derived interesting partial results. Let us pose two exercises regarding this problem.

Exercise 14.4.6. Extend Corollary 7.2.6 to regular bipartite multigraphs.

Exercise 14.4.7. Let G be a bipartite graph, and let $\mathbf{L}(G)$ be the lattice in $\mathbb{Z}^E$ generated by the incidence vectors of the perfect matchings of G, and $\mathbf{H}(G)$ the linear span of $\mathbf{L}(G)$ in $\mathbb{R}^E$. Show that $\mathbf{L}(G) = \mathbf{H}(G) \cap \mathbb{Z}^E$ [Lov85]. Hint: Use Exercise 14.4.6.

The result of Exercise 14.4.7 does not extend to arbitrary graphs, as shown by [Lov85]: the Petersen graph provides a counterexample. The general case is treated in [Lov87]. Related problems can be found in [JuLe88, JuLe89] and [Rie91], where lattices corresponding to the 2-matchings of a graph and lattices corresponding to the bases of a matroid are examined.

For some practical applications in which n is very large even algorithms for determining an optimal matching with complexity $O(n^3)$ are not fast enough; in this case, one usually resorts to approximation techniques. In general, these techniques will not find an optimal solution but just a reasonable approximation; to make up for this, they have the advantage of being much faster. We refer the interested reader to [Avi78, Avi83] and to [GrKa88]. Two alternatives to using heuristics for large values of n are either to use appropriate LP-relaxations to determine minimal perfect matchings on suitable sparse subgraphs, or to use post-optimization methods. We refer to [GrHo85] and to [DeMe91]; one of the best practical methods at present seems to be the one given in [ApCo93].

14.5 The Chinese postman

This section is devoted to an interesting application of optimal matchings in K_{2n}. The following problem due to Kwan [Kwa62] concerns a postman who has to deliver the mail for a (connected) system of streets: our postman wants to minimize the total distance he has to walk by setting up his tour suitably. This problem is nowadays generally known as the *Chinese postman problem.*

Problem 14.5.1 (Chinese postman problem, CPP). Let $G = (V, E)$ be a connected graph, and let $w\colon E \to \mathbb{R}_0^+$ be a length function on G. We want to find a closed walk C of minimal length $w(C)$ which contains each edge of G at least once.[6]

[6]Note that we view the edges of our graph as (segments of) streets here, and the vertices as intersections (or dead ends), so that each edge certainly needs to be traversed to deal with the houses in this street; in this rather simplistic model we neglect the need for having to cross the street to deliver the mail to houses on opposite sides. Hence it might be more realistic to consider the directed case and use the complete orientation of G; see Exercise 14.5.6.

If G should be Eulerian, the solution of the CPP is trivial: any Euler tour C will do the job. Recall that G is Eulerian if and only if each vertex of G has even degree (Theorem 1.3.1) and that an Euler tour C can then be constructed with complexity $O(|E|)$ (Example 2.5.2).

If G is not Eulerian, we use the following approach. Let X be the set of all vertices of G with odd degree. We add a set E' of edges to G such that the following three conditions are satisfied:

(a) Each edge $e' \in E'$ is parallel to some edge $e \in E$; we extend w to E' by putting $w(e') = w(e)$.
(b) In (V, E'), precisely the vertices of X have odd degree.
(c) $w(E')$ is minimal: $w(E') \leq w(E'')$ for every set E'' satisfying (a) and (b).

Then $(V, E \,\dot\cup\, E')$ is an Eulerian multigraph, and any Euler tour induces a closed walk of minimal length $w(E)+w(E')$ in G. It is rather obvious that any solution of CPP can be described in this way. We now state – quite informally – the algorithm of Edmonds and Johnson [EdJo73] for solving the CPP. Note that $|X|$ is even by Lemma 1.1.1.

Algorithm 14.5.2. Let $G = (V, E)$ be a connected graph with a length function $w\colon E \to \mathbb{R}_0^+$.

Procedure CPP$(G, w; C)$

(1) $X \leftarrow \{v \in V\colon \deg v \text{ is odd}\}$;
(2) Determine $d(x, y)$ for all $x, y \in X$.
(3) Let H be the complete graph on X with weight function $d(x, y)$. Determine a perfect matching M of minimal weight for (H, d).
(4) Determine a shortest path W_{xy} from x to y in G and, for each edge in W_{xy}, add a parallel edge to G (for all $xy \in M$). Let G' be the multigraph thus defined.
(5) Determine an Euler tour C' in G' and replace each edge of C' which is not contained in G by the corresponding parallel edge in G. Let C be the closed walk in G arising from this construction.

