MATHEMATICAL ENGINEERING TECHNICAL REPORTS

A Weighted $K_{t,t}$-Free t-Factor Algorithm for Bipartite Graphs

Kenjiro TAKAZAWA

(Communicated by Kazuo MUROTA)

METR 2008–07 February 2008

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: http://www.keisu.t-u-tokyo.ac.jp/research/techrep/index.html

A Weighted $K_{t,t}$-Free t-Factor Algorithm for Bipartite Graphs

Kenjiro TAKAZAWA*†

February, 2008

Abstract

For a simple bipartite graph and an integer $t \geq 2$, we consider the problem of finding a minimum-weight t-factor under the restriction that it contains no complete bipartite graph $K_{t,t}$ as a subgraph. When $t = 2$, this problem amounts to the minimum-weight square-free 2-factor problem in a bipartite graph, which is NP-hard. We propose, however, a strongly polynomial algorithm for a certain case where the weight vector is vertex-induced on any subgraph isomorphic to $K_{t,t}$. The algorithm adapts the unweighted algorithms of Hartvigsen and Pap, and a primal-dual approach to the minimum-cost flow problem. The algorithm is fully combinatorial, and thus provides a dual integrality theorem, which is tantamount to Makai's theorem dealing with maximum-weight restricted t-matchings.

1 Introduction

Let $G = (V, E)$ be a simple undirected graph, that is, G has neither parallel edges nor self-loops. Throughout this paper, we assume that the given graphs are simple. For a vector $b \in \mathbf{Z}_+^V$, an edge set $M \subseteq E$ is said to be a *b-matching* if every vertex $v \in V$ is incident to at most $b(v)$ edges in M, and a *b-factor* if every vertex $v \in V$ is incident to exactly $b(v)$ edges in M. If $b(v) = t$ for every $v \in V$, we simply refer to b-matchings/factors as t-matchings/factors. For instance, a 2-matching is a vertex-disjoint collection of cycles and paths, and a 2-factor is a vertex-disjoint collection of cycles that cover all vertices in V. If a b-factor exists in a graph, it is a maximum b-matching.

---

*Department of Mathematical Informatics, Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan, and Research Institute for Mathematical Sciences, Kyoto University, Kyoto 606-8502, Japan. E-mail: takazawa@misojiro.t.u-tokyo.ac.jp.

†Supported by Grant-in-Aid for JSPS Fellows.

Let us denote a cycle of length k by C_k. For a b-matching/factor M with $b(v) \leq 2$ for each $v \in V$, we say that M is *C_k-free* if M contains no cycles of length k or less. The C_k-free 2-factor problem is to find a C_k-free 2-factor in a given graph. Note that the case where $k \leq 2$ is exactly the classical simple 2-factor problem, which can be solved efficiently.

One important aspect of the C_k-free 2-factor problem is that it is a relaxation of the Hamilton cycle problem. From this point of view, it is easily seen that this problem is NP-hard when $|V|/2 \leq k \leq |V| - 1$. Moreover, Papadimitriou showed that the problem is NP-hard when $k \geq 5$ (see [1]). On the other hand, for the case where $k = 3$, an augmenting path algorithm is given by Hartvigsen [11]. The C_4-free 2-factor problem is left open.

The weighted C_k-free 2-factor problem is to find a C_k-free 2-factor that minimizes the total weight of its edges for a given weighted graph. The problem is NP-hard when $k \geq 5$, which follows from the NP-hardness of the unweighted problem, and so is the case where $k = 4$ [23]. The weighted C_3-free 2-factor problem is unsettled. Polyhedral structures of C_k-free 2-factors are studied in Cunningham and Wang [3]. Related works also appear in [2, 14, 20].

We now focus on bipartite graphs. Note that it suffices to consider the cases where k is even. While the C_6-free 2-factor problem in bipartite graphs is NP-hard [10], C_4-free 2-factors in bipartite graphs are tractable and studied actively. We remark that a C_4-free 2-matching/factor in a bipartite graph, which is a 2-matching/factor that does not contain C_4, is often referred to as a *square-free 2-matching/factor*. Since the complement of an $(n - 3)$-connected bipartite graph is a square-free 2-matching, the theory of square-free 2-matching can be applied to the vertex-connectivity augmentation problem.

The first result on square-free 2-matchings was due to Hartvigsen [12], who proposed a characterization of graphs that admit square-free 2-factors and a combinatorial algorithm for finding one. This was followed by a min-max formula by Z. Király [15]. Then Hartivgsen [13], the journal version of [12], presented a full description of the algorithm and a constructive proof for the min-max formula.

As for the weighted C_k-free 2-factor problem in bipartite graphs, the NP-hardness when $k \geq 6$ follows from that of the unweighted problem. Moreover, Z. Király proved that the weighted square-free 2-factor problem is also NP-hard (see [6]).

An attractive generalization of the square-free 2-factor problem is the $K_{t,t}$-free t-factor problem, proposed by Frank [6]. In a bipartite graph, t-matching/factor is said to be *$K_{t,t}$-free* if it contains no $K_{t,t}$ as a subgraph. Note that the case where $t = 2$ is exactly the square-free 2-factor problem. Using a general framework of Frank and Jordán [7] on covering crossing bi-supermodular functions on pairs of sets, Frank [6] provided a min-max formula for $K_{t,t}$-free t-matchings, which extends Z. Király's formula [15] for the spacial case of $t = 2$. Through this approach, one can compute the size of the maximum $K_{t,t}$-free t-matching in polynomial time by the ellipsoid method or a combinatorial method by Fleiner [5]. Moreover, one can find a maximum $K_{t,t}$-free t-matching combinatorially by applying Végh and Benczúr's algorithm for covering pairs of sets [22]. A direct approach to this problem was done by Pap [17, 18, 19]. He gave a combinatorial proof for Frank's min-max formula, which implies a polynomial-time algorithm. We remark that applying Pap's algorithm to the case when $t = 2$ results in an algorithm different from Hartvigsen's algorithm.

The weighted $K_{t,t}$-free t-factor problem in bipartite graphs has also been considered. As mentioned, this problem is NP-hard when $t = 2$. However, Makai [16] showed a linear programming description of maximum weight $K_{t,t}$-free t-matchings and proved its dual integrality for a certain class of weight vectors called *vertex-induced*. For a weight vector $w \in \mathbf{R}^E$ and a subgraph H of G, w is said to be vertex-induced on H if there exists a function $\pi_H : V(H) \to \mathbf{R}$ such that $w(uv) = \pi_H(u)+\pi_H(v)$ for every $uv \in E(H)$. Here, $V(H)$ and $E(H)$ denote the vertex set and edge set of H, respectively, and uv denotes an edge connecting $u, v \in V(H)$. The class considered by Makai [16] is that w is vertex-induced on any subgraph isomorphic to $K_{t,t}$. Applying the ellipsoid method to Makai's description, one obtains a polynomial algorithm for this class of weighted bipartite graphs, which could be made strongly polynomial by Frank and Tardos' method [8].

This paper presents a combinatorial primal-dual algorithm to find a minimum-weight $K_{t,t}$-free t-factor in a weighted bipartite graph whose weight vector is vertex-induced on any subgraph isomorphic to $K_{t,t}$. The primal part of the algorithm is a variant of Hartvigsen's and Pap's algorithms, while the dual part is based on the framework of a primal-dual approach to the minimum-cost flow problem [4, 21]. The algorithm is fully combinatorial, so the output of the algorithm is integer if the weight vector is integer. Thus, the algorithm implies a theorem on dual integrality of an LP-formulation for the problem, which is tantamount to Makai's one [16]. The complexity of the algorithm is $\mathrm{O}(tn^2D)$, where n is the number of vertices and D is the time to execute a shortest path algorithm with nonnegative length. Incorporating Fredman and Tarjan's implementation of Dijkstra's algorithm [9], we get a strongly polynomial complexity $\mathrm{O}(tn^2m + tn^3 \log n)$, where m is the number of edges.

