MATHEMATICAL ENGINEERING TECHNICAL REPORTS

A Weighted Even Factor Algorithm

Kenjiro TAKAZAWA

(Communicated by Satoru IWATA)

METR 2005–17 July 2005

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.i.u-tokyo.ac.jp/mi/mi-e.htm

A Weighted Even Factor Algorithm

Kenjiro TAKAZAWA*

July, 2005

Abstract

An even factor in a digraph is a collection of vertex-disjoint dipaths and even dicycles, which generalizes a path-matching introduced by Cunningham and Geelen (1997). In a restricted class of digraphs, called odd-cycle-symmetric, Pap (2005) presented a combinatorial algorithm to find a maximum even factor. In a certain class of weighted digraphs, called odd-cycle-symmetric, Király and Makai (2004) showed a linear programming that describes the maximum weight even factor problem, and proved the dual integrality.

In this paper, we present a primal-dual algorithm to find a maximum weight even factor for an odd-cycle-symmetric weighted digraph. This algorithm is based on the weighted matching algorithm and Pap's even factor algorithm. The running time of the algorithm is $\mathrm{O}(n^2 m)$, where n and m are the numbers of the vertices and arcs, respectively. This is the first fully combinatorial strongly polynomial algorithm even for the weighted path-matching problem. The algorithm also gives a constructive proof for the dual integrality.

1 Introduction

As a common generalization of the matching and matroid intersection problems, Cunningham and Geelen [1] introduced the notion of *path-matching* in undirected graphs. In [1], they showed a min-max formula, a totally dual integral polyhedral description, and polynomial-time solvability of the path-matching problem, which are similar to those in the matching and matroid intersection problems. Frank and Szegő [7] simplified the min-max formula and presented its combinatorial proof. Spille and Szegő [14] proved that the path-matching problem has a property which generalizes the Edmonds-Gallai structure of matchings.

The proof of polynomial-time solvability in [1] depends on the ellipsoid method. Cunningham and Geelen [3] presented an algebraic algorithm to find a maximum path-matching, which lead us to another proof of polynomial-time solvability. So, a polynomial-time combinatorial algorithm to find a maximum path-matching had been desired. Spille and Weismantel [15] proposed to generalize Edmonds' matching algorithm [6] to path-matchings.

For a combinatorial approach to path-matching, Cunningham and Geelen [2] introduced the *even factor problem* in digraphs, which is yet a generalization of the path-matching problem. It is shown by Cunningham and Geelen [3] that finding a maximum even factor is NP-hard in general digraphs. With an algebraic approach, however, Cunningham and Geelen [2] showed it can be done in polynomial time if the digraph is *weakly symmetric*. A digraph is said to be weakly symmetric if every arc in any dicycle has the reverse arc. Pap and Szegő [11] extended the min-max formula in [7] to the even factor problem in weakly symmetric digraphs. This formula is known to hold in a broader class of digraphs, called *odd-cycle-symmetric*. A digraph is odd-cycle-symmetric if each arc in any odd dicycle has the reverse arc. Pap and Szegő [11] also showed that the Edmonds-Gallai

*Department of Mathematical Informatics, Graduate School of Information Science and Technology, University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan. E-mail: `takazawa@misojiro.t.u-tokyo.ac.jp`

decomposition can be extended to even factors. Pap [10] gave a fully combinatorial algorithm to find a maximum even factor in odd-cycle-symmetric graphs, which provides an algorithmic proof of the min-max formula and the Edmonds-Gallai-type structure. This algorithm has some properties similar to Edmonds' matching algorithm [6].

Meanwhile, Cunningham and Geelen [2] also addressed a weighted generalization. They considered weakly symmetric digraphs and such weight vectors that an arc e and its reverse arc $\bar{e}$ (if exists) have the same weight. If a digraph $G = (V, A)$ is weakly symmetric and a weight vector $w \in \mathbb{R}^A$ satisfies this property, the weighted digraph (G, w) is called weakly symmetric. In [2], they presented an algebraic polynomial-time algorithm to compute a maximum weight even factor in a weakly symmetric weighted digraph, and proved the integrality of a polytope similar to the perfect matching polytope. Király and Makai [8] extended this result to an odd-cycle-symmetric weighted digraph. A weighted digraph (G, w) is called odd-cycle-symmetric if G is odd-cycle-symmetric, and, for every odd cycle C, the sum of the weight of the arcs in C is the same as that of its reverse cycle $\bar{C}$. For an odd-cycle-symmetric weighted digraph (G, w), they presented a linear programming that describes the maximum weight even factor problem, and proved the dual integrality.

