_k\mu_k$ and that individual $x_j^{k+1}u_j^{k+1}$ are not necessarily equal to

$\sigma_k\mu_k$. However as $\sigma_k\mu_k$ approaches 0, the $N_2(\theta)$ neighborhood becomes narrower forcing the corresponding iterates to stay closer and closer to the central path and finally tapering into the solution. The short step algorithm is given as follows.

Notation. To avoid notational clutter we shall denote $\omega^k = (x^k, v^k, u^k)$

The short step algorithm is stated as follows. This will be followed by a detailed theoretical analysis of the algorithm resulting into a complete proof of the polynomial time complexity of the short step algorithm.

Short Step Algorithm
Given $\theta, \delta \in (0,1)$ such that $\frac{\theta^2+\delta^2}{2^{\frac{3}{2}}(1-\theta)} \leq \left(1-\frac{\delta}{\sqrt{n}}\right)\theta$
Set $\sigma = \left(1-\frac{\delta}{\sqrt{n}}\right)\theta$ and choose $\omega^0 = (x^0, u^0, v^0) \in N_2(\theta)$
avg = $\frac{{x^0}^T u^0}{n}$, set eps=tol
while *avg>eps* **do**
    solve 1.89 with $\omega = \omega^k$ for $\Delta\omega = (\Delta x, \Delta v, \Delta u)$
    $\omega^{k+1} = \omega^k + \Delta\omega$
    avg = $\frac{{x^{k+1}}^T u^{k+1}}{n}$
end
end

The following theorem is stated which provides theoretical credibility to the algorithm and all the ideas we have discussed so far.

Some remarks will follow this theorem in an attempt to summarize the key ideas at play here. We shall then finally state the theorem which proves the polynomial time complexity of this algorithm. Before the proof of the these theorems we shall look at two technical lemmas which shall be used in the proof. We shall state those technical lemmas along with their proofs and then continue with the proof of the main theorem.

Theorem 1.3.6. Given the choices of θ and σ as in the short step algorithm, we have the following,

1. $\omega^{k+1} = (x^{k+1}, v^{k+1}, u^{k+1}) \in N_2(\theta)$ for all k and
2. $\mu(\omega^{k+1}) = \frac{{x^{k+1}}^T v^{k+1}}{n} = \sigma\mu(\omega^k)$
3. $\mu(\omega^k) = \left(1-\frac{\delta}{\sqrt{n}}\right)^n \mu(\omega_0)$

Remark. Note that in this algorithm we reduce μ_k by a constant factor σ =. Now starting at ω^k if were to solve the system 1.85-1.87 using the Newtons method then the resulting solution would have $x_j^{k+1}u_j^{k+1} = \sigma\mu(\omega^k)$ for all j. Now the theorem states that just one iteration of Newtons method for the system corresponding to $\sigma\mu(\omega^k)$, we have, $\mu(\omega^{k+1}) = \sigma\mu(\omega^k)$. Is there any contradiction because we are getting the same reduction in the average value in just one Newton iteration that we would have expected having the solved the system to its solution. The answer is that there is no contradiction, for the simple reason that even though the reduction in average is same, the individual terms $x_j^{k+1}u_j^{k+1}$ are not equal to the average value. A complete solution to the system would have forced individual terms to be equal to the average value.

We shall now state two technical lemmas and their proofs.

Lemma 1.2. Let u and v be any two vectors in $\mathbb{R}^n$ with $u^T v \geq 0$. Then

$$|||UVe| \leq 2^{\frac{-3}{2}} ||u+v||^2 \tag{1.94}$$

where $U = diag(u_1, \ldots, u_n)$ and $V = diag(v_1, \ldots, v_n)$.

Proof. For scalers α and β we have $\sqrt{|\alpha\beta|} \leq \frac{1}{2}|\alpha+\beta|$. Define $\mathcal{P} = \{i|\ u_i v_i \geq 0\}$ and $\mathcal{M} = \{i|\ u_i v_i \geq 0\}$. Since $u^T v \geq 0$, we have

$$0 \leq u^T v = \sum_{u_i v_i \geq 0} u_i v_i + \sum_{u_i v_i < 0} u_i v_i = \sum_{i\in\mathcal{P}} |u_i v_i| - \sum_{i\in\mathcal{M}} |u_i v_i| = ||[u_i v_i]_{i\in\mathcal{P}}||_1 - |||[u_i v_i]_{\mathcal{M}}||_1 \tag{1.95}$$

