volume per month. Neither refined oil nor blended cooking oil can be stored. The hardness of the cooking oil must be between $H_{\min}$ and $H_{\max}$. The hardness of each raw oil i is H_i, and hardness blends linearly. Presently the volume of raw oil i in storage is S_i, and the amount of each oil in storage at the end of T months should again be S_i. Formulate an LP model to determine the manufacturing plan for each month, over the next T months, that maximizes net income (revenue minus holding cost).

8 Primal-Dual Interior Point Methods

An *interior-point method* moves through the interior of the feasible set, rather than along the boundary as in the simplex method. Leonid Khachiyan was the first to show (in 1979) that an interior-point method for LP can have polynomial worst-case complexity. He showed this for his *ellipsoid algorithm*, which unfortunately is impractical because it requires very high-precision calculations.

The first practical polynomial-time interior-point method is due to Karmarkar (1984). It is a primal method based on the idea of distorting the solution space in each iteration with a projective scaling transformation that moves the current iterate to the center of the feasible set. A steepest descent direction is then selected in the transformed space. This allows one to take a longer step in each iteration before hitting the boundary of the feasible set, perhaps resulting in many fewer iterations than the simplex method.

Following Karmarkar's result, a number of investigators examined the connection between his method and known nonlinear programming methods. It was first shown to be a Newton barrier method that uses a well-chosen barrier paramater. Subsequent research reformulated projective scaling methods as primal-dual methods that solve the Kuhn-Tucker-Karush conditions for primal and dual optimality. *Path-following* methods accomplish this by maintaining primal and dual feasibility and working toward complementary slackness.

Path-following methods mimic the effect of projective scaling by modifying a Newton direction to place the next iterate nearer the center of the feasible set. If there is proper balance between a pure Newton direction (which achieves a maximum rate of improvement) and a centering direction (which allows a longer step in the next iteration), these methods achieve polynomial worst-case complexity.

We will briefly describe two path-following methods and show how polynomial complexity is proved.

8.1 Primal-Dual Formulation

We will address the LP

$$\begin{aligned} &\min\ c^T x \\ &Ax = b,\ \ x \geq 0 \end{aligned} \tag{46}$$

It is convenient to write the dual with slack variables:

$$\begin{aligned} &\max\ b^T u \\ &A^T u + s = c,\ \ s \geq 0 \end{aligned} \tag{47}$$

We first review a necessary and sufficient condition for the optimality of x.

Corollary 17 *A feasible solution x is optimal for (46) if and only if there is a feasible solution (u, s) of (47) satisfying the complementary slackness condition $s^T x = 0$.*

Proof. Suppose x is optimal. Then by strong duality, there is an optimal solution (u, s) of the dual that satisfies $c^T x = u^T b$. But we have $c^T x \geq u^T Ax = u^T b$ because $x \geq 0$, $u^T A \leq c^T$ and $Ax = b$. So $c^T x = u^T Ax$, which implies $(c^T - u^T A)x = s^T x = 0$. For the converse, suppose x is primal feasible, (u, s) is dual feasible, and $s^T x = 0$. Then $(c^T - u^T A)x = 0$, which implies $c^T x = u^T Ax = u^T b$. Due to weak duality, x and (u, s) are optimal in the primal and dual, respectively. □

We can therefore solve (46) by solving the following primal-dual system:

$$\begin{aligned} A^T u + s &= c \\ Ax &= b \\ x^T s &= 0 \\ x, s &\geq 0 \end{aligned} \tag{48}$$

We will solve the system by a modified Newton's method that stays strictly inside the feasible set by making sure that $x, s > 0$ at all times. This results in an *interior point* method because the iterates are always interior to the feasible set. Because the optimal solution is on the bounary, the method terminates within a specified tolerance of the optimal point rather than exactly at the optimal point.

8.2 Newton Direction

Newton's method solves a system of equations $F(x) = 0$ by creating a first-order approximation of the function $F(x)$ at the current iterate x:

$$F(x + \Delta x) \approx F(x) + \nabla F(x)\Delta x$$

where the *gradient* $\nabla F(x)$ is the row vector of partial derivatives of $F(x)$ evaluated at x. It then sets this approximation equal to zero and solves the equations

$$\nabla F(x)\Delta x = -F(x) \tag{49}$$

for the *Newton direction* Δx. The next iterate is obtained by moving in this direction:

$$x^{\text{new}} = x + \alpha \Delta x$$

where α is the step size. The choice of step size varies with the particular version of Newton's method used.

This method is applied to (48) by writing the system in the form

$$\begin{aligned} F(x,u,s) &= \begin{bmatrix} A^T u + s - c \\ Ax - b \\ XSe \end{bmatrix} = 0 \\ x, s &\geq 0 \end{aligned} \tag{50}$$

It is common practice in the interior point literature to write $x^T s = 0$ as $XSe = 0$, where $X = \text{diag}(x_1, \ldots, x_n)$, $S = \text{diag}(s_1, \ldots, s_n)$, and $e = (1, \ldots, 1)$. The equation (49) becomes

$$\nabla F(x,u,s) \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe \end{bmatrix} \tag{51}$$

where $\nabla F(x,u,s)$ is the *Jacobian* of $F(x,u,s)$. So if (x,u,s) is the current iterate, the next iterate is

$$(x^{\text{new}}, u^{\text{new}}, s^{\text{new}}) = (x,u,s) + \alpha(\Delta x, \Delta u, \Delta x)$$

The method terminates when we satisfy the optimality conditions within a given tolerance.

