Chapter 24
Pivotal Interior-Point Method

The simplex method and the interior-point method are two diverging and competitive types of methods for solving LP problems. The former moves on the underlying polyhedron, from vertex to adjacent vertex, along edges until an optimal vertex is reached while the latter approaches an optimal point by moving across interior of the polyhedron.

Although the basic ideas, motivations and development tracks of the two methods appear quite different, attempts will be made in this chapter to combine the two methods to take advantages of both, in a natural manner.

In view of that the interior-point method has been seriously restricted in applications since it can not be “warmly started”, and provides only an approximate optimal solution (see Sect. 9.5), three pivotal interior-point algorithms will be derived. These algorithms yield from standard interior-point algorithms by adding inner steps. Presented in the last section of this chapter, on the other hand, the so-called “feasible-point simplex algorithm” is established along another line, which might be the first simplex-like algorithm that acrosses the interior of the polyhedron.

24.1 Pivotal Affine Interior-Point Method

In this section, firstly an interior-point algorithms is designed by forming a search direction based on affine-scaling. This search direction is equivalent to that used in Dikin’s affine algorithm in theory. However, the new framework allows to introduce inner pivotal steps to create a better interior-point algorithm.

Let $\bar{x}$ be the current interior-point. Consider the dual problem of (9.25), i.e.,

$$\begin{array}{ll} \max & g = b^{\mathrm{T}} y, \\ \text{s.t.} & (\bar{X} A^{\mathrm{T}} \vdots I) \begin{pmatrix} y \\ z \end{pmatrix} = \bar{X} c, \; z \geq 0. \end{array} \tag{24.1}$$

P.-Q. PAN, *Linear Programming Computation*, DOI 10.1007/978-3-642-40754-3_24,

Note that there is a 1-to-1 correspondence between columns of the unit matrix I and indices of z.

At first, we will realize the "dual elimination" by orthogonal transformations (Sect. 25.1.3). Since A is of full row rank, hence $\bar{X}A^{\mathrm{T}}$ is of full column rank, there exists the QR factorization

$$\bar{X}A^{\mathrm{T}} = (Q_1, Q_3)\begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1 R_1, \tag{24.2}$$

where (Q_1, Q_3) is orthogonal, partitioned as $Q_1 \in \mathcal{R}^{n\times m}$ and $Q_3 \in \mathcal{R}^{n\times(n-m)}$, and $R_1 \in \mathcal{R}^{m\times m}$ is nonsingular upper triangular.

The following result reveals that the search direction in x'-space, used in the Dikin's affine algorithm, can be obtained alternatively by using the matrix Q_3.

Proposition 24.1.1. *$\Delta x'$, defined by (9.27), is equal to $Q_3 Q_3^{\mathrm{T}}\bar{X}c$.*

Proof. Substituting (24.2) to (9.27) and noting $Q_1^{\mathrm{T}}Q_1 = I$ and $Q_1Q_1^{\mathrm{T}} + Q_3Q_3^{\mathrm{T}} = I$ gives

$$\Delta x' = (I - Q_1R_1(R_1^{\mathrm{T}}Q_1^{\mathrm{T}}Q_1R_1)^{-1}R_1^{\mathrm{T}}Q_1^{\mathrm{T}})\bar{X}c = (I - Q_1Q_1^{\mathrm{T}})\bar{X}c = Q_3(Q_3^{\mathrm{T}}\bar{X}c). \tag{24.3}$$

□

Thereby, we are led to the following variant of the affine algorithm.

Algorithm 24.1.1 (Variant of Algorithm 9.2.1). The same as Algorithm 9.2.1, except for its step 1 replaced by

1. Compute $\Delta x'$ by (24.3).

As they differ only in the way to compute the same search direction, Algorithm 24.1.1 and Dikin's algorithm are equivalent. The computational efforts involved in them depend on how to implement, the sparsity of A, and the number $n - m$, compared with m, and etc. We will not go into details here because, after all, what we are really interested in is not the algorithm itself but a variant, as derived as follows.

As the affine method with long step turned out to be superior to that with short step in practice, it is attractive to go further along this line by introducing inner pivotal steps to decrease the objective value as much as possible, with reasonable costs.

We begin with premultiplying the augmented matrix of the equality constraints of (24.1) by $Q^{\mathrm{T}} = [Q_1, Q_3]^{\mathrm{T}}$. Such doing leads to a so-called *triangular form*, i.e.,

$$Q^{\mathrm{T}}(\bar{X}A^{\mathrm{T}} \vdots I \mid \bar{X}c) = \begin{pmatrix} R_1 & Q_1^{\mathrm{T}} & \mid Q_1^{\mathrm{T}}\bar{X}c \\ 0 & Q_3^{\mathrm{T}} & \mid Q_3^{\mathrm{T}}\bar{X}c \end{pmatrix}, \tag{24.4}$$

which represents the linear system equivalent to the dual equality constraints. Based on Proposition 24.1.1, it is known that the south-east submatrix $(Q_3^{\mathrm{T}} \mid Q_3^{\mathrm{T}}\bar{X}c)$ gives the projection $\Delta x'$, defined by (24.3), i.e.,

$$\Delta x' = (I - Q_1R_1(R_1^{\mathrm{T}}Q_1^{\mathrm{T}}Q_1R_1)^{-1}R_1^{\mathrm{T}}Q_1^{\mathrm{T}})\bar{X}c = (I - Q_1Q_1^{\mathrm{T}})\bar{X}c = Q_3(Q_3^{\mathrm{T}}\bar{X}c). \tag{24.5}$$

Then, update $\bar{x}$ by (9.30), i.e.,

$$\hat{x} = \bar{x} - \lambda\bar{X}\Delta x'/\max(\Delta x'), \tag{24.6}$$

where $\lambda \in (0, 1)$ a *stepsize*. If it goes to the next iteration at this point, the resulting is just the same as Algorithm 24.1.1.

On the contrary, we will carry out a series of inner iterations in x'-space, starting from

$$N = \emptyset, \quad B = A$$

Assume that $\Delta x'_B = \Delta x' \not\leq 0$. Update the $\bar{x}'_B$ by the following formula:

$$\hat{x}'_B = e - \lambda\Delta x'_B/\max(\Delta x'_B),$$

and determine an index q such that

$$q = \arg\max\{\Delta x'_j \mid j = 1, \cdots, n\}.$$

It is not difficult to show that $(\bar{X}A^{\mathrm{T}} \vdots e_q)$ is of full column rank. If the QR factorization of it is available, then the orthogonal projection of the objective gradient $\bar{X}c$ onto the null space of

$$\begin{pmatrix} A\bar{X} \\ e_q^{\mathrm{T}} \end{pmatrix}$$

can be computed analogously as before, and it is thereby able to update the solution once more in x'-space. Note that the q-indexed component of the projection equals 0, hence the q-indexed component of the solution remains unchanged.

Assume that after $k < n - m$ inner iterations, there are index sets

$$N = \{1, \cdots, k\}, \quad B = \{k+1, \cdots, n\}.$$

Let the QR factorization $(\bar{X}A^{\mathrm{T}} \vdots I_N) = QR$ be available. Premultiplying by Q^{T} the augmented matrix of equality constraints of (24.1) gives

$$\begin{aligned} & Q^{\mathrm{T}}(\bar{X}A^{\mathrm{T}} \vdots I_N \vdots I_B|\bar{X}c) \\ &= \begin{pmatrix} Q_1^{\mathrm{T}}\bar{X}A^{\mathrm{T}} & Q_1^{\mathrm{T}}I_N & Q_1^{\mathrm{T}}I_B & | \ Q_1^{\mathrm{T}}\bar{X}c \\ Q_2^{\mathrm{T}}\bar{X}A^{\mathrm{T}} & Q_2^{\mathrm{T}}I_N & Q_2^{\mathrm{T}}I_B & | \ Q_2^{\mathrm{T}}\bar{X}c \\ Q_3^{\mathrm{T}}\bar{X}A^{\mathrm{T}} & Q_3^{\mathrm{T}}I_N & Q_3^{\mathrm{T}}I_B & | \ Q_3^{\mathrm{T}}\bar{X}c \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & Q_1^{\mathrm{T}}I_B & | \ Q_1^{\mathrm{T}}\bar{X}c \\ 0 & R_{22} & Q_2^{\mathrm{T}}I_B & | \ Q_2^{\mathrm{T}}\bar{X}c \\ 0 & 0 & Q_3^{\mathrm{T}}I_B & | \ Q_3^{\mathrm{T}}\bar{X}c \end{pmatrix}. \end{aligned} \tag{24.7}$$

where the orthogonal matrix $Q = (Q_1, Q_2, Q_3)$ is partitioned as $Q_1 \in \mathcal{R}^{n\times m}$, $Q_2 \in \mathcal{R}^{n\times k}$ and $Q_3 \in \mathcal{R}^{n\times(n-m-k)}$, and $R_{11} \in \mathcal{R}^{(m+k)\times(m+k)}$ is nonsingular upper triangular. The preceding is the kth triangular form, whose south-east submatrix gives projection

$$\Delta x'_B = (Q_3^{\mathrm{T}} I_B)^{\mathrm{T}}(Q_3^{\mathrm{T}} \bar{X} c). \tag{24.8}$$

