Vol. 48 No. 10 IPSJ Journal Oct. 2007

Recommended Paper

Optimal Balanced Semi-Matchings for Weighted Bipartite Graphs

Yuta Harada,† Hirotaka Ono,† Kunihiko Sadakane†
and Masafumi Yamashita†

The matching of a bipartite graph is a structure that can be seen in various assignment problems and has long been studied. The semi-matching is an extension of the matching for a bipartite graph $G = (U \cup V, E)$. It is defined as a set of edges, $M \subseteq E$, such that each vertex in U is an endpoint of exactly one edge in M. The load-balancing problem is the problem of finding a semi-matching such that the degrees of each vertex in V are balanced. This problem is studied in the context of the task scheduling to find a "balanced" assignment of tasks for machines, and an $O(|E||U|)$ time algorithm is proposed. On the other hand, in some practical problems, only balanced assignments are not sufficient, e.g., the assignment of wireless stations (users) to access points (APs) in wireless networks. In wireless networks, the quality of the transmission depends on the distance between a user and its AP; shorter distances are more desirable. In this paper, We formulate the min-weight load-balancing problem of finding a balanced semi-matching that minimizes the total weight for weighted bipartite graphs. We then give an optimal condition of weighted semi-matchings and propose an $O(|E||U||V|)$ time algorithm.

1. Introduction

Finding a maximum matching in an undirected graph is one of the most traditional problems in the field of combinational optimization and has been intensively studied [3]. A matching is a set of edges sharing no vertices with each other. Actually, the maximum matching problem for bipartite graphs is one of the most classic problems and is known to have simple efficient algorithms [6],[7].

In this paper, we are concerned with a variation of the matching problem on a bipartite graph $G = (U \cup V, E)$, which is called the *semi-matching* problem. A semi-matching is defined as a set of edges, $M \subseteq E$, such that each vertex in U is an endpoint of exactly one edge in M. Suppose U and V represent set of tasks and set of machines, respectively. An edge between a task and a machine shows that the machine can process the task. In this setting, a semi-matching gives an assignment of the tasks to the machines. In such an assignment problem, finding a balanced assignment is often considered under the assumption that machines work independently in parallel [2]. This problem can be interpreted as the load-balancing problem, that is, the problem of obtaining a semi-matching in which the degrees of each V vertex are balanced [5]. For the problem, an algorithm which runs in $O(|E||U|)$ time is proposed.

Although the above problem is studied for unweighted graphs, some assignment problems should be considered under the setting with weights. As an example, we consider the problem of assigning wireless stations (users) to access points (APs) in wireless networks. A balanced assignment of users to APs is appropriate in wireless networks, otherwise users connected to an overloaded AP cannot expect effective communication. However the transmission quality also depends on the distance between a user and its AP. This implies that only balanced assignments do not always guarantee high-quality communication, and we also need to consider a goodness measure of the communication quality.

In considering the above discussion, we formulate the problem of finding a balanced semi-matching in which the total weight is minimized for weighted bipartite graphs, called *the minimum weight load-balancing problem.* We give an optimal condition of min-weight balanced semi-matchings and then propose an $O(|E||U||V|)$ time algorithm.

It should be noted that the objective of our problem is not to minimize the maximum weighted cost (load) of the semi-matching but to minimize the total weight of the bal-

† Graduate School of Information Science and Electrical Engineering, Kyushu University

An extended abstract of this paper was presented at Hinokuni Symposium 2006 and recommended to be submitted to IPSJ Journal by Manager of IPSJ Kyushu Branch

anced semi-matching. (The "balanced" is in terms of the (unweighted) degrees.) In passing, the former problem is equivalent to the restricted scheduling of unrelated parallel machines, which is known to be NP-hard[8]. This remains to be NP-hard even if the degree of every vertex in U is exactly 2, because the graph orientation problem[1], a special case of the semi-matching by regarding the edges of the orientation as U, is NP-hard.

The rest of the paper is organized as follows. Section 2 introduces the load-balancing problem and an algorithm proposed by Harvey, et al.[5]. In Section 3, we first formulate the min-weight load-balancing problem. We then discuss an optimal condition of the weighted balanced semi-matching. Based on this condition, we propose an algorithm. Section 4 concludes the paper.

2. Load-Balancing Problem

2.1 Preliminaries

Let $G = (U \cup V, E)$ be a simple bipartite graph, where U and V denotes a set of vertices and $E \subseteq U \times V$ denote a set of edges between U and V. Throughout the paper, let $m = |E|$, $n_1 = |U|$, $n_2 = |V|$ and $n = n_1 + n_2$ for the input graph. By $\{u, v\}$ for $u \in U$ and $v \in V$ we denote the edge with ends in u and v. Let $\delta(v) = \{\{u, v\} \in E\}$ and $\deg(v) = |\delta(v)|$ for a vertex $v \in V$, that is, $\delta(v)$ represents a set of edges having a vertex v as an endpoint and $\deg(v)$ is the degree of a vertex v. Similarly, $\delta(u)$ and $\deg(u)$ are defined for a vertex $u \in U$.

A *semi-matching* $M \subseteq E$ is defined as a set of edges such that each vertex in U is an endpoint of exactly one edge in M. Edge $e \in M$ and $e \notin M$ are called a *matching edge* and a *non-matching edge*, respectively. Let $\delta_M(v) = \{\{u, v\} \in M\}$ and $\deg_M(v) = |\delta_M(v)|$ for a semi-matching M. We similarly use $\delta_M(u)$ and $\deg_M(u)$ for $u \in U$.

Given a semi-matching M, we define $cost_M(v) = \deg^2_M(v)$ as the cost of a vertex $v \in V$. The total cost of a semi-matching M is defined as $T(M) = \sum_{v \in V} cost_M(v)$.

2.2 Load-Balancing Problem

The load-balancing problem[5] is given as follows:

The Load-Balancing Problem

Input: A simple bipartite graph $G = (U \cup V, E)$,

Output: A semi-matching $M \subseteq E$ minimizing the total cost $T(M)$.