This paper is organized as follows. Section 2 describes a maximum-cardinality square-free 2-matching algorithm, and Section 3 extends it to a minimum-weight square-free 2-factor algorithm. Section 4 provides a further extension of the algorithm to the weighted $K_{t,t}$-free t-factor problem. Finally, Section 5 discusses the relation between minimum-weight $K_{t,t}$-free t-factors and maximum-weight $K_{t,t}$-free t-matchings.

Before closing this section, let us prepare some notations and definitions used in the following sections. Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. An edge connecting $u, v \in V$ is denoted by uv. For a vertex $v \in V$, $\delta v \subseteq E$ denotes the set of edges incident to v. For $Z \subseteq V$, the subgraph induced by Z is denoted by $G[Z] = (Z, E[Z])$, that is, $E[Z] = \{uv \mid u, v \in Z,\ uv \in E\}$. For a subgraph H of G, $V(H)$ and $E(H)$ denote the vertex set and edge set of H, respectively, i.e., $H = (V(H), E(H))$. A cycle is a subgraph $(\{v_1, \dots, v_k\}, \{e_1, \dots, e_k\})$ where $v_i \neq v_j$ if $i \neq j$, $e_i = v_i v_{i+1}$ for $i = 1, \dots, k-1$ and $e_k = v_k v_1$. A cycle consisting of k edges is denoted by C_k.

When we denote a graph by $G = (U, V; E)$, we mean that G is bipartite, that is, the vertex set and edge set of G are $U \cup V$ and E, respectively, and $E[U]$ and $E[V]$ are empty. For subgraph H of G, $U(H)$ (resp. $V(H)$) denotes the set of vertices in U (resp. V) that belong to H. A complete bipartite graph $K_{s,t}$ is a simple bipartite graph $(U, V; E)$ with $|U| = s$, $|V| = t$ and $E = \{uv \mid u \in U,\ v \in V\}$. Recall that $K_{2,2}$ is isomorphic to C_4, and is often called a *square*. For a subgraph H of G, a component in H isomorphic to $K_{2,2}$ is called a *square-component* and the number of square-components in H is denoted by $c(H)$. For a bipartite graph G, let $\mathcal{S}_t$ denote the family of all its subgraphs isomorphic to $K_{t,t}$. We often abbreviate $\mathcal{S}_2$ as $\mathcal{S}$.

For a directed graph $G = (V, A)$ with vertex set V and edge set A, we denote an edge e from u to v by uv, as far as it causes no confusion whether e is directed or undirected. For $e = uv \in A$, the initial and terminal vertex of e are denoted by $\partial^+ e$ and $\partial^- e$, respectively, that is, $\partial^+ e = u$ and $\partial^- e = v$. A path is a subgraph $(\{v_1, \dots, v_k\}, \{e_1, \dots, e_{k-1}\})$ where $v_i \neq v_j$ if $i \neq j$ and $e_i = v_i v_{i+1}$ for $i = 1, \dots, k-1$.

For two sets $F_1, F_2 \subseteq E$, the symmetric difference $(F_1 \setminus F_2) \cup (F_2 \setminus F_1)$ is denoted by $F_1 \triangle F_2$. For a vector $x \in \mathbf{R}^E$ and $F \subseteq E$, define $x(F) = \sum_{e \in F} x(e)$.

2 A Maximum Square-Free 2-Matching Algorithm

This section describes an algorithm to find a maximum square-free 2-matching in bipartite graphs. The algorithm is based on algorithms of Hartvigsen [12, 13] and Pap [17, 18, 19], but different from both. Our algorithm uses the shortest augmenting path, whereas Pap's one does not involve the length of augmenting paths. Using the shortest path yields some simplicity, especially in the shrinking procedure, which makes the algorithm suitable for a weighted extension.

Let $G = (U, V; E)$ be a bipartite graph and $M \subseteq E$ be a square-free 2-matching in G. First, construct an auxiliary directed graph $G_M = (U, V; A)$ in the following manner. Define the directed edge set A by

$$A = \{uv \mid u \in U,\ v \in V,\ uv \in E \setminus M\} \cup \{vu \mid v \in V,\ u \in U,\ uv \in M\}.$$

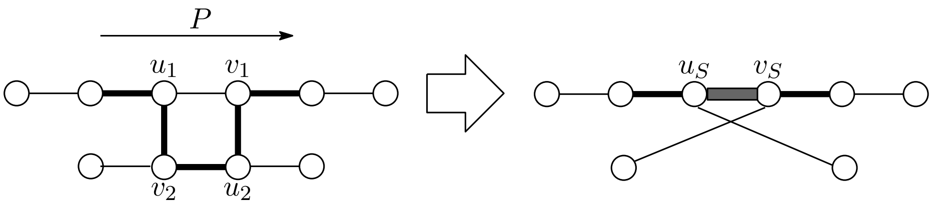

Figure 1: Shrinking of a square (bold line: M-edge).

Where it causes no confusion, we identify the undirected edge uv in G and the directed edge uv (or vu) in G_M. We also define two distinguished subsets $U^\circ \subseteq U$ and $V^\circ \subseteq V$ by

$$U^\circ = \{u \mid u \in U,\, |\delta u \cap M| < 2\},\ V^\circ = \{v \mid v \in V,\, |\delta v \cap M| < 2\}.$$

Then, find a shortest path P from U° to V° and consider the edge set $M' = M \triangle E(P)$. Observe that M' is a 2-matching with $|M'| = |M| + 1$. Hence, if M' is square-free, then M' is a larger square-free 2-matching. We refer to the procedure to obtain M' as an *augmentation*.

What if, however, M' contains squares? Suppose $M \triangle E(P)$ contains a square S. Since P is the shortest U°-V° path, we have that $|M \cap E(S)| = 3$, a detailed discussion of which will appear in Proposition 2.1. Denote $U(S) = \{u_1, u_2\}$, $V(S) = \{v_1, v_2\}$ and $\{u_1 v_1\} = E(P) \cap E(S)$ (see Figure 1). Then, what we do is to "shrink" S. Identify u_1 and u_2 to obtain a new vertex u_S, and v_1 and v_2 to obtain a new vertex v_S. Then, delete all edges in $E(S)$ and connect u_S and v_S by an M-edge. If an edge in $E \setminus E(S)$ had been incident to u_1 or u_2 (resp. v_1 or v_2), the edge is incident to u_S (resp. v_S) in the resulting graph. We allow parallel edges to appear in this procedure. If an edge had belonged to M, it also belongs to M in the new graph, and otherwise it does not. We denote the resulting graph by $\tilde{G} = (\tilde{U}, \tilde{V}; \tilde{E})$ and refer to the new M-edge $u_S v_S$ as a *shrunk square*. Note that it follows from $|M \cap E(S)| = 3$ that the number of M-edges in the parallel edges incident to u_S (or v_S) is at most one, so M remains simple whereas $\tilde{G}$ may not. In addition, the new M is a 2-matching in $\tilde{G}$ and may contain a square that includes shrunk squares.

If more than one square appears in $M \triangle E(P)$, we shrink the square which is "closest" to U°. That is, we shrink the square whose non-M-edge appears the first in P. We refer to the procedure to obtain a new graph and 2-matching as $\mathsf{Shrink}(M, P)$.