Assume that $\Delta x'_B \not\leq 0$. Since $\Delta x'_N = (Q_3^{\mathrm{T}} I_N)^{\mathrm{T}}(Q_3^{\mathrm{T}} \bar{X} c) = 0$ and $\bar{x}'_N$ remains unchanged, what only needs to do is updating by

$$\hat{x}'_B = \bar{x}'_B - \lambda\alpha\Delta x'_B. \tag{24.9}$$

where

$$\alpha = \bar{x}'_q/\Delta x'_q = \min\{\bar{x}'_j/\Delta x'_j \mid \Delta x'_j > 0,\ j \in B\}, \tag{24.10}$$

which is the stepsize from the current solution to the nearest boundary. It is clear that the new solution is again an interior point. Then move q from B to N, and go to the $(k+1)$th inner iteration.

The forgoing process terminates when $k = n-m$ or $Q_3^{\mathrm{T}} \bar{X} c = 0$, hence the $\Delta x'_B$ defined by (24.8) vanishes. In this case, substituting $\bar{z}_B = 0$ to the dual equality constraints, represented by (24.7), leads to the upper triangular system

$$\begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \begin{pmatrix} y \\ z_N \end{pmatrix} = \begin{pmatrix} Q_1^{\mathrm{T}} \bar{X} c \\ Q_2^{\mathrm{T}} \bar{X} c \end{pmatrix}. \tag{24.11}$$

Assume that $(\bar{y}, \bar{z}_N)$ is the solution to this system.

If $\bar{z}_N \geq 0$, it is not difficult to show that $\bar{x}$ is the optimal solution to the following problem:

$$\begin{array}{ll} \min & c^{\mathrm{T}} x, \\ \text{s.t.} & Ax = b, \\ & x_B \geq 0, \quad x_N \geq \bar{x}_N. \end{array}$$

If $\lambda \in (0, 1)$ is sufficiently close to 1, then $\bar{x}_N$ can be arbitrarily close to 0, in principle. If λ is predetermined to be close to 1, therefore, $\bar{x}$ can be regarded as an approximate optimal solution to the original problem, and the solution process terminates.

In the other case when $\bar{z}_N \not\geq 0$, the inner iterations are finished. The related solution in the original space is computed by

$$\bar{x} = \bar{X}\bar{x}',$$

and a new $\bar{X}$ can be formed to be ready to go to the next outer iteration.

The trick of the algorithm lies in that the QR factors in each inner iteration can be obtained via recurrence, need not to compute from scratch. In fact, the last $n-m-k-1$ components of the q-indexed column of (24.7) can be eliminated by Givens rotations. Assume that $\tilde{Q} \in \mathcal{R}^{(n-m-k)\times(n-m-k)}$ is the product of Givens rotations such that

$$\tilde{Q} Q_3^{\mathrm{T}} e_q = \eta e_1,$$

then $\hat{Q}^{\mathrm{T}} = [I \ \vdots\ \hat{Q}^{\mathrm{T}}]^{\mathrm{T}} Q^{\mathrm{T}}$ is just the wanted factor for the $(k+1)$th inner iteration. It is then seen that the $(k+1)$ and kth triangular forms are the same, except for the submatrix, associated with Q_3. On the other hand, nevertheless, at the beginning of each round of outer iterations, it is necessary to compute the QR factors from scratch since $\bar{X}$ changed.

The overall steps can be summarized to the following algorithm.

Algorithm 24.1.2 (Pivotal affine interior-point algorithm). Given $\lambda \in (0, 1)$. Initial: interior point $\bar{x} > 0$. This algorithm solves the standard LP problem.

1. Set $k = 0$, and compute triangular form (24.4).
2. Compute $\Delta x' = Q_3(Q_3^{\mathrm{T}} \bar{X} c)$.
3. Stop if $\Delta x' \leq 0$ (lower unbounded).
4. Determine α and q such that
   $\alpha = \bar{x}'_q / \Delta x'_q = \min\{\bar{x}'_j / \Delta x'_j \mid \Delta x'_j > 0, j \in B\}$.
5. If $\alpha \neq 0$, update: $\bar{x}'_B = \bar{x}'_B - \lambda\alpha\Delta x'_B$ (24.9).
6. Set $k = k + 1$, and update (B, N) by bringing q from B to N.
7. Go to step 10 if $k = n - m$ or $Q_3^{\mathrm{T}} \bar{X} c = 0$.
8. Eliminate the $(m + k + 1)$ to nth components of the q-indexed column of the triangular form by Givens rotations.
9. Go to step 2.
10. Solve the upper triangular system (24.11).
11. Stop if $\bar{z}_N \geq 0$ (approximate optimality achieved).
12. Set $\bar{x} = \bar{X}\bar{x}'$.
13. Go to step 1.

Note This Algorithm contains steps 2–9 as its inner steps.

The algorithm, developed by Pan (2013) is slightly different from the preceding algorithm, as the former uses update

$$\hat{x}'_B = \bar{x}'_B - \alpha\Delta x'_B$$

rather than (24.9). Thereby, the resulting iterate is not interior but boundary point. If the optimality condition is not satisfied after a round of inner iterations finished, it goes back to a nearby interior point to start the next round of outer iterations as follows.

Assume that $\bar{x}$ is the interior point at the beginning of the outer iteration, and $\hat{x}$ is the end boundary point of the inner iterations. The interior point used for the next outer iteration is determined by

$$\bar{x} = \bar{x} + \mu(\hat{x} - \bar{x}),$$

where $\mu \in (0, 1)$ ($\mu = 0.95$ is taken by Pan).

Table 24.1 Iteration and time ratios

	AIP/VAIP		AIP/PAIP	
Problem	Iterations	Time	Iterations	Time
Set 1 (16)	0.98	0.32	6.32	1.47
Set 2 (10)	0.92	0.25	3.55	1.52
Average (26)	0.95	0.26	4.56	1.52

There is no available numerical results with Algorithm 24.1.2. As it is close to Pan's original algorithm, we cite his numerical results to give the reader a clue on its performance.

The associated computational experiments were carried out on a Pentium III 550E PC with Windows 98 operating system, 168 MB inner storage and about 16 decimal precision. Visual Fortran 5.0 compiler was used. There were following three dense codes involved:

1. AIP: affine Algorithm 9.2.1.
2. VAIP: Algorithm 24.1.1.
3. PAIP: Algorithm 24.1.2.

The preceding codes are tested on the 26 smallest (by $m + n$) Netlib standard problems. The first set involves 16 smaller problems, and the second set are the rest 10 problems (Appendix B: Table B.4, problems AFIRO-DEGEN2).

Table 24.1 lists iterations and time ratios:

From the bottom line of the preceding Table, it is seen that total iteration and time ratios of AIP to VAIP are 0.95 and 0.26, respectively. As expected, the latter performs worse than the former. However, PAIP outperforms AIP significantly: the total iteration and time ratios of AIP to PAIP are 4.56 and 1.52, respectively. Therefore, the pivotal inner iterations appear effective.

It is not surprising that the time ratio of AIP to PAIP is much less than their iteration ratio (1.52 vs. 4.56), since each iteration of the latter is more time consuming, due to the use of the orthogonal transformation. Fortunately, the so-called "dual elimination" allows to use the Gaussian elimination instead (see Lemma 25.1.1). In particular, such doing is advantageous for sparse computations. On the other hand, of course, the associated search direction will no longer be the desired *orthogonal* projection. It is not known how such an algorithm will perform.

24.2 Pivotal Affine Face Interior-Point Method

In this section, two interior-point variants of the affine face method (Sect. 22.4) will be derived. Firstly, an interior-point variant is designed by directly using the initial search direction of the affine face method. Then, it is modified further by introducing pivotal inner iterations.