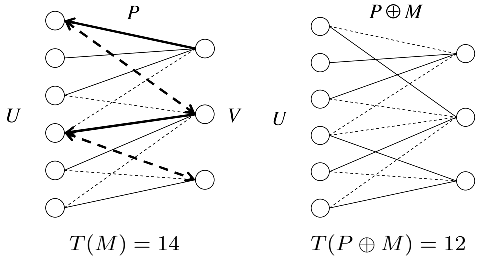


Fig. 1 Example of the switching operation along a cost-reducing path P.

We will call a semi-matching M minimizing $T(M)$ a *balanced semi-matching* from here on.

The load-balancing problem can be represented as restricted cases of scheduling on unrelated machines. If we respectively regard U, V as a set of tasks and machines, a balanced semi-matching M corresponds to a balanced assignment of tasks for machines.

2.3 Properties of Balanced Semi-Matchings

We introduce paths that characterize the optimality of balanced semi-matchings and their properties.

2.3.1 Alternating Path

For a given semi-matching M in G, define an *alternating path* as a sequence of edges $P = (\{v_1, u_1\}, \{u_1, v_2\}, \ldots, \{u_{k-1}, v_k\})$ with $v_i \in V$, $u_i \in U$, and $\{v_i, u_i\} \in M$ for each i. For convenience, we treat an alternating path as a sequence of vertices $P = (v_1, u_1, \ldots, u_{k-1}, v_k)$. For the path P, let $\overline{P}$ denote the path of its reverse order, that is, $\overline{P} = (v_k, \ldots, v_1)$.

We define the notation $A \oplus B$ as the symmetric difference of sets A and B; i.e. $A \oplus B = (A \setminus B) \cup (B \setminus A)$. If P is an alternating path with respect to a semi-matching M, then we can obtain a new semi-matching $P \oplus M$ by switching matching and non-matching edges along P. This switching operation decreases the degree of v_1 by one and increases the degree of v_k by one, but does not affect the degrees of any other vertices.

2.3.2 Cost-Reducing Path

In an alternating path $P = (v_1, \ldots, v_k)$ with respect to M, if $\deg_M(v_1) > \deg_M(v_k) + 1$ then P is called a *cost-reducing path*. This is because by switching edges along a cost-reducing path P, the total cost $T(M)$ decreases literally; that is, $T(P \oplus M) < T(M)$. The following Eq. (1) shows the correctness of that. **Figure 1** shows an example of a cost-reducing path and its switching operation, where bold lines represent the matching edges, and dotted lines rep-

Algorithm $\mathcal{A}_{SM1}$:

Step 0: Set a set of edges $M = \emptyset$, a set of vertices $S = U$.

Step 1: Select a vertex $u \in U$ from S and let $S = S \setminus \{u\}$ if $S \neq \emptyset$. Otherwise output M.

Step 2: Build an alternating search tree T rooted at u, where edges in M are directed from V to U and edges not in M are directed from U to V.

Step 3: Find a path $P = (u, \ldots, v)$ such that $\deg_M(v)$ is as small as possible in the tree T. Such a path P is called an *augmenting path.*

Step 4: Extend M by switching matching and non-matching edges along an augmenting path P, and go to Step 1.

Fig. 2 Procedure of $\mathcal{A}_{SM1}$.

resent the non-matching edges. By switching the matching and non-matching edges along the arrow direction (left), we obtain a new semi-matching (right). Throughout the paper, we use a similar manner in the figures.

$$T(M) - T(P \oplus M) = 2(\deg_M(v_1) - \deg_M(v_k) - 1) \qquad (1)$$

2.3.3 Optimality of Balanced Semi-Matchings

The following theorem about the optimality of balanced semi-matchings is proved in Ref. 5).

Theorem 1 [5] *A semi-matching M is a balanced semi-matching if and only if no cost-reducing path with respect to M exists.* □

We immediately have the next Theorem from the proof of Theorem 1, by utilizing the convexity of the quadratic function *cost*.

Theorem 2 *If a semi-matching M is a balanced semi-matching then maximum degree $\max_{v \in V}\{\deg_M(v)\}$ is minimized.* □

2.4 Algorithm: $\mathcal{A}_{SM1}$

We introduce an algorithm $\mathcal{A}_{SM1}$ [5] that solves the load-balancing problem in **Fig. 2**. $\mathcal{A}_{SM1}$ is a variation of Hungarian Algorithm [7] that is originally used to find a maximum bipartite matching.

The following lemma and theorem were proved about the operation of $\mathcal{A}_{SM1}$.

Lemma 3 [5] *No cost-reducing path is created in G while $\mathcal{A}_{SM1}$ executes.* □

Theorem 4 [5] *$\mathcal{A}_{SM1}$ produces a balanced semi-matching M in $O(mn_1)$ time.* □

3. Min-Weight Load-Balancing Problem

3.1 Preliminaries for Our Problem

In this section, we consider the semi-matching problem for the weighted simple bipartite graph $G = (U \cup V, E, w)$, where w denotes a positive weight function $w : E \to \boldsymbol{R}^+$. Each edge $\{u, v\} \in E$ has a weight $w(\{u, v\})$.

For a given semi-matching M, we define $w(M)$ as the total weight, that is to say, $w(M) = \sum_{\{u,v\} \in M} w(\{u, v\})$. For an alternating/augmenting path P in M, the weight increment w_P is defined as follows:

$$w_P = \sum_{e \in P \setminus M} w(e) - \sum_{e \in P \cap M} w(e)$$

w_P represents the increasing amount of total weight from $w(M)$ by switching operation along the alternating/augmenting path P.

We give some definitions of fundamental paths. In an alternating path $P = (v_1, \ldots, v_k)$, if $v_i \neq v_j$ for $^\forall v_i, ^\forall v_j (i \neq j)$ then P is called a *simple path.* If the initial vertex v_1 is equal to the end vertex v_k but no other vertices are equal to each other, we call P a *simple cycle.* In $P = (w_1, w_2, \ldots, w_{k-1}, w_k)$, moreover, we define a *subpath* as $P(w_i, w_j) = (w_i, \ldots, w_j)$ for $1 \leq i < j \leq k$. $P_1 \cdot P_2$ represents the concatenation of two paths for $P_1 = (w_i, \ldots, w_j)$ and $P_2 = (w_j, \ldots, w_k)$, i.e., $P_1 \cdot P_2 = (w_i, \ldots, w_j, \ldots, w_k)$.