In this paper, we present a combinatorial primal-dual algorithm to find a maximum weight even factor in an odd-cycle-symmetric weighted digraph (G, w). This algorithm combines Pap's algorithm [10] and the weighted matching algorithm of Edmonds [5] (see [9], for example). The resulting algorithm runs in $\mathrm{O}(n^2 m)$ time, where n and m are the numbers of the vertices and arcs, respectively. This is the first fully combinatorial strongly polynomial algorithm even for the weighted path-matching problem.

Besides a maximum weight even factor, this algorithm computes a dual optimal solution, which is integral if the weight vector w is integral. Thus, this algorithm also gives a constructive proof for the dual integrality of the linear programming in [8]. This situation is in contrast to the situation in matchings. The weighted matching algorithm of Edmonds [5] does not imply the total dual integrality, which was established by Cunningham and Marsh [4] with the aid of a primal algorithm, followed by a simpler proof given by Schrijver [12, 13].

2 The Weighted Even Factor Problem

Throughout this paper, digraphs are assumed to have no loops. A *path* is the arc set of a directed walk which does not revisit any vertex. A *cycle* is the arc set of a directed walk which ends in the starting vertex and does not revisit any internal vertex. For a cycle C, let $\bar{C}$ denote the reverse cycle of C (if exists), and $V(C)$ denote the set of vertices which appear in C. A path or cycle is called *odd* (resp. *even*) if the number of the arcs is odd (resp. even). An *even factor* is an arc set which consists of vertex-disjoint paths and even cycles.

For an arc e, let $\partial^- e$ (resp. $\partial^+ e$) denote the terminal (resp. initial) vertex of e. For a vertex v, $\delta^-(v)$ (resp. $\delta^+(v)$) denotes the set of arcs entering (resp. leaving) v. For a vertex set U, let $A[U]$ denote the set of arcs whose end vertices are both contained in U, and $G[U]$ denote the subgraph induced by U, that is, $G[U] := (U, A[U])$.

An arc is called *symmetric* if there is a reverse arc in the arc set of the digraph. A digraph is said to be *symmetric* if every arc is symmetric. A digraph is *odd-cycle-symmetric* if every arc in any odd cycle is symmetric. A weighted digraph (G, w) of a digraph $G = (V, A)$ and a weight vector $w \in \mathbb{R}^A$ is said to be *odd-cycle-symmetric*, if G is odd-cycle-symmetric and w satisfies that $w(C) = w(\bar{C})$ for every odd cycle C, where $w(C) := \sum_{e \in C} w_e$. Similarly, for a vector $x \in \mathbb{R}^A$ and an arc set $F \subseteq A$, in general we denote $x(F) = \sum_{e \in F} x_e$.

Let (G, w) be odd-cycle-symmetric. Király and Makai [8] presented a linear programming which describes the maximum weight even factor problem:

$$
\begin{aligned}
\text{(P)} \quad & \text{maximize} && wx && && (2.1)\\
& \text{subject to} && x\left(\delta^-(v)\right) \le 1 && (\forall v \in V), && (2.2)\\
& && x\left(\delta^+(v)\right) \le 1 && (\forall v \in V), && (2.3)\\
& && x\left(A[U]\right) \le |U| - 1 && (\forall \text{odd sets } U \subseteq V), && (2.4)\\
& && x \ge 0. && && (2.5)
\end{aligned}
$$

The dual problem of (P) is given by

$$
\begin{aligned}
\text{(D)} \quad & \text{minimize} && \sum_{v\in V}\left(\pi^-_v + \pi^+_v\right) + \sum_{U\colon \text{odd set}} y_U(|U|-1) && (2.6)\\
& \text{subject to} && \pi^-_{\partial^- e} + \pi^+_{\partial^+ e} + \sum_{\substack{\text{odd set } U,\\ e\in A[U]}} y_U \ge w_e \quad (\forall e \in A), && (2.7)\\
& && \pi^-_v,\ \pi^+_v,\ y_U \ge 0. && (2.8)
\end{aligned}
$$

Király and Makai [8] showed that if the weighted digraph (G, w) is odd-cycle-symmetric, both (P) and (D) have integral optimal solutions.