Step (2) can be performed using Algorithm 3.8.1; however, if $|X|$ is small, it might be better to run Dijkstra's algorithm several times. Determining shortest paths explicitly in step (4) can be done easily by appropriate modifications of the algorithms already mentioned; see Exercise 3.8.3 and 3.6.3. In the worst case, steps (2) and (4) need a complexity of $O(|V|^3)$. Step (3) can be executed with complexity $O(|X|^3)$ by Result 14.4.5; note that determining a perfect matching of minimal weight is equivalent to determining an optimal matching for a suitable auxiliary weight function; see Section 14.1. Finally, step (5) has complexity $O(|E'|)$ by Example 2.5.2. Thus we get a total complexity of $O(|V|^3)$.

It still remains to show that the algorithm is correct. Obviously, the construction in step (4) adds, for any matching M of H, a set E' of edges to G which satisfies conditions (a) and (b) above; the closed walk in G arising from this construction has length $w(E) + d(M)$, where $d(M)$ is the weight of M with respect to d. Thus it is reasonable to choose a matching M of minimal weight in step (3). However, it is not immediately clear that there cannot be some other set E' of edges leading to a solution of even smaller weight. We need the following lemma.

Lemma 14.5.3. *Let $G = (V, E)$ be a connected graph with length function $w\colon E \to \mathbb{R}_0^+$. Moreover, let H be the complete graph on a subset X of V of even cardinality; the edges of H are assigned weight $d(x, y)$, where d denotes the distance function in G with respect to w. Then, for each perfect matching M of H with minimal weight and for each subset E_0 of E for which any two vertices of X have the same distance in G and in (V, E_0), the inequality $d(M) \le w(E_0)$ holds.*

Proof. Let $M = \{x_1y_1, \dots, x_ny_n\}$ be a perfect matching with minimal weight in H. Then $d(M) = d(x_1, y_1) + \ldots + d(x_n, y_n)$. Moreover, let P_i be a shortest path from x_i to y_i in (V, E_0) (for $i = 1, \dots, n$). By hypothesis, $w(P_i) = d(x_i, y_i)$. We claim that no edge e with $w(e) \ne 0$ can be contained in more than one of the paths P_i; if we prove this claim, the assertion of the lemma follows. Suppose our claim is wrong. Then we may assume

$$P_1 = x_1 \xrightarrow{P_1'} u \xrightarrow{e} v \xrightarrow{P_1''} y_1 \quad \text{and} \quad P_2 = x_2 \xrightarrow{P_2'} u \xrightarrow{e} v \xrightarrow{P_2''} y_2,$$

which implies

$$\begin{aligned} d(x_1, y_1) + d(x_2, y_2) &= d(x_1, u) + w(e) + d(v, y_1) + d(x_2, u) + w(e) + d(v, y_2) \\ &> d(x_1, u) + d(u, x_2) + d(y_1, v) + d(v, y_2) \\ &\ge d(x_1, x_2) + d(y_1, y_2). \end{aligned}$$

But then replacing x_1y_1 and x_2y_2 by x_1x_2 and y_1y_2 in M would yield a perfect matching of smaller weight, a contradiction. □

Theorem 14.5.4. *Algorithm 14.5.2 calculates with complexity $O(|V|^3)$ a solution of the CPP.*

Proof. We already know that Algorithm 14.5.2 yields a closed walk of length $w(E) + d(M)$ containing each edge of G, where $d(M)$ is the minimal weight of a perfect matching of (H, d).

Now suppose that E' is an arbitrary set of edges satisfying conditions (a) to (c). Then E' induces a closed walk of weight $w(E) + w(E')$ which contains all edges of G. We have to show $w(E') \ge d(M)$. Suppose Z is a connected component of (V, E') containing at least two vertices. Then we must have $Z \cap X \ne \emptyset$: otherwise, we could omit all edges of E' which are contained in Z and the remaining set of edges would still satisfy (a) and (b). As X is the set of vertices of (V, E') with odd degree, $|Z \cap X|$ has to be even by Lemma 1.1.1. Thus the connected components of (V, E') induce a partition $X_1, \dots, X_k$ of

X into sets of even cardinality so that any two vertices in X_i are connected by a path in E'.