Thus

$$\begin{aligned} ||UVe|| &= \left(|||[u_i v_i]_{i\in\mathcal{P}}||^2 + |||[u_i v_i]_{i\in\mathcal{M}}||^2\right) \\ &\leq \left(|||[u_i v_i]_{i\in\mathcal{P}}||_1^2 + |||[u_i v_i]_{i\in\mathcal{M}}||_1^2\right) \end{aligned} \tag{1.96}$$

Now from 1.95, use the fact that $a - b \geq 0$ gives $a^2 + b^2 \leq 2a^2$, then for $a = |||[u_i v_i]_{i\in\mathcal{P}}||_1$ and $|||[u_i v_i]_{\mathcal{M}}||_1$ and 1.96 we have

$$\begin{aligned} ||UVe||^2 \leq |||[u_i v_i]_{\mathcal{P}}||_1^2 + |||[u_i v_i]_{\mathcal{M}}||_1^2 \leq 2|||[u_i v_i]_{\mathcal{P}}||_1^2 \\ \Rightarrow ||UVe|| < 2^{\frac{1}{2}} \left(|||[u_i v_i]_{\mathcal{P}}||_1^2\right)^{\frac{1}{2}}. \end{aligned} \tag{1.97}$$

Now consider a vector $v = [a_1 b_1, \ldots, a_n b_n]^T$. Then from the scaler inequality stated at the beginning we have

$$||v||_1 = \left(|a_1 b_1| + \ldots + |a_n b_n|\right) \leq \frac{1}{4}\left(\left(a_1 + b_1\right)^2 + \ldots + \left(a_n + b_n\right)^2\right) = \frac{1}{4}|||[\left(a_i + b_i\right)^2]|||_1 \tag{1.98}$$

Use 1.98 in 1.97 to obtain

$$||UVe|| \leq 2^{\frac{1}{2}} \left|\left|[\frac{1}{4}(u_i + v_i)]_{i\in\mathcal{P}}\right|\right|_1 = 2^{-\frac{3}{2}} \sum_{i\in\mathcal{P}} \left(u_i + v_i\right)^2 \leq 2^{-\frac{3}{2}} \sum_{i=1}^{n} \left(u_i + v_i\right)^2 = 2^{-\frac{3}{2}} ||u+v||^2 \tag{1.99}$$

□

Lemma 1.3. $\left|\left|\Delta X \Delta U\, e\right|\right| \leq 2^{-\frac{3}{2}} \left|\left|(XU)^{-\frac{1}{2}}\left(-XUe + \sigma\mu e\right)\right|\right|^2$

Proof. The last Newton block equation is $U\Delta x + X\Delta u = \left(-XUe + \sigma\mu e\right)$. Multiply this by $(XS^{-\frac{1}{2}})$ where $X = diag(x_1, \ldots, x_n)$ and $U = diag(u_1, \ldots, u_n)$. Since X and U are diagonal matrices and their entries x_i, u_i are strictly positive, the products involving negative powers of X and U are well defined and the products are commutative. Let $D = X^{\frac{1}{2}} U^{-\frac{1}{2}}$ and we obtain

$$D^{-1}\Delta x + D\Delta u = (XU)^{-\frac{1}{2}}\left(-XUe + \sigma\mu e\right) \tag{1.100}$$

Now apply previous lemma with $u = D^{-1}\Delta x$ and $v = D\Delta u$. Since D is a diagonal matrix, the matrix UVe from the lemma is given by $\Delta X \Delta U\, e$ and so we have

$$\left|\left|\Delta X \Delta U\, e\right|\right| = \left|\left|\left(D^{-1}\Delta x\right)\left(D\Delta u\right)\right|\right| \leq 2^{-\frac{3}{2}} \left|\left|D^{-1}\Delta x + D\Delta u\right|\right|^2 = 2^{-\frac{3}{2}} \left|\left|\left(XU\right)^{-\frac{1}{2}}\left(-XUe + \sigma\mu e\right)\right|\right|^2 \tag{1.101}$$

$$= 2^{-\frac{3}{2}} \sum_{j=1}^{n} \frac{-x_j u_j + \sigma\mu}{x_j u_j} \tag{1.102}$$

$$\leq 2^{-\frac{3}{2}} \frac{||XUe - \sigma\mu e||}{min_j x_j u_j} \tag{1.103}$$

□

We now complete the proof of the theorem.