Then, we recursively execute the above procedures. Here, we have to take care that the U°-V° path does not contain shrunk squares. In order to achieve this, we search a U°-V° path in a subgraph $\tilde{G}'_M$ of $\tilde{G}_M$ obtained by

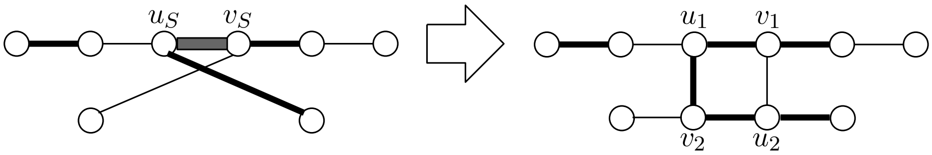

Figure 2: Expanding of a square (bold line: M-edge).

deleting all the shrunk squares. Then, set $b \in \{1,2\}^{\tilde{U}\cup\tilde{V}}$ by

$$b(v) = \begin{cases} 1 & (v \text{ is an end vertex of a shrunk square deleted from } \tilde{G}_M), \\ 2 & (\text{otherwise}), \end{cases} \tag{1}$$

and modify the definition of U° and V° by

$$U^\circ = \{u \mid u \in \tilde{U}, |\delta u \cap M| < b(u)\},\ V^\circ = \{v \mid v \in \tilde{V}, |\delta v \cap M| < b(v)\}. \tag{2}$$

This means that what we deal with is a square-free b-matching M in $\tilde{G}'_M$. Thus, one would see that the shrunk squares get neither incident to each other nor nested.

After an augmentation, we expand every shrunk squares to obtain the original bipartite graph G. Let $u_S v_S$ be a shrunk square that is obtained by shrinking S with $U(S) = \{u_1, u_2\}$ and $V(S) = \{v_1, v_2\}$. Now, replace the vertices u_S, v_S and edge $u_S v_S$ by $K_{2,2}$ induced by $U(S) \cup V(S)$. An edge incident to u_S or v_S is connected to a vertex in $U(S) \cup V(S)$ to which the edge had been incident before shrinking S. Next, determine M-edges. An M-edge before expanding S also belongs to M. Then, pick up three edges in $E(S)$ to be in M so that M forms a 2-matching. Figure 2 illustrates an example of expanding a shrunk square. By expanding every shrunk square, we obtain the original bipartite graph G and a new square-free 2-matching M of one larger size.

The procedures are summarized below.

Algorithm Maximum Square-Free 2-Matching

Step 0: Set $M = \emptyset$ and $\tilde{G} = G$.

Step 1: If $|M| = 2\min\{|U|, |V|\}$, then halt. (M is a square-free 2-factor.)

Step 2: Construct an auxiliary directed graph $\tilde{G}'_M$. In $\tilde{G}'_M$, define b by (1) and search for a shortest path from U° to each vertex. Let $R \subseteq \tilde{U} \cup \tilde{V}$ be the set of the reachable vertices from U°. If $V^\circ \cap R = \emptyset$, then expand each shrunk square and halt. (M is a maximum square-free 2-matching.)

Step 3: Let P be the shortest path from U° to V°. If $M \triangle \tilde{E}(P)$ contains a square in $\tilde{G}'_M$, then execute $\mathsf{Shrink}(M, P)$ and go to Step 2.

Step 4: Replace M by $M \triangle \tilde{E}(P)$ and expand every shrunk square. Then, go to Step 1.

Here, we show that if $M \triangle \tilde{E}(P)$ contains a square S then $|M \cap \tilde{E}(S)| = 3$.

Proposition 2.1. *Let S be a square in $\tilde{G}'_M$ that appears in $M \triangle \tilde{E}(P)$. Then, it holds that $|M \cap \tilde{E}(S)| = 3$.*

Proof. Since $\tilde{E}(S) \subseteq M \triangle \tilde{E}(P)$, the edges in $\tilde{E}(S)$ can be partitioned into two parts, $\tilde{E}_M = M \cap \tilde{E}(S)$ and $\tilde{E}_P = \tilde{E}(P) \cap \tilde{E}(S)$. We prove that $|\tilde{E}_P| = 1$.

As M is square-free in $\tilde{G}'_M$, we have that $|\tilde{E}_P| \geq 1$. As P visits each vertex at most once and the edges of M and $E \setminus M$ lie alternately in P, we have that $|\tilde{E}_P| \leq 2$. Hence, it suffices to show that $|\tilde{E}_P| \neq 2$.

Denote $U(S) = \{u_1, u_2\}$ and $V(S) = \{v_1, v_2\}$. To the contrary we assume, without loss of generality, that $\tilde{E}_P = \{u_1v_1, u_2v_2\}$ and u_1v_1 appears earlier than u_2v_2 in P. Let us denote the subpath of P which is from the initial vertex of P to v_1 by P_1, and which is from u_2 to the terminal vertex of P by P_2. Here, connecting P_1, u_2v_1 and P_2, we obtain another U°-V° path, which is shorter than P. This contradicts that P is the shortest U°-V° path in $\tilde{G}'_M$. □

Now, what is left is to prove that M is maximum when the algorithm halts in Step 2. The following is a min-max formula for square-free b-matchings.

Theorem 2.2 (Z. Király [15]). *Let $G = (U, V; E)$ be a bipartite graph and $b \in \{0, 1, 2\}^{U \cup V}$. Then, the size of the maximum square-free b-matching in G is equal to*

$$\min_{Z \subseteq U \cup V} \{b(U \cup V \setminus Z) + |E[Z]| - c(G[Z])\}.$$

Let us view the weak duality and equality conditions for the formula. For $G = (U, V; E)$ and $b \in \{0, 1, 2\}^{U \cup V}$, define a function $f_{G,b} : 2^{U \cup V} \to \mathbf{R}$ by

$$f_{G,b}(Z) = b(U \cup V \setminus Z) + |E[Z]| - c(G[Z]).$$

Let M be a square-free b-matching and $Z \subseteq U \cup V$. For Z, partition M into three sets M_1, M_2, M_3, where

$$M_1 = M \cap E[Z], \quad M_2 = M \cap E[(U \cup V) \setminus Z], \quad M_3 = M \setminus (M_1 \cup M_2).$$

Then, it follows that

$$|M_1| \leq |E[Z]| - c(G[Z]), \qquad 2|M_2| + |M_3| \leq b(U \cup V \setminus Z). \tag{3}$$

Hence, it holds that

$$|M| \leq |M_1| + 2|M_2| + |M_3| \leq f_{G,b}(Z). \tag{4}$$

By (3), we have the following conditions for (4) to hold with equality.

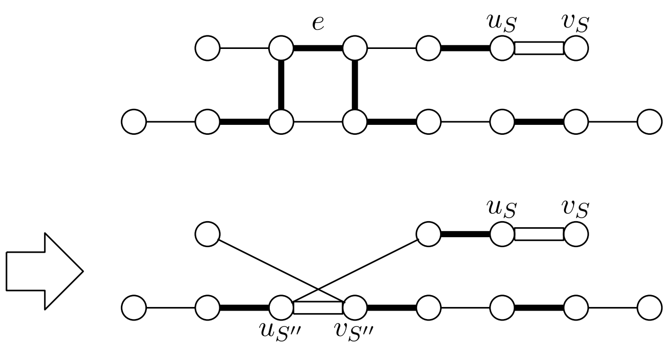

Figure 3: Reachability of u_S (bold line: M-edge).

Condition (a). In $G[Z]$, every edge except for one edge in each square-component belongs to M.

Condition (b). $M_2 = \emptyset$.

Condition (c). $\forall v \in (U \cup V) \setminus Z$, $|M_3 \cup \delta v| = b(v)$.

In what follows, we abbreviate $f_{G,b}$ as f_G, since b is always defined by (1). We prove that there exists $Z \subseteq U \cup V$ such that $|M| = f_G(Z)$, which yields a verification of our algorithms and an alternative proof for Theorem 2.2. The argument below is an adaptation of the combinatorial proof for Theorem 2.2 of Pap [17, 18, 19] to our algorithm.

The following is a key observation to our verification.

