4.6 The Chinese Postman Problem

Chinese Postman Problem

In his job, a postman picks up mail at the post office, delivers it, and then returns to the post office. He must, of course, cover each street in his area at least once. Subject to this condition, he wishes to choose his route in such a way that walks as little as possible. This problem is known as the **Chinese postman problem**, since it was first considered by a Chinese mathematician, Guan in 1960.

Graphic Model

We refer to the street system as a weighted graph $(G, \mathbf{w})$ whose vertices represent the intersections of the streets, whose edges represent the streets (one-way or two-way), and the weight represents the distance between two intersections, of course, a positive real number. A closed walk that covers each edge at least once in G is called a **postman tour**. Clearly, the Chinese postman problem is just that of finding a minimum-weight postman tour. We will refer to such a postman tour as an **optimal tour**.

There are many real-world situations that can be reduced as the Chinese postman problem. For example, a driver of a watering car or a garbage truck, he wishes to choose his route in such a way that traverses as little as possible. In this section, we introduce an efficient algorithm for solving the Chinese postman problem, due to Edmonds and Johnson (1973).

First consider simple case that G is eulerian.

Then any an Euler circuit is an optimal tour since it traverses each edge exactly once. The Chinese postman problem is easily solved in this case, since there exists an efficient algorithm determining an Euler circuit in an eulerian graph, no matter that it is directed or undirected.

We here consider G **a directed eulerian graph**. Then G is strongly connected, and so there must be a spanning out-tree rooted at x_0 in G for any $x_0 \in V(G)$ (by the exercise 2.1.10). **Making use of Dijsktra's algorithm, we can find a spanning out-tree T rooted at x_0**. Based on this tree T, Edmonds and Johnson's algorithm can find an Euler directed circuit in G.

Edmonds and Johnson's Algorithm

1. Choose arbitrarily a vertex x_0 in G, find a spanning out-tree T rooted at x_0 in G, and set $P_0 = x_0$.

2. Suppose that a directed trail $P_i = x_i a_i x_{i-1} a_{i-1} \cdots a_2 x_1 a_1 x_0$ has been chosen. Then choose an edge a_{i+1} from $E(G) \setminus \{a_1, a_2, \cdots, a_i\}$ in such a way that

(i) $\psi_G(a_{i+1}) = (x_{i+1}, x_i)$;

(ii) $a_{i+1} \notin E(T)$ unless there is no alternative.

3. Stop when Step 2 can no longer be implemented.

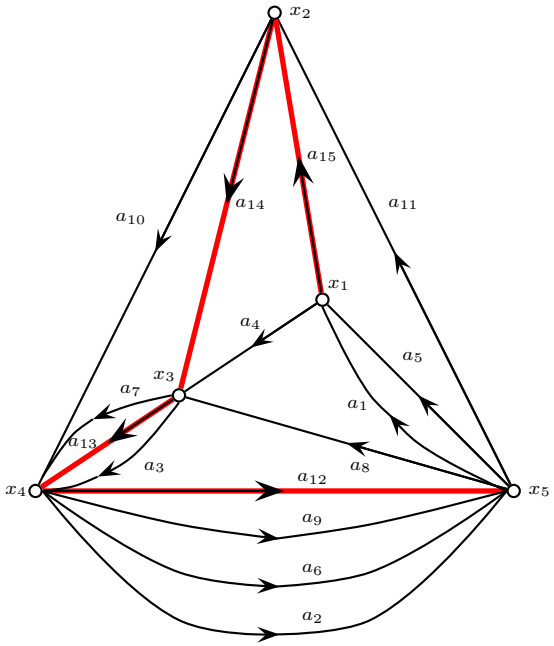

Figure 4.14: A spanning out-tree T rooted at x_1 in G

Theorem 4.11 **If G is an eulerian digraph, then any directed trail in G constructed by the above algorithm is an Euler directed circuit in G.**

Proof: Let G be an eulerian digraph, and let $P_n = x_n a_n x_{n-1} a_{n-1} \cdots a_2 x_1\ a_1 x_0$ be a directed trail in G constructed by the above algorithm. Since G is eulerian, G is balanced by Theorem 1.7, and so $x_n = x_0$.