Proof. We must first show that $\omega^{k+1} \in \mathcal{F}^0$. This can be shown as follows. Assuming $\omega^k \in N_2(\theta)$ we first show that $\Omega^{k+1} \in \mathcal{F}^0$. We have

$$Ax^{k+1} = Ax^k + A\Delta x \tag{1.104}$$

$$A^T y^{k+1} + u^{k+1} = A^T v + u + A^T \Delta v + \Delta u \tag{1.105}$$

From the Newton's update equations 1.89 we have $A\Delta x = 0$ and $A^T \Delta y + \Delta u = 0$. Since $\omega^k \in \mathcal{F}^0$ we have $Ax^k = b$ and $A^T v + u = c$. Thus we get $Ax^{k+1} = b$ and $A^T y^{k+1} + u^{k+1} = c$. Now in order to prove that $\omega^{k+1} \in N_2(\theta)$ we must show that $||X^{k+1}U^{k+1}e - \mu^{k+1}e|| \leq \theta\mu^{k+1}$. Let us proceed to show that. We have

$$X^{k+1}U^{k+1}e = (X^k + \Delta X)(U^k + \Delta U)e \tag{1.106}$$

$$= X^k \Delta x + U^k \Delta x + \Delta X \Delta U e + X^k U^k e. \tag{1.107}$$

$$= \sigma\mu(\omega^k)e - X^k U^k e + \Delta X \Delta U e + X^k U^k e \text{ (from 1.89)} \tag{1.108}$$

$$= \mu(\omega^{k+1})e + \Delta X \Delta U e, \text{ (since } \sigma\mu(\omega^k) = \mu_{k+1}) \tag{1.109}$$

Thus we have

$$X^{k+1}U^{k+1}e - \mu(\omega^{k+1})e = \Delta X \Delta U e \tag{1.110}$$

$$||X^{k+1}U^{k+1}e - \mu(\omega^{k+1})e|| = ||\Delta X \Delta U e|| \tag{1.111}$$

We will now use the lemma to put a bound on the right hand side of 1.111. We have from 1.103

$$||\Delta X \Delta U e|| \leq 2^{-\frac{3}{2}} \frac{||X^k U^k e - \mu(\omega^{k+1})e||}{min_j x_j^k u_j^k} \tag{1.112}$$

Since $\omega^k \in N_2(\theta)$, we have

$$||X^k U^k - \mu(\omega^k)|| \leq \theta\mu(\omega^k) \tag{1.113}$$

Hence we have $x_j^k u_j^k \geq (1-\theta)\mu(\omega^k)$ for all j. In particular we have $min_j x_j^k u_j^k \geq (1-\theta)\mu(\omega^k)$. Now we can bound the numerator as follows,

$$||X^k U^k e - \mu^{k+1}e||^2 \tag{1.114}$$

$$= ||X^k U^k e - \mu(\omega^k)e + \mu(\omega^k)e - \sigma\mu(\omega^k)e||^2 \tag{1.115}$$

$$= ||X^k U^k e - \mu(\omega^k)e||^2 + 2(1-\sigma)\mu(\omega^k)e^T(X^k U^k e - \mu(\omega^k)e) + (1-\sigma)^2\mu(\omega^k)^2 e^T e \tag{1.116}$$

$$\leq \theta\mu(\omega^k)^2 + (1-\sigma)^2\mu(\omega^k)^2 n \tag{1.117}$$

Note that $e^T e = n$ and since $\omega^k \in N_2(\theta)$. Now $\sigma = \left(1 - \frac{\delta}{\sqrt{n}}\right)$ with $\delta = 0.4$ as discussed earlier. This gives $(1-\sigma)^2 = \frac{\delta^2}{n}$ and we get the bound on $||\Delta X \Delta U e||$ as

$$||\Delta X \Delta U e|| \leq \frac{\theta\mu(\omega^k)^2 + (1-\sigma)^2\mu(\omega^k)^2 n}{2^{\frac{3}{2}}(1-\theta)\mu(\omega^k)} = \frac{\theta^2 + \delta^2}{2^{\frac{3}{2}}(1-\theta)}\mu(\omega^k) \tag{1.118}$$