Theorem 2.1 (Király and Makai [8]). *If a weighted digraph (G, w) is odd-cycle-symmetric, the linear programming problem (P) has an integral optimal solution. Moreover, the dual problem (D) also has an integral optimal solution such that $\mathcal{F} = \{U \mid y_U > 0\}$ is a laminar family and $G[U]$ is strongly connected and symmetric for each $U \in \mathcal{F}$.*

The following definitions are of *alternating walk/path*, which appear in Pap's even factor algorithm [10]. They can be considered as extensions of those in matchings.

Definition 2.2. *Let M be a fixed even factor in a digraph $G = (V, A)$. A sequence $W = (v_0, e_1, v_1, \ldots, v_{k-1}, e_k, v_k)$ is an M-*alternating walk *if there is no arc in M which leaves v_0, $e_i = v_{i-1}v_i \in A \setminus M$ if i is odd, and $e_i = v_i v_{i-1} \in M$ if i is even. A vertex v_i is called* odd *(resp.* even*) if i is odd (resp. even).*

Definition 2.3. *An M-alternating walk is an M-*alternating path *if its even vertices are pairwise distinct, and its odd vertices are pairwise distinct.*

Using these definitions, we define an *M-alternating tree.*

Definition 2.4. *The union of M-alternating paths is called an M-*alternating tree *if these M-alternating paths start from a common vertex r. The vertex r is called a* root*. The set of arcs in P is denoted by $A[P]$.*

Note that the underlying graph of an M-alternating tree is *not* necessarily a tree, and there may exist a vertex that is both odd and even.

We are now ready to present how to construct *alternating forests*. Fix an even factor M. The aim of the construction is to find an *augmenting path* P and take the symmetric difference of M and $A[P]$, which is denoted by $M \triangle A[P]$.

First, choose each vertex r such that $\delta^+(r) \cap M = \emptyset$ to be a root. Then, grow an r-rooted M-alternating tree from each root r. Each vertex cannot be contained in multiple M-alternating trees, unless it is contained in two trees, even in one and odd in the other. Suppose an r-rooted M-alternating tree reaches an odd vertex u such that $\delta^-(u) \cap M = \emptyset$. The M-alternating path P from r to u is called an *augmenting path* if $M' = M \triangle A[P]$ does not contain any odd cycle. This