Proposition 2.3. *Let $\mathcal{S}' = \{S \mid S \in \mathcal{S},\ S$ is shrunk into $u_S v_S$ in $\tilde{G}$ $\}$. Then, for u_S of $S \in \mathcal{S}'$, $\tilde{G}'_M$ has a path from U° to u_S consisting of edges that was in a shortest path used in shrinking a square $S' \in \mathcal{S}'$ and closer to U° than $u_{S'}$.*

Proof. The proof is by induction on the number of shrinkings. The statement is obvious immediately after shrinking S.

Now, suppose P is a path from U° to u_S that satisfies the condition in the statement and consider subsequent shrinkings. A shrinking that does not delete any edge in $\tilde{E}(P)$ does not matter. If a shrinking deletes an edge $e \in \tilde{E}(P)$, then it holds that $e \in M$. For, if $e \notin M$, a square S_e appears when e turns to be an M-edge by Proposition 2.1, which contradicts that e had been in a shortest path used in shrinking $S' \in \mathcal{S}'$ and closer to U° than $u_{S'}$. Let $e \in M \cap \tilde{E}(P)$ be deleted in a subsequent shrinking (an example is shown in Fig. 3). Here, $\partial^- e \in \tilde{U}$ is shrunk into $u_{S''}$, which is reachable from U°. On the other hand, u_S is reachable from $u_{S''}$ by tracing the subpath of P that had connected $\partial^- e$ and u_S. Therefore, the statement is maintained in shrinking S''. □

Suppose no U°-V° path is found in Step 2 and denote the current graph $\tilde{G}'_M$ by $G_0 = (U_0, V_0; E_0)$. Recall that we deleted all the shrunk squares from $\tilde{G}_M$ to obtain $\tilde{G}'_M$. Let $R \subseteq U_0 \cup V_0$ be the set of vertices reachable from U° in $\tilde{G}'_M$, and define $Z_0 = (U_0 \cap R) \cup (V_0 \setminus R)$. For Z_0, partition M into three sets M_1, M_2, M_3, where

$$M_1 = M \cap E[Z_0], \quad M_2 = M \cap E[(U_0 \cup V_0) \setminus Z_0], \quad M_3 = M \setminus (M_1 \cup M_2).$$

Then, by the definition of Z_0, it holds that

$$|M_1| = |E[Z_0]|, \quad |M_2| = 0, \quad M_3 = b(U_0 \cup V_0 \setminus Z_0), \quad c(G[Z_0]) = 0,$$

and hence $|M| = f_{G_0}(Z_0)$. By (4), it follows that M is the maximum square-free b-matching in G_0 and Z_0 minimizes f_{G_0}.

Next, consider to expand a shrunk square. Let $Z_0^* \subseteq U_0 \cup V_0$ be a minimal minimizer of f_{G_0}. That is, $f(Z_0^*) < f(Z')$ holds for all $Z' \subsetneq Z_0^*$. Let $u_S v_S$ be a shrunk square in G_0 and denote the bipartite graph obtained by expanding $u_S v_S$ by $G_1 = (U_1, V_1; E_1)$. Consider whether $u_S, v_S \in Z_0^*$ or not.

By Proposition 2.3, G_0 has a path P from U° to u_S which consists of edges that was in a shortest path used in shrinking a square and closer to U° than the square. Denote the initial vertex of P by u_0. Since $u_0 \in U^\circ$, Condition (c) implies that $u_0 \in Z_0^*$. Moreover, carefully looking at Conditions (a) and (b), we have that $U_0(P) \subseteq Z_0^*$ and $V_0(P) \subseteq (U_0 \cup V_0) \setminus Z_0^*$, which implies that $u_S \in Z_0^*$.

Therefore, we have two cases: $v_S \in Z_0^*$ or not.

Case 1 ($v_S \in Z_0^*$). Define $Z_1 \subseteq U_1 \cup V_1$ by $Z_1 = (Z_0^* \setminus \{u_S, v_S\}) \cup U(S) \cup V(S)$. Then, it is easily seen that

$$b((U_1 \cup V_1) \setminus Z_1) = b((U_0 \cup V_0) \setminus Z_0^*), \qquad E_1[Z_1] = E_0[Z_0^*] + 4.$$

Moreover, it holds that $c(G_1[Z_1]) = c(G_0[Z_0^*]) + 1$. For, by Condition (a) no edge in $E[Z_0^*]$ is incident to u_S, and if an edge in $E_0[Z_0^*]$ is incident to v_S then $Z_0^* \setminus \{v_S\}$ also satisfies Conditions (a)–(c) and minimizes f_{G_0}, which contradicts the minimality of Z_0^*. Hence, the subgraph induced by $U_1(S) \cup V_1(S)$ is a square-component in $G_1[Z_1]$, which implies that $c(G_1[Z_1]) = c(G_0[Z_0^*]) + 1$.
Therefore, we have that $f_{G_1}(Z_1) = f_{G_0}(Z_0^*) + 3$. Since the size of M increases by three in expanding a shrunk square, it holds that $|M| = f_{G_1}(Z_1)$. Thus, it follows from (4) that M is a maximum square-free b-matching in G_1 and Z_1 is a minimizer of f_{G_1}.

Case 2 ($v_S \notin Z_0^*$). Define $Z_1 \subseteq U_1 \cup V_1$ by $Z_1 = (Z_0^* \setminus \{u_S\}) \cup U(S)$. Then, it holds that

$$b((U_1 \cup V_1) \setminus Z_1) = b((U_0 \cup V_0) \setminus Z_0^*) + 3,$$
$$E_1[Z_1] = E_0[Z_0^*], \quad c(G_1[Z_1]) = c(G_0[Z_0^*]),$$

and hence $|M| = f_{G_1}(Z_1)$. Then, by (4), we have that M is a maximum square-free b-matching in G_1 and Z_1 is a minimizer of f_{G_1}.

Applying the above argument repeatedly, we obtain a square-free 2-matching M in the original graph G and $Z \subseteq U \cup V$ such that $|M| = f_G(Z)$.

3 A Weighted Square-Free 2-Factor Algorithm

This section deals with the weighted square-free 2-factor problem. Let (G, w) be a weighted bipartite graph with $G = (U, V; E)$ and $w \in \mathbf{R}_+^E$. Throughout this section, we assume that $|U| = |V|$. We also assume that w is vertex-induced on any square. That is, we assume that, for any square S with $U(S) = \{u_1, u_2\}$ and $V(S) = \{v_1, v_2\}$, there exists a potential function $\pi_S : U(S) \cup V(S) \to \mathbf{R}$ such that $w(u_i v_j) = \pi_S(u_i) + \pi_S(v_j)$ for any $i, j \in \{1, 2\}$. In other words, it holds that $w(u_1v_1) + w(u_2v_2) = w(u_1v_2) + w(u_2v_1)$. We propose an algorithm to find a square-free 2-factor M that minimizes $w(M)$ if exists, or otherwise determine that no square-free 2-factor exists in G. The algorithm, based on the primal-dual framework of the minimum-cost flow algorithm [4, 21], extends Algorithm Maximum Square-Free 2-Matching.

Let $x \in \mathbf{R}^E$. The following is a linear programming relaxation of an integer program for the minimum-weight square-free 2-factor problem:

$$\begin{array}{llll} \text{(P)} & \text{minimize} & wx & \\ & \text{subject to} & x(\delta v) = 2 & (\forall v \in U \cup V), \\ & & x(E(S)) \leq 3 & (\forall S \in \mathcal{S}), \\ & & 0 \leq x(e) \leq 1 & (\forall e \in E). \end{array}$$

One would see that the incidence vector of a square-free 2-factor is a feasible solution for (P). In what follows, we often identify an edge set M and its incidence vector x.