Consider reduced problem (22.1). Introduce notation

$$\bar{X} = \mathrm{diag}(\bar{x}_1, \cdots, \bar{x}_n, 1). \tag{24.12}$$

The initial search direction is defined by (22.30) with $k = n + 1$ or $B = A$, or denoted by

$$\Delta = -e_{n+1} + \bar{X}^2 A^{\mathrm{T}} \bar{y}, \quad (A\bar{X}^2 A^{\mathrm{T}})\bar{y} = -e_{m+1}. \tag{24.13}$$

which is downhill with respect to the objective function, as well as points to the interior of the feasible region. It is suitable to be a search direction for designing an interior-point algorithm. What is needed is just starting from an interior point and taking λ times of the original stepsize, where λ is a positive number less than 1 (e.g., 95–99 %).

Using the preceding notation, the overall steps can be put into the following interior-point algorithm.

Algorithm 24.2.1 (Affine face interior-point algorithm). Given $\lambda \in (0, 1)$. Initial: interior point $\bar{x} > 0$. This algorithm approximately solves the reduced problem (22.1).

1. Compute $A\bar{X}^2 A^T = LL^T$.
2. Solve $L^T \bar{y} = -(1/\nu)e_{m+1}$ for $\bar{y}$, where ν is the $(m + 1)$th diagonal of L.
3. Compute $\Delta = -e_{n+1} + \bar{X}^2 A^T \bar{y}$.
4. Stop if $J = \{j \in A \mid \Delta_j < 0\} = \emptyset$ (lower unbounded).
5. Determine $\alpha = \lambda \min_{j \in J} -\bar{x}_j/\Delta_j$.
6. Update: $\bar{x} = \bar{x} + \alpha\Delta$.
7. Go to step 1.

Was astonished, the author found that the preceding algorithm is the same as Dikin's affine algorithm (hence Algorithm 24.1.1), essentially. The only difference lies in that the former solves the reduced problem while the latter solves the standard problem. In fact, if Dikin handled the reduced rather than standard problem, he would have derived Algorithm 24.2.1.

On the other hand, the two algorithms differ computationally. Algorithm 24.2.1 should be preferable, as it saves the computational effort by solving a triangular system in each iteration. It is more than that. In fact, Algorithm 24.2.1 can be improved by incorporate pivotal inner iterations.

The resulting algorithm can be obtained by modifying Algorithm 22.4.1 easily.

Algorithm 24.2.2 (Pivotal affine face interior-point algorithm). Given $\lambda \in (0, 1)$. Initial: interior point $\bar{x} > 0$. This algorithm solves the reduced problem (22.1).

The same as Algorithm 22.4.1, except for step 7 replaced by

7. Update $\bar{x}_B = \bar{x}_B + \lambda\alpha\Delta_B$.

It is not difficult to modify the preceding conformably by including some termination criterion on precision.

Note that the vertex optimal solution can be computed by setting $\bar{x}_N = 0$ at the end of the solution precess, if needed.

Example 24.2.1. Solve the following problem by Algorithm 24.2.2:

$$
\begin{array}{ll}
\min & f = x_6, \\
\text{s.t.} & -4x_1 + 3x_2 + x_3 - 2x_4 + 2x_5 = 5, \\
& 3x_1 - x_2 + 2x_3 - 3x_4 - 4x_5 = -8, \\
& x_1 + x_2 + 2x_3 + x_4 + 3x_5 = 12, \\
& -2x_1 + 3x_3 + 2x_4 + x_5 - x_6 = 0, \\
& x_j \geq 0, \quad j = 1, \cdots, 6.
\end{array}
$$

Set $\lambda = 99/100$. Initial interior point: $\bar{x} = (1, 2, 1, 1, 2, -1)^{\mathrm{T}}$.

Answer Iteration 1:

1. $k = 6$, $B = \{1, 2, 3, 4, 5, 6\}$, $N = \emptyset$. $\bar{X}_B = \mathrm{diag}(1, 2, 1, 1, 2, -1)^{\mathrm{T}}$, face matrix B and the Cholesky factor of $B\bar{X}_B^2 B^{\mathrm{T}}$ are

$$
B = \begin{pmatrix} -4 & 3 & 1 & -2 & 2 & \\ 3 & -1 & 2 & -3 & -4 & \\ 1 & 1 & 2 & 1 & 3 & \\ -2 & & -3 & 2 & 1 & -1 \end{pmatrix},
$$

$$
L = \begin{pmatrix} 1{,}399/134 & & & \\ -2{,}362/411 & 7{,}176/919 & & \\ 1{,}985/471 & -1{,}327/373 & 916/207 & \\ 1{,}181/822 & -302/85 & -800/331 & 815/672 \end{pmatrix}.
$$

2. $\bar{y} = (-100/5{,}011, -2{,}485/5{,}193, -215/579, -1{,}219/1{,}793)^{\mathrm{T}}$.
3. $\Delta_B = (-583/1, 587, -2, 127/2, 168, 509/1, 591, -356/1, 393, 278/865, -1, 385/3, 339)^{\mathrm{T}} \neq 0$.
5. $J = \{1, 2, 4\} \neq \emptyset$.
6. $\alpha = (99/100) \min\{-1/(-583/1{,}587), -2/(-2{,}127/2{,}168), -1/(-356/1{,}393)\}$ $= (99/100)(1{,}745/856) = 2{,}333/1{,}156$, $p = 2$.
7. $\bar{x}_B = (1, 2, 1, 1, 2, -1)^{\mathrm{T}} + (2, 333/1, 156)(-583/1, 587, -2, 127/2{,}168, 509/1{,}591, -356/1, 393, 278/865, -1, 385/3, 339)^{\mathrm{T}}$ $= (323/1, 249, 750/37, 499, 1, 593/968, 261/539, 2, 005/757, -2, 019/1, 099)^{\mathrm{T}}$.
8. $B = \{1, 3, 4, 5, 6\}$, $N = \{2\}$.
9. $L = \begin{pmatrix} 882/145 & & & \\ -290/49 & 307/43 & & \\ 1{,}450/441 & -959/279 & 3{,}244/737 & \\ 145/98 & -7{,}775/2{,}199 & -1{,}184/473 & 2{,}500/2{,}447 \end{pmatrix}.$
10. $k = 5$.

Iteration 2:

2. $\bar{y} \;= (-361/1{,}907, -1{,}405/1{,}907, -1{,}039/1{,}907, -845/882)^{\mathrm{T}}$.
3. $\Delta_B = (-156/1{,}907, 232/1{,}907, 244/1{,}907, -184/1{,}907, -277/6{,}603)^{\mathrm{T}} \neq 0$.
5. $J \;= \{1, 5\} \neq \emptyset$.
6. $\alpha \;= (99/100)\min\{-(323/1,249)/(-156/1,907), -(2,005/757)/(-184/1,907)\}$
   $= (99/100)(7{,}761/2{,}455) = 917/293, \; p = 1$.
7. $\bar{x}_B \;= (323/1,249, 1,593/968, 261/539, 2,005/757, -2,019/1{,}099)^{\mathrm{T}} + (917/293)(-156/1{,}907, 232/1{,}907, 244/1{,}907, -184/1{,}907, -277/6{,}603)^{\mathrm{T}}$
   $= (109/42{,}149, 1{,}688/833, 583/659, 1{,}117/476, -5{,}360/2{,}723)^{\mathrm{T}}$.
8. $B \;= \{3, 4, 5, 6\}, \; N = \{1, 2\}$.
9. $L \;= \begin{pmatrix} 3{,}524/769 & & & \\ -1{,}435/274 & 1{,}211/172 & & \\ 1{,}435/274 & -3{,}433/1{,}235 & 1{,}187/491 & \\ 769/3{,}524 & -2{,}201/577 & -499/322 & 1 \end{pmatrix}$.
10. $k \;= 4 = m + 1$.
11. $\bar{z}_N = (156/1{,}907, 717/1{,}907)^{\mathrm{T}} \geq 0$.
12. The approximate basic optimal solution and optimal value are

$$\bar{x} \approx (109/42{,}149, 750/37{,}499, 1{,}688/833, 583/659, 1{,}117/476)^{\mathrm{T}},$$

$$\bar{x}_6 \approx -5{,}360/2{,}723.$$

The outcome is close to the exact basic optimal solution and optimal value, i.e.,

$$x^* = (0, 0, 79/39, 34/39, 92/39)^{\mathrm{T}}, \; x_6^* = -77/99.$$

The error in components of the approximate optimal solution is about 0.01, while that in the approximate optimal value is about 0.05.