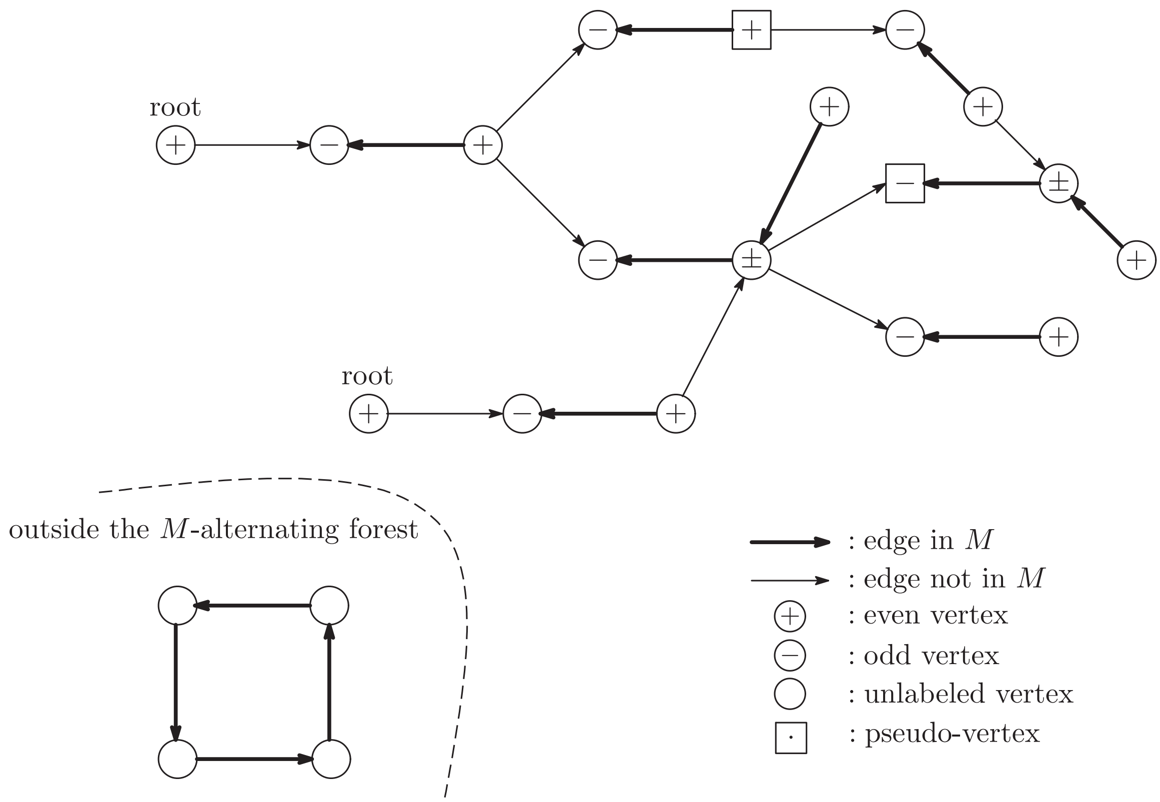

Figure 1: An M-alternating forest.

naming comes from the fact that M' is also an even factor and $|M'| = |M| + 1$. This operation of taking the symmetric difference is called an *augmentation*.

If M' contains an odd cycle C, we avoid the creation of the odd cycle as follows. Suppose $P = (v_0, e_1, v_1, \ldots, v_{k-1}, e_k, v_k)$ with $r = v_0$ and $u = v_k$. Then we denote $P_i = (v_0, e_1, v_1, \ldots, v_{i-1}, e_i, v_i)$ for each $i \leq k$. Let e_j be the arc which is the remotest arc from r in $P \cap C$. Then, take the symmetric difference of M and $A[P_{j-2}]$, and contract $G[V(C)]$. Such a contraction is said to be *shrinking*, and the resulting vertex $v(V(C))$ is called a *pseudo-vertex*. We call the subgraph obtained by growing M-alternating trees and contracting odd cycles an *M-alternating forest*. Recall that the underlying graph of an M-alternating forest is not necessarily a forest. Vertices outside an M-alternating forest are called *unlabeled*. An example of an M-alternating forest is shown in Figure 1.

3 A Primal-Dual Algorithm

In this section, we describe an algorithm to find a maximum weight even factor in a odd-cycle-symmetric weighted digraph (G, w). Since we are concerned with a maximum weight even factor, we assume without loss of generality that the weight w is positive.

For each $e \in A$, define $c_e := \pi^-_{\partial^- e} + \pi^+_{\partial^+ e} + \sum_{U: e \in A[U]} y_U - w_e$. Then, the complementary slackness conditions of the primal and dual linear programs are as follows:

$$
\begin{aligned}
&c_e x_e = 0 && (\forall e \in A), && (3.1)\\
&(|U| - 1 - x(A[U]))\, y_U = 0 && (\forall \text{odd sets } U \subseteq V), && (3.2)\\
&\big(1 - x(\delta^-(v))\big)\, \pi_v^- = 0 && (\forall v \in V), && (3.3)\\
&\big(1 - x(\delta^+(v))\big)\, \pi_v^+ = 0 && (\forall v \in V). && (3.4)
\end{aligned}
$$

The algorithm begins with a pair of primal and dual feasible solutions that satisfies the conditions (3.1)–(3.3). It maintains the feasibility and (3.1)–(3.3). The optimality is achieved when the condition (3.4) is satisfied.