Consider the dual problem of (P). Let $p \in \mathbf{R}^{U \cup V}$, $q \in \mathbf{R}^E$ and $r \in \mathbf{R}^{\mathcal{S}}$. The dual problem is given by

$$\begin{array}{llll} \text{(D)} & \text{maximize} & 2(p(U) - p(V)) - q(E) - 3r(\mathcal{S}) & \\ & \text{subject to} & p(u) - p(v) - q(e) - \displaystyle\sum_{S\colon e \in E(S)} r(S) \leq w(e) & (\forall e = uv \in E), \\ & & q, r \geq 0. & \end{array}$$

The complementary slackness conditions of (P) and (D) are

$$x(e) > 0 \Rightarrow p(u) - p(v) - q(e) - \sum_{S\colon e \in E(S)} r(S) = w(e), \tag{6}$$

$$q(e) > 0 \Rightarrow x(e) = 1, \tag{7}$$

$$r(S) > 0 \Rightarrow x(E(S)) = 3. \tag{8}$$

In what follows, we present an algorithm to find feasible solutions for (P) and (D) which satisfy (6)–(8) by extending Algorithm Maximum Square-Free 2-Matching. Roughly speaking, we maintain a square-free 2-matching M, construct an auxiliary directed graph $\tilde{G}_M$, search for a U°-V° path P in its subgraph $\tilde{G}'_M$, and then augment M by substituting $M \triangle \tilde{E}(P)$ for M or shrink a square. In these procedures, we also take dual solutions into account. In particular, a significant difference from Algorithm Maximum Square-Free 2-Matching is that we do not expand a shrunk square $u_S v_S$ after an augmentation if $r(S) > 0$, and such a shrunk square is used in the subsequent searching for a U°-V° path.

Now, let us consider the details. Let $\tilde{G}_M = (\tilde{U}, \tilde{V}; A)$ be an auxiliary directed graph, which may have resulted from repeated shrinking and expanding of squares. Recall that the M-edges (including all shrunk squares) are oriented in the direction from $\tilde{V}$ to $\tilde{U}$, and other edges in the opposite direction. For $\tilde{G}_M$, a length function $l : A \to \mathbf{R}$ is defined by

$$l(e) = \begin{cases} w(e) - p(u) + p(v) & (e \notin M \text{ and corresponds to } uv \in E), \\ -w(e) + p(u) - p(v) & (e \in M \text{ and corresponds to } uv \in E), \\ r(S) & (e \text{ is a shrunk square } u_S v_S). \end{cases}$$

Remark that p is defined on $U \cup V$, the vertex set of the original bipartite graph G, while l is defined on A, the edge set of $\tilde{G}_M$.

In the auxiliary graph $\tilde{G}_M$, we establish the following optimality criterion.

Theorem 3.1. *Let M be a 2-factor in $\tilde{G}_M$ such that each square in M contains shrunk squares, and let $p \in \mathbf{R}^{U \cup V}$ and $r \in \mathbf{R}^{\mathcal{S}}$. If the following* (9)–(11) *hold, then we can expand M to obtain a minimum-weight square-free 2-factor in (G, w) and determine q so that (p, q, r) forms an optimal solution for* (D):

$$r \geq 0 \text{ and } r(S) > 0 \text{ only if } S \text{ is shrunk}; \tag{9}$$

$$\forall e \in A,\ l(e) \geq 0; \tag{10}$$

$$\forall \text{shrunk square } u_S v_S,\ \forall e = uv \in E(S),\ p(u) - p(v) - r(S) = w(e). \tag{11}$$

Proof. We prove the theorem by showing how to construct feasible solutions for (P) and (D) that satisfy (6)–(8).

Let $e = uv \in E$ be an edge not shrunk in $\tilde{G}_M$. If $e \notin M$, then by (10) we have that $l(e) = w(e) - p(u) + p(v) \geq 0$. Now, set $q(e) = 0$ to have (6) and (7) hold in e. If $e \in M$, then $l(e) = -w(e) + p(u) - p(v) \geq 0$ by (10). Set $q(e) = l(e)$, which gets (6) and (7) to hold in e.

Let $e = uv \in E$ belong to $E(S)$ of a shrunk square $u_S v_S$ in $\tilde{G}_M$. For such e, set $q(e) = 0$. Then, by (11), we have that (6) and (7) hold in e regardless whether $x(e) = 0$ or $x(e) = 1$ after expanding S.

Now we have determined $q(e)$ on every $e \in E$. From the above construction, one would appreciate the feasibility of (p, q, r) for (D). Moreover, by expanding all shrunk squares in $\tilde{G}_M$, we obtain a square-free 2-factor in G, a feasible solution for (P). For this pair of solutions for (P) and (D), it follows from (9) that (8) holds. Therefore, (6)–(8) hold for this pair of solutions for (P) and (D). □

Now, let us describe the minimum-weight square-free 2-factor algorithm. The algorithm keeps a square-free 2-matching M and a dual solution (p, r) that satisfy (9)–(11), and increases $|M|$ until it attains the maximum.

Algorithm Minimum-Weight Square-Free 2-Factor

Step 0: Set $M = \emptyset$, $p = 0$, $r = 0$ and $\tilde{G} = G$.

Step 1: If $|M| = 2|\tilde{U}|$, then expand every shrunk square and halt. (M is a minimum-weight square-free 2-factor.)

Step 2: Construct an auxiliary directed graph $\tilde{G}_M = (\tilde{U}, \tilde{V}; A)$ and delete shrunk squares that are created after the latest augmentation to obtain a new graph $\tilde{G}'_M$. Then, in $\tilde{G}'_M$, define b by (1) and search for a shortest path with respect to (w.r.t.) l from U° to each vertex. Let $R \subseteq \tilde{U} \cup \tilde{V}$ be the set of the reachable vertices from U° and define $d : \tilde{U} \cup \tilde{V} \to \mathbf{R}$ by

$$d(v) = \begin{cases} \text{distance from } U^\circ \text{ to } v \text{ w.r.t. } l & (v \in R), \\ \max\{d(v) \mid v \in R\} & (\text{otherwise}). \end{cases}$$

If $V^\circ \cap R = \emptyset$, then halt. (No square-free 2-factor exists.)

Step 3: Let P be the shortest path (w.r.t l) from U° to V°. If more than one shortest path exists, select a path with the minimum number of edges. If P contains a shrunk square, apply Dual-Update (described below), expand every shrunk square in P, and then go to Step 2.

Step 4: If $M \triangle \tilde{E}(P)$ contains a square without shrunk squares, apply Dual-Update, execute Dual-Shrink(M, P) (described below), and then go to Step 2.

Step 5: Apply Dual-Update, replace M by $M \triangle \tilde{E}(P)$, and expand every shrunk square S with $r(S) = 0$. Then, go to Step 1.

We remark that a shrunk square $u_S v_S$ with $r(S) > 0$ is not expanded in Step 5, and belongs to $\tilde{G}'_M$ after the augmentation.

In the procedure Dual-Update, we change the dual solution as follows:

$$p(v) := p(v) - d(v) \quad (v \in U \cup V),$$
$$r(S) := \begin{cases} r(S) - d(u_S) + d(v_S) & (S \text{ is shrunk}), \\ r(S) & (\text{otherwise}), \end{cases}$$

where $d(v)$ for a vertex $v \in (U \cup V) \setminus (\tilde{U} \cup \tilde{V})$ that is shrunk into $v_S \in \tilde{U} \cup \tilde{V}$ is defined by $d(v_S)$.