Unfortunately, the Newton direction tends to hit the boundary of the feasible set before we can move very far. For this reason, the Newton direction is modified to take us nearer the center of the feasible set, by approximately following the *central path*.

8.3 Central Path

Rather than solve the optimality conditions (50) we will solve a modified system that does not require strict complementary slackness:

$$\begin{aligned} F(x,u,s) &= \begin{bmatrix} A^T u + s - c \\ Ax - b \\ XSe \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau e \end{bmatrix} \\ x, s &\geq 0 \end{aligned} \tag{52}$$

It can be shown that this system has a unique solution for any given $\tau > 0$ (if the interior of the feasible set is nonempty). If (x_τ, u_τ, s_τ) is the solution for a given τ, the path paramaterized by τ is the *central path*. More precisely, the central path is $\{(x_\tau, u_\tau, s_\tau) \mid \tau > 0\}$.

Our strategy is to modify the Newton directions to follow approximately the central path. This allows us to take longer steps in each iteration and perhaps reach a near-optimal solution in fewer iterations. The parameter τ will tend to zero, at which point we have complemenatry slackness, and the original problem is solved. We therefore maintain primal and dual feasibility while striving for complementary slackness, which results in a primal-dual method.

We will adjust τ as we go along by writing it $\tau = \sigma\mu$, where σ is the *centering parameter* and μ measures the distance from complementary slackness. Specifically,

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_j s_j = x^T s / n$$

where (x, u, s) is the current iterate. Thus $\mu = 0$ when complementary slackness is achieved. The equations (51) therefore become

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -XSe + \sigma\mu e \end{bmatrix} \tag{53}$$

The centering paramater $\sigma = 0$ when we take the Newton direction. This results in an *affine scaling* algorithm, for reasons that cannot be explained here. The paramater is $\sigma = 1$ when we take a *centering direction*, which is the Newton direction toward the point where all the products $x_j s_j$ are equal to μ. Movement in the centering direction makes little progress toward reducing μ in the current iteration. However, by moving us toward the center of the feasible set, it may allow us to take a long improving step in the next iteration. Primal-dual algorithms find some reasonable compromise between $\sigma = 0$ and $\sigma = 1$.

8.4 Path-Following Methods

Path-following methods typically keep the iterates within a specified neighborhood of the central path. If this neighborhood and the stepsize are defined properly, the methods have polynomial worst-case complexity.

We first develop a *short-step* method that uses a neighborhood defined by the Euclidean norm $||v||_2 = (\sum_j v_j^2)^{1/2}$. We then summarize a *long-step* method that uses an L_∞ norm.

In the short-step method, the diameter of the neighborhood is parameterized by $\theta \in (0, 1]$. Recall that the central path is defined by the fact that, at each point on the path, all of the products $x_j s_j$ are equal to μ. Distance from the path can therefore be measured by the Euclidean norm of the deviations from μ. So the neighborhood for a given θ is

$$\mathcal{N}_2(\theta) = \{(x, u, s) \in \mathcal{S}_0 \mid ||XSe - \mu e||_2 \leq \theta\mu\}$$

where $\mathcal{S}_0$ is the interior of the feasible set.

The short-step method uses a step size of $\alpha = 1$ and a constant centering parameter $\sigma \in (0, 1)$. It begins with a point in $\mathcal{N}_2(\theta)$. If (x, u, s) is the current iterate, then the next iterate is

$$(x^{\text{new}}, u^{\text{new}}, s^{\text{new}}) = (x, u, s) + (\Delta x, \Delta u, \Delta s)$$

where $(\Delta x, \Delta u, \Delta s)$ solves (53). If we set $\theta = 0.4$ and $\sigma = 1 - 0.4/\sqrt{n}$, it can be shown that each iterate of the short-step method stays within $\mathcal{N}_2(0.4)$. We will show here only that the method converges linearly to the solution and derive the convergence rate. In the next section, we will prove polynomial complexity.