An initial pair of primal and feasible solutions satisfying (3.1)–(3.3) is given by

$$
\begin{aligned}
&x_e := 0 && (\forall e \in A), && (3.5)\\
&y_U := 0 && (\forall \text{odd sets } U \subseteq V), && (3.6)\\
&\pi_v^- := 0 && (\forall v \in V), && (3.7)\\
&\pi_v^+ := \max_{e \in A} w_e && (\forall v \in V). && (3.8)
\end{aligned}
$$

Let $A^= := \{e \in A \mid c_e = 0\}$ and $G^= := (V, A^=)$. Throughout the algorithm, we keep $x_e = 0$ for $e \in A \setminus A^=$, in order to maintain (3.1).

Here is the steps of the algorithm.

Step 0 (Initialization): Start with the primal and dual solutions given by (3.5)–(3.8). $M := \emptyset$, $A^= := \arg\max\{w_e\}$.

Step 1: Construct an M-alternating forest in $G^=$. If an augmenting path P is found, then go to Step 2. Otherwise, go to Step 3.

Step 2 (Augmentation): Update the primal solution by $M := M \triangle A[P]$. If (3.4) is satisfied, then go to Step 4. Otherwise, go to Step 1.

Step 3 (Dual Change): Apply the dual change given below by (3.9)–(3.11). If (3.4) is satisfied, then go to Step 4. Otherwise, update $G^=$, expand each pseudo-vertex $v(U)$ with $y_U = 0$, and then go to Step 1.

Step 4 (Expand): Expand each pseudo-vertex $v(U)$. The current pair of primal and dual solutions is optimal.

We now describe the dual change in Step 3. Define the following sets:

$$
\begin{aligned}
\mathcal{U}^+ &:= \{\text{odd set } U \mid U \text{ is the vertex set of a shrunk blossom}\\
&\qquad\qquad\qquad \text{represented by an even pseudo-vertex}\},\\
\mathcal{U}^- &:= \{\text{odd set } U \mid U \text{ is the vertex set of a shrunk blossom}\\
&\qquad\qquad\qquad \text{represented by an odd pseudo-vertex}\},\\
\mathcal{U}^\circ &:= \{\text{odd set } U \mid U \text{ is the vertex set of a shrunk blossom}\\
&\qquad\qquad\qquad \text{represented by an unlabeled pseudo-vertex}\},\\
V^+ &:= \{v \in V \mid v \text{ is even or } v \in U \text{ for some } U \in \mathcal{U}^+\},\\
V^- &:= \{v \in V \mid v \text{ is odd or } v \in U \text{ for some } U \in \mathcal{U}^-\},\\
V^\circ &:= \{v \in V \mid v \text{ is unlabeled or } v \in U \text{ for some } U \in \mathcal{U}^\circ\},\\
A^+ &:= \{e \in A \mid \partial^+ e \in V^+, \partial^- e \notin V^-\}.
\end{aligned}
$$

The dual variables are changed as

$$\pi_v^+ := \pi_v^+ - \delta \quad (\text{for } v \in V^+), \tag{3.9}$$

$$\pi_v^- := \pi_v^- + \delta \quad (\text{for } v \in V^-), \tag{3.10}$$

$$y_U := \begin{cases} y_U + \delta & (\text{for } U \in \mathcal{U}^+), \\ y_U - \delta & (\text{for } U \in \mathcal{U}^-), \end{cases} \tag{3.11}$$

where

$$\delta := \min\{\delta_1, \delta_2, \delta_3\},$$

$$\delta_1 := \begin{cases} \min\limits_{v \in V^+} \pi_v^+ & (\text{if } V^+ \neq \emptyset), \\ \infty & (\text{if } V^+ = \emptyset), \end{cases}$$

$$\delta_2 := \begin{cases} \min\limits_{U \in \mathcal{U}^-} y_U & (\text{if } \mathcal{U}^- \neq \emptyset), \\ \infty & (\text{if } \mathcal{U}^- = \emptyset), \end{cases}$$

$$\delta_3 := \begin{cases} \min\limits_{e \in A^+} c_e & (\text{if } A^+ \neq \emptyset), \\ \infty & (\text{if } A^+ = \emptyset). \end{cases}$$

We verify the validity of this algorithm. First, we show that an augmentation does not ruin the feasibility nor the conditions (3.1)–(3.3).