24.3 Pivotal D-Reduced Gradient Interior-Point Method

In this section, we derive a pivotal interior-point algorithm with the deficient-basis framework, without discussion on the related theoretical problems.

It is noted that the search direction, used in the D-reduced gradient method (Sect. 21.5), is uphill with respect to the dual objective function, as well as points to the interior of the feasible region. Thereby, the direction is suitable to be used to design an interior-point algorithm by the known trick: starting from an interior point, and cutting down the stepsize to λ times of the original, where λ is positive number less than 1 (e.g., 0.95–0.999).

The according steps are put in the following algorithm.

Algorithm 24.3.1 (D-reduced gradient interior-point algorithm). Given $\mu > 0$, $\lambda \in (0,1)$, $\epsilon > 0$. Initial: interior point $\bar{z} > 0$. This algorithm solves the D-reduced problem (17.1).

1. Compute ω by (21.15).
2. Stop if $J = \{j \in A \mid \omega_j > 0\} = \emptyset$ (infeasible problem).
3. Compute $\beta = \bar{z}_q/\bar{\omega}_q = \min_{j \in J} \bar{z}_j/\bar{\omega}_j$.
4. Stop if $\beta < \epsilon$ (approximate optimality achieved).
5. If $\beta \neq 0$, update $\bar{z} = \bar{z} - \lambda\beta\omega$.
6. Go to step 1.

Nevertheless, our preliminary test indicates that the preceding algorithm converges quite slow (if does).

As in the previous two sections, we incorporate pivotal inner iterations to it. The resulting algorithm can be obtained by modifying Algorithm 21.6.1 easily.

Algorithm 24.3.2 (Pivotal D-reduced gradient interior-point algorithm). Given $\mu > 0, \lambda \in (0,1)$. Initial: $\bar{z}_N > 0$. This algorithm solves D-reduced problem (17.1).

The same as Algorithm 21.6.1, except for steps 4 and 5 replaced respectively by

4. Determine β and q such that $\beta = \bar{z}_q/\bar{\omega}_q = \min\limits_{j \in J} \bar{z}_j/\bar{\omega}_j$, and compute $\lambda\beta$.
5. If $\beta \neq 0$, update $\bar{z}_N = \bar{z}_N - \lambda\beta\bar{\omega}_N$.

It is not difficult to modify the preceding algorithm by introducing some precision tolerance.

Example 24.3.1. Solve the following problem by Algorithm 24.3.2:

$$
\begin{array}{lrl}
\min & f = & 2x_1 + x_2 + 3x_3 + x_4 + 2x_5 + 4x_6, \\
\text{s.t.} & & x_1 + 3x_2 - 2x_3 + 3x_4 - 4x_5 + 2x_6 = 0, \\
& & - 2x_2 \qquad\quad - x_4 - 2x_5 + x_6 = 0, \\
& & -2x_1 + x_2 + 2x_3 - 4x_4 + 3x_5 - 3x_6 = 0, \\
& & + 2x_2 + 3x_3 \qquad\quad - 2x_5 - x_6 = 1, \\
& & x_j \geq 0, \quad j = 1, \cdots, 6.
\end{array}
$$

Answer For convenience of comparison, the related tableau will be given for each iteration.

Initial tableau:

x_1	x_2	x_3	x_4	x_5	x_6	RHS
1	3	−2	3	−4	2	
	−2		−1	−2	1	
−2	1	2	−4	3	−3	
	2	3		−2	−1	1
2	1	3	1	2	4	

$\lambda = 95/100,\ \mu = 1.\ r = 4.$

$$N_{R'} = \begin{pmatrix} 1 & 3 & -2 & 3 & -4 & 2 \\ 0 & -2 & 0 & -1 & -2 & 1 \\ -2 & 1 & 2 & -4 & 3 & -3 \end{pmatrix}.$$

Iteration 1:

1. $k \quad = 0;\ B, R{=}\emptyset,\ N{=}\{1,\cdots,6\},\ R' = \{1,2,3\}.\ \bar{z}_N = (2,1,3,1,2,4)^{\mathrm{T}}.$
2. $\omega_N = (0,2,3,0,-2,-1)^{\mathrm{T}} - N_{R'}^{\mathrm{T}} N_{R'} (1/2,1,1/3,1,1/2,1/4)^{\mathrm{T}}$
   $= (-19/2,-191/12,101/6,-325/12,187/12,-41/3)^{\mathrm{T}},\ \bar{\omega}_N = \omega_N.$

x_1	x_2	x_3	x_4	x_5	x_6	RHS
1	3	−2	3	−4	2	
	−2		−1	−2	1	
−2	1	2	−4	3	−3	
−19/2	−191/12	101/6	−325/12	187/12	−41/3	1
2	1	3	1	2	4	

3. $J \quad = \{3,5\}.$
4. $\beta \quad = (95/100)\min\{3/(101/6), 2/(187/12)\} = (95/100)(24/187)$
   $= 114/935,\ q = 5.$
5. $\bar{z}_N \ = (2,1,3,1,2,4)^{\mathrm{T}}$
   $- (114/935)(-19/2,-191/12,101/6,-325/12,187/12,-41/3)^{\mathrm{T}}.$
6. $\bar{a}_q(R') = (-4,-2,3)^{\mathrm{T}}.$
7. $\bar{a}_q(R') \neq 0.$
8. $p \quad = 1,\ \tau = -1/4, \hat{B}_{\hat{R}}^{-1} = (-1/4).$
9. $k \quad = 1,\ B = \{5\},\ R = \{1\}.N = \{1,2,3,4,6\},\ R' = \{2,3\}.$
10. $\bar{\omega}_N = (-19/2,-191/12,101/6,-325/12,-41/3)^{\mathrm{T}}$
    $- (1,3,-2,3,2)^{\mathrm{T}}(-1/4)(187/12)$
    $= (-269/48,-203/48,217/24,-739/48,-47/8)^{\mathrm{T}}.$

x_1	x_2	x_3	x_4	x_5	x_6	RHS
−1/4	−3/4	1/2	−3/4	1	−1/2	
−1/2	−7/2	1	−5/2			
−5/4	13/4	1/2	−7/4		−3/2	
−269/48	−203/48	217/24	−739/48		−47/8	1
1,696/537	644/219	886/935	1,609/374	1/10	5,298/935	−114/935

Iteration 2:

3. $J = \{3\}.$
4. $\beta = (95/100)\min\{(886/935)/(217/24)\} = (95/100)(875/8{,}349)$
   $= 205/2{,}059, q = 3.$

5. $\bar{z}_N = (1{,}696/537, 644/219, 886/935, 1{,}609/374, 5{,}298/935)^{\mathrm{T}}$
$\quad - (205/2{,}059)(-269/48, -203/48, 217/24, -739/48, -47/8)^{\mathrm{T}}$
$\quad = (1{,}349/363, 9{,}359/2{,}784, 47/992, 4{,}102/703, 2{,}513/402)^{\mathrm{T}}$.
6. $\bar{a}_q(R) = (-1/4)(-2) = 1/2,\ \bar{a}_q(R') = (0,2)^{\mathrm{T}} - (-2,3)^{\mathrm{T}}(-2) = (-4,8)^{\mathrm{T}}$.
7. $\bar{a}_q(R') \neq 0$.
8. $p = 3,\ \tau = (2 - 3(1/2))^{-1} = 2, v = -2(1/2) = -1, d^{\mathrm{T}}$
$\quad = -2(3)(-1/4) = 3/2$.
$\quad U = (-1/4) - (-1)3(-1/4) = -1, \quad \hat{B}_{\hat{R}}^{-1} = \begin{pmatrix} -1 & -1 \\ 3/2 & 2 \end{pmatrix}$.
9. $k = 2,\ B = \{5,3\},\ R = \{1,3\}.N = \{1,2,4,6\},\ R' = \{2\}$.
10. $\bar{\omega}_N = (-19/2, -191/12, -325/12, -41/3)^{\mathrm{T}}$
$\quad - \begin{pmatrix} 1 & 3 & 3 & 2 \\ -2 & 1 & -4 & -3 \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} -1 & -1 \\ 3/2 & 2 \end{pmatrix}^{\mathrm{T}} (187/12, 101/6)^{\mathrm{T}}$
$\quad = (17, -63, 65/4, 85/4)^{\mathrm{T}}$.