It is at this stage that the choice of θ and δ such that $\frac{\theta^2+\delta^2}{2^{\frac{3}{2}}(1-\theta)} \leq \left(1 - \frac{\delta}{\sqrt{n}}\right)$ as mentioned in the short step algorithm gets motivated. We thus have finally

$$||\Delta X \Delta U e|| \leq \left(1 - \frac{\delta}{\sqrt{n}}\right)\mu(\omega^k) \tag{1.119}$$

which finally gives us the bound

$$||X^{k+1}U^{k+1}e - \mu(\omega^{k+1})e|| \leq (1 - \frac{\delta}{\sqrt{n}})\mu(\omega^k) \tag{1.120}$$

. We have thus shown that $\omega^{k+1} \in N_2(\theta)$ and this completes the proof of part 1.

2. We have from 1.109

$$X^{k+1}U^{k+1}e = \mu(\omega^{k+1})e + \Delta X \Delta U e \tag{1.121}$$

Now since $A^T\Delta x + \Delta u = 0$ and $A\Delta x = 0$ from 1.89, multiplying the first of two equations with Δx^T we get $\Delta x^T \Delta u = 0$. Pre-multiply 1.121 by e^T to get

$$\frac{1}{n}e^T X^{k+1}U^{k+1}e = \mu(\omega^{k+1}) = \sigma\mu(\omega^k) \tag{1.122}$$

3. We thus have from (2), $\mu(\omega^{k+1}) = \left(1 - \frac{\delta}{\sqrt{n}}\right)\mu(\omega_k)$ This holds true for all k and thus we have by recursion

$$\mu(\omega^k) = \left(1 - \frac{\delta}{\sqrt{n}}\right)^k \mu(\omega_0) \tag{1.123}$$

□

Theorem 1.3.7. To attain the average duality gap of ϵ it takes $\mathcal{O}(\sqrt{n}|log\epsilon|)$ Newton iterations.

Proof. Conclusion 3 of the theorem states then $\mu^k = \left(1 - \frac{\delta}{\sqrt{n}}\right)^k \mu^0$ or equivalently $\frac{\mu^k}{\mu^0} = \left(1 - \frac{\delta}{\sqrt{n}}\right)^k$ Now $\left(1 - \frac{\delta}{\sqrt{n}}\right)^k \leq \epsilon \iff klog\left(1 - \frac{\delta}{\sqrt{n}}\right) \leq log\epsilon$ Using the inequality $log(1+x) \leq x$, $x \geq -1$, we have $-k\frac{\delta}{\sqrt{n}} \leq log\epsilon \Rightarrow klog\left(1 - \frac{\delta}{\sqrt{n}}\right) \leq log\epsilon$. Finally we have $-k\frac{\delta}{\sqrt{n}} \leq log\epsilon \iff k \geq \frac{\sqrt{n}}{\delta}|log\epsilon|$. Thus,

$$\boxed{\frac{\mu^k}{\mu^0} \leq \epsilon \text{ in } \mathcal{O}(\sqrt{n}|log\epsilon|) \text{ Newton's iterations}}$$

□

1.3.7. Computational Experiments

To verify the algorithm and its working, we solved randomly generated instances of linear programming problem in the standard form.

The matrix A was a randomly generated matrix using the rand(m,n) command on MATLAB and the objective function coefficient matrix c was also generated using the same command. Thus we have the constraint matrix and the objective function coefficient vector generated at random. The algorithm requires the starting point (x^0, u^0, v^0) to lie in the neighborhood $N_2(\theta)$, for which we generate them in such a way that $(x^0, u^0, v^0) \in \mathcal{F}^0$. So in order to have a starting point we first generate a random vector $x, u \in \mathbb{R}^n$ and take its absolute value to satisfy the inequality constraints. Using this x, we obtain the vector $b = Ax$. The vector v is evaluated such that $A^Tv + u = c$. We thus have a randomly generated instance of linear program with a starting point $(x^0, u^0, v^0) \in \mathcal{F}^0$. Now we must take care of the fact that $\omega^0 \in N_2(\theta)$.

We can fix a $\mu > 0$ and randonly generated u or x we can calculate the other variable such that $x^0 = \frac{\mu.}{u^0}$ (this is MATLAB notation which means that $x_j^0 = \frac{\mu}{u_j^0}$). This ensures that we start on the central path because

then $x_j^0 u_j^0 = \mu$. Starting on the central path trivially implies starting in $N_2(\theta)$.