Proposition 3.1. *If the primal and dual feasible solutions before an augmentation satisfy the conditions* (3.1)–(3.3)*, the solutions after the augmentation are also feasible and satisfy* (3.1)–(3.3).

Proof. **(Feasibility)** An augmentation does not interrupt the dual feasibility because it only changes the primal solution. The primal feasibility is ensured by the fact that $M \triangle A[P]$ is an even factor.

(Condition (3.1)**)** In an augmentation, we change x_e only if $e \in A^= = \{e \in A \mid c_e = 0\}$, which implies that the condition (3.1) holds.

(Condition (3.2)**)** For an odd set U, the value y_U is positive only if $G[U]$ is shrunk. With $G[U]$ being shrunk, we have $x(A[U]) = |U| - 1$ and $x(A[U])$ is not changed by an augmentation.

(Condition (3.3)**)** The only possibility for an augmentation to violate (3.3) is that $x(\delta^-(v))$ changes from zero to one for a vertex v with $\pi_v^- > 0$. The value $\pi_v^- > 0$ means that v had been in V^-, which implies $x(\delta^-(v)) = 1$, before the augmentation. As the value $x(\delta^-(v))$ does not decrease through the algorithm, there does not exist a vertex v with $x(\delta^-(v)) = 0$ and $\pi_v^- > 0$.

□

Next, we describe that the value δ is determined so that the dual change does not interrupt the dual feasibility and the complementary slackness conditions (3.1)–(3.3).

Proposition 3.2. *If the primal and dual feasible solutions before a dual change satisfy the conditions* (3.1)–(3.3)*, the primal and dual solutions after the dual change are also feasible and satisfy* (3.1)–(3.3).

Proof. **(Feasibility)** As a dual change does not change the primal solution, it suffices to check the dual feasibility. As $\delta \leq \delta_1 = \min_{v \in V^+} \pi_v^+$ and $\delta \leq \delta_2 = \min_{U \in \mathcal{U}^-} y_U$, the condition (2.8) holds after a dual change. Furthermore, we can see the condition (2.7) holds from the fact that c_e decreases only if $e \in A^+$ and that $\delta \leq \delta_3 = \min_{e \in A^+} c_e$.

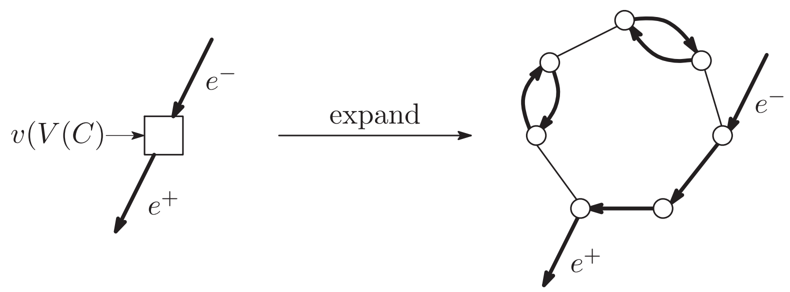

Figure 2: Expanding $v(V(C))$ with $|C| = 7$.

(Condition (3.1)**)** It suffices to show that c_e does not change in a dual change for every arc e with $x_e = 1$. If $x_e = 1$, the both ends of e are unlabeled or $\partial^+ e$ is even and $\partial^- e$ is odd, which implies c_e does not change.

(Condition (3.2)**)** In a dual change, the value y_U is changed only if $G[U]$ is shrunk, which implies $x(A[U]) = |U| - 1$.

(Condition (3.3)**)** The value π_v^- is changed only if $v \in V^-$, which implies $\delta^-(v) = 1$.

□

Finally, we describe how to expand pseudo-vertices. The following proposition assures the validity of shrinking and expanding.

Proposition 3.3. *If all arcs in an odd cycle C are contained in $A^=$, then all arcs in $\bar{C}$ are also contained in $A^=$.*

Proof. Note that $c(C) = c(\bar{C})$, which follows from the odd-cycle-symmetry of (G, w).