x_1	x_2	x_3	x_4	x_5	x_6	RHS
1	−4		1	1	1	
2	−10		1		3	
−5/2	13/2	1	−7/2		−3	
17	−63		65/4		85/4	1
1,349/363	9,359/2,784	47/992	4,102/703	1/10	2,513/402	−905/4,086

Iteration 3:

3. $J = \{1,4,6\}$.
4. $\lambda\beta = (95/100)\min\{(1{,}696/537)/17, (4{,}102/703)/(65/4), (2{,}513/402)/(85/4)\}$
$\quad = (95/100)(473/2{,}546) = 473/2{,}680,\ q = 1$.
5. $\bar{z}_N = (1{,}349/363, 9{,}359/2{,}784, 4{,}102/703, 2{,}513/402)^{\mathrm{T}}$
$\quad - (473/2{,}680)(17, -63, 65/4, 85/4)^{\mathrm{T}}$
$\quad = (577/806, 9{,}398/649, 1{,}528/515, 1{,}608/643)^{\mathrm{T}}$.
6. $\bar{a}_q(R) = \begin{pmatrix} -1 & -1 \\ 3/2 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \end{pmatrix} = \begin{pmatrix} 1 \\ -5/2 \end{pmatrix}$,
$\quad \bar{a}_q(R') = 0 - (-2,0)(1,-2)^{\mathrm{T}} = 2$.
7. $\bar{a}_q(R') \neq 0$.
8. $p = 2.\ \tau = (0 - (-2,0)(1,-5/2)^{\mathrm{T}})^{-1} = 1/2$.
$\quad v = -(1/2)(1,-5/2)^{\mathrm{T}} = (-1/2, 5/4)^{\mathrm{T}}$.
$\quad d^{\mathrm{T}} = -(1/2)((-2,0)B_R^{-1}) = -(1/2)(2,2) = (-1,-1)$.
$\quad U = \begin{pmatrix} -1 & -1 \\ 3/2 & 2 \end{pmatrix} - (-1/2, 5/4)^{\mathrm{T}}(2,2) = \begin{pmatrix} 0 & 0 \\ -1 & -1/2 \end{pmatrix}$,

$$\hat{B}_{\hat{R}}^{-1} = \begin{pmatrix} 0 & 0 & -1/2 \\ -1 & -1/2 & 5/4 \\ -1 & -1 & 1/2 \end{pmatrix}.$$

9. $k \;= 2,\ B = \{5,3,1\},\ R = \{1,3,2\}.N = \{2,4,6\},\ R' = \emptyset.$
10. $\bar{\omega}_N = (-191/12, -325/12, -41/3)^{\mathrm{T}}$
$$- \begin{pmatrix} 3 & 3 & 2 \\ 1 & -4 & -3 \\ -2 & -1 & 1 \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} 0 & 0 & -1/2 \\ -1 & -1/2 & 5/4 \\ -1 & -1 & 1/2 \end{pmatrix}^{\mathrm{T}} (187/12, 101/6, -19/2)^{\mathrm{T}}$$
$= (22, 31/4, -17/4)^{\mathrm{T}}.$

x_1	x_2	x_3	x_4	x_5	x_6	RHS
	1		1/2	1	−1/2	
1	−5		1/2		3/2	
	−6	1	−9/4		3/4	
	22		31/4		−17/4	1
577/806	9,398/649	47/992	1,528/515	1/10	1,608/643	−1,143/2,872

Iteration 4:

3. $J \;= \{2,4\}.$
4. $\lambda\beta = (95/100) \min\{(8{,}645/597)/22, (1{,}528/515)/(31/4)\}$
$= (95/100)(1{,}557/4{,}067) = 559/1{,}537,\ q = 4.$
5. $\bar{z}_N = (9{,}398/649, 1{,}528/515, 1{,}608/643)^{\mathrm{T}} - (559/1{,}537)(22, 31/4, -17/4)^{\mathrm{T}}$
$= (6, 460/997, 301/2, 029, 3, 395/839)^{\mathrm{T}}.$

x_1	x_2	x_3	x_4	x_5	x_6	RHS
	1		1/2	1	−1/2	
1	−5		1/2		3/2	
	−6	1	−9/4		3/4	
	22		31/4		−17/4	1
577/806	6,460/997	47/992	301/2,029	1/10	3,395/839	−636/835

6. $$\bar{a}_q(R) = \begin{pmatrix} 0 & 0 & -1/2 \\ -1 & -1/2 & 5/4 \\ -1 & -1 & 1/2 \end{pmatrix} \begin{pmatrix} 3 \\ -4 \\ -1 \end{pmatrix} = \begin{pmatrix} 1/2 \\ -9/4 \\ 1/2 \end{pmatrix}.$$
7. $\bar{a}_q(R') = 0.$
11. $\max\{1/2, -9/4, 1/2\} = 1/2 > 0,\ s = 1,\ p = 1.$

Iteration 5:

1. $k \;= 0;\; B, R = \emptyset,\; N = \{1, \cdots, 6\},\; R' = \{1, 2, 3\}.$
2. $\omega_N = (0, 2, 3, 0, -2, -1)^{\mathrm{T}}$
   $\quad - N_{R'}^{\mathrm{T}} N_{R'}(806/577, 997/6{,}460, 992/47, 2{,}029/301{,}10, 839/3{,}395)^{\mathrm{T}}$
   $\quad = (9{,}606/67, 1{,}965/23, -22{,}001/110, 22{,}051/69, -23{,}926/57,$
   $\quad 6{,}767/25)^{\mathrm{T}},$
   $\bar{\omega}_N = \omega_N.$

x_1	x_2	x_3	x_4	x_5	x_6	RHS
1	3	−2	3	−4	2	
	−2		−1	−2	1	
−2	1	2	−4	3	−3	
9,606/67	1,965/23	−22,001/110	22,051/69	−23,926/57	6,767/25	1
577/806	6,460/997	47/992	301/2,029	1/10	3,395/839	−636/835

3. $J \;= \{1, 2, 4, 6\}.$
4. $\lambda\beta = (95/100)\min\{(577/806)/(9, 606/67), (6, 460/997)/(1, 965/23),$
   $(301/2, 029)/(22, 051/69), (3, 395/839)/(6, 767/25)\}$
   $= (95/100)(37/79, 707) = 8/18, 141,\; q = 4.$
5. $\bar{z}_N \;= (577/806, 6, 460/997, 47/992, 301/2, 029, 1/10, 3, 395/839)^{\mathrm{T}}$
   $- (8/18, 141)(9, 606/67, 1, 965/23, -22, 001/110, 22, 051/69,$
   $- 23, 926/57, 6, 767/25)^{\mathrm{T}}$
   $= (295/452, 1, 604/249, 189/1, 394, 104/14, 021, 1, 074/3, 767, 1,$
   $347/343)^{\mathrm{T}}.$
6. $\bar{a}_q(R') = (3, -1, -4)^{\mathrm{T}}.$
7. $\bar{a}_q(R') \neq 0.$
8. $p \;= 3.\; \tau = -1/4. \hat{B}_{\hat{R}}^{-1} = (-1/4).$
9. $k \;= 1,\; B = \{4\},\; R = \{3\}. N = \{1, 2, 3, 5, 6\},\; R' = \{1, 2\}.$
10. $\bar{\omega}_N = (9, 606/67, 1, 965/23, -22, 001/110, -23, 926/57, 6, 767/25)^{\mathrm{T}}$
    $- (-2, 1, 2, 3, -3)^{\mathrm{T}}(-1/4)(22, 051/69)$
    $= (-25, 331/1, 543, 15, 045/91, -10, 457/260, -7, 743/43, 6,$
    $478/209)^{\mathrm{T}}.$

x_1	x_2	x_3	x_4	x_5	x_6	RHS
−1/2	15/4	−1/2		−7/4	−1/4	
1/2	−9/4	−1/2		−11/4	7/4	
1/2	−1/4	−1/2	1	−3/4	3/4	
−25,331/1,543	15,045/91	−10,457/260		−7,743/43	6,478/209	1
295/452	1,604/249	189/1,394	104/14,021	1,074/3,767	1,347/343	−1,022/1,341

Iteration 6:

3. $J \;= \{2, 6\}$.
4. $\beta \;= (95/100) \min\{(1{,}604/249)/(15{,}045/91), (1{,}347/343)/(6{,}478/209)\}$
   $= (95/100)(239/6{,}134) = 311/8{,}402,\ q = 2.$
5. $\bar{z}_N \;= (295/452, 1{,}604/249, 189/1{,}394, 1{,}074/3{,}767, 1{,}347/343)^{\mathrm{T}}$
   $- (311/8{,}402)(-25{,}331/1{,}543, 15{,}045/91, -10{,}457/260,$
   $- 7{,}743/43, 6{,}478/209)^{\mathrm{T}}$
   $= (1{,}893/1{,}502, 439/1{,}363, 2{,}019/1{,}243, 1{,}821/262, 1{,}957/704)^{\mathrm{T}}.$
6. $\bar{a}_q(R) = (-1/4)(1) = -1/4,\ \bar{a}_q(R') = (3, -2)^{\mathrm{T}} - (3, -1)^{\mathrm{T}}(-1/4) = (15/4,$
   $- 9/4)^{\mathrm{T}}.$
7. $\bar{a}_q(R') \neq 0.$
8. $p \;= 1,\ \tau = (3 - 3(-1/4))^{-1} = 4/15, v = -(4/15)(-1/4) = 1/15,$
   $d^{\mathrm{T}} = -(4/15)(3)(-1/4) = 1/5.$
   $U \;= (-1/4) - (1/15)3(-1/4) = -1/5, \quad \hat{B}_{\hat{R}}^{-1} = \begin{pmatrix} -1/5 & 1/15 \\ 1/15 & 4/15 \end{pmatrix}.$
9. $k \;= 2,\ B = \{4, 2\},\ R = \{3, 1\}.N = \{1, 3, 5, 6\},\ R' = \{2\}.$
10. $\bar{\omega}_N = (9{,}606/67, -22{,}001/110, -23{,}926/57, 6{,}767/25)^{\mathrm{T}}$
    $- \begin{pmatrix} -2 & 2 & 3 & -3 \\ 1 & -2 & -4 & 2 \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} -1/5 & 1/15 \\ 1/15 & 4/15 \end{pmatrix}^{\mathrm{T}} (22{,}051/69, 1{,}965/23)^{\mathrm{T}}$
    $= (1{,}570/279, -1{,}763/97, -9{,}777/95, 14{,}664/349)^{\mathrm{T}}.$

x_1	x_2	x_3	x_4	x_5	x_6	RHS
−2/15	1	−2/15		−7/15	−1/15	
1/5		−4/5		−19/5	8/5	
7/15		−8/15	1	−13/15	11/15	
1,570/279		−1,763/97		−9,777/95	14,664/349	1
1,893/1,502	439/1,363	2,019/1,243	104/14,021	1,821/262	1,957/704	−553/692

Iteration 7:

3. $J \;= \{1, 6\}$.
4. $\lambda\beta = (95/100) \min\{(1{,}893/1{,}502)/(1{,}570/279), (1{,}957/704)/$
   $(14{,}664/349)\}$
   $= (95/100)(426/6{,}439) = 436/6{,}937,\ q = 6.$
5. $\bar{z}_N \;= (1{,}893/1{,}502, 2{,}019/1{,}243, 1{,}821/262, 1{,}957/704)^{\mathrm{T}}$
   $- (436/6{,}937)(1{,}570/279, -1{,}763/97, -9{,}777/95, 14{,}664/349)^{\mathrm{T}}$
   $= (437/482, 3{,}118/1{,}127, 5{,}287/394, 113/813)^{\mathrm{T}}.$
6. $\bar{a}_q(R) = \begin{pmatrix} -1/5 & 1/15 \\ 1/15 & 4/15 \end{pmatrix} \begin{pmatrix} -3 \\ 2 \end{pmatrix} = \begin{pmatrix} 11/15 \\ -1/15 \end{pmatrix},$
   $\bar{a}_q(R') = 1 - (-1, -2)(11/15, -1/15)^{\mathrm{T}} = 8/5.$
7. $\bar{a}_q(R') \neq 0.$

8. $p \;= 2,\ \tau = (1-(-1,-2)(11/15,-1/15)^{\mathrm{T}})^{-1} = 5/8.$

$v \;= -(5/8)(11/15,-1/15)^{\mathrm{T}} = (-11/24, 1/24)^{\mathrm{T}}.$

$d^{\mathrm{T}} = -(5/8)((-1,-2)B_R^{-1}) = -(5/8)(-1/5,-3/5) = (1/8, 3/8).$

$$U = \begin{pmatrix} -1/5 & 1/15 \\ 1/15 & 4/15 \end{pmatrix} - (-11/24, 1/24)^{\mathrm{T}}(-1/5,-3/5)$$

$$= \begin{pmatrix} -7/24 & -5/24 \\ 5/24 & 7/24 \end{pmatrix}.$$

$$\hat{B}_{\hat{R}}^{-1} = \begin{pmatrix} -7/24 & -5/24 & -11/24 \\ 5/24 & 7/24 & 1/24 \\ 1/8 & 3/8 & 5/8 \end{pmatrix}.$$

9. $k \;= 2,\ B = \{4,2,6\},\ R = \{3,1,2\}. N = \{1,3,5\},\ R' = \emptyset.$

10. $$\bar{\omega}_N = (9{,}606/67, -22{,}001/110, -23{,}926/57)^{\mathrm{T}} - \begin{pmatrix} -2 & 2 & 3 \\ 1 & -2 & -4 \\ 0 & 0 & -2 \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} -7/24 & -5/24 & -11/24 \\ 5/24 & 7/24 & 1/24 \\ 1/8 & 3/8 & 5/8 \end{pmatrix}^{\mathrm{T}} \cdot (22{,}051/69, 1{,}965/23, 6{,}767/25)^{\mathrm{T}}$$

$$= (515/1{,}373, 25{,}888/9{,}137, -9{,}428/3{,}017)^{\mathrm{T}}.$$

x_1	x_2	x_3	x_4	x_5	x_6	RHS
−1/8	1	−1/6		−5/8		
1/8		−1/2		−19/8	1	
3/8		−1/6	1	7/8		
515/1,373		25,888/9,137		−9,428/3,017		1
437/482	439/1,363	3,118/1,127	104/14,021	5,287/394	113/813	−1,755/2,036

Iteration 8:

3. $J \;= \{1,3\}.$

4. $\lambda\beta = (95/100)\min\{(437/482)/(515/1{,}373), (3{,}118/1{,}127)/(25{,}888/9{,}137)\}$
$= (95/100)(2{,}780/2{,}847) = 500/539,\ q = 3.$

5. $\bar{z}_N = (437/482, 3{,}118/1{,}127, 5{,}287/394)^{\mathrm{T}} - (500/539)(515/1{,}373, 25{,}888/9{,}137, -9{,}428/3{,}017)^{\mathrm{T}}$
$= (733/1{,}312, 214/1{,}547, 9{,}350/573)^{\mathrm{T}}.$

x_1	x_2	x_3	x_4	x_5	x_6	RHS
−1/8	1	−1/6		−5/8		
1/8		−1/2		−19/8	1	
3/8		−1/6	1	7/8		
515/1,373		25,888/9,137		−9,428/3,017		1
733/1,312	439/1,363	214/1,547	104/14,021	9,350/573	113/813	−2,450/1,369

6. $\bar{a}_q(R) = \begin{pmatrix} -7/24 & -5/24 & -11/24 \\ 5/24 & 7/24 & 1/24 \\ 1/8 & 3/8 & 5/8 \end{pmatrix} \begin{pmatrix} 2 \\ -2 \\ 0 \end{pmatrix} = \begin{pmatrix} -1/6 \\ -1/6 \\ -1/2 \end{pmatrix}.$
7. $R' \quad = \emptyset.$
11. $\max\{-1/6, -1/6, -1/2\} \leq 0.$
13. Setting $\bar{x}_3 = 9{,}137/25{,}888$ gives

$$\begin{aligned} \bar{x}_B &= (9{,}137/25{,}888)(1/6, 1/6, 1/2)^{\mathrm{T}} \\ &= (9{,}137/155{,}328, 9{,}137/155{,}328, 9{,}137/51{,}776)^{\mathrm{T}}. \end{aligned}$$

Thus, the approximate basic optimal solution and optimal value are

$$\begin{aligned} \bar{x} &\approx (0, 9{,}137/155{,}328, 9{,}137/25{,}888, 9{,}137/155{,}328, 0, 9{,}137/51{,}776)^{\mathrm{T}}, \\ \bar{f} &\approx (1, 3, 1, 4)(9{,}137/155{,}328, 9{,}137/25{,}888, 9{,}137/155{,}328, 9{,}137/ \\ &\quad 51{,}776)^{\mathrm{T}} = 9{,}137/4{,}854. \end{aligned}$$

On the other hand, the exact basic optimal solution and optimal value are

$$x^* = (0, 1/17, 6/17, 1/17, 0, 3/17)^{\mathrm{T}}, \quad f^* = 32/17.$$

The errors are less than 10^{-5}, which are accumulated from the computation of $\bar{\omega}$.