Remark. We reiterate the fact that μ is the duality gap and is used as the measure of optimality. And all of this relates to the theoretical results we have studied through the course of these notes that for a convex optimization problem with Slater's qualification, strong duality holds and the duality gap can be used as a measure of optimality. We run the code till the duality gap drops to 10^{-6}. In this write up we show two complete problem instances and their solutions.

The first problem is such that it has 10 constraints and 5 variables. The second problem is such that it has 5 constraints and 10 variables. We have shown result for the case 2 with starting point in the neighborhood $N_2(\theta)$. The correctness of our solution was tested along with the result for the same problem using the package $glpk$ on Octave for solving linear programming problems. The results of the problem obtained from our code and the octave package have been summarized. The matrix A and vector c for the first problem are shown in the figure and for the second problem are shown in figure in the appendix. A com-

optimal x*	shortstep.m	glpk
	0.91222	0.91222
	0.86828	0.86828
	0.44031	0.44031
	0.33919	0.33919
	0.76990	0.76990

Table 1.2. **Primal Optimal Solution, shortstep.m vs glpk**

parison between the values of x^* and u^* is given in the table below. It should be noticed that the product $x_j^* u_j^* \approx 10^{-6}$. This corresponds to the complimentary slackness as is expected at the optimal solution. The

optimal x*	optimal u*
0.91222	0.00000095593
0.86828	0.00000100431
0.44031	0.00000198045
0.33919	0.00000257086
0.76990	0.00000113264

Table 1.3. **Solution to problem 1 (optimal primal and dual variable)**

starting average was $\frac{1}{n} x^{0^T} u^0 = 0.38894$, whereas the duality gap at the optimal was found to be equal to 0.00000087202 (see the appendix for the dual variable v^*).

For the second problem with 5 constraints and 10 variables, the constraint matrix A and coefficient vector c are shown in the appendix. The optimal solution to the problem as obtained from our code is given in the table. Notice that 5 constraints for a 10 variable cause 5 basic feasible solutions and 5 non basic feasible solutions. We would expect our optimal value of x to have 5 non zero components and as many components with a value of 0. The corresponding dual variables should be non zero or zero so that their product is zero for the complimentary slackness condition to hold true.

optimal x^*	optimal u^*
0.40990055151	0.0000022588
0.00000087504	1.0580844813
0.51448340783	0.0000017996
0.47590772591	0.0000019455
0.00000168410	0.5497705082
0.00000120789	0.7665195543
0.00000314474	0.2944190185
0.00000077267	1.1982755178
0.17555384948	0.0000052740
0.86271356090	0.0000010732

Table 1.4. **Solution to problem 2 (optimal primal and dual variable)**

Finally we have tabulated the number of iterations taken to converge to the optimal solution for higher dimensional randomly generated instances of linear programming.

Constraints (m)	Variables (n)	No. Iterations
10	30	167
30	50	216
50	70	254
70	90	289

Table 1.5. **Higher Dimensional Results, No. of variables greater than No. of constraints**

Constraints (m)	Variables (n)	No. Iterations
30	10	91
50	30	165
70	50	212
90	70	253

Table 1.6. **Higher Dimensional Results, No. of constraints greater than No. of variables**