As every arc in C is contained in $A^=$, we have $c_e = 0$ for each $e \in C$. Thus, $c(\bar{C}) = c(C) = 0$. From the feasibility of the current solution, it holds that $c_e \geq 0$ for each $e \in A$. Hence, $c(\bar{C}) = 0$ implies that $c_e = 0$ for every $e \in \bar{C}$. □

From Proposition 3.3, in expanding $v(V(C))$ we can choose any arc in $C \cup \bar{C}$ to be contained in M. The proof of the following proposition tells us which arcs to be chosen.

Proposition 3.4 (Pap [10]). *Suppose $G = (V, A)$ is odd-cycle-symmetric, C is an odd cycle in G, G' is a digraph obtained by shrinking $V(C)$, M is an even factor in G'. Then, there is an even factor of size $|M| + |C| - 1$ in G.*

Proof. Suppose the pseudo-vertex $v(V(C))$ has a leaving arc e^+ and an entering arc e^- in M. As C is odd, there exists an even path P from $\partial^- e^-$ to $\partial^+ e^+$ in C. In addition, there exist cycles of length two that cover every vertex exposed by P exactly once. The union of M, P and these cycles is an even factor in G of size $|M| + |C| - 1$. (An example is shown in Figure 2.)

If there does not exist e^+ and/or e^-, it can be proved by a similar argument. □

4 Complexity Analysis

In this section, we discuss the time complexity of the algorithm. Recall that $n = |V|$ and $m = |A|$.

Lemma 4.1. *The dual change does not degenerate, that is, $\delta > 0$.*

Proof. ($\delta_1 > 0$) Suppose $\delta_1 = 0$, which means there exists a vertex $v \in V^+$ with $\pi_v^+ = 0$. Throughout the algorithm, the condition (3.4) is violated only by the root vertices, which are the elements of V^+. Since an augmentation does not decrease the number of arcs in M leaving each vertex, a root r has been in V^+ all through the algorithm. Hence, π_r^+ has been decreased in every dual change. Then it follows that all the root vertices achieve $\min_{v \in V^+}\{\pi_v^+\} = 0$. Thus the condition (3.4) holds, which suggests that the algorithm would have terminated.

($\delta_2 > 0$) After a dual change, each pseudo-vertex $v(U)$ with $y_U = 0$ is expanded. Therefore, the only possibility for a pseudo-vertex $v(U)$ to have the value $y_U = 0$ is that $G[U]$ has been shrunk since the last augmentation, which implies $U \in \mathcal{U}^+$. Thus, $U \in \mathcal{U}^-$ implies $y_U > 0$.

($\delta_3 > 0$) If $c_e = 0$ for some $e \in A^+$, then e is contained in $G^=$. Then, the arc e can be added to an alternating tree which contains $\partial^+ e$ as an even vertex, which implies $\partial^- e \in V^-$. Therefore, there does not exist an arc $e \in A^+$ with $c_e = 0$.

□

Lemma 4.1 tells us how many dual changes can happen between augmentations.

Proposition 4.2. *There are at most $4n/3$ dual changes between augmentations.*

Proof. ($\delta = \delta_1$) It follows from the proof of Lemma 4.1 that the algorithm terminates once $\delta = \delta_1$ happens.

($\delta = \delta_2$) For some $U \in \mathcal{U}^-$, the value y_U becomes zero and $G[U]$ is expanded, so that $|\mathcal{U}^-|$ decreases at least by one. After an augmentation, there can be at most $n/3$ odd pseudo-vertices and all new pseudo-vertices created between augmentations are even. Hence, $\delta = \delta_2$ can happen at most $n/3$ times between augmentations.

($\delta = \delta_3$) For some $e \in A^+$, the reduced cost c_e becomes zero, so that e is added to $A^=$ and $\partial^- e$ comes to be contained in V^-. The result is that $|V^-|$ increases at least by one, or an augmenting path is found. Hence, $\delta = \delta_3$ can happen at most n times between augmentations.

□

It takes $\mathrm{O}(m)$ time to determine δ, so the dual changes between augmentations require $\mathrm{O}(nm)$ time. The construction of an M-alternating forest takes $\mathrm{O}(n^2)$ time. In finding an M-alternating path P, we have to check whether $M \triangle A[P]$ contains an odd cycle. This exploration requires $\mathrm{O}(n)$ time and happens $\mathrm{O}(n)$ times between augmentations. Thus, the most time consuming part of this algorithm is the dual changes. Since there are at most n augmentations, the total time complexity of the algorithm is $\mathrm{O}(n^2 m)$.