Let $x, y \in X_i$, and let P_{xy} be the path from x to y in E'. Then P_{xy} must be a shortest path from x to y in G: otherwise, the edges of P_{xy} could be replaced by the edges of a shortest path from x to y, which would yield a set E'' of edges satisfying (a) and (b) and $w(E'') < w(E')$. Now, trivially, P_{xy} is also a shortest path from x to y in (V, E'). Denote the connected component of (V, E') corresponding to X_i by Z_i, and let E'_i be the set of edges of E' which have both end vertices in Z_i. Moreover, let H_i be the complete graph on Z_i with weights $d(x, y)$ (where d is the distance function in G or in (Z_i, E'_i)). Then Lemma 14.5.3 yields $d(M_i) \leq w(E'_i)$ for each perfect matching M_i of minimal weight in H_i. Obviously, $M_1 \cup \ldots \cup M_k$ is a perfect matching of H, and $E' = E'_1 \cup \ldots \cup E'_k$. Hence we obtain the desired inequality

$$w(E') \; = \; w(E'_1) + \ldots + w(E'_k) \; \geq \; d(M_1) + \ldots + d(M_k) \; \geq \; d(M). \quad \square$$

Example 14.5.5. Let G be the graph displayed in Figure 14.6. Then $X = \{x, y, z, w\}$, so that we get the complete graph H shown in Figure 14.7. The edges xw and yz form a perfect matching of minimal weight of H; the corresponding paths are (x, a, w) and (y, x, z). Hence we replace the corresponding edges in G by two parallel edges each; this yields the multigraph G' in Figure 14.8. Now it is easy to find an Euler tour in G', for example $(x, y, b, w, c, z, x, y, a, x, a, w, a, z, x)$ with length $30 + 4 = 34$.

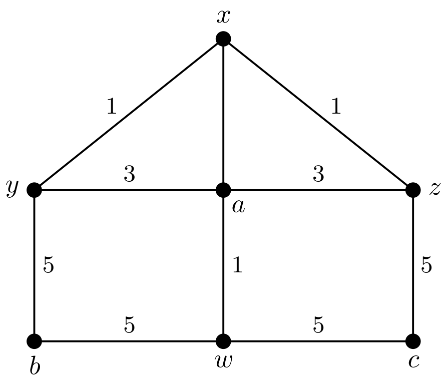

Fig. 14.6. A graph

Exercise 14.5.6. We consider the directed version of the CPP: let G be a digraph with a nonnegative length function w; we want to find a directed closed walk of minimal length containing each edge of G at least once. Hint: Reduce this problem to the problem of determining an optimal circulation [EdJo73].

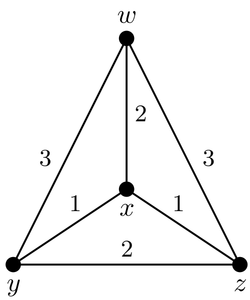

Fig. 14.7. The complete graph H

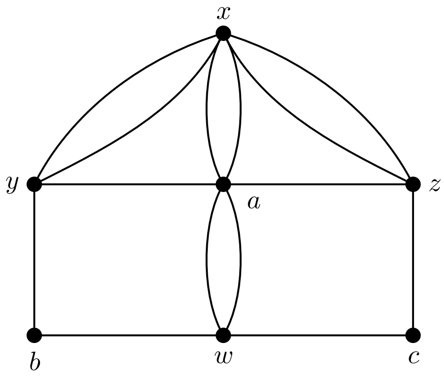

Fig. 14.8. The corresponding Eulerian multigraph

Theorem 14.5.4 and Exercise 14.5.6 (together with a corresponding algorithm for determining an optimal circulation) show that there are good algorithms for the CPP for directed graphs as well as for undirected graphs. In contrast, the CPP for mixed graphs is NP-complete, so that most likely there is no polynomial solution; see [Pap76] or [GaJo79]. A cutting plane algorithm for the mixed CCP is in [NoPi96], and some applications of the CPP are discussed in [Bar90].