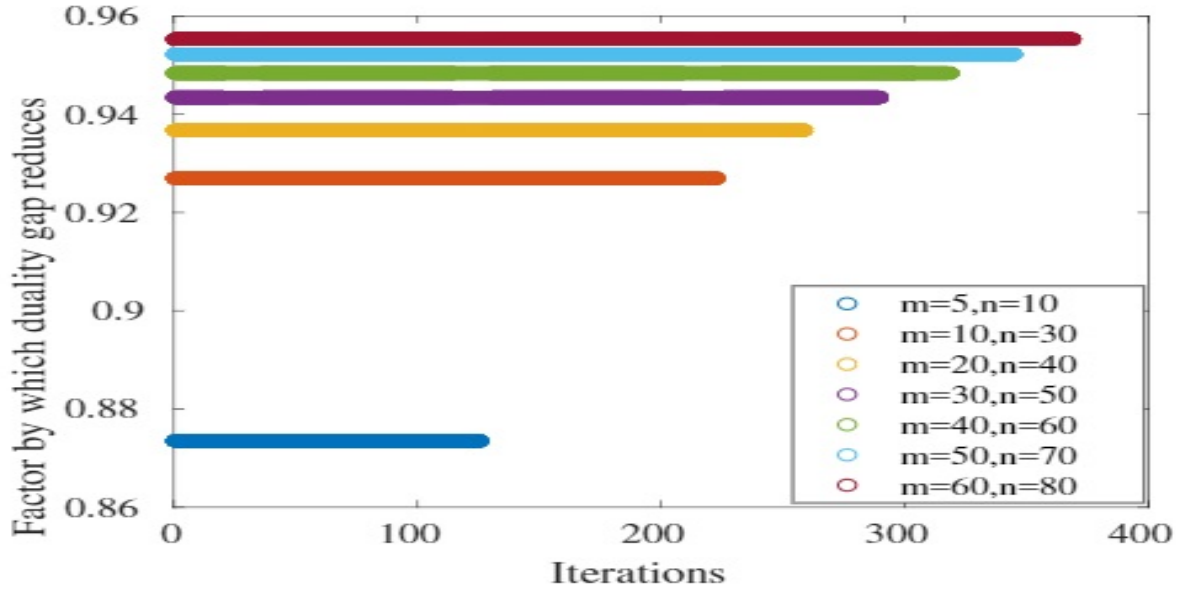
The table shows the values of $\sigma = 1 - \frac{0.4}{\sqrt{n}}$.

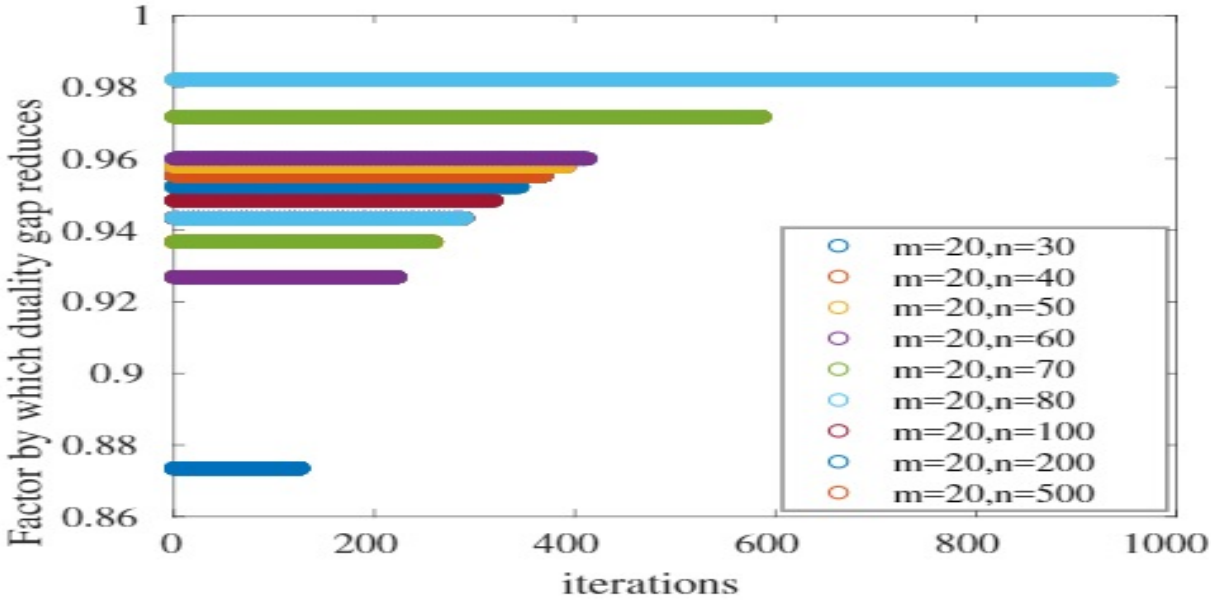
This is the factor which we would expect our iterates to have their averages $\frac{1}{n}x^{k+1T}u^{k+1}$ reduced from the average $\frac{1}{n}x^{kT}u^{k}$. We validate this theoretical assertion now.