Suppose, now, that P_n is not an Euler circuit of G. Then there is $b_1 \in E(G)$, but $b_1 \notin E(P_n)$. Let $\psi_G(b_1) = (x_i, x_j)$. Then, by Step 2 (ii), without loss of generality suppose $b_1 \in E(T)$. Since x_i is balanced in G and P_n, there is $b_2 \in E(G)$, but $b_2 \notin E(P_n)$ with $\psi_G(b_2) = (x_k, x_i) \in E(T)$. Similarly, there is $b_3 \in E(G)$, but

$b_3 \notin E(P_n)$ with $\psi_G(b_3) = (x_l, x_k) \in E(T)$ and so on. Thus, there is a sequence of edges $b_1, b_2, b_3, \cdots$, which can be traced back to $x_0 = x_n$ along an (x_0, x_j)-path of T in reverse direction. Since x_n is balanced in G and P_n, there is $a \in E(G)$ with head x_n, but $a \notin E(P_n)$. This contradicts Step 3. Therefore, P_n is an Euler circuit of G. ∎

Example 4.6.1 Consider the digraph G in Figure 4.14. Since G is connected and balanced, by Theorem 1.7, G is eulerian. A spanning out-tree T rooted at x_1 in G is denoted by heavy edges. An Euler circuit constructed by Edmonds and Johnson's algorithm is as follows:

$$\begin{aligned} P = \quad & x_1\ a_{15}\ x_2\ a_{14}\ x_3\ a_{13}\ x_4\ a_{12}\ x_5\ a_{11}\ x_2\ a_{10}\ x_4\ a_9 \\ & x_5\ a_8\ x_3\ a_7\ x_4\ a_6\ x_5\ a_5\ x_1\ a_4\ x_3\ a_3\ x_4\ a_2\ x_5\ a_1\ x_1. \end{aligned}$$

∎

We, now, suppose that G is not eulerian. The digraph G in Figure 4.15 has no postman tour since it contains no directed path from $\{y_1, y_2, y_3\}$ to $\{x_1, x_2, x_3\}$. Thus, we first discuss the existence of post-tours.

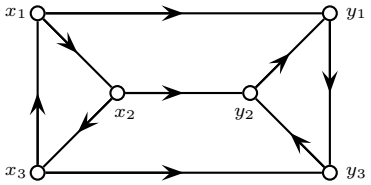

Figure 4.15: A digraph that contains no postman tour

Theorem 4.12 A digraph G contains a postman tour if and only if G is strongly connected.

Proof: The necessity holds clearly. We suppose that G is strongly connected. Then G must contains a directed cycle. Choose a closed directed walk C that contains as many edges of G as possible. If C were not a postman tour, there should be $a \in E(G) \setminus E(C)$. Let $\psi_G(a) = (x, y)$ and choose arbitrarily a vertex u in C. There are a (u, x)-path P and a (y, u)-path Q. Then $C' = C \oplus (P + a) \oplus Q$ is a closed directed walk in G that contains edges of G more than C does, a contradiction. ∎

Suppose, now, that $(G, \mathbf{w})$ **is a non-balanced and strongly connected weighted digraph**, P is a postman tour in G. Thus, P must traverse each edge of G once or more. We denote by $p(a)$ the number of times that the edge a repeatedly occurs in P. Let G^* denote the supergraph of G obtained by adding extra $p(a)$ copies of a for each edge $a \in E(G)$ to G. Clearly, G^* is balanced and the postman tour P in G corresponds to an Euler circuit in G^*.

Thus, a basic outline of solving the Chinese postman problem can be described as follows.

For a given strongly connected weighted digraph $(G, \mathbf{w})$,

(i) construct a balanced supergraph G^* of G such that the added edges have as little sum of weight as possible;

(ii) finding an Euler directed circuit in G^*.

Edmonds and Johnson's algorithm described above solves (ii). We will introduce an algorithm for solving (i), also due to Edmonds and Johnson (1973).