14.6 Matchings and shortest paths

This section deals with applications of matchings to the problem of determining shortest paths in a network on an *undirected* graph without cycles of negative length. We remind the reader that our usual transformation to the directed case – replacing a graph G by its complete orientation – will not work in this situation, because an edge $e = \{u, v\}$ of negative weight $w(e)$ in (G, w) would yield a directed cycle $u \longrightarrow v \longrightarrow u$ of negative length $2w(e)$ in $(\vec{G}, w)$, whereas all the algorithms given in Chapter 3 apply only to graphs withouts such cycles. We describe a solution for this path problem below; it is due to Edmonds [Edm67a].

The first step consists of transforming the given problem to the problem of determining an f-factor in an appropriate auxiliary graph; this problem was already mentioned at the end of Section 13.5. In our case, the only values $f(v)$ will take are 1 and 2; however, the auxiliary graph might contain loops. Note that a loop $\{v, v\}$ adds 2 to the degree $\deg v$ of a vertex v. In what follows, we call a path from s to t an *$\{s, t\}$-path.*

Lemma 14.6.1. *Let $N = (G, w)$ be a network on a graph $G = (V, E)$ with respect to a weight function $w \colon E \to \mathbb{R}$, and assume that there are no cycles of negative length in N. Let s and t be two vertices of G, and let G' be the graph which results from adding the loop $\{v, v\}$ to G for each vertex $v \neq s, t$. Extend the weight function w to G' by putting $w(v, v) = 0$. Then each $\{s, t\}$-path P in G may be associated with an f-factor $F = F(P)$ in G', where f is given by*

$$f(s) = f(t) = 1 \quad \textit{and} \quad f(v) = 2 \textit{ for all } v \neq s, t, \tag{14.9}$$

so that the weight of P always equals that of the corresponding f-factor F. Moreover, the problem of determining a shortest $\{s, t\}$-path in (G, w) is equivalent to determining a minimal f-factor in (G', w).

Proof. Given an $\{s, t\}$-path P in G, put

$$F \;=\; P \cup \{\{v, v\} : v \text{ is not contained in } P\}\,.$$

Obviously, F is an f-factor for G', as the loop $\{v, v\}$ increases the degree of v in F to 2 whenever v is not contained in P. By our definition of w for loops, $w(F) = w(P)$.

Conversely, let F be an f-factor for G'; we want to construct an $\{s, t\}$-path P from F. As s has degree 1 in F, there is exactly one edge sv_1 in F. Now v_1 has degree 2 in F, so that there exists precisely one further edge in F incident with v_1, say v_1v_2; note that this edge cannot be a loop. Continuing in this manner, we construct the edge sequence of a path P with start vertex s in G. As the only other vertex of degree 1 in F is t, t must be the end vertex of P.

Note that it is quite possible that there are not only loops among the remaining edges of F: these edges might contain one or more cycles. In other words, in general we will have $F \neq F(P)$, so that the correspondence given above is not a bijection. However, our assumption that there are no cycles of negative length in (G, w) guarantees at least $w(P) \leq w(F)$, which proves the final assertion. □

Next we show how one may reduce the determination of a minimal f-factor for the special case where $f(v) \in \{1, 2\}$ to the determination of a minimal perfect matching in an appropriate auxiliary graph whose size is polynomial in the size of the original graph. As already mentioned in Section 13.5, the general existence problem for arbitrary f-factors can be reduced to the general existence problem for perfect matchings; see [Tut54].

Lemma 14.6.2. *Let $G = (V, E)$ be a graph (where loops are allowed), and let $f : V \to \mathbb{N}$ be a function with $f(v) \in \{1, 2\}$ for all $v \in V$. Then the f-factors of G correspond to perfect matchings of a suitable auxiliary graph H with at most $5|E|$ edges and at most $2|V| + 2|E|$ vertices. If there also is a weight function $w : E \to \mathbb{R}$ on G given, a weight function w on H can be defined in such a way that the weight $w(F)$ of an f-factor F is always equal to the weight $w(M)$ of the corresponding perfect matching M.*

