OPTIMUM MATCHINGS IN WEIGHTED BIPARTITE GRAPHS

CARLOS E. VALENCIA AND MARCOS C. VARGAS

Abstract. Given an integer weighted bipartite graph \( G = (U \sqcup V, E), w : E \to \mathbb{Z} \) we consider the problems of finding all the edges that occur in some minimum weight matching of maximum cardinality and enumerating all the minimum weight perfect matchings. Moreover, we construct a subgraph \( G_{cs} \) of \( G \) which depends on an \( \epsilon \)-optimal solution of the dual linear program associated to the assignment problem on \( (G, w) \) that allows us to reduced this problems to their unweighed variants on \( G_{cs} \). For instance, when \( G \) has a perfect matching and we have an \( \epsilon \)-optimal solution of the dual linear program associated to the assignment problem on \( (G, w) \), we solve the problem of finding all the edges that occur in some minimum weight perfect matching in linear time on the number of edges. Therefore, starting from scratch we get an algorithm that solves this problem in time \( O(\sqrt{nm} \log(nW)) \), where \( n = |U| \geq |V|, m = |E|, \) and \( W = \max\{|w(e)| : e \in E\} \).

1. Introduction

Given an integer weighted bipartite graph \( \{G, w\} \), that is, a bipartite graph \( G = (U \sqcup V, E) \) with bipartitions \( U \) and \( V \) and an integer weight function \( w : E \to \mathbb{Z} \) over the edges of \( G \). A matching of \( G \) is a set of edges \( M \) which are vertex disjoint. Moreover, a matching is called perfect if it covers all the vertices of \( G \). A bipartite graph is called feasible if it has at least one perfect matching. The weight of a matching \( M \) is given by

\[
    w(M) = \sum_{uv \in M} w(uv).
\]

A matching of maximum cardinality and minimum weight will be called optimum matching. In this paper we reserve the symbols \( m = |E|, n = |U| \geq |V| \) and \( W \) to denote the maximum absolute weight.

Matchings in graphs is one of the most important topics in combinatorial optimization and has been extensively studied since the early nineteenth century until now. Moreover, it have given origin to several key ideas in combinatorial optimization. For bipartite graphs with no weights there are algorithms for finding maximum cardinality matchings \([11, 12]\), for enumerating all the perfect matchings \([13, 14]\), for finding all the edges that occur in some maximum cardinality matching \([15, 16, 17]\), etc. Finding matchings have many applications in mathematics, computer sciences, operations research, biochemistry, electrical engineering, etc. For weighted bipartite graphs there are several algorithms that solve the assignment problem, which consist of finding a perfect matching of minimum weight, see for instance \([5, 6, 7, 8, 9, 10]\). The reader can consult \([1, 2, 3, 4]\) and the references therein for more information about assignment problems.

As we mentioned before, several authors have worked on the problem of finding all the edges that occur in some perfect matching of a bipartite graph and also on the problem of enumerating all the perfect matchings of a bipartite graph. In this article we solve efficiently the weighted variants of these problems.
problems. More precisely, we give algorithms for finding all the edges that occur in some minimum weight perfect matching and for enumerating all the minimum weight perfect matchings of a weighted bipartite graph \( \{G, w\} \). Moreover, given an optimal solution \( P \) of the dual linear program associated to the assignment problem on \( \{G, w\} \) we construct a subgraph \( G_{cs}(P) \) of \( G \) such that if we can solve a problem involving the perfect matchings of \( G \), then we can use the subgraph \( G_{cs}(P) \) to solve the weighted variant of this problem. For instance, since we can enumerate all the perfect matchings of a bipartite graph, then we can enumerate all the minimum weight perfect matchings of a weighted bipartite graph. The following result, which will be proven later, provides the relation between the optimum perfect matchings of \( \{G, w\} \) and the perfect matchings of \( G_{cs}(P) \).

**Theorem 2.5** If \( P \) is an optimal solution of the dual linear program associated to the assignment problem on \( \{G, w\} \), then the set of minimum weight perfect matchings of \( \{G, w\} \) is equal to the set of perfect matchings of \( G_{cs}(P) \).

The article is organized as follows: In Section 2 we present the linear program that models the assignment problem and its dual. We define what is an optimal solution and an \( \epsilon \)-optimal solution of these linear programs. Also, we construct the subgraph \( G_{cs}(P) \) from an optimal and \( \epsilon \)-optimal solution \( P \) of this dual linear program. In Section 3 we use \( G_{cs}(P) \) to finding all the edges that occur in some minimum weight perfect matching and enumerating all the minimum weight perfect matchings. Also, as an application, we solve the problem of finding a perfect matching where some edges are preferred. Finally, in section 4 we give some strategies to deal with these problems when the bipartite graph has no perfect matchings and we are interested in optimum matchings.