The procedure Dual-Shrink(M, P) is twofold: update of p in two vertices; and Shrink(M, P). We have that $M \triangle \tilde{E}(P)$ contains squares in $\tilde{G}'_M$. For such a square S, it holds that $|M \cap \tilde{E}(S)| = 3$, which is proven later (Proposition 3.2). Let S be the nearest one from U° among the squares in $M \triangle \tilde{E}(P)$, and denote $\tilde{U}(S) = \{u_1, u_2\}$ and $\tilde{V}(S) = \{v_1, v_2\}$. Without loss of generality, we assume $u_1v_1 \in \tilde{E}(P) \setminus M$. Then, update the values $p(u_2)$ and $p(v_2)$ by

$$p(u_2) := p(u_2) - l(u_2v_1), \qquad p(v_2) := p(v_2) + l(u_1v_2),$$

and call Shrink(M, P).

Now, let us confirm the validity of the algorithm. Note that (9)–(11) hold at the beginning of the algorithm. We prove that the conditions are maintained throughout the algorithm.

Proposition 3.2. *Throughout the algorithm, the following* (i) *and* (ii) *hold*:

(i) $l(e) \geq 0$ *for each edge* e *in* $\tilde{G}'_M$,

(ii) *if a square* S *without shrunk squares appears in* $M \triangle \tilde{E}(P)$, *it holds that* $|M \cap \tilde{E}(S)| = 3$.

Proof. We prove that (ii) holds under the assumption of (i), and (i) is maintained when (ii) holds. Then, since (i) holds at the beginning of the algorithm, (i) and (ii) inductively hold throughout the algorithm.

Let S be a square without shrunk squares such that $\tilde{E}(S) \subseteq M \triangle \tilde{E}(P)$. Denote $\tilde{U}(S) = \{u_1, u_2\}$, $\tilde{V}(S) = \{v_1, v_2\}$, and $\tilde{E}_P = \tilde{E}(P) \cap \tilde{E}(S)$. By the argument in the proof for Proposition 2.1, it suffices to show that $|\tilde{E}_P| \neq 2$ in order to prove (ii).

Assume to the contrary that $\tilde{E}_P = \{u_1v_1, u_2v_2\}$ and u_1v_1 appears earlier than u_2v_2 in P. Then, it holds that $\sum_{e \in \tilde{E}(S)} l(e) = 0$ since w is vertex-induced on S. Hence, it follows from (i) that $l(e) = 0$ for all $e \in \tilde{E}(S)$. Now, as is described in the proof for Proposition 2.1, we have another U°-V° path P', which is obtained by taking v_1u_2 as a shortcut for P. It follows from (i) and $l(v_1u_2) = 0$ that P', which has fewer edges than P, is no longer than P w.r.t. l. This contradicts the choice of P.

Next, we prove that (i) is maintained under the assumption of (ii). Consider Dual-Update. Pick up a directed edge $e \in A$. By the definition of d, it holds that $d(\partial^- e) \leq d(\partial^+ e) + l(e)$. If $e = uv$ is in the direction from $\tilde{U}$ to $\tilde{V}$, the shift of $l(e)$ in Dual-Update is $-(-d(u)) - d(v) = d(\partial^+ e) - d(\partial^- e) \geq -l(e)$. If $e = vu$ is not shrunk and in the direction of $\tilde{V}$ to $\tilde{U}$, i.e., $e \in M$, then the shift of $l(e)$ is $-d(u) - (-d(v)) = -d(\partial^- e) + d(\partial^+ e) \geq -l(e)$. Finally, if $e = v_S u_S$ is a shrunk square, the shift of $l(e)$ is $-d(u_S) + d(v_S) = -d(\partial^- e) + d(\partial^+ e) \geq -l(e)$. Therefore, in any case we have that $l(e) \geq 0$ after Dual-Update. Moreover, for a shortest U°-V° path P, the above inequalities hold with equality for each $e \in \tilde{E}(P)$ and hence $l(e) = 0$ after Dual-Update. Thus, in an augmentation using P, in which $l(e)$ changes to $-l(e)$ for $e \in \tilde{E}(P)$, (i) is maintained.

Consider Dual-Shrink(M, P). Since (ii) holds, the procedure Dual-Shrink(M, P) is valid. In Dual-Shrink(M, P), l changes only on the edges in $\delta u_2 \cup \delta v_2$. One would easily see that $l(u_1 v_2)$ and $l(u_2 v_1)$ become zero by the update of $p(u_2)$ and $p(v_2)$. As we have applied Dual-Update just before Dual-Shrink(M, P), we have $l(u_1 v_1) = 0$. Moreover, since w is vertex-induced on S, it holds that $l(u_2 v_2) - l(u_1 v_1) = l(u_1 v_2) + l(u_2 v_1)$, which implies that $l(u_2 v_2)$ also becomes zero. Meanwhile, for an edge $e \in (\delta u_2 \cup \delta v_2) \setminus E(S)$, we have that $e \notin M$. Hence, the shift of $l(e)$ is equal to $l(u_2 v_1)$ for $e \in \delta u_2$ and equal to $l(u_1 v_2)$ for $e \in \delta v_2$. Therefore, $l(e) \geq 0$ is kept for every edge in $\delta u_2 \cup \delta v_2$ in Dual-Shrink(M, P). □

The above argument induces the following corollaries.

Corollary 3.3. *After* Dual-Update, $l(e) = 0$ *for every edge* $e \in \tilde{E}(P)$.

Corollary 3.4. *When we shrink a square* S, $l(e) = 0$ *for every* $e \in \tilde{E}(S)$.

It follows from Corollary 3.4 that (11) holds for S when we shrink S, which is the purpose of the update of $p(u_2)$ and $p(v_2)$ in Dual-Shrink(M, P).

Proposition 3.5. *Let* $u_S v_S$ *be a shrunk square created after the latest augmentation. Then,* $d(u_S) = 0$ *until the next augmentation.*

Proof. It follows from Proposition 2.3 that u_S is reachable from U° in $\tilde{G}'_M$ by traversing edges that had been in a shortest U°-V° path. By Corollary 3.3, such an edge e has its length $l(e) = 0$ in Dual-Update executed when $e \in \tilde{E}(P)$, and $l(e)$ remains to be zero until the next augmentation. □

Proposition 3.6. *Throughout the algorithm,* (9)–(11) *hold.*

Proof. **Condition** (9)**.** Since we change $r(S)$ only if S is shrunk and expand $u_S v_S$ only if $r(S) = 0$, it holds that $r(S) = 0$ for every non-shrunk square S. Moreover, we have seen in the proof for Proposition 3.2 that $r(S) = l(u_S v_S) \geq 0$ for every shrunk square $u_S v_S$ in $\tilde{G}'_M$. As for a shrunk square not in $\tilde{G}'_M$, in other words created after the latest shrunk, $d(u_S) = 0$ by Proposition 3.5, which implies $r(S) \geq 0$ after a Dual-Update. In Dual-Shrink(M, P), $r(S)$ is not changed since Dual-Shrink(M, P) is executed for a square containing no shrunk square.

Condition (10)**.** Already proved.

Condition (11)**.** By Corollary 3.4, (11) holds when S is shrunk. Consider the shift of $p(u) - p(v) - r(S)$ in subsequent Dual-Update for $e = uv \in E(S)$. The variables are changed as follows:

$$p(u) := p(u) - d(u_S), \qquad p(v) := p(v) - d(v_S),$$
$$r(S) := r(S) - d(u_S) + d(v_S).$$

Then, $p(u) - p(v) - r(S)$ does not change in Dual-Update. In Dual-Shrink(M, P), the variables concerned are not changed.

□

By Theorem 3.1 and Proposition 3.6, if the algorithm halts in Step 1, then we have a minimum-weight square-free 2-factor M and a dual optimal solution. If the algorithm halts in Step 2, G has no square-free 2-factor. This is shown by a similar argument to that in Section 2.

Let us discuss the complexity of the algorithm. Recall that $|U \cup V| = n$ and $|E| = m$. The following is an easy observation, but plays a key role in analyzing the complexity.