For $x \in V(G)$, let $\rho(x) = d_G^-(x) - d_G^+(x)$, and let

$$X = \{x \in V(G) : \ \rho(x) > 0\}, \quad \text{and} \quad Y = \{y \in V(G) : \ \rho(y) < 0\}.$$

Since G is non-balanced, by Theorem 1.1, we have $X \neq \emptyset,\ Y \neq \emptyset$ and

$$\sum_{x \in X} \rho(x) = -\sum_{y \in Y} \rho(y).$$

Denote by $\rho(G)$ the above value.

For example, for the digraph G in Figure 4.16 (a), we have

$$\rho(x_1) = -1,\ \rho(x_2) = 0,\ \rho(x_3) = 2,\ \rho(x_4) = 1,\ \rho(x_5) = -2,$$

and so

$$X = \{x_3, x_4\}, \quad Y = \{x_1, x_5\}, \quad \text{and} \quad \rho(G) = 3.$$

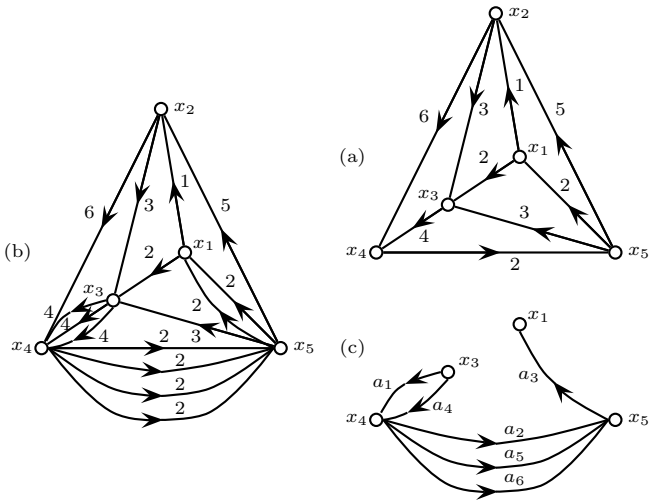

Figure 4.16: (a) A weighted digraph $(G, \mathbf{w})$; (b) $(G^*, \mathbf{w}^*)$; (c) $H = G^*[E^*]$

Suppose that, subject to (i), a supergraph G^* and the collection E^* of added edges have been chosen. Let $H = G^*[E^*]$. Then H **consists of $\rho(G)$ edge-disjoint (x, y)-paths for some $x \in$ and $y \in Y$.** See, for example, Figure 4.16, a supergraph G^* of G in (a) is in (b), where E^* consists of the curved lines, H is in (c), which consists of three edge-disjoint directed paths:

$$P_1 = x_3\, a_1\, x_4\, a_2\, x_5\, a_3\, x_1, \quad P_2 = x_3\, a_4\, x_4\, a_5\, x_5, \quad P_3 = x_4\, a_6\, x_5.$$

Conversely, the set of edges E^* of any $\rho(G)$ edge-disjoint (X, Y)-paths with minimum weight (if some edge a is used m times, then the weight of a must be computed m times) is a solution of (i) (since, in the present case, G^* is balanced).

Thus, the solution of (i) is reduced to choosing ρ edge-disjoint paths $P_1, P_2, \cdots, P_\rho$ from X to Y in G such that the sum of their weights $\mathbf{w}(P_1) + \mathbf{w}(P_2) + \cdots + \mathbf{w}(P_\rho)$ is as little as possible, where $\rho = \rho(G)$.

Settled Method

To the end, we construct a cost-capacity network $N = (G'_{x_0 y_0}, \mathbf{b}, \mathbf{c})$, where G' is obtained from G by adding two new vertices x_0 and y_0, then joining x_0 to each vertex x in X by a directed edge with cost 0 and capacity $\rho(x)$; joining each vertex y in Y to y_0 by a directed edge with cost 0 and capacity $-\rho(y)$; $\mathbf{b}(a) = \mathbf{w}(a)$ and $\mathbf{c}(a) = \infty$ for each $a \in E(G)$.