Proof. Our transformation consists of two steps. First, the given f-factor problem for G is transformed to an equivalent problem for an auxiliary graph H' for which each non-loop edge is incident with at least one vertex v satisfying $f(v) = 1$. Thus let $e = uv \in E$ be an edge with $u \neq v$ and $f(u) = f(v) = 2$. We subdivide e by introducing two new vertices u_e, v_e; replace the edge e by the path

$$P_e : \ u \longrightarrow u_e \longrightarrow v_e \longrightarrow v;$$

and extend f by putting $f(u_e) = f(v_e) = 1$. By performing this operation for all edges $e = uv$ with $f(u) = f(v) = 2$ and $u \neq v$, we obtain the desired graph H'. Now let F be an f-factor in G. Then F yields an f-factor F' in H' as follows: we replace each edge $e = uv \in F$ with $f(u) = f(v) = 2$ and $u \neq v$ by the edges uu_e and vv_e; moreover, we add for each edge $e = uv$ with $f(u) = f(v) = 2$ and $u \neq v$ which is not in F the edge $u_e v_e$ to F'. Under this operation, each f-factor in H' actually corresponds to an f-factor in G. We can also make sure that the weights of corresponding f-factors F and F' are equal: for each edge $e = uv$ with $f(u) = f(v) = 2$ and $u \neq v$, we define the weights of the edges on P_e as

$$w(uu_e) \ = \ w(vv_e) \ = \ \frac{w(e)}{2} \quad \text{and} \quad w(u_e v_e) \ = \ 0.$$

In the second step of the transformation, we define a graph H which results from H' by splitting each vertex v with $f(v) = 2$ into two vertices:[7] we replace v by two vertices v' and v''; we replace each edge $e = uv$ with $u \neq v$ by two edges $e' = uv'$ and $e'' = uv''$; finally, each loop $\{v, v\}$ with $f(v) = 2$ is replaced by the edge $v'v''$. These operations are well-defined because of the transformations performed in the first step: H' does not contain any edges $e = uv$ with $f(u) = f(v) = 2$ and $u \neq v$. Let us denote the resulting graph by H.

It is now easy to see that the f-factors F' of H' correspond to the perfect matchings M of H. Note that at most one of the two parts of a split edge

[7]Note that this condition can hold only for *old* vertices, that is, vertices which were contained in G.

$e = uv$ (with $f(v) = 2$) can be contained in a perfect matching M of H, since we must have $f(u) = 1$ in that case. Note that this correspondence between f-factors and perfect matchings is, in general, not bijective: if F' contains two edges $e_1 = u_1v$ and $e_2 = u_2v$ (where $f(v) = 2$ and $f(u_1) = f(u_2) = 1$), M might contain either u_1v' and u_2v'' or u_1v'' and u_2v'. Thus, in general, there are several perfect matchings of H which correspond to the same f-factor of H'. However, the weights of corresponding f-factors and perfect matchings agree if we put

$$w(e') \;=\; w(e'') \;=\; w(e)$$

for split edges e' and e''. □

By performing the transformations of Lemmas 14.6.1 and 14.6.2 successively, we obtain the desired reduction of the determination of a shortest path between two vertices s and t in an undirected network (G, w) without cycles of negative length to the determination of a perfect matching of minimal weight in an appropriate auxiliary graph H (with respect to a suitable weight function). As the number of vertices of H is linear in the number of vertices of G, Result 14.4.5 yields the following conclusion.

Theorem 14.6.3. *Let $N = (G, w)$ be a network on a graph $G = (V, E)$, where $w\colon E \to \mathbb{R}$, and let s and t be two vertices of G. If N does not contain cycles of negative length, one may determine with complexity $O(|V|^3)$ a shortest path from s to t.* □

Example 14.6.4. Consider the network (G, w) given in Figure 14.9. The bold edges form a path

$$P:\; s \text{ --- } c \text{ --- } b \text{ --- } t$$

of length $w(P) = 0$, which corresponds to the f-factor

$$F \;=\; \{\{a, a\}, sc, cb, bt\}$$

of weight $w(F) = 0$ in the graph G' shown in Figure 14.10, where $f(a) = f(b) = f(c) = 2$ and $f(s) = f(t) = 1$. Again, F consists of the bold edges.

Now we perform the transformations of Lemma 14.6.2. First, when H' is constructed, the edges $e = ab$ and $g = bc$ are divided into paths of length 3. We obtain the auxiliary graph H' with the f-factor

$$F' \;=\; \{\{a, a\}, sc, cc_g, bb_g, bt, a_eb_e\}$$

corresponding to F, where $f(a) = f(b) = f(c) = 2$ and $f(v) = 1$ for all other vertices v. Note that F' indeed has weight $w(F') = 0$. Figure 14.11 shows H' and F'; as usual, F' consists of the bold edges.