Proposition 3.7. *A shrunk square created in Step 4 is not expanded until the next augmentation.*

Proof. We search a U°-V° path P in $\tilde{G}'_M$, which does not contain shrunk squares created after the latest augmentation, and a shrunk square expanded by the next augmentation is contained in P. □

The bottleneck part of the algorithm lies in Step 2, determining the distance from U° to every vertex. It follows from Proposition 3.2 that this can be computed by a shortest path algorithm with nonnegative length. By Proposition 3.7, we apply a shortest path algorithm $\mathrm{O}(n)$ times between augmentations. Since augmentations happen at most n times throughout the algorithm, the total complexity of the algorithm is $\mathrm{O}(n^2 D)$, where D is the time to execute a shortest path algorithm with nonnegative length. Among a number of implementations of such an algorithm, incorporating Fredman and Tarjan's version of Dijkstra's algorithm [9], we get a strongly polynomial complexity $\mathrm{O}(n^2 m + n^3 \log n)$.

Theorem 3.8. Algorithm Minimum-Weight Square-Free 2-Factor *runs in* $\mathrm{O}(n^2 m + n^3 \log n)$ *time.*

We should remark here that Algorithm Minimum-Weight Square-Free 2-Factor is fully combinatorial, that is, it consists of only addition, subtraction, and comparison. Thus, the algorithm leads to the following integrality theorem.

Theorem 3.9. *Let (G, w) be a weighted bipartite graph such that G admits a square-free 2-factor and $w \in \mathbf{R}_+^E$ is integer and vertex-induced on any square. Then, the linear program* (P) *has an integral optimal solution. Moreover, the dual problem* (D) *also has an integral optimal solution (p, q, r) such that the elements in $\{S \mid S \in \mathcal{S},\ r(S) > 0\}$ are pairwise disjoint.*

4 Extension to $K_{t,t}$-Free t-Factors

We can naturally extend Algorithm Minimum-Weight Square-Free 2-Factor to the minimum-weight $K_{t,t}$-free t-factor problem. Let (G, w) be a weighted bipartite graph with $G = (U, V; E)$ and $w \in \mathbf{R}_+^E$. Assume that $|U| = |V|$ and w is vertex-induced on any $K_{t,t}$ in G.

Let $x \in \mathbf{R}^E$, $p \in \mathbf{R}^{U \cup V}$, $q \in \mathbf{R}^E$ and $r \in \mathbf{R}^{\mathcal{S}_t}$. The following is a linear programming relaxation of an integer program for the minimum-weight $K_{t,t}$-free t-factor problem:

$$
\begin{array}{llll}
(\mathrm{P}_t) & \text{minimize} & wx & \\
& \text{subject to} & x(\delta v) = t & (\forall v \in U \cup V), \\
& & x(E(S)) \le t^2 - 1 & (\forall S \in \mathcal{S}_t), \\
& & 0 \le x(e) \le 1 & (\forall e \in E).
\end{array}
$$

The dual problem of (P_t) is

$$
\begin{array}{lll}
(\mathrm{D}_t) & \text{maximize} & t(p(U) - p(V)) - q(E) - (t^2 - 1) r(\mathcal{S}_t) \\
& \text{subject to} & p(u) - p(v) - q(e) - \displaystyle\sum_{S\colon e \in E(S)} r(S) \le w(e) \quad (\forall e = uv \in E), \\
& & q, r \ge 0.
\end{array}
$$

We describe an algorithm for the minimum-weight $K_{t,t}$-free t-factor problem by mentioning the differences from Algorithm Minimum-Weight Square-Free 2-Factor. First, let us remark Dual-Shrink(M, P). As an extension of Proposition 3.2, we have the following.

Proposition 4.1. *Throughout the algorithm, the following* (i) *and* (ii) *hold*:

(i) *$l(e) \ge 0$ for each edge e in $\tilde{G}'_M$,*

(ii) *if a $K_{t,t}$, denoted by S, appears in $M \triangle \tilde{E}(P)$, then $|M \cap \tilde{E}(S)| = t^2 - 1$.*

Let S be a $K_{t,t}$ in $M \triangle \tilde{E}(P)$ closest from U°. Denote $\tilde{U}(S) = \{u_1, \ldots, u_t\}$ and $\tilde{V}(S) = \{v_1, \ldots, v_t\}$, and suppose $u_1 v_1 \notin M$. We update the dual valuable as follows:

$$
p(u_i) := p(u_i) - l(u_i v_1), \quad p(v_i) := p(v_i) + l(u_1 v_i) \quad (i = 2, \ldots, t).
$$

Since w is vertex-induced on S, we have $l(e) = 0$ for every $e \in \tilde{E}(S)$ by this update.

Then, we shrink S in the following manner. Identify the vertices $\tilde{U}(S)$ to obtain a new vertex u_S, and $\tilde{V}(S)$ to obtain a new vertex v_S. Delete all edges in $\tilde{E}(S)$ and connect u_S and v_S by an M-edge. If an edge in $\tilde{E} \setminus \tilde{E}(S)$ had been incident to $\tilde{U}(S)$ (resp. $\tilde{V}(S)$), the edge is incident to u_S (resp. v_S) in the resulting graph. If an edge had belonged to M, it also belongs to M in the new graph, and otherwise it does not. We refer to the new M-edge $u_S v_S$ as a *shrunk* $K_{t,t}$.

Next, in an auxiliary directed graph $\tilde{G}'_M$, the vector b is defined by

$$b(v) = \begin{cases} 1 & (v \text{ is an end vertex of a shrunk } K_{t,t} \text{ deleted from } \tilde{G}_M), \\ 2 & (v \text{ is an end vertex of a shrunk } K_{t,t} \text{ in } \tilde{G}'_M), \\ t & (\text{otherwise}). \end{cases} \tag{12}$$

Then, U° and V° are determined by (2) according to (12).

Now, we are ready to present a full description of an algorithm to find a minimum-weight $K_{t,t}$-free t-factor.

Algorithm Minimum-Weight $K_{t,t}$-Free t-Factor

Step 0: Set $M = \emptyset$, $p = 0$, $r = 0$ and $\tilde{G} = G$.

Step 1: Let k be the number of shrunk $K_{t,t}$ in $\tilde{G}$. If $|M| = 2k + t(|\tilde{U}| - k)$, then expand every shrunk $K_{t,t}$ and halt. (M is a minimum-weight $K_{t,t}$-free t-factor.)

Step 2: Construct an auxiliary directed graph $\tilde{G}_M = (\tilde{U}, \tilde{V}; A)$. In $\tilde{G}_M$, delete every shrunk $K_{t,t}$ that is created after the latest augmentation to obtain a new graph $\tilde{G}'_M$. Then, in $\tilde{G}'_M$, define b by (12) and search for a shortest path w.r.t. l from U° to each vertex. Let $R \subseteq \tilde{U} \cup \tilde{V}$ be the set of the reachable vertices from U° and define $d : \tilde{U} \cup \tilde{V} \to \mathbf{R}_+$ by

$$d(v) = \begin{cases} \text{distance from } U^\circ \text{ to } v \text{ w.r.t. } l & (v \in R), \\ \max\{d(v) \mid v \in R\} & (\text{otherwise}). \end{cases}$$

If $V^\circ \cap R = \emptyset$, then halt. (No $K_{t,t}$-free t-factor exists.)

Step 3: Let P be the shortest path (w.r.t l) from U° to V°. If more than one shortest path exists, select a path with the minimum number of edges. If P contains a shrunk $K_{t,t}$, apply Dual-Update, expand every shrunk $K_{t,t}$ in P, and then go to Step 2.

Step 4: If $M \triangle \tilde{E}(P)$ contains a $K_{t,t}$ without a shrunk $K_{t,t}$, apply Dual-Update, execute Dual-Shrink(M, P), and go to Step 2.