3.2 Min-Weight Load-Balancing Problem

We give the min-weight load-balancing problem as follows:

The Min-Weight Load-Balancing Problem

Input: A weighted simple bipartite graph $G = (U \cup V, E, w)$,

Output: A balanced semi-matching $M \subseteq E$ minimizing the total weight $w(M)$.

We call a balanced semi-matching M minimizing the total weight $w(M)$ a *min-weight balanced semi-matching.*

As mentioned in Introduction, the min-weight load-balancing problem is useful for the assignment of wireless stations (users) to access points (APs) in wireless networks. In wireless networks composed of multiple APs, each user needs to choose an AP to connect itself to. It is known that the following negative effects may arise [4),9)].

(1) An overload of many users to a few specific APs deteriorates the throughput of each user in inverse proportion to the number of users connecting to them.
(2) As the distance between the user and the connected AP becomes longer, the communication quality becomes worse linearly.

In considering case (1), a balanced assignment of users to APs is appropriate to prevent the throughput degradation. However, distances between users and APs depend on the communication quality by (2), and this implies that balanced assignments do not always guarantee high-quality communication. Thus we consider that appropriate assignments are balanced assignments such that distances from users to APs are as short as possible.

By regarding U, V, E and w as a set of users, APs, communication links and distances between users and APs respectively, the min-weight load-balancing problem provides a min-weight balanced semi-matching M that is a good assignment of users to APs from the view points of both (1) and (2).

3.3 Properties of Min-Weight Balanced Semi-Matchings

We give some definitions of min-weight balanced semi-matchings and their properties.

3.3.1 Cost-Preserving Path · Cost-Preserving Cycle

In an alternating path $P = (v_1, u_1, \ldots, u_{k-1}, v_k)$, if $\deg_M(v_1) = \deg_M(v_k) + 1$ or $v_1 = v_k$ we call P a *cost-preserving path* or *cost-preserving cycle*, respectively.

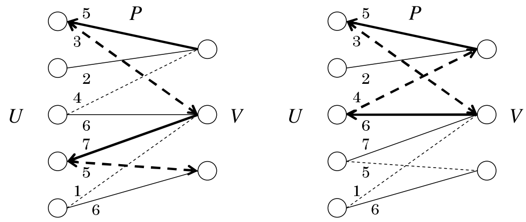
The switching operations along a cost-preserving path/cycle P preserves the total cost $T(M)$; that is, $T(P \oplus M) = T(M)$. The Eq. (1) clearly proves that the total cost $T(M)$ is preserved if $\deg_M(v_1) = \deg_M(v_k) + 1$. In the case of $v_1 = v_k$, $\deg_{P \oplus M}(v) = \deg_M(v)$ holds for any $v \in V$ because P is a cycle. Therefore the switching operation also does not affect the total cost in the case of $v_1 = v_k$.

3.3.2 Weight-Reducing Path · Weight-Reducing Cycle

If a cost-preserving path P has a negative weight increment, i.e. $w_P < 0$, P is called a *weight-reducing path.* Similarly, we call P a *weight-reducing cycle* for a cost-preserving cycle P.

Although switching along a weight-reducing path/cycle P does not affect the total cost $T(M)$, the total weight of the obtained semi-matching is smaller than the one of the previous semi-matching M; that is, $w(P \oplus M) < w(M)$. This is obvious from the following equation.

$$w(P \oplus M) = w(M) + w_P \qquad (2)$$



$\deg_M(v_1) = \deg_M(v_k) + 1$ $\qquad$ $v_1 = v_k$

Fig. 3 Weight-reducing path and cycle.

Figure 3 shows examples of weight-reducing path and cycle. We adopt similar line types to Fig. 1. Numbers by edges represent their weights.

3.3.3 Optimality of Min-Weight Balanced Semi-Matchings

We give an optimal condition of min-weight load-balanced semi-matchings by proving the next theorem.

Theorem 5 *In a balanced semi-matching M, M is a min-weight balanced semi-matching if and only if there exists no weight-reducing path/cycle with respect to M.*

Proof Let G be an input bipartite graph for the load-balancing problem and M be a balanced semi-matching in G. If weight-reducing paths/cycles with respect to M exist, M is not evidently minimum because the total weight $w(M)$ can be reduced. In the rest of the proof, we show that weight-reducing paths/cycles always exist in M when $w(M)$ is not minimum. Let M be a balanced semi-matching whose total weight $w(M)$ is not minimum, and let O be a min-weight balanced semi-matching that minimizes the size of the symmetric difference $|M \oplus O|$. Let G' be a subgraph defined by $G' = (U \cup V, M \oplus O)$.

Lemma 6 *If G' has a vertex $v_1 \in V$ with $\deg_M(v_1) > \deg_O(v_1)$, a cost-preserving path $P = (v_1, \ldots, v_k)$ with $\deg_M(v_k) < \deg_O(v_k)$ exists in G'.*

Proof We construct an alternating path $P = (v_1, \ldots, v_k)$ in G' from an arbitrary vertex $v_1 \in V$ with $\deg_M(v_1) > \deg_O(v_1)$ as follows: First set $P = (v_1)$ and $Q_M = Q_O = \emptyset$. We extend Q_M and Q_O by following the path alternately; Q_M and Q_O are the sets of edges in $M \setminus O$ and $O \setminus M$ respectively. (1) In each vertex $v_i \in V$, let $E(v_i) = \delta_{M \setminus O}(v_i) \setminus Q_M$, i.e., the set of edges connected to v_i but not in P. If both $E(v_i) \neq \emptyset$ and $\deg_M(v_i) \geq \deg_O(v_i)$ hold for v_i, we extend P by adding an arbitrary edge $\{v_i, u_i\} \in E(v_i)$. (Edge $\{v_i, u_i\}$ is inserted into Q_M.) Otherwise (i.e., v_i does not satisfy one of the two conditions), we stop the construction of P as v_i is the end vertex of P. This v_i satisfies $\deg_M(v_i) < \deg_O(v_i)$ as explained later. The initial vertex v_1 satisfies the above two conditions. (2) In each vertex $u_i \in U$, we follow the unique edge $\{u_i, v_{i+1}\} \in O \setminus M$ and add it into Q_O. Such an edge always exists by the definition of a semi-matching.