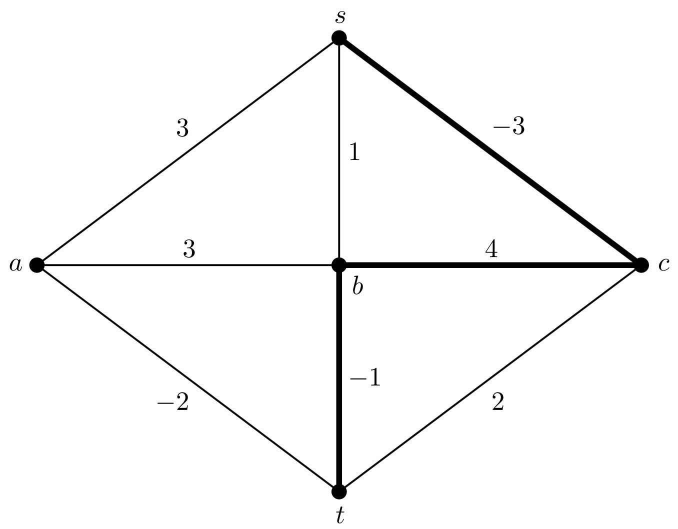

Fig. 14.9. A path in G

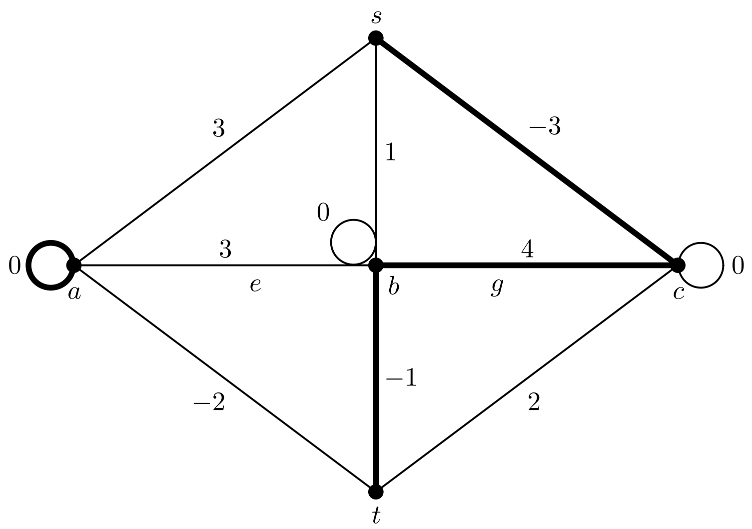

Fig. 14.10. The corresponding f-factor in G'

Finally, in the second step of the transformation, the three vertices a, b, c with $f(a) = f(b) = f(c) = 2$ are divided into two vertices each. This yields the graph H shown in Figure 14.12 and the perfect matching

$$K = \{aa', sc', c''c_g, b''b_g, b't, a_eb_e\}$$

of weight $w(K) = 0$ corresponding to the f-factor F'.

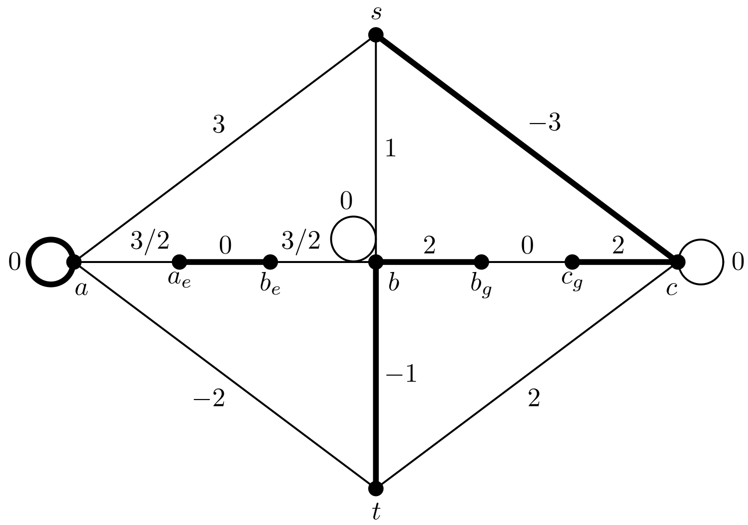

Fig. 14.11. The corresponding f-factor in H'

Exercise 14.6.5. Determine an $\{s, t\}$-path of shortest length as well as the corresponding f-factors and a corresponding perfect matching of minimal weight for the network of Example 14.6.4.