Figure 4.17 (a) illustrates G' and $N = (G'_{x_0 y_0}, \mathbf{b}, \mathbf{c})$, where G is shown in Figure 4.16 (a).

Thus, each unit of (x_0, y_0)-flow $\mathbf{f}_0$ in N denotes an (x, y)-path P_0 in G, where $x \in X$ and $y \in Y$, $\mathbf{w}(P_0) = \mathbf{b}(\mathbf{f}_0)$. Since both $E^+_{G'}(x_0)$ and $E^-_{G'}(y_0)$ are (x_0, y_0)-cuts in G' and admit capacity $\rho(G)$, but all of other (x_0, y_0)-cuts admit capacity ∞, it follows that $E^+_{G'}(x_0)$ and $E^-_{G'}(y_0)$ are minimum (x_0, y_0)-cuts in G'. By Theorem 4.1, **there exists a maximum (x_0, y_0)-flow $\mathbf{f}$ with value $\mathrm{val}\,\mathbf{f} = \rho(G)$.** Thus,

finding a solution of (i) is reduced to finding a minimum-cost maximum-flow in N,

The latter has been solved by Klein's algorithm described in the preceding section.

Summing up the above statement, for a given weighted digraph $(G, \mathbf{w})$, we can describe Edmonds-Johnson algorithm for solving the Chinese postman problem as follows.

Edmonds-Johnson's Algorithm

1. Construct G' and $N = (G'_{x_0y_0}, \mathbf{b}, \mathbf{c})$;

2. Find a minimum-cost maximum-flow in N;

3. Construct a supergraph G^* of G;

4. Find an Euler directed circuit in G^*, which is an optimal postman tour in $(G, \mathbf{w})$.

Example 4.6.2 Consider the weighted digraph $(G, \mathbf{w})$ in Figure 4.16 (a). G' and $N = (G'_{x_0y_0}, \mathbf{b}, \mathbf{c})$ are shown in Figure 4.17 (a). A minimum-cost maximum-flow $\mathbf{f}$ is shown in Figure 4.17 (b), where $\mathbf{f}(a)$ denotes the times that the edge a appears in E^*. A supergraph G^* of G is shown in Figure 4.14 or Figure 4.16 (b); an optimal tour is

$$P = (x_1, x_2, x_3, x_4, x_5, x_2, x_4, x_5, x_3, x_4, x_5, x_1, x_3, x_4, x_5, x_1)$$

with weight $\mathbf{w}(P) = 44$. ∎

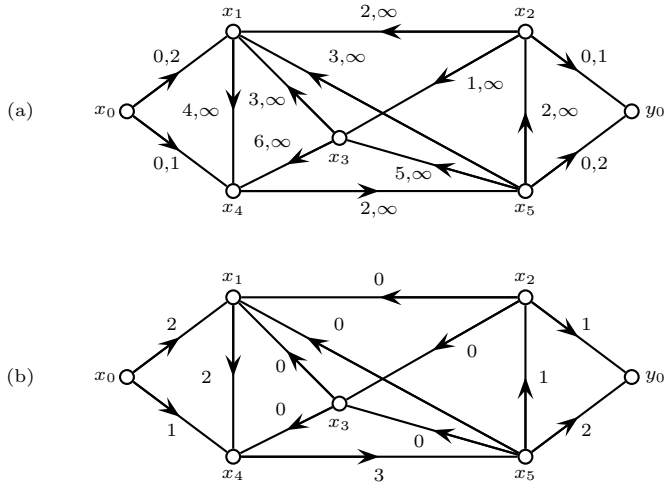

Figure 4.17: (a) $N = (G'_{x_0y_0}, \mathbf{b}, \mathbf{c})$; (b) a minimum-cost maximum-flow in N

It is not difficult to see the algorithm is efficient since execution of its each step can be completed in a polynomial time, the detail is left to the reader as an exercise (the exercise 4.6.3).