Now we show that the constructed path $P = (v_1, \ldots, v_k)$ satisfies $\deg_M(v_k) < \deg_O(v_k)$. If $\deg_M(v_k) \geq \deg_O(v_k)$ does not hold in (1), P is obviously an alternating path with $\deg_M(v_k) < \deg_O(v_k)$. We then consider the case where $E(v_k) = \emptyset$. If $v_1 = v_k$ holds, i.e., P is a simple but not necessarily elementary cycle, the indegree and the outdegree of $v_k(= v_1)$ on P are equal. This implies $\deg_M(v_1) = \deg_O(v_1)$, which contradicts the condition $\deg_M(v_1) > \deg_O(v_1)$. Thus, $v_1 \neq v_k$ holds, and P is not a cycle but a path, which leads to $|\delta_{Q_M}(v_1)| = |\delta_{Q_O}(v_1)| + 1$, $|\delta_{Q_M}(v_k)| = |\delta_{Q_O}(v_k)| - 1$, and $|\delta_{Q_M}(v_i)| = |\delta_{Q_O}(v_i)|$ for $i = 2, \ldots, k-1$. Since $\delta_{M \setminus O}(v_k) \subseteq Q_M$ holds by $E(v_k) = \emptyset$, $|\delta_{M \setminus O}(v_k)| \leq |\delta_{Q_M}(v_k)|$ holds. These equations conduce $\deg_{M \setminus O}(v_k) = |\delta_{M \setminus O}(v_k)| \leq |\delta_{Q_M}(v_k)| < |\delta_{Q_O}(v_k)| \leq |\delta_{O \setminus M}(v_k)| = \deg_{O \setminus M}(v_k)$, which shows P is an alternating path with $\deg_M(v_k) < \deg_O(v_k)$.

Next, we show P is a cost-preserving path; $\deg_M(v_1) = \deg_O(v_k) + 1$. Let us consider $\overline{P}$, the reverse of the path P, which is an alternating path with respect to O. Notice that both P and $\overline{P}$ are not cost-reducing paths because M and O are balanced semi-matchings. This implies that $\deg_M(v_1) \leq \deg_M(v_k) + 1$ and $\deg_O(v_k) \leq \deg_O(v_1) + 1$. These and $\deg_M(v_1) > \deg_O(v_1)$ and $\deg_M(v_k) < \deg_O(v_k)$ yield $\deg_M(v_1) = \deg_M(v_k) + 1$ and $\deg_O(v_k) = \deg_O(v_1) + 1$, which mean that both P and $\overline{P}$ are cost-preserving paths. □

Lemma 7 *If $\deg_M(v) = \deg_O(v)$ holds for all vertices $v \in V$ in G', a (cost-preserving) cycle $P = (v_1, \ldots, v_k)$ exists in G'.*

Proof From an arbitrary vertex $v_1 \in V$ in G', we build a path $P = (v_1, \ldots, v_k)$ as follows. Q_M, Q_O and $E(v_i)$ are defined as Lemma 6. (1) In each vertex $v_i \in V$, if $E(v_i) \neq \emptyset$ then we extend P by adding an arbitrary edge $\{v_i, u_i\} \in E(v_i)$ and insert it into Q_M. Otherwise let v_i be the end vertex of P, and output P. (2) In each vertex $u_i \in U$, we trace the unique edge $\{u_i, v_{i+1}\} \in O \setminus M$ and add it into Q_O. If $v_{i+1} = v_1$ holds, we set v_{i+1} as the end vertex and stop the construction.

We say the constructed $P = (v_1, \ldots, v_k)$ is a cycle. P output in (1) satisfies $E(v_k) = \emptyset$. Assume $v_k \neq v_1$; i.e., P is not a cycle but a path. By the similar augment of Lemma 6, $\deg_M(v_k) < \deg_O(v_k)$ holds, which contradicts $\deg_M(v_k) = \deg_O(v_k)$. Thus, P is a cycle. It is clear that P output in (2) is also a cycle. □

A cost-reducing path/cycle P with respect to M always exists in G' from Lemmas 6 and 7. Let us consider the case of $w_P > 0$. Because $w_P = -w_{\overline{P}}$, $w_{\overline{P}} < 0$ holds, $\overline{P}$ is a weight-reducing path/cycle with respect to O, however this contradicts the optimality of O. If $w_P = w_{\overline{P}} = 0$, by switching $\overline{P}$, we can obtain O' such that $|M \oplus O'| < |M \oplus O|$. This contradicts the minimality of $|M \oplus O|$. These contradictions conduce $w_P < 0$ and P is a weight-reducing path/cycle with respect to M, which completes the proof. □

Algorithm *WSM*:
Step 0-2: Same as $\mathcal{A}_{SM1}$.
Step 3: Apply the breadth-first search for T to find an *augmenting path* $P = (u, \ldots, v)$ that gives $\min\{w_P | P \in \mathcal{P}_u\}$ where $\mathcal{P}_u = \{(u, \ldots, v) | \deg_M(v)$ is minimum$\}$.
Step 4: Same as $\mathcal{A}_{SM1}$.

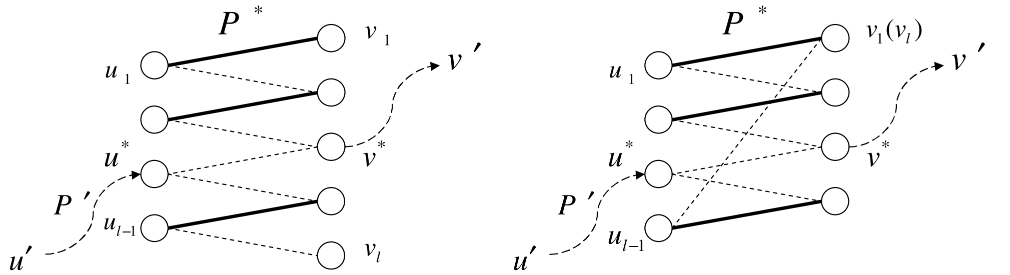
Fig. 4 Procedure of *WSM*.