Theorem 4.3. *Given an odd-cycle-symmetric weighted digraph (G, w), the algorithm finds a maximum weight even factor in $\mathrm{O}(n^2 m)$ time.*

5 Concluding Remarks

We have presented a fully combinatorial algorithm to find a maximum weight even factor in an odd-cycle-symmetric weighted digraph (G, w) in $\mathrm{O}(n^2 m)$ time. This algorithm obtains primal and dual optimal solutions simultaneously.

In the dual changes, we only use the operations of addition, subtraction, and comparison. Therefore, if the weight vector w is integral, the dual optimal solution obtained by the algorithm is also integral. Moreover, the family $\mathcal{F} = \{U \mid y_U > 0\}$ is the collection of shrunk odd sets, which implies that $\mathcal{F}$ is a laminar family and $G[U]$ is strongly connected and symmetric for all $U \in \mathcal{F}$. Thus, this algorithm provides a constructive proof of Theorem 2.1.

Acknowledgements

I am very obliged to Satoru Iwata for his happy suggestions and reading the draft of the paper.

References

[1] Cunningham, W. H., and Geelen, J. F.: The optimal path-matching problem, *Combinatorica*, vol. 17 (1997), 315–337.

[2] Cunningham, W. H., and Geelen, J. F.: Vertex-disjoint directed paths and even circuits, manuscript, 1998.

[3] Cunningham, W. H., and Geelen, J. F.: Combinatorial algorithms for path-matching, manuscript, 2000.

[4] Cunningham, W. H., and Marsh, III, A. B.: A primal algorithm for optimum matching, *Mathematical Programming Study*, vol. 8 (1978), 50–72.

[5] Edmonds, J.: Maximum matching and a polyhedron with 0,1-vertices, *Journal of Research of the National Bureau of Standards*, *Section B*, vol. 69 (1965), 125–130.

[6] Edmonds, J.: Paths, trees, and flowers, *Canadian Journal of Mathematics*, vol. 17 (1965), 449–467.

[7] Frank, A., and Szegő, L.: Note on the path-matching formula, *Journal of Graph Theory*, vol. 41 (2002), 110–119.

[8] Király, T., and Makai, M.: On polyhedra related to even factors, in Bienstock, D., and Nemhauser, G., eds., *Integer Programming and Combinatorial Optimization*: *Proceedings of the 10th International IPCO Conference*, LNCS 3064, Springer-Verlag, 2004, 416–430.

[9] Nemhauser, G. L., and Wolsey, L. A.: *Integer and Combinatorial Optimization*, John Wiley and Sons, New York, 1988.

[10] Pap, G.: A combinatorial algorithm to find a maximum even factor, in Jünger, M., and Kaibel, V., eds., *Integer Programming and Combinatorial Optimization*: *Proceedings of the 11th International IPCO Conference*, LNCS 3509, Springer-Verlag, 2005, 66–80.

[11] Pap, G., and Szegő, L.: On the maximum even factor in weakly symmetric graphs, *Journal of Combinatorial Theory*, *Series B*, vol. 91 (2004), 201–213.

[12] Schrijver, A.: Min-max results in combinatorial optimization, in Bachem, A., Grötschel, M., and Korte, B., eds., *Mathematical Programming—The State of the Art*, Springer-Verlag, Berlin, 1983, 439–500.

[13] Schrijver, A.: Short proofs on the matching polyhedron, *Journal of Combinatorial Theory*, *Series B*, vol. 34 (1983), 104–108.

[14] Spille, B., and Szegő, L.: A Gallai-Edmonds-type structure theorem for path-matchings, *Journal of Graph Theory*, vol. 46 (2004), 93–102.

[15] Spille, B., and Weismantel, R.: A generalization of Edmonds' matching and matroid intersection algorithms, in Cook, W. J., and Schulz, A. S., eds., *Integer Programming and Combinatorial Optimization*: *Proceedings of the 9th International IPCO Conference*, LNCS 2337, Springer-Verlag, 2002, 9–20.