4.7 Construction of Squared Rectangles

In this section, we will introduce a famous combinatorial problem, squared rectangles (squares), arising from recreational mathematics whose solution is related to connectivity of planar graphs and theory of network flows.

A **squared rectangle (square)** is a rectangle (square) dissected into a finitely many (but at least two) squares. A squared rectangle is called to be **perfect** if no two of the squares in the dissection have the same size. The **order** of a squared rectangle is the number of squares into which it is dissected. A squared rectangle is **simple** if it does not contains a rectangle which is self squared. Clearly, every squared rectangle is composed of ones that are simple.

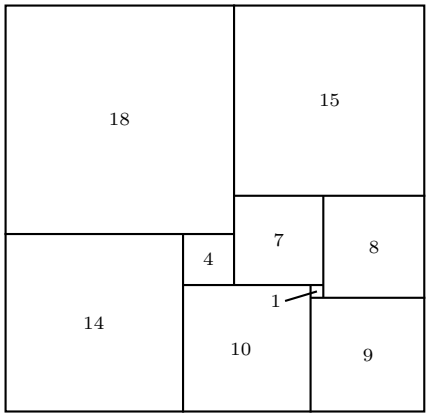

Figure 4.18: The perfect squaring of the 33×32 rectangle of order 9

Figure 4.18 shows a perfect rectangle of order 9, due to by Moroń (1925), in which the digit associated with a square is the length of its side.

Since the beginning of the 20th century, the problem of squared rectangles (squares) has received much attention. In 1940, with the aid of theory of graphs, Brooks *et al.* developed a systematic method for constructing of a squared rectangle, proved the lowest order of a perfect squared rectangle is 9 and gave a list of perfect squared rectangles of order 9 through 11. In 1964, Bouwkamp et al. gave a list of all perfect squared rectangles of order 9 through 18 by computers. The number of perfect squared rectangles of lower order is as follows.

Order	9	10	11	12	13	14	15	16	17	18
Number	2	6	22	67	213	744	2609	9016	31427	110384

For a long time no perfect square was known, and it was conjectured that such squares might not exist. In 1939, Sprague was the first to publish an example of a perfect 4205-square of order 55 obtained by a make-up of several known perfect squared rectangles. In 1940, in this manner Brooks *et al.* constructed a perfect 608-square of order 26. Wilson constructed a perfect 112-square of order 25 by a computer. In 1978, Duijvestijn found a perfect 112-square of order 21 also with the help of a computer, shown in Figure 4.19. There are none below order 21 and only one of order 21.

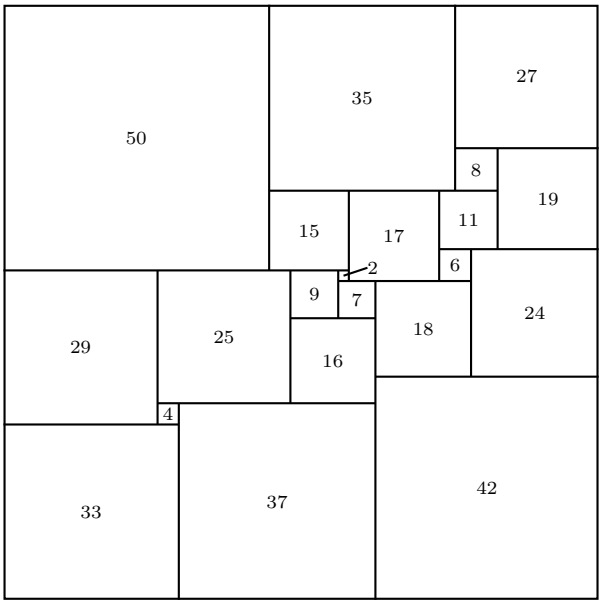

Figure 4.19: The perfect 112-square of order 21

The number of known perfect squares of lower order is as follows.

Order	21	22	23	24	25	26	27	28	29	30	31
Number	1	?	?	?	8	28	6	?	?	?	4

We now introduce the method of Brooks *et al.* for constructing a perfect squared rectangle.