The convergence rate is based on a general result about directions that solve (53). For convenience, given a step size α let

$$\begin{aligned}(x(\alpha), u(\alpha), s(\alpha)) &= (x, u, s) + \alpha(\Delta x, \Delta u, \Delta s)\\ \mu(\alpha) &= x(\alpha)^T s(\alpha)/n\end{aligned}$$

Lemma 18 *If the direction $(\Delta x, \Delta x, \Delta s)$ is defined by (53), then we have $\Delta x^T \Delta s = 0$ and*

$$\mu(\alpha) = (1 - \alpha(1 - \sigma))\mu \tag{54}$$

Proof. $\Delta x^T \Delta s = -\Delta x^T A^T \Delta u = 0$, where the first equation is due to the first row of (53) and the second equation to the second row. To show (54), we have from the third row of (53) that

$$e^T S\Delta x + e^T X\Delta s = -e^T XSe + \sigma\mu e^T e$$

Using the fact that $\mu = x^T s/n$, this yields

$$s^T \Delta x + x^T \Delta a = -x^T s + \sigma x^T s = -(1-\sigma) x^T s \tag{55}$$

Now we have

$$x(\alpha)^T s(\alpha) = (x + \alpha \Delta x)^T (s + \alpha \Delta s) = x^T s + \alpha (s^T \Delta x + x^T \Delta s) + \alpha^2 \Delta x^T \Delta s$$

Due to (55) and the fact that $\Delta x^T \Delta s = 0$, this yields

$$x(\alpha)^T s(\alpha) = (1 - \alpha(1-\sigma)) x^T s$$

which proves the lemma. □

Because $\alpha = 1$ in the short-step algorithm, we have from Lemma 18 that

$$\mu^{\text{new}} = (1 - \alpha(1-\sigma))\mu = \sigma\mu = \left(1 - \frac{0.4}{\sqrt{n}}\right)\mu \tag{56}$$

Convergence is therefore linear with rate $1 - 0.4/\sqrt{n}$. This is enough to establish polynomial complexity (next section).

The long-step method uses the neighborhood

$$\mathcal{N}_{-\infty}(\gamma) = \{(x, u, s) \in \mathcal{S}_0 \mid x_j s_j \geq \gamma\mu, \text{ all } j\}$$

where γ is small, something like 0.001. While $\mathcal{N}_2$ keeps the iterates close to the central path, $\mathcal{N}_{-\infty}$ is much larger, allowing iterates to move quite close to the boundary. This allows us to take longer steps, thus perhaps solving the problem more quickly. However, the provable convergence rate is less attractive.

Rather than use a constant step size $\alpha = 1$, the long-step method takes α to be as large as possible while keeping the next iterate inside $\mathcal{N}_{-\infty}$. The centering parameter σ is not crucial, so long as it remains inside a fixed interval within $(0, 1)$. Given this, it can be shown that

$$\mu^{\text{new}} \leq \left(1 - \frac{\delta}{n}\right)\mu$$

where δ is a constant independent of n. This can be compared with (56). The thoretically provable convergence rate slows more rapidly as n increases than does the rate for the short-step method.

A third type of algorothm is a *predictor-corrector* method that alternates between $\sigma = 0$ and $\sigma = 1$. It can be based on either an $\mathcal{N}_2$ or an $\mathcal{N}_{-\infty}$ neighborhood. Practical algorigthms tend to be modified predictor-corrector methods with well-chosen heuristics for selecting σ, α, and other parameters.

8.5 Polynomial Complexity

The complexity results depend on a basic lemma. Let the iterates be (x^k, u^k, s^k) for $k = 0, 1, \ldots$, and let $\mu_k = (x^k)^T s^k / n$.

Lemma 19 *For a given $\epsilon > 0$, suppose that*

$$\mu_{k+1} \leq \left(1 - \frac{\delta}{n^\omega}\right) \mu_k, \quad \textit{all } k \tag{57}$$

for some $\delta, \omega > 0$, and that

$$\mu_0 \leq 1/e^\kappa \tag{58}$$

for some $\kappa > 0$. Then there exists

$$K = \mathcal{O}\left(n^\omega |\log \epsilon|\right)$$

such that $\mu_k \leq \epsilon$ for all $k \geq K$.

Proof. Taking logs in (57), we get

$$\log \mu_{k+1} \leq \log \left(1 - \frac{\delta}{n^\omega}\right) + \log \mu_k$$

Applying the formula recursively, we get

$$\log \mu_{k+1} \leq k \log \left(1 - \frac{\delta}{n^\omega}\right) + \log \mu_0 \leq k \log \left(1 - \frac{\delta}{n^\omega}\right) + \kappa \log \frac{1}{\epsilon}$$

where the second inequality is from (58). Because $\log(1+r) \leq r$ for $r > -1$, this yields

$$\log \mu_{k+1} \leq k \left(-\frac{\delta}{n^\omega}\right) + \kappa \log \frac{1}{\epsilon}$$

So we have the desired result $\mu_k \leq \epsilon$ if

$$k \left(-\frac{\delta}{n^\omega}\right) + \kappa \log \frac{1}{\epsilon} \leq \log \epsilon$$

But this holds if

$$k \geq K = (1 + \kappa) \frac{n^\omega}{\delta} \log \frac{1}{\epsilon}$$

and the lemma follows. □

The convergence rate (56) for the short-step method allows us to apply Lemma 19 with $\delta = 0.4$ and $\omega = 1/2$. We can apply the lemma to the long-step method with $\omega = 1$.

Theorem 20 *The short-step and long-step primal-dual methods have polynomial worst-case complexity.*