3.4 Algorithm: *WSM*

Utilizing the optimality condition shown in the previous subsection, we propose an algorithm for solving the min-weight load-balancing problem. The algorithm extends the idea of algorithm $\mathcal{A}_{SM1}$ introduced in Section 2. In order to adapt the optimality condition, we add a new condition for the weight increment of an augmenting path P in Step 3. **Figure 4** shows our algorithm *WSM*. Note that since the new requirement for P is just additional, the semi-matching found by *WSM* is also balanced by Lemma 3.

The correctness of *WSM* is guaranteed by the next lemma.

Lemma 8 *Neither a weight-reducing path nor cycle is created in G during the execution of WSM.*



Case (1-1) $v_1 \neq v_l$ Case (1-2) $v_1 = v_l$

Fig. 5 Case (1) ($|Q| = 1$).

Proof Show by the contradiction. Let $P^* = (v_1, u_1, \ldots, u_{l-1}, v_l)$ denote the first weight-reducing path or cycle in G created by *WSM*. Let M' be the set of matching edges in the execution of the algorithm just before P^* is created. Here, we assume the path $P' = (u', \ldots, v')$ is found as the augmenting path for M', and *WSM* applies the switch operation along P'. We define the resulting set of matching edges as M^*. Let $Q = (M^* \setminus M') \cap P^*$. By the definition of P^*, $|Q| \geq 1$. $Q \subseteq P'$ holds because P^* is a path created by P'. In this situation, we show the contradictions by induction about $|Q|$ when P^* is a weight-reducing path ($v_1 \neq v_l$) and cycle ($v_1 = v_l$) for each case.

(1) First, we consider the case of $|Q| = 1$. Let $Q = \{\{v^*, u^*\}\}$. Since P' includes it, let $P' = (u', \ldots, u^*, v^*, \ldots, v')$. **Figure 5** shows an example of this case.

(1-1) Case of $v_1 \neq v_l$. Let $\deg_{M'}(v_1) = d$. Since P^* is a weight-reducing path, $\deg_{M'}(v_l) = d-1$ and $w_{P^*} < 0$. Additionally, $\deg_{M'}(v) \in \{d-1, d\}$ for any $v \in V$ existing on P^* because no cost-reducing paths with respect to M^* exist by Lemma 3. Since P' has an edge $\{u^*, v^*\} \in Q$, $P^*(v_1, v^*) \cdot P'(v^*, v')$ is an alternating path in M'. Because $P^*(v_1, v^*) \cdot P'(v^*, v')$ is not a cost-reducing path by the property of $\mathcal{A}_{SM1}$ (Lemma 3), $\deg_{M'}(v') \geq \deg_{M'}(v_1) - 1 = d-1$ holds. If $\deg_{M'}(v') \geq d$ then $P'(u', u^*) \cdot P^*(u^*, v_l)$ should be chosen as an augmenting path other than P' because $\deg_{M'}(v_l) < \deg_{M'}(v')$, contradiction. In the other case of $\deg_{M'}(v') = d-1$, i.e., $\deg_{M'}(v_l) = \deg_{M'}(v')$. Since $P^*(v_1, v^*) \cdot P'(v^*, v')$ is not a weight-reducing path,

$$w_{P^*(v_1,v^*)\cdot P'(v^*,v')} = w_{P^*(v_1,v^*)} + w_{P'(v^*,v')} \geq 0$$

holds. Also, we have

$$w_{P^*(v_1,v^*)\cdot P^*(v^*,v_l)} = w_{P^*(v_1,v^*)} + w_{P^*(v^*,v_l)} < 0$$

by $w_{P^*} < 0$. These inequalities yield

$$w_{P^*(v^*,v_l)} < -w_{P^*(v_1,v^*)} \leq w_{P'(v^*,v')}.$$

This inequality is transformed as

$$w_{P^*(v^*,v_l)} = -w(\{v^*, u^*\}) + w_{P^*(u^*,v_l)} < w_{P'(v^*,v')}.$$

Then,

$$w_{P^*(u^*,v_l)} < w(\{u^*, v^*\}) + w_{P'(v^*,v')} = w_{P'(u^*,v')}.$$

By adding $w_{P'(u',u^*)}$ to both sides, we obtain

$$w_{P'(u',u^*)\cdot P^*(u^*,v_l)} = w_{P'(u',u^*)} + w_{P^*(u^*,v_l)} < w_{P'(u',u^*)} + w_{P'(u^*,v')} = w_{P'}.$$

The weight increment of $P'(u', u^*) \cdot P^*(u^*, v_l)$ is less than that of P' by the above inequality. Note that $P'(u', u^*) \cdot P^*(u^*, v_l)$ is also a candidate of an augmenting path because $\deg_{M'}(v_l) = \deg_{M'}(v')$. This contradicts the condition of Step 3 of *WSM*; $w_{P'}$ is minimum.

(1-2) Case of $v_1 = v_l$. The following inequalities hold by $w_{P^*} < 0$, that is,

$$w_{P^*(v_1,v^*)} - w(\{v^*, u^*\}) + w_{P^*(u^*,v_l)} < 0.$$

By $v_1 = v_l$,

$$w_{P^*(u^*,v_l)\cdot P^*(v_1,v^*)} < w(\{v^*, u^*\}).$$

By adding $w_{P'(u',u^*)}$ and $w_{P'(v^*,v')}$ to both sides,

$$w_{P'(u',u^*)\cdot P^*(u^*,v_l)\cdot P^*(v_1,v^*)\cdot P'(v^*,v')} < w_{P'}.$$

This inequality also contradicts the condition of P' as above. Thus, weight-reducing path P^* with $|Q| = 1$ does not exist by (1-1) and (1-2).