24.4 Feasible-Point Simplex Method

On the conventional simplex framework, the method presented in this section will utilize a new pivot rule to generates a sequence of feasible points, which are not necessarily vertices or interior points. If the initial is an interior point, however, it becomes an interior-point algorithm. It might be the first simplex-like method that may go across the interior of the feasible region.

We are concerned with the bounded-variable problem (7.13), i.e.,

$$\begin{aligned} \min \quad & c^T x \\ s.t \quad & Ax = b, \quad l \leq x \leq u, \end{aligned} \tag{24.14}$$

where $A \in R^{m \times n} (m < n)$; rank $A = m$. The values of components of l and u are assumed to be finite, hence the problems is bounded. For problems with infinite bounds, one may use numbers in large enough module instead.

Let B be the current (standard) basis and N the associated nonbasis, defined by

$$B = \{1, \cdots, m\}, \qquad N = A \backslash B.$$

24.4.1 Column Pivot Rule and Optimality Condition

Let $\bar{x}$ be current feasible solution, whose nonbasic components are not necessarily on either lower or upper bound. The reduced costs are

$$\bar{c}_B = 0, \qquad \bar{c}_N = c_N - N^T\bar{y}, \qquad B^T\bar{y} = c_B. \tag{24.15}$$

Introduce nonbasic index sets

$$N_1 = \{j \in N \mid \bar{c}_j < 0\}, \qquad N_2 = \{j \in N \mid \bar{c}_j > 0\}. \tag{24.16}$$

Recall that Algorithm 7.4.1 selects an entering index based on $\bar{c}$ only. In contrast, we not only consider $\bar{c}$, but also take into account the possible ranges the nonbasic components of the current feasible solution are allowed to change. To this end, introduce notation

$$\delta_j = \begin{cases} u_j - \bar{x}_j, & j \in N_1, \\ \bar{x}_j - l_j, & j \in N_2, \\ \min\{u_j - \bar{x}_j, \bar{x}_j - l_j\} & j \in N \setminus N_1 \cup N_2. \end{cases} \tag{24.17}$$

So, δ_j is the distance from $\bar{x}_j$ to one of the associated its bounds that will be violated if $\bar{x}_j$ changes to decrease the objective value.

Then the following rule is applicable.

Rule 24.4.1 (Column pivot rule) Select a nonbasic index q such that

$$|\bar{c}_q|\delta_q = \max\ \{\ |\bar{c}_j|\delta_j \mid j \in N\ \}. \tag{24.18}$$

Under the preceding rule, the objective value will decrease the most (by amount $|\bar{c}_q|\delta_q$), ignoring presence of broking basic variables.

In this context, the following optimal condition is relevant.

Proposition 24.4.1. *$\bar{x}$ is an optimal solution if it holds that*

$$|\bar{c}_j|\delta_j = 0, \qquad \forall\ j \in N. \tag{24.19}$$

Proof. Note that quantities $|\bar{c}_j|\delta_j$ $(j \in N)$ are upper bounds of the amount by which the objective value can decrease as the value of x_j changes. Thus, condition (24.19) implies that the objective value can not decrease any further, and the proposition is valid. □

Furthermore, it is clear that $\bar{x}$ is a basic optimal solution if

$$\delta_j = 0, \qquad \forall\ j \in N. \tag{24.20}$$

The preceding optimality condition might be suitable for applications when a vertex solution is required.

24.4.2 Search Direction

If the optimality condition is not fulfilled, it is possible to decrease the objective value.

Assume that a nonbasic index q has been selected to enter the basis. For a search direction, consider vector Δx defined by

$$\Delta x_B = \operatorname{sign}(\bar{c}_q)\bar{a}_q, \qquad B\bar{a}_q = a_q. \tag{24.21}$$

$$\Delta x_j = \begin{cases} 0, & j \in N;\ j \neq q \\ -\operatorname{sign}(\bar{c}_q), & j = q. \end{cases} \tag{24.22}$$

We have the following Lemma, ensuring the eligibility of Δx to be a search direction.

Lemma 24.4.1. *Assume that q is determined by (24.18). Vector Δx is a downhill with respective to the objective in the null of A.*

Proof. Note that $\bar{c}_q \neq 0$, since, otherwise, $|\bar{c}_q|\delta_q = 0$ implies that the optimality condition (24.19) holds, as leading to a contradiction.

From (24.21) and (24.22), it follows that

$$A\Delta x = \operatorname{sign}(\bar{c}_q)BB^{-1}a_q - \operatorname{sign}(\bar{c}_q)a_q = 0 \tag{24.23}$$

Thus, Δx is in the null of A.

Further, it holds by (24.15) that

$$\bar{c}_q = c_q - c_B^T B^{-1} a_q,$$

which together with (24.21), (24.22), (24.18) and (24.16) gives

$$\begin{aligned} c^T \Delta x &= -\operatorname{sign}(\bar{c}_q)c_q + \operatorname{sign}(\bar{c}_q)c_B^T B^{-1} a_q \\ &= -\operatorname{sign}(\bar{c}_q)(c_q - c_B^T B^{-1} a_q) \\ &= -\operatorname{sign}(\bar{c}_q)\bar{c}_q \\ &< 0. \end{aligned} \tag{24.24}$$

Therefore, Δx is downhill with respect to the objective function. □

24.4.3 Stepsize and Row Pivot Rule

Using Δx as a search direction, we are led to the line search scheme below:

$$\hat{x} = \bar{x} + \alpha \Delta x,$$

or equivalently,

$$\hat{x}_B = \bar{x}_B + \alpha \Delta x_B, \tag{24.25}$$

$$\hat{x}_q = \bar{x}_q - \text{sign}(\bar{c}_q)\alpha, \tag{24.26}$$

$$\hat{x}_j = \bar{x}_j, \quad j \in N;\ j \neq q, \tag{24.27}$$

where stepsize α is to be determined.

Introduce notation

$$\gamma_i = \begin{cases} (u_i - \bar{x}_i)/\Delta x_i & \text{if } \Delta x_i > 0, \text{ and } i \in B, \\ \\ (l_i - \bar{x}_i)/\Delta x_i & \text{if } \Delta x_i < 0, \text{ and } i \in B, \end{cases} \tag{24.28}$$

Then the largest possible value of α is derived subject to $l \leq \hat{x} \leq u$, i.e.,

$$\bar{\alpha} = \min\{\delta_q,\ \min\{\gamma_i \mid \Delta x_i \neq 0,\ i \in B\}\}. \tag{24.29}$$

However, the algorithm will not take α itself as a stepsize, but a smaller one instead, i.e.,

$$\alpha = \lambda \bar{\alpha}, \qquad 0 < \lambda < 1. \tag{24.30}$$

Therefore, a new solution $\hat{x}$ can be computed via (24.25)–(24.30).

There are two cases arising:

(i) $\alpha = \delta_q$. It is then clear that there is no need for any basis change.
(ii) $\alpha < \delta_q$. A basis change is performed. The following row rule is used to determine a leaving index to match the entering index q.

Rule 24.4.2 (Row pivot rule) Select a basic index p such that

$$\gamma_p = \alpha,\ p \in B,\ \Delta x_p \neq 0. \tag{24.31}$$

Lemma 24.4.2. *$\hat{x}$ defined by (24.25)–(24.27) with α given by (24.28)–(24.30) is a feasible solution.*

Proof. Firstly, it is easy to verify that $\hat{x}$ defined by (24.25)–(24.27) satisfies the equality constraints $A\hat{x} = b$ for any real number α. In addition, by the feasibility of $\bar{x}$, (24.25)–(24.27), and (24.30), $\hat{x}$ satisfies $l \leq \hat{x} \leq u$, and is hence feasible. □

However, if α vanishes, $\hat{x}$ is not really new but the same as $\bar{x}$. This happens only when the current solution is degenerate (see Definition 7.4.1).

Theorem 24.4.3. *If feasible solution $\bar{x}$ is nondegenerate, then the associated stepsize is positive, and the new iterate $\hat{x}$ is a nondegenerate feasible solution, associated with a strictly lower objective value than the old.*

Proof. From Lemma 24.4.2, it is clear that the new iterate $\hat{x}$ is a feasible solution.