We solve LP instances for constant value of $m = 20$ and increasing values of $n = 30, \ldots, 500$. We then solve instances of LP with increasing values of $m = 5, \ldots, 60$ and increasing values of $n = 10, \ldots, 80$. As the value of n increases, the number of iterations go up as seen on the $x-$ axis. On the $y-$ axis we see that for given n we have constant value of reduction in the average, equal to the expected reduction as shown in the table earlier.

number of variables (n)	σ
10	0.87351
30	0.92697
40	0.93675
50	0.94343
60	0.94836
70	0.95219
80	0.95528

Table 1.7. **n vs** σ



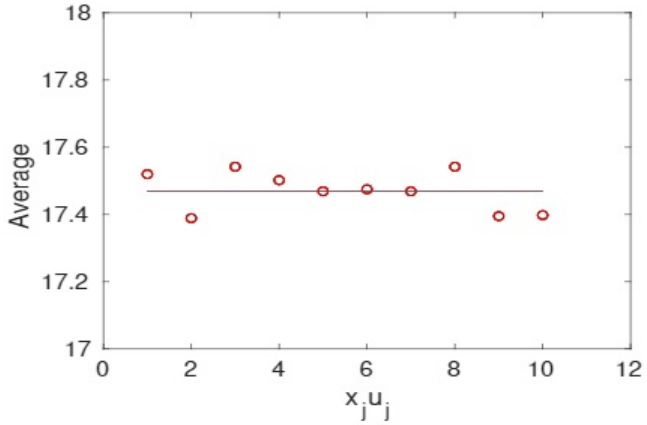


We start on the central path, but since we do not completely solve the perturbed system the theoretical assertion is that we do not stay on the central path. We have shown that the average drop observed computationally matches the theory. **We now show that even though the average drop is exactly as what we would expect had we stayed on the central path, the iterates are such that $x_j^{k+1}u_j^{k+1} = \sigma\mu^k$ does not hold true for all j.** Instead it is $\frac{1}{n}x_j^{k+1T}u_j^{k+1}$ **which equals** $\sigma\mu^k$. And thus the assertion that we do not stay on the central path but stay 'close' to it has been shown in the following example.

We consider a randomly generated instance of LP with $m = 5$ and $n = 10$ such that the starting point is on the central path. Start with an average $\mu^0 = 20$ and consider one iteration.

$\mu(\omega^0) = 20$	$x_j^0u_j^0$	$x_j^1u_j^1$	$\mu(\omega^1)$	σ
20	20	17.520	17.470	0.87351
	20	17.389	17.470	
	20	17.542	17.470	
	20	17.502	17.470	
	20	17.469	17.470	
	20	17.475	17.470	
	20	17.469	17.470	
	20	17.542	17.470	
	20	17.395	17.470	
	20	17.398	17.470	

Table 1.8. **Iterates not on the central path**



We end this discussion with some comments on the difference between the classical Simplex Method and Interior Point Methods [8] For reference consider the KKT conditions associated with a linear programming problem.

KKT Conditions

$$\begin{array}{c} A^Tv + u = c \\ Ax = b \\ x_ju_j = 0,\ j = 1,\ldots,n \\ (x,u) \geq 0 \end{array}$$

Simplex algorithm (G. Dantzig, 1948) is one of the most popular algorithms in the history of computa-

tional mathematics across various disciplines. Interior point methods maintain the KKT conditions 1, 2 and 4 and try to satisfy the condition 3 in the limit. Simplex method maintains conditions 1, 2 and 3 throughout the process while trying to find the optimal vertex in the correct orthant.

Bibliography

[1] Jorge Noecedal, Stephen J. Wright. *Numerical Optimization, second edition*. Springer.

[2] Mokhtar S. Bazaraa, Hanif D. Sherali, C.M.Shetty, *Nonlinear Programming: Theory and Algorithms 3rd Edition*. John Wiley and Sons, Inc, 2006.

[3] Anthony V. Fiacco, Garth P. Mccormick *Nonlinear programming, Sequential Unconstrained Minimization Techniques*. SIAM, Classics in Applied Mathematics, 1990.

[4] Anders Forsgreen, Philip Gill and Margaret Wright. `Interior Point Methods for Nonlinear Optimization` SIAM Review, Vol 44, No 4, pp-525-597.

[5] Stephen J. Wright `Primal Dual Interior Point Methods` SIAM

[6] Stephen Boyd and Lieven Vanderberghe `Convex Optimization` Cambridge University Press

[7] Ryan Tibshirani
`https://www.youtube.com/watch?v=s8-O28F-YE4`.