(2) Assuming that P^* is not created with any $|Q| \leq k-1$, we show that P^* is also not created in the case of $|Q| = k$. If $\deg_{M'}(v') \geq d$ then the contradiction also arises as in case (1). Thus we consider the case of $\deg_{M'}(v') = d-1$.

(2-1) Case of $v_1 \neq v_l$. Let $\{v_1^*, u_1^*\}, \{v_2^*, u_2^*\}, \ldots, \{v_k^*, u_k^*\}$ be edges in Q, where they appear in P^* in this order, i.e., $P^* = (v_1, \ldots, v_1^*, u_1^*, \ldots, v_k^*, u_k^*, \ldots, v_l)$. We consider the following two cases (a) and (b), according to the traced order of edges in P'. We show examples of the case of $|Q| = 3$ in **Fig. 6**.

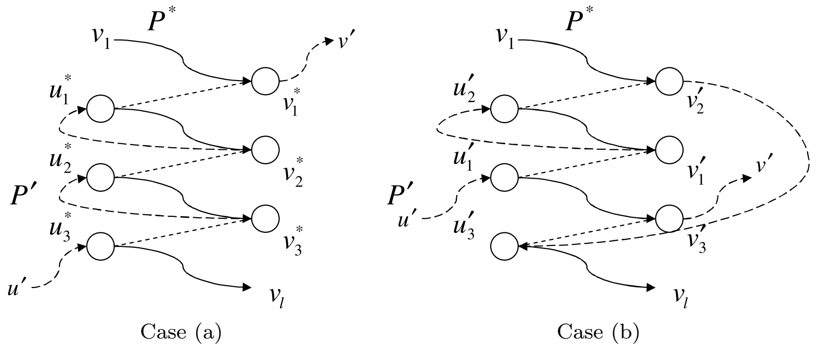


Fig. 6 Case (2-1) ($|Q| = 3, v_1 \neq v_l$).

(a) Consider the case that P' traces edges of Q in the order of $\{u_k^*, v_k^*\}, \ldots, \{u_1^*, v_1^*\}$, that is, $P' = (u', \ldots, u_k^*, v_k^*, \ldots, u_1^*, v_1^*, \ldots, v')$. In this case, a cycle $P^*(u_i^*, v_{i+1}^*) \cdot P'(v_{i+1}^*, u_i^*)$ exists for each $i = 1, 2, \ldots, k-1$. Because these cycles are not weight-reducing cycles, we have $w_{P^*(u_i^*, v_{i+1}^*) \cdot P'(v_{i+1}^*, u_i^*)} \geq 0$, that is,

$$w_{P^*(u_i^*, v_{i+1}^*)} \geq -w_{P'(v_{i+1}^*, u_i^*)}, \qquad (3)$$

for each $i = 1, 2, \ldots, k-1$. Since an edge $\{v_1^*, u_1^*\} \in Q$ is on P', $P^*(v_1, v_1^*) \cdot P'(v_1^*, v')$ exists for M', and it is also not a weight-reducing path. Thus, $w_{P^*(v_1, v_1^*) \cdot P'(v_1^*, v')} \geq 0$, that is,

$$w_{P^*(v_1, v_1^*)} \geq -w_{P'(v_1^*, v')}. \qquad (4)$$

The inequality $w_{P^*} < 0$ is decomposed as

$$w_{P^*(v_1, v_1^*)} + \sum_{i=1}^{k-1} \{w(\{v_i^*, u_i^*\}) + w_{P^*(u_i^*, v_{i+1}^*)}\} + w_{P^*(v_k^*, v_l)} < 0.$$

Moreover by using (3) and (4),

$$-w_{P'(v_1^*, v')} + \sum_{i=1}^{k-1} \{w(\{v_i^*, u_i^*\}) - w_{P'(v_{i+1}^*, u_i^*)}\} - w(\{v_k^*, u_k^*\}) + w_{P^*(u_k^*, v_l)} < 0.$$

Then,

$$w_{P^*(u_k^*, v_l)} < w(\{u_k^*, v_k^*\}) + \sum_{i=1}^{k-1} \{-w(\{u_i^*, v_i^*\}) + w_{P'(v_{i+1}^*, u_i^*)}\} + w_{P'(v_1^*, v')} = w_{P'(u_k^*, v')}.$$

By adding $w_{P'(u', u_k^*)}$, we have

$$w_{P'(u', u_k^*) \cdot P^*(u_k^*, v_l)} < w_{P'}.$$

This inequality contradicts the condition of the augmenting path P'.

(b) In the cases except for (a), P' contains a subpath $P'(u_i^*, v_{i+1}^*)$ for some i. Let p be the number of edges included in Q on $P'(u_i^*, v_{i+1}^*)$, and $\{u_1', v_1'\}, \{u_2', v_2'\}, \ldots, \{u_p', v_p'\}$ be these edges that $P'(u_i^*, v_{i+1}^*)$ traces in order, i.e., $P'(u_i^*, v_{i+1}^*) = P(u_1', v_p') = (u_1', v_1', \ldots, u_p', v_p')$.

For a subpath $P'(v_j', u_{j+1}')$ of each j, a cycle $P'(v_j', u_{j+1}') \cdot P^*(u_{j+1}', v_j')$ exists for M' if u_{j+1}' appears earlier than v_j' on P^*. The number of edges of Q included on this cycle is at most $k-2$, because $\{u_1^*, v_1^*\}$ and $\{u_k^*, v_k^*\}$ cannot be contained. This conduces $w_{P'(v_j', u_{j+1}') \cdot P^*(u_{j+1}', v_j')} \geq 0$ by the assumption of the induction; weight-reducing cycles are not created. Therefore, $-w_{P^*(u_{j+1}', v_j')} \leq w_{P'(v_j', u_{j+1}')}$ holds. By adding $w(\{u_j', v_j'\})$,

$$-\{w_{P^*(u_{j+1}', v_j')} - w(\{v_j', u_j'\})\} \leq w(\{u_j', v_j'\}) + w_{P'(v_j', u_{j+1}')}.$$