Step 5: Apply Dual-Update, replace M by $M \triangle \tilde{E}(P)$ and expand every shrunk $S \in \mathcal{S}_t$ with $r(S) = 0$. Then, go to Step 1.

Let us discuss the complexity. Recall that $n = |U \cup V|$, $m = |E|$ and D is the time for a shortest paths algorithm with nonnegative length. In Step 4, we check whether $M \triangle \tilde{E}(P)$ contains a $K_{t,t}$, which takes $\mathrm{O}(m)$ time, smaller than the complexity of a shortest path algorithm. Hence, it takes $\mathrm{O}(nD)$ time between augmentations. Since augmentations happen at most $tn/2$ times, the total complexity is $\mathrm{O}(tn^2 D)$, which gets to $\mathrm{O}(tn^2 m + tn^3 \log n)$ by employing $D = \mathrm{O}(m + n \log n)$ [9].

Theorem 4.2. Algorithm Minimum-Weight $K_{t,t}$-Free t-Factor *runs in* $\mathrm{O}(tn^2 m + tn^3 \log n)$ *time.*

As was true for Algorithm Minimum-Weight Square-Free 2-Factor, Algorithm Minimum-Weight $K_{t,t}$-Free t-Factor is fully combinatorial, and thus implies the following integrality theorem.

Theorem 4.3. *Let* (G, w) *be a weighted bipartite graph such that* G *admits a* $K_{t,t}$*-free* t*-factor and* $w \in \mathbf{R}_+^E$ *is integer and vertex-induced on any* $K_{t,t}$*. Then, the linear program* (P_t) *has an integral optimal solution. Moreover, the dual problem* (D_t) *also has an integral optimal solution* (p, q, r) *such that the elements in* $\{S \mid S \in \mathcal{S}_t,\ r(S) > 0\}$ *are pairwise disjoint.*

5 Concluding Remarks

This paper has dealt with the minimum-weight $K_{t,t}$-free t-factor problem in bipartite graphs, whereas Makai [16] considered the maximum-weight $K_{t,t}$-free t-matching problem. Let us close this paper by mentioning that these two problems are equivalent.

In fact, the two problems are polynomially reducible to each other. Given an instance (G, w) of the minimum-weight $K_{t,t}$-free t-factor problem such that $G = (U, V; E)$ and w is vertex-induced on any $K_{t,t}$, consider a weight vector $w' \in \mathbf{R}^E$ defined by $w'(e) = L - w(e)$, where L is a sufficiently large number. Note that w' is vertex-induced on any $K_{t,t}$ in G. Then, a maximum-weight $K_{t,t}$-free t-matching in (G, w') is a maximum cardinality $K_{t,t}$-free t-matching and hence gives a solution of the minimum-weight $K_{t,t}$-free t-factor problem in (G, w).

Conversely, let us given an instance (G, w) of the maximum-weight $K_{t,t}$-free t-matching problem, where $G = (U, V; E)$ and w is vertex-induced on any $K_{t,t}$. We also assume that $|U| \geq t$ and $|V| \geq t$, as is to be expected. Then, construct a new weighted graph (G', w') as follows. If $|U| \neq |V|$, add isolated dummy vertices to have $|U| = |V|$. For any pair of vertices $u \in U$ and $v \in V$, add $2t + 2$ vertices $u_0, u_1, \ldots, u_t, v_0, v_1, \ldots v_t$ and edges

$$\{uv_0, u_0 v\} \cup (\{u_i v_j \mid i \in \{0, 1, \ldots, t\},\ j \in \{0, 1, \ldots, t\}\} \setminus \{u_0 v_0\}).$$

Then, define a new weight vector w' by

$$w'(e) = \begin{cases} L - w(e) & (e \in E), \\ L & (e\colon \text{new edge}), \end{cases}$$

where L is a sufficiently large number. Now, observe that w' is vertex-induced on any $K_{t,t}$ in G' and G' admits a $K_{t,t}$-free t-factor. Moreover, for a minimum-weight $K_{t,t}$-free t-factor M in (G', w'), $M \cap E$ is a maximum-weight $K_{t,t}$-free t-matching in (G, w).

Acknowledgements

The author is grateful to Satoru Iwata for discussions on this topic.

References

[1] Cornuéjols, G., and Pulleyblank, W.: A matching problem with side conditions, *Discrete Mathematics*, 29 (1980), 135–159.

[2] Cunningham, W. H.: Matching, matroids, and extensions, *Mathematical Programming*, 91 (2002), 515–542.

[3] Cunningham, W. H., and Wang, Y.: Restricted 2-factor polytopes, *Mathematical Programming*, 87 (2000), 87–111.

[4] Edmonds, J., and Karp, R. M.: Theoretical improvements in algorithmic efficiency for network flow problems, *Journal of the ACM*, 19 (1972), 248–264.

[5] Fleiner, T.: Uncrossing a family of set-pairs, *Combinatorica*, 21 (2001), 145–150.

[6] Frank, A.: Restricted t-matchings in bipartite graphs, *Discrete Applied Mathematics*, 131 (2003), 337–346.

[7] Frank, A., and Jordán, T.: Minimal edge-coverings of pairs of sets, *Journal of Combinatorial Theory*, *Series B*, 65 (1995), 73–110.

[8] Frank, A., and Tardos, E.: An application of simultaneous diophantine approximation in combinatorial optimization, *Combinatorica*, 7 (1987), 49–65.

[9] Fredman, M. L., and Tarjan, R. E.: Fibonacci heaps and their uses in improved network optimization algorithms, *Journal of the ACM*, 34 (1987), 596–615.

[10] Geelen, J. F.: The C_6-free 2-factor problem in bipartite graphs is NP-complete, unpublished, 1999.

[11] Hartvigsen, D.: *Extensions of Matching Theory*, Ph.D. thesis, Carnegie Mellon University, 1984.

[12] Hartvigsen, D.: The square-free 2-factor problem in bipartite graphs, in Cornuéjols, G., Burkard, R. E., and Woeginger, G. J., eds., *Integer Programming and Combinatorial Optimization: Proceedings of the 7th International IPCO Conference*, LNCS 1610, Springer-Verlag, 1999, 234–241.

[13] Hartvigsen, D.: Finding maximum square-free 2-matchings in bipartite graphs, *Journal of Combinatorial Theory, Series B*, 96 (2006), 693–705.

[14] Hell, P., Kirkpatrick, D., Kratochvíl, J., and Kříž, I.: On restricted two-factors, *SIAM Journal on Discrete Mathematics*, 1 (1988), 472–484.

[15] Király, Z.: C_4-free 2-factors in bipartite graphs, Technical report, TR-2001-13, Egerváry Research Group, 1999.

[16] Makai, M.: On maximum cost $K_{t,t}$-free t-matchings of bipartite graphs, *SIAM Journal on Discrete Mathematics*, 21 (2007), 349–360.

[17] Pap, G.: Alternating paths revisited II: restricted b-matchings in bipartite graphs, Technical report, TR-2005-13, Egerváry Research Group, 2005.

[18] Pap, G.: *A Constructive Approach to Matching and Its Generalizations*, Ph.D. thesis, Eötvös Loránd University, 2006.

[19] Pap, G.: Combinatorial algorithms for matchings, even factors and square-free 2-factors, *Mathematical Programming*, 110 (2007), 57–69.

[20] Russell, M.: *Restricted Two-Factors*, Master's thesis, University of Waterloo, 2001.

[21] Tomizawa, N.: On some techniques useful for solution of transportation network problems, *Networks*, 1 (1971), 173–194.

[22] Végh, L. A., and Benczúr, A. A.: Primal-dual approach for directed vertex connectivity augmentation and generalizations, in *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms*, ACM-SIAM, 2005, 186–194.

[23] Vornberger, O.: Easy and hard cycle covers, Preprint, Universität Paderborn, 1980.