[8] Ryan Tibshirani
`https://www.youtube.com/watch?v=UfPvb`$_K$78*ik*

[9] Rodrigo Eustaquio, Elizabeth Karas and Ademir Alves Ribeiro
`Constraint qualifications for nonlinear programming`

[10] F. J. Gould and Jon W. Tolle
`A necessary and sufficient qualification for constrained optimization`, SIAM Journal on Applied Mathematics, 20:164-172, 1971

[11] J. Abadie
`On the Kuhn Tucker theorem`,Nonlinear programming, pages 19-36. North Holland, Amsterdam, 1967.

[12] M. Guinard.
`Generalized Kuhn Tucker conditins for mathematical programming problems in a Banach space`,SIAM Journal on Control, 7:232-241,1969.

[13] O Mangasarian and S Fromovitz
`The Fritz John necessary optimality conditions in the presence of equality and inequality constraints`,J. Math.Anal. Appl., 17(1967), pg 37-47.

[14] H. KUHN AND A. TUCKER,
Nonlinear programming,Proc. Second Berkeley Symposium on Mathematical Statistics and Probability, J. Neyman, ed., University of California Press, Berkeley, 1951, pp. 481-492.

[15] Maziar Sanjabi (https://math.stackexchange.com/users/479764/maziar-sanjabi), Image for a strict local minimizer that is not an isolated minimizer, URL (version: 2017-09-13): https://math.stackexchange.com/q/2428496

Appendix

Newton's Method

Consider the unconstrained minimization problem

$$\text{minimize}_x \quad f(x)$$

with $x \in \mathbb{R}^n$. Assume that $f \in \mathcal{C}^2$.

The intuitive idea behind the Newton's method is to consider the second order (quadratic) approximation of the function $f(x)$ at each subsequent iterate x^k and find its minimum to obtain x^{k+1}. Finding the minimum of a quadratic function is really the generalization of finding the minimum of a one variable quadratic function $ax^2 + bx + c$ which occurs at $x = \frac{-b}{2a}$ assuming $a > 0$. For a quadratic function $f(x) = \frac{1}{2}x^T Hx + b^T x$, assuming the matrix H which is the hessian of the given function is positive definite, the minimum occurs at $\bar{x} = H^{-1}b$.

Use Taylor's series to approximate f at x^k by a quadratic function as

$$f(x) \approx f_q(x) = f(x^k) + g^{k^T}(x - x^k) + \frac{1}{2}(x - x^k)^T H^k (x - x^k) \tag{1.124}$$

where $g^k = \nabla f(x^k)$ and $H^k = \nabla^2 f(x^k)$. Then we have

$$x^{k+1} = arg\, min_x\, f_q(x) \tag{1.125}$$

The stationary point can be found as

$$\nabla f_q(x) = H^k(x - x^k) + g^k = 0 \tag{1.126}$$

$$\Rightarrow x = x^k - (H^k)^{-1}g^k = x^{k+1} \tag{1.127}$$

The point $x = x^{k+1}$ is a local minimum provided H^k is a positive definite matrix.

Thus the classical Newton's method consists of taking steps of the form $x^{k+1} = x^k - \alpha^k d^k$ where $d^k = (H^k)^{-1}g^k$ and $\alpha^k = 1$. Note that α^k can also be found using backtracking line search []. There are however certain difficulties associated with the classical Newton's method. The hessian at each iteration may not necessarily be invertible. Further, it is possible that the hessian may be negative invertible in which case the direction d^k may not be a descent direction. In order to avoid such difficulties, a modified version of the Newton's method is used which is summarized next.

Modified Newton's Method. Let x^k be the current iterate and $d^k_{PN} = -(H^k)^{-1}g^k$ be the pure Newton direction. Find the smallest $\tau_k \geq 0$ such that the smallest eigen value of the matrix $(H^k + \tau_k I)$ is positive. Thus the matrix $(H^k + \tau_k I)$ is positive definite and the direction $d^k = (H^k + \tau_k I)^{-1}g^k$ is a descent direction.

The τ_k can be found using a modified Cholesky factorization algorithm as the one given in [] which yields both τ_k and the Cholesky factorization such that $L^k L^{k^T} = H^k + \tau_k I$. Given the Cholesky factorization the direction d^k can be found by simple forward and backward substitution. Given x^k and d^k, x^{k+1} can be found as $x^{k+1} = x^k + \alpha^k d^k$ where α^k can be using backtracking line search.