Then,

$$-w_{P^*(u_{j+1}', u_j')} \leq w_{P'(u_j', u_{j+1}')} \qquad (5)$$

holds. In the other case, that is, v_j' appears earlier than u_{j+1}' on P^* by contraries, a path $P^*(v_1, v_j') \cdot P'(v_j', u_{j+1}') \cdot P^*(u_{j+1}', v_l)$ exists for M'. Also the number of edges of Q on this path is at most $k-2$, so it is not a weight-reducing path. Thus, $w_{P^*(v_1, v_j') \cdot P'(v_j', u_{j+1}') \cdot P^*(u_{j+1}', v_l)} \geq 0$, i.e.,

$$-w_{P^*(v_1, v_j')} - w_{P^*(u_{j+1}', v_l)} \leq w_{P'(v_j', u_{j+1}')} \qquad (6)$$

holds. And by

$$w_{P^*} = w_{P^*(v_1, v_j') \cdot P^*(v_j', u_{j+1}') \cdot P^*(u_{j+1}', v_l)} < 0,$$

$$w_{P^*(v_j', u_{j+1}')} < -w_{P^*(v_1, v_j')} - w_{P^*(u_{j+1}', v_l)} \qquad (7)$$

By combining Eqs. (6) and (7), we obtain

$$w_{P^*(u_j', u_{j+1}')} < w_{P'(u_j', u_{j+1}')}. \qquad (8)$$

Either Eqs. (5) or (8) is satisfied in each subpath $P'(u_j', v_{j+1}')$ for $1 \leq j < p$, and at least one j is of the latter case. By summing these equations up,

$$w_{P^*(u'_1,u'_p)} < \sum_{j=1}^{p-1} w_{P'(u'_j,u'_{j+1})} = w_{P'(u'_1,u'_p)}.$$

By adding $w(\{u'_p, v'_p\})$, $w_{P'(u',u'_1)}$ and $w_{P'(v'_p,v')}$,

$$w_{P'(u',u'_1)\cdot P^*(u'_1,v'_p)\cdot P'(v'_p,v')} < w_{P'}.$$

This inequality contradicts the condition of an augmenting path P', in consequence P^* is not created when $v_1 \neq v_l$.

(2-2) Case of $v_1 = v_l$. As in the case of $v_1 \neq v_l$, let $P^* = (v_1, \ldots, v_1^*, u_1^*, \ldots, v_k^*, u_k^*, \ldots, v_l)$. We then mention that a path $P'(u_i^*, u_{i+1}^*)$ with respect to M' exists for some i because P^* is a cycle. Therefore the proof of case (b) of (2-1) has already proved this case.

These show P^* of $|Q| = k$ is also not created.

By the induction of (1) and (2), a weight-reducing path/cycle P^* does not exist in M^* for all values of $|Q|$, which shows no weight-reducing path/cycle exists in the final semi-matching produced by *WSM*. □

The above Lemma 8 guarantees the correctness of our algorithm. We next discuss the time complexity.

Lemma 9 *An augmenting path P found in Step 3 of WSM is a simple path.*

Proof By Lemma 8, no weight-reducing path exists while *WSM* executes, which implies all existing cycles have nonnegative weight increments. Namely, any augmenting path P contains no cycle; P is a simple path. □

Lemma 10 *Suppose we execute Step 3 of WSM. For $u \in U$, let $P_1 = (u, \ldots, v_1)$, $P_2 = (u, \ldots, v_2)$ as two different paths having a common end vertex, i.e., $v_1 = v_2$. If $w_{P_1} < w_{P_2}$, then P_2 is not a subpath of any augmenting path in the algorithm.*

Proof Let T be the alternating search tree rooted u in arbitrary iteration of the algorithm. Let T_1 and T_2 be subtrees of T rooted v_1 and v_2 respectively, and let V_1 and V_2 be the sets of vertices belonging to T_1 and T_2, respectively. We show the contradiction by assuming that an augmenting path P contains P_2, i.e., $P = P_2 \cdot P' = (u, \ldots, v_2, \ldots, v)$, where P' is the path existing in T_2.

(1) First let us consider the case that P' is also in T_1. In this case, $P_1 \cdot P'$ exists in T. Thus, $w_{P_1 \cdot P'} < w_{P_2 \cdot P'} = w_P$ holds by $w_{P_1} < w_{P_2}$. This contradicts the condition of the augmenting path P; w_P is minimum.

(2) In the other case that P' does not exist in T_1. **Figure 7** shows an example of search tree T. Since $v_1 = v_2$, we can trace a subpath of P' from v_1 in T_1, but it ends at some

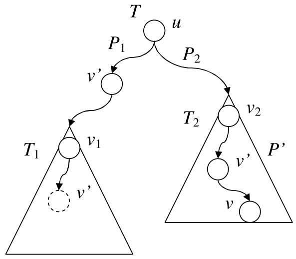


Fig. 7 Search tree T (Lemma 10(2)).

$u' \in U$ because P' is not in T_1. (The subpath ends at a vertex in U but not in V, because if we can reach some vertices in V then we can reach a unique vertex of U by following the matching edge.) Let v' be the node next to u' in P'. This v' exists in P_1, otherwise we can extend the above subpath from v_1. In this situation, a cycle $P_1(v', v_1) \cdot P'(v_2, v')$ exists in the graph because $v_1 = v_2$. Since it is not a weight-reducing cycle by Lemma 8, we have $w_{P_1(v',v_1)\cdot P'(v_2,v')} \geq 0$, i.e., $-w_{P_1(v',v_1)} \leq w_{P'(v_2,v')}$. This and $w_{P_1} < w_{P_2}$ yields $w_{P_1} - w_{P_1(v',v_1)} < w_{P_2} + w_{P'(v_2,v')}$, that is $w_{P_1(u,v')} < w_{P_2\cdot P'(v_2,v')}$. By adding $w_{P'(v',v)}$ to both sides, $w_{P_1(u,v')\cdot P'(v',v)} < w_{P_2\cdot P'(v_2,v)} = w_P$ holds. This contradicts the condition of the augmenting path P.