The main result of this paper is based on the following key observation that follows directly from the Complementary Slackness Theorem, which can be found in [3, Theorem 4.5]. Given a linear program in standard form with decision variables \( x = (x_1, \ldots, x_r) \), cost vector \( c = (c_1, \ldots, c_r) \), \( q \times r \) constraint matrix \( A \) with columns \( A_j \), and dual variables \( p = (p_1, \ldots, p_q) \). Then the optimal primal solutions of this linear program are characterized by the following result:

**Corollary 1.1.** If \( p \) is an optimal dual solution, then a feasible primal solution \( x \) is optimal if and only if \( x_j = 0 \) for all \( j \) such that \( (c_j - p \cdot A_j) \neq 0 \).

2. The subgraph \( G_{cs} \) for feasible bipartite graphs

In this section we define the main object of this paper: the subgraph \( G_{cs} \). Before we define it, we present the linear program that models the assignment problem, and its dual. After that, we define what means optimal and \( \epsilon \)-optimal solutions of the linear assignment program and its dual.

The assignment problem can be modeled by the following linear program, which we call the assignment program.

\[
\begin{align*}
\text{minimize:} & \quad \sum_{uv \in E} M(uv)w(uv) \\
\text{subject to:} & \quad \sum_{v \in N(u)} M(uv) = 1 \quad \forall u \in U, \\
& \quad \sum_{u \in N(v)} M(uv) = 1 \quad \forall v \in V, \\
& \quad M(uv) \geq 0 \quad \forall uv \in E,
\end{align*}
\]

(2.1)

where \( M \) represents the incidence vector of the matching, see [3 Section 7.8]. Since the matrix that defines this linear program is the incidence matrix of the bipartite graph \( G \), then it is totally unimodular and therefore we can assume that \( M(uv) \in \{0, 1\} \) for all \( uv \in E \); see [4 chapter 18]. The dual linear
program of 2.1 that we call dual assignment program, is given by:

\[
\text{max: } \sum_{u \in U} \pi(u) + \sum_{v \in V} p(v) \\
\text{subject to: } \pi(u) + p(v) \leq w(uv) \quad \forall uv \in E.
\]

(2.2)

The dual variables \( \pi : U \to \mathbb{R} \) are associated to the constraints \( \sum_{v \in N(u)} M(uv) = 1 \) for each \( u \in U \) and the dual variables \( p : V \to \mathbb{R} \) to the constraints \( \sum_{u \in N(v)} M(uv) = 1 \) for each \( v \in V \). Therefore, the pair \( P = (\pi, p) \), also called dual prices or simply prices, is the set of dual variables of (2.2). Derived from the complementary slackness theorem, we get the following result.

**Proposition 2.1.** Let \( M \) be a perfect matching of \( G \) and \( P = (\pi, p) \) prices of the dual assignment program. Then \( M \) is a minimum weight perfect matching of \( G \), and \( P \) are optimal prices of (2.2) if and only if

\[
\pi(u) + p(v) \leq w(uv) \quad \forall uv \in E,
\]

\[
\pi(u) + p(v) = w(uv) \quad \forall uv \in M.
\]

Now, let \( \epsilon > 0 \), \( M \) be a perfect matching of \( G \), and \( P = (\pi, p) \) be dual prices of (2.2). We say that \( M \) and \( P \) are \( \epsilon \)-optimal solutions of the assignment program if and only if

\[
\pi(u) + p(v) \leq w(uv) + \epsilon \quad \forall uv \in E,
\]

\[
\pi(u) + p(v) = w(uv) \quad \forall uv \in M.
\]

**Proposition 2.2.** Let \( M \) be a perfect matching and \( P = (\pi, p) \) dual prices that are \( \epsilon \)-optimal. If \( M^* \) is a perfect matching of minimum weight of \( G \), then \( w(M) \leq w(M^*) + n\epsilon \).

**Proof.** Directly from the definition of \( \epsilon \)-optimal solutions, we get that

\[
w(M) = \sum_{uv \in M} w(uv) = \sum_{uv \in M} (\pi(u) + p(v)) \leq \sum_{uv \in M^*} (w(uv) + \epsilon) = w(M^*) + n\epsilon. \]

Since we have integral weights, it follows from Proposition 2.2 that if \( M \) and \( P \) are \( \epsilon \)-optimal solutions for \( \epsilon < 1/n \), then \( M \) is of minimum weight.