We first show how a simple digraph D can be associated with a given squared rectangle R of order n. A horizontal line segment of the dissection of R is called a **horizontal dissector** of R. For example, In Figure 4.20 (a), the horizontal dissectors are indicated by solid line segments.

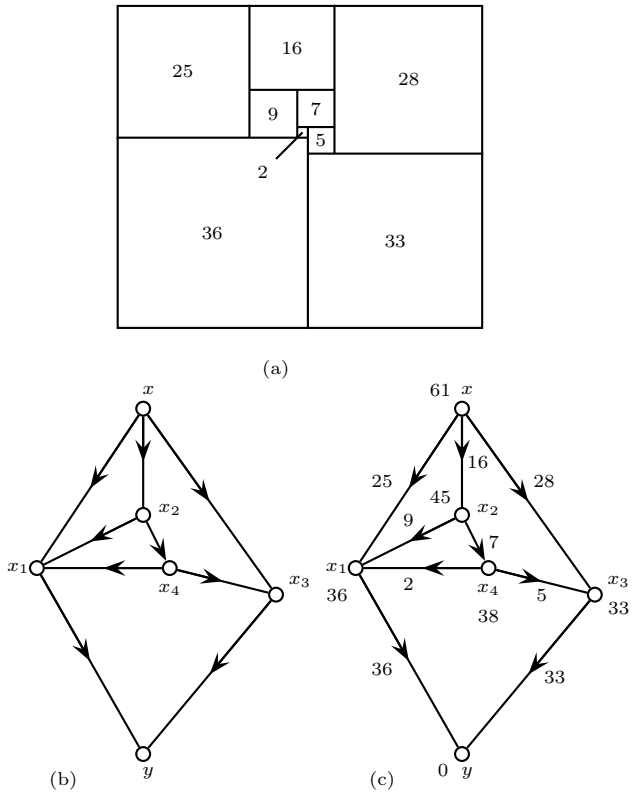

Figure 4.20: A squared rectangle and an associated digraph

Let $H_1, H_2, \cdots, H_m$ be all horizontal dissectors of R. Define a simple digraph $D = (V, E)$ as follows. $V = \{x_1, x_2, \cdots, x_m\}$ and $(x_i, x_j) \in E$ if and only if H_i and H_j are the upper and lower sides of some square of in R. The vertices corresponding to the upper and lower sides of R are denoted by x and y, respectively. Clearly, $\varepsilon(D) = n$. Figure 4.20 (b) shows the digraph D associated with the squared rectangle R in Figure 4.20 (a).

Define a vector $\mathbf{p} \in \mathscr{V}(D)$ with $\mathbf{p}(x_i)$ equal to the height (above the lower side of R) of the corresponding horizontal dissector, see Figure 4.20 (c). We can regard D as a network with enough large capacity and the source x and the sink y. It is

easily verified (the exercise 4.7.2) that the bond-vector $\mathbf{g} \in \mathscr{E}(D)$ defined by

$$\mathbf{g}(a) = \mathbf{p}(x_i) - \mathbf{p}(x_j), \quad \forall\ a = (x_i, x_j) \in E(D) \tag{4.14}$$

is an (x, y)-flow. See Figure 4.20 (c), for example, the digits nearby edges determine an (x, y)-flow $\mathbf{g}$ with value 69.

Let D be the digraph corresponding to a squared rectangle R, and let G be the underlying graph of D. The graph $G + xy$ is called the **horizontal graph** of R. Brooks *et al.* showed that the horizontal graph of any simple squared rectangle is a 3-connected planar graph. They also showed that, conversely, if H is a 3-connected planar graph and $xy \in E(H)$, then an (x, y)-flow defined by any bond-vector in $\mathscr{E}(H - xy)$ determines a squared rectangle. Thus, a possible way of searching for perfect rectangles of order n is to

1. list all 3-connected planar graphs with $n + 1$ edges, and

2. for each such graph H and each edge xy of H, determine an (x, y)-flow defined by a bond-vector in $\mathscr{E}(H - xy)$.