By cases (1) and (2), P_2 is not a subgraph of any augmenting path when $w_{P_1} < w_{P_2}$. □

Theorem 11 *WSM finds a min-weight balanced semi-matching M in $O(mn_1n_2)$ time.*

Proof It is clear that a semi-matching M obtained by *WSM* is a min-weight balanced semi-matching from Theorem 5 and Lemma 8. We give the time complexity of *WSM*. By making use of the property of Lemma 10, we cut off the search from some vertices. If the breadth first search is done, the number of vertices in V existing in arbitrary odd depth k is bounded by at most $n_2(= |V|)$. This is because a candidate of augmenting paths containing v is guaranteed to be unique by Lemma 10; the one whose weight increment is minimum. On the other hand, the number of U vertices in even depth $k+1$ is at most $n_1(= |U|)$ by the definition of a semi-matching, and at most $m(= |E|)$ edges exist from depth k to $k+2$. Considering that the depth of a search tree is at most $2n_2 - 1$ by Lemma 9, building a search tree requires at most $O(n_2 + m \cdot (2n_2 - 2)/2) = O(mn_2)$ time. Same as algorithm $\mathcal{A}_{SM1}$, it has exactly n_1 iterations. Conclusively the total running time required by the algorithm is at most $O(mn_2 \cdot n_1)$

time. □

4. Conclusion

We formulated the minimum weight load-balancing problem for weighted bipartite graphs, and characterized the optimality of weighted semi-matchings by weight-reducing paths/cycles. As an application for the problem, we gave assigning users to APs appropriately in wireless networks. We then proposed an $O(mn_1n_2)$ time algorithm that finds an optimal semi-matching by keeping non-existence property of weight-reducing paths/cycles. As a future work, we expect further improvements of the running time, for example, by utilizing more elaborate data structures.

Acknowledgments This work is supported in part by the Grant-in-Aid of the Ministry of Education, Science, Sports and Culture of Japan.

References

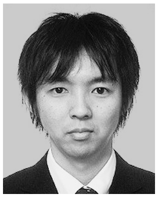
1) Asahiro, Y., Miyano, E., Ono, H. and Zenmyo, K.: Graph Orientation Algorithms to Minimize the Maximum Outdegree, *Proc. Twelfth Computing: The Australasian Theory Symposium* (*CATS2006*), Hobart, Australia. CRPIT, 51, Gudmundsson, J. and Jay, B. (Eds.), ACS, pp.11–20 (2006).
2) Bruno, J.L., Coffman, E.G. and Sethi, R.: Scheduling independent tasks to reduce mean finishing time, *Comm. ACM*, Vol.17, pp.382–387 (1974).
3) Edmonds, J.: Paths, trees, and flowers, *Canadian Journal of Mathematics*, Vol.17, pp.449–467 (1965).
4) Fukuda, Y., Abe, T. and Oie, Y.: Decentralized Access Point Selection Architecture for Wireless LANs, *Proc. Wireless Telecommunications Symposium*, pp.137–145 (2004).
5) Harvey, N.J.A., Ladner, R.E., Lovasz, L. and Tamir, T.: Semi-Matchings for Bipartite Graphs and Load Balancing, *Workshop on Algorithms and Data Structures* (*WADS*), pp.294–308 (2003).
6) Hopcroft, J.E. and Karp, R.M.: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, *SIAM Journal on Computing 2*, pp.225–231 (1973).
7) Kuhn, H.W.: The Hungarian method for the assignment problem, *Naval Res. Logist. Quart. 2*, pp.83–97 (1955).
8) Lenstra, J.K., Shmoys, D.B. and Tardos, É.: Approximation algorithms for scheduling unrelated parallel machines, *Mathematical Programming*, Vol.46, pp.259–271 (1990).
9) Wu, H. and Peng, Y.: Performance of Reliable Transport Protocol over IEEE802.11 Wireless LAN: Analysis and Enhancement, *IEEE INFOCOM*, pp.599–607 (2002).

(Received October 16, 2006)
(Accepted July 3, 2007)
(Online version of this article can be found in the IPSJ Digital Courier, Vol.3, pp.693–702.)

Editor's Recommendation

The paper was selected as a candidate for a recommendation paper; it was ranked one of the bests as the results of 1) the review when it was submitted and 2) the evaluation by the session chair when it was presented. We assigned two reviewers to pre-review the paper and forwarded conditions for us to recommend it to IPSJ. The authors then agreed to revise so that all the comments are incorporated with. We thus recommended this paper.

(Manager of IPSJ Kyushu Branch
Dr. Hirofumi Amano)



Yuta Harada received his B.E. degree from the Department of Electrical Engineering and Computer Science, Kyushu University in 2006. In the same year, he joined the Graduate School of Information Science and Electrical Engineering, Kyushu University.

Hirotaka Ono received his B.E., M.E. and Doctor of Informatics degrees from Kyoto University in 1997, 1999 and 2002 respectively. He is currently an assistant professor of the Department of Computer Science and Communication Engineering of Kyushu University. His research interests include combinatorial optimization, logical analysis of data and distributed algorithms. He is a member of the Information Processing Society of Japan and the Operation Research Society of Japan.

Kunihiko Sadakane received B.S., M.S., and Ph.D. degrees from the Department of Information Science, the University of Tokyo in 1995, 1997 and 2000, respectively. He was a research associate at Graduate School of Information Sciences, Tohoku University in 2000–2003. He has been an associate professor of the Department of Computer Science and Communication Engineering, Kyushu University. His research interests include algorithms and data structures for text compression and text retrieval. He is a member of IPSJ.

Masafumi Yamashita received his B.E. and M.E. degrees from Kyoto University in 1974 and 1977, respectively. He received his Doctor of Engineering degree from Nagoya University in 1981. From 1980 to 1985, he was with the Department of Information and Computer Sciences, Toyohashi University of Technology. In 1985, he joined the Faculty of Engineering at Hiroshima University as an associate professor, and was a professor from 1992 to 1998. He has held visiting appointments several times with Simon Fraser University, B.C., Canada and University of Wisconsin-Milwaukee, WI, USA. His research interests include distributed algorithms/systems, parallel algorithms, and cooperative systems. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan, the Information Processing Society of Japan, SIAM Japan, IEEE, and ACM.