Exercise 14.6.6. Discuss the transformation method given above for the case in which (G, w) contains cycles of negative length. What will go wrong then?

Now consider a network (G, w) on a digraph G which does not contain directed cycles of negative length. Then the problem of determining a shortest directed path from s to t can be transformed to the problem of determining a perfect matching of minimal weight in a bipartite graph – that is, to the assignment problem; see [HoMa64] and also [AhMO93, Chapter 12.7]. As we have already seen two efficient algorithms for determining shortest paths for this case in Chapter 3, we will not present this transformation here. In practice, the reverse approach is more common: the assignment problem is often solved using the SP-problem (without negative weights) as an auxiliary procedure.

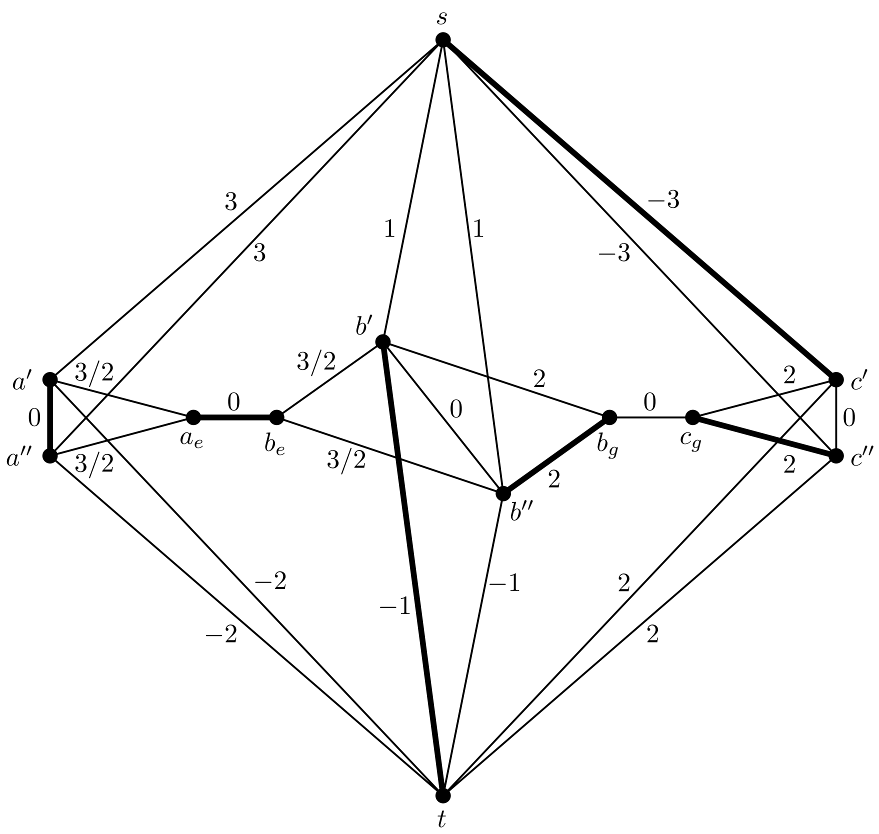

Fig. 14.12. A corresponding perfect matching in H

We conclude this section with one more application of matching theory to a problem concerning shortest paths, which is taken from [Gro85]. Consider a network $N = (G, w)$ on a graph G, where w is a nonnegative weight function. Let us call a path P in G *odd* if P contains an odd number of edges, so that P has odd length in the graph theoretical sense; *even* paths contain an even number of vertices.

We want to find a shortest odd path between two given vertices s and t. This problem can be reduced to determining a perfect matching of minimal weight in a suitable auxiliary graph G', which again results from G by splitting vertices: each vertex $v \neq s, t$ of G is replaced by two vertices v' and v'', and an edge $v'v''$ of weight $w(v'v'') = 0$ is added to E. Moreover, each edge of G of the form sv or tv is replaced by the edge sv' or tv', respectively; and each edge uv with $u, v \neq s, t$ is replaced by two edges $u'v'$ and $u''v''$. Using similar arguments as for the proofs of Lemmas 14.6.1 and 14.6.2, one obtains the following result; the details will be left to the reader as an exercise.