Note that δ_q is positive, since, otherwise, $|\bar{c}_q|\delta_q = 0$ implies that the optimality condition (24.19) holds. Moreover, the nondegeneracy assumption implies that γ_i, defined by (24.28), are positive for all $i \in B$ with $\Delta x_i \neq 0$. These along with (24.29) and (24.30) leads to $0 < \alpha < \alpha$. For $\hat{x}$ defined by (24.25)–(24.27), therefore, the basic components are still not on their bounds, no matter whether the basis change is performed ($\alpha < \delta_q$) or not.

Furthermore, by $\alpha > 0$, (24.25)–(24.27) and Lemma 24.4.1 it holds that $c^T\hat{x} < c^T\bar{x}$, which completes the proof. □

24.4.4 Formulation of the Algorithm

Some computational considerations should be incorporated in the implementation of the algorithm. In practice, for instance, what required is often an *approximate* optimal solution only. Therefore, the algorithm will use the following condition in place of (24.19) instead.

Definition 24.4.1. $\bar{x}$ is an ϵ-optimal solution if

$$|\bar{c}_j| \leq \epsilon_1 \qquad \text{or} \qquad \delta_j \leq \epsilon_2, \qquad \forall\, j \in N, \tag{24.32}$$

where $\epsilon_1 > 0$ and $\epsilon_2 > 0$ are predetermined tolerances.

Accordingly, index sets N_1 and N_2, defined by (24.16), are now redefined by

$$N_1 = \{j \in N \mid \bar{c}_j < -\epsilon_1\}, \qquad N_2 = \{j \in N \mid \bar{c}_j > \epsilon_2\}. \tag{24.33}$$

The overall steps can be put into the following algorithm.

Algorithm 24.4.1 (Feasible-point simplex algorithm). Given $0 < \lambda < 1$, $\epsilon_1,\ \epsilon_2 > 0$ and $M \gg 1$. Initial: basis B, associated with feasible solution $\bar{x}$. This algorithm solves the bounded-variable problem.

1. Solve $B^T\bar{y} = c_B$ for $\bar{y}$.
2. Compute $\bar{c}_N = c_N - N^T\bar{y}$.
3. Compute $\delta_j,\ j \in N$ by (24.17).
4. Stop if (24.32) is satisfied.
5. Determine index q such that $|\bar{c}_q|\delta_q = \max\{\,|\bar{c}_j|\delta_j \mid j \in N\,\}$.

6. Solve $B\bar{a}_q = a_q$ for $\bar{a}_q$.
7. Compute Δx by (24.21) and (24.22).
8. Determine $\bar{\alpha} = \min\{\delta_q, \min\{\gamma_i \mid \Delta x_i \neq 0, i \in B\}\}$.
9. Compute $\alpha = \lambda\bar{\alpha}$.
10. Stop if $\alpha > M$.
11. Update $\bar{x}$ by (24.25)–(24.27).
12. Go to Step 1 if $\alpha = \delta_q$.
13. Determine index p such that $\gamma_p = \alpha, \ p \in B, \ \Delta x_p \neq 0$.
14. Update basis B by replacing its p-th column with q-th column.
15. Go to step 1.

Theorem 24.4.4. *Assume termination of Algorithm 24.4.1. It terminates at either*

(i) step 4, achieving an ϵ-optimal basic solution; or
(ii) step 10, declaring lower unboundedness.

The meanings of the exits of the preceding Algorithm is clear. At this stage, however, it has not been possible to rule out the possibility of infiniteness. As it solved a large number of test problems in our computational experiments (see below), we claim that Algorithm 24.4.1 should be regarded as finite practically, just like the standard simplex algorithm.

Even if it is still open whether Algorithm 24.4.1 is finite or not, on the other hand, the following result is valid.

Theorem 24.4.5. *Assume that the initial feasible solution is nondegenerate. Then all subsequent stepsizes are positive, and hence iterates are all nondegenerate feasible solutions.*

Proof. It is enough to consider a current iteration.

By Lemma 24.4.2, the new iterate $\hat{x}$ satisfies $A\hat{x} = b$. From the feasibility of $\bar{x}$, (24.25)–(24.27), and (24.30), it follows that $\bar{x}$ satisfies $l \leq \hat{x} \leq u$, and hence is a feasible solution.

By (3.3), moreover, δ_q is positive, since, otherwise, the iteration would have terminated at Step 3(3). In addition, the nondegeneracy assumption implies that γ_i, defined by (24.28), are positive for all $i \in B$ satisfying $\Delta x_i \neq 0$. This fact along with (24.29) and (24.30) gives $0 < \alpha < \alpha$. Consequently, from (24.25)–(24.27) it follows that $\hat{x}$ is again nondegenerate.

Furthermore, by $\alpha > 0$, (24.25)–(24.27) and Lemma 24.4.1, it holds that $c^T\hat{x} < c^T\bar{x}$. This completes the proof. □

If the initial feasible solution is nondegenerate, Theorem 24.4.5 implies that the objective value strictly decreases in the solution process. Moreover, the algorithm has the following feature.

Proposition 24.4.2. *Assume that the initial feasible solution is nondegenerate. An on-bound component of $\bar{x}$ could become an interior component; but any interior component never becomes an on-bound component.*

The preceding implies that Algorithm 24.4.1 is an interior-point algorithm if an initial interior-point is used. On the other hand, solutions generated by Algorithm 24.4.1 are not vertices in general, even if the initial one is. Consequently, the algorithm goes across the interior of the feasible region.

Finally, we introduce the concept of an *approximate* optimal *basic* solution in place of (24.20), alternatively:

Definition 24.4.2. $\bar{x}$ is an ϵ-optimal basic solution if

$$\delta_j < \epsilon, \qquad \forall\ j \in N, \tag{24.34}$$

where $\epsilon > 0$ is a predetermined tolerance.

It is noted that if condition (24.34) is used as termination criterion, $\bar{c}_q = 0$ could holds, and hence the associated Δx is not downhill. In this case, the algorithm continues using Δx to move to an optimal *basic* solution, even if optimality has already been achieved (see Proposition 24.4.1).

24.4.5 Phase-1 and Purification

Any Phase-I approach for the bounded-variable problem can be used to provide an initial feasible solution. If one wants the algorithm starting from an interior point, the approach described at the end of Sect. 7.4 applies.

As for obtaining an exact optimal basic solution, the following simple purification can be incorporated to Algorithm 24.4.1.

Assume that Algorithm 24.4.1 terminates at step 4 with an ϵ-optimal basic solution. The purification is done by moving nonbasic components of the solution onto their respective nearest bounds with the basic components unchanged. If the resulting solution, say x^0, satisfies $Ax^0 = b$ within some tolerance, it is clearly an optimal basic solution. In the other case, a standard two-phase simplex algorithm can be used to attain a basic optimal solution, hopefully within few iterations.

24.4.6 Computational Results

Computational experiments have been performed to gain an insight into the behavior of Algorithm 24.4.1. A summary of the associated numerical results are offered in this subsection.

Implemented, and compared are the following three codes:

1. MINOS: MINOS 5.51 with full pricing.
2. FPS: Two-Phase code based on Algorithm 24.4.1.
3. FPSP: Two-Phase code based on Algorithm 24.4.1 with the purification.

Table 24.2 Ratio

	MINOS/FPSP		MINOS/FPS		FPSP/FPS	
Problem	Itns	Time	Itns	Time	Itns	Time
Kennington(16)	6.6	3.2	9.0	3.5	1.4	1.1
BPMPD(15)	2.8	3.4	9.5	9.0	3.3	2.6
Average(31)	3.3	3.4	9.4	6.6	2.8	2.0

The first set of test problems included all 16 problems from Kennington and the second included all 17 problems from BPMPD that were more than 500KB in compressed form (Appendix B: Tables B.2–B.3).

In Table 24.2, a comparison between the three codes is made.

From the table, it is seen that FPSP and FPS outperformed MINOS remarkably, with average iteration ratios 6.6 and 9.0, and time ratios 3.2 and 3.5 for the 16 Kennington problems. They outperformed MINOS, by average iterations ratios 2.8 and 9.5, and time ratios 3.4 and 9.0 for BPMPD problems. For the entire set of the 31 test problems, FPSP and FPS defeated MINOS by average iterations ratios 3.3 and 9.4, and time ratios 3.4 and 6.6.

In summary, the feasible-point simplex algorithm is significantly superior to the standard simplex algorithm with the test set (see Appendix E for more details).