We now introduce how to compute such an (x, y)-flow in D. Suppose $\mathbf{g} \in \mathscr{B}(D)$ is an (x, y)-flow with value σ. Then

$$\sum_{a \in E(D)} m_x(a)\mathbf{g}(a) = \sigma, \tag{4.15}$$

and for any $x_i \in V(D) \setminus \{x, y\}$,

$$\sum_{a \in E(D)} m_{x_i}(a)\mathbf{g}(a) = 0. \tag{4.16}$$

By Theorem 2.7, $\mathbf{g}$ is orthogonal to every cycle-vector of $\mathscr{E}(D)$, that is,

$$\mathbf{C}\,\mathbf{g}^T = 0, \tag{4.17}$$

where $\mathbf{C}$ is a basis matrix of the cycle-space $\mathscr{C}(D)$ corresponding to a spanning tree T of D and $\mathbf{g}^T$ is the transpose of the vector $\mathbf{g}$. Equations (4.15) – (4.17) together give the matrix equation

$$\begin{pmatrix} \mathbf{K} \\ \mathbf{C} \end{pmatrix} \mathbf{g}^T = \begin{pmatrix} \sigma \\ \mathbf{O} \end{pmatrix}, \tag{4.18}$$

where $\mathbf{K}$ is the matrix obtained from the incidence matrix $\mathbf{M}$ of D by deleting the row m_y. The equation can be solved using Cramér's rule. Note that, since (the exercise 2.3.1)

$$\det \begin{pmatrix} \mathbf{K} \\ \mathbf{C} \end{pmatrix} = \pm\varsigma(D),$$

we can obtain an integral solution if $\sigma = \varsigma(D)$. Thus, in computing the (x, y)-flow $\mathbf{g}$, it is convenient to take $\operatorname{val}\mathbf{g} = \varsigma(D)$.

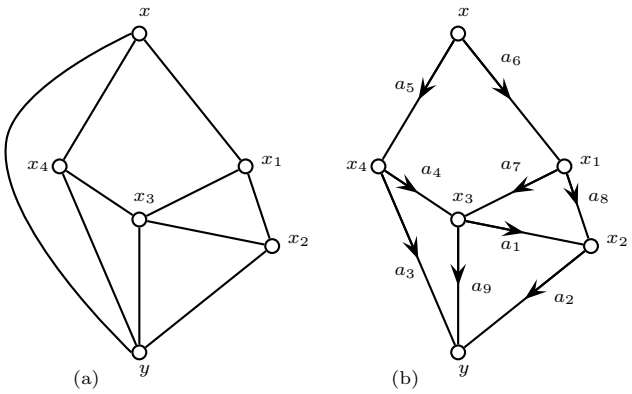

Figure 4.21: An illustration in Example 4.7.1

We illustrate the above procedure.

Example 4.7.1 Consider the 3-connected planar graph in Figure 4.21 (a). On deleting the edge xy and orienting each edge we obtain the digraph D of Figure 4.21 (b). The matrix

$$\mathbf{K} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

It has been computed in Example 2.3.3 that $\varsigma(D) = \det \mathbf{K}\mathbf{K}^T = 66$.

Choose the spanning tree T of D induced by the set of edges $\{a_5, a_6, a_7, a_8, a_9\}$. Then the basis matrix $\mathbf{C}$ of the cycle-space $\mathscr{C}(D)$ corresponding to a spanning tree T of D is

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 1 & 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 \end{pmatrix}.$$

Suppose that the required bond-vector $\mathbf{g} \in \mathscr{E}(D)$ is

$$\mathbf{g} = (g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9),$$

where

$$g_i = \mathbf{g}(a_i), \quad \text{for } i = 1, 2, \cdots, 9.$$

We obtain the following nine equations, as in (4.18):

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 1 & 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \\ g_8 \\ g_9 \end{pmatrix} = \begin{pmatrix} 66 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

The solution to this system of equations is given by

$$\mathbf{g} = (g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9) = (36, 30, 14, 16, 20, 2, 18, 28, 8).$$

The squared rectangle based on this (x, y)-flow is just the one in Figure 4.18 with all dimensions doubled.