Theorem 14.6.7. *Let $N = (G, w)$ be a network on a graph G, where w is a nonnegative weight function. Moreover, let s and t be two vertices of G, and let G' be the auxiliary graph described above. Then the odd $\{s, t\}$-paths P in G correspond bijectively to the perfect matchings M in G', and the length of P is equal to the weight of the matching M corresponding to P under this bijection. In particular, the shortest odd $\{s, t\}$-paths correspond bijectively to the perfect matchings of minimal weight in G'.* □

Example 14.6.8. Let (G, w) be the network shown in Figure 14.13, where all edges $e \in E$ have weight $w(e) = 1$. Then the bold edges form an $\{s, t\}$-path

$$P: \ s \longrightarrow u \longrightarrow v \longrightarrow t$$

of length 3, which corresponds to the perfect matching

$$K = \{su', u''v'', v't, a'a'', b'b'', c'c''\}$$

in the auxiliary graph G'; see Figure 14.14.

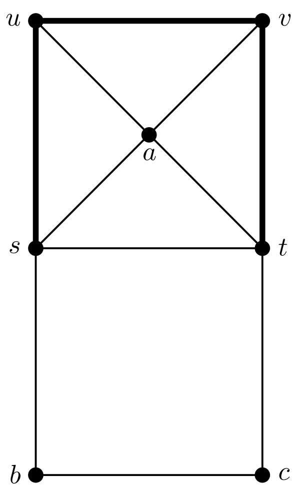

Fig. 14.13. A path of odd length in G

Exercise 14.6.9. Find a transformation similar to the one used in Theorem 14.6.7 which allows to find a shortest even $\{s, t\}$-path in (G, w) and apply this transformation to Example 14.6.8.

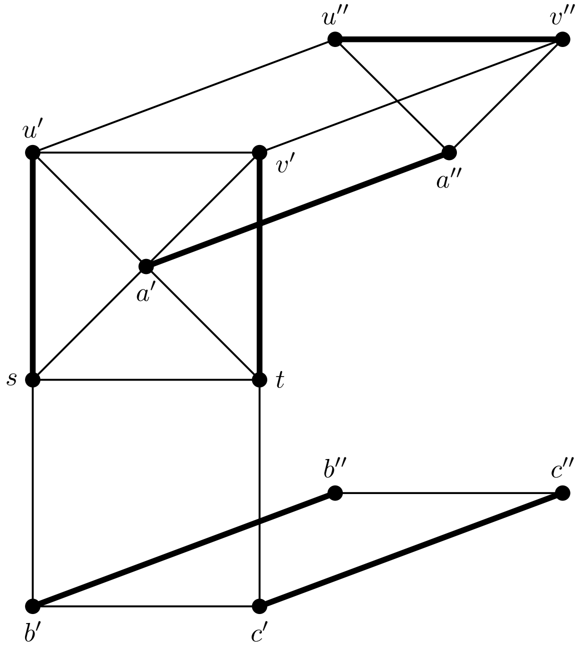

Fig. 14.14. The corresponding perfect matching in G'

14.7 Some further problems

In this final section of the chapter, we briefly mention some further problems concerning matchings, beginning with problems with side constraints. Such problems occur in practice, for example, when planning the schedules for bus drivers, when designing school time tables, or even when analyzing biomedical pictures; see [Bal85], [EvIS76], and [ItRo78]. We restrict our attention to rather simple – or at least seemingly simple – types of side constraints.

Problem 14.7.1 (restricted perfect matching, RPM). Let $G = (V, E)$ be a graph, and let $E_1, \dots, E_k$ be subsets of E and $b_1, \dots, b_k$ be positive integers. Does there exist a perfect matching M of G satisfying the conditions

$$|M \cap E_i| \ \leq \ b_i \qquad \text{for } i = 1, \dots, k? \tag{14.10}$$

If we want to fix the number k of constraints, we use the notation RPMk.

Exercise 14.7.2. Show that RPM1 can be solved with complexity $O(|V|^3)$ [ItRo78]. Hint: Reduce the problem to the determination of an optimal matching for the complete graph H on V with respect to a suitable weight function.