2.1. The auxiliary subgraph \( G_{cs} \). Given a weighted bipartite graph \( \{G, w\} \) and optimal prices \( P = (\pi, p) \), we define \( G_{cs}(P) \) as follows:

**Definition 2.3.** Let \( G_{cs}(P) = (U \cup V, E_{cs}) \), where \( E_{cs}(P) = \{uv \in E \mid \pi(u) + p(v) = c(uv)\} \)

Note that \( G_{cs}(P) \) is obtained by removing the edges \( uv \) of \( G \) such that \( w(uv) - \pi(u) - p(v) \neq 0 \).

Directly from the definition of \( G_{cs} \) we get the following result:

**Proposition 2.4.** The subgraph \( G_{cs}(P) \) can be constructed in linear \( O(m) \) time.

Now, let \( \mathcal{M}(G, w) \) the set of all the minimum weight perfect matchings of \( \{G, w\} \) and \( \mathcal{M}(G_{cs}(P)) \) the set of all the perfect matchings of \( G_{cs}(P) \). The following theorem is a key result of this paper, and gives a very important relation between this two sets.

**Theorem 2.5.** If \( P = (\pi, p) \) are optimal prices, then \( \mathcal{M}(G, w) = \mathcal{M}(G_{cs}(P)) \).

**Proof.** Since \( P \) are optimal prices, then by corollary 1.1, a perfect matching \( M \) of \( G \) is of minimum weight if and only if \( uv \notin M \) for all edge \( uv \) such that \( w(uv) - \pi(u) - p(v) \neq 0 \), which happens if and only if \( M \) is in \( G_{cs}(P) \). \( \square \)
Note that Theorem 2.3 implies that $M_1(G_{cs}(P))$ is the same for all optimal prices $P$. Regardless of this fact, the following example shows that the subgraph $G_{cs}(P)$ can be different for different optimal prices.

**Example 2.6.** Let $G$ be a bipartite graph with $U = \{u_0, u_1, u_2\}$ and $V = \{v_0, v_1, v_2\}$ as in figure 1.

\begin{figure}[h]
\centering
\begin{subfigure}{0.3\linewidth}
\centering
\begin{tikzpicture}
\node (u0) at (0,0) [circle,draw] {$u_0$};
\node (u1) at (1,0) [circle,draw] {$u_1$};
\node (u2) at (2,0) [circle,draw] {$u_2$};
\node (v0) at (2,1) [circle,draw] {$v_0$};
\node (v1) at (2,-1) [circle,draw] {$v_1$};
\node (v2) at (2,0) [circle,draw] {$v_2$};
\draw (u0) -- (v0) node[midway, above] {$1$};
\draw (u0) -- (v1) node[midway, above] {$1$};
\draw (u0) -- (v2) node[midway, above] {$2$};
\draw (u1) -- (v1) node[midway, above] {$1$};
\draw (u2) -- (v2) node[midway, above] {$2$};
\end{tikzpicture}
\caption{$G$}
\end{subfigure}
\begin{subfigure}{0.3\linewidth}
\centering
\begin{tikzpicture}
\node (u0) at (0,0) [circle,draw] {$u_0$};
\node (v0) at (1,0) [circle,draw] {$v_0$};
\node (u1) at (0,1) [circle,draw] {$u_1$};
\node (v1) at (1,1) [circle,draw] {$v_1$};
\node (u2) at (0,0) [circle,draw] {$u_2$};
\node (v2) at (1,0) [circle,draw] {$v_2$};
\draw (u0) -- (v0) node[midway, above] {$1$};
\draw (u0) -- (v1) node[midway, above] {$3$};
\draw (u0) -- (v2) node[midway, above] {$1$};
\draw (u1) -- (v1) node[midway, above] {$2$};
\draw (u2) -- (v2) node[midway, above] {$1$};
\end{tikzpicture}
\caption{$G_{cs}(P_1)$}
\end{subfigure}
\begin{subfigure}{0.3\linewidth}
\centering
\begin{tikzpicture}
\node (u0) at (0,0) [circle,draw] {$u_0$};
\node (v0) at (1,0) [circle,draw] {$v_0$};
\node (u1) at (0,1) [circle,draw] {$u_1$};
\node (v1) at (1,1) [circle,draw] {$v_1$};
\node (u2) at (0,0) [circle,draw] {$u_2$};
\node (v2) at (1,0) [circle,draw] {$v_2$};
\draw (u0) -- (v0) node[midway, above] {$1$};
\draw (u0) -- (v1) node[midway, above] {$3$};
\draw (u0) -- (v2) node[midway, above] {$1$};
\draw (u1) -- (v1) node[midway, above] {$2$};
\draw (u2) -- (v2) node[midway, above] {$1$};
\end{tikzpicture}
\caption{$G_{cs}(P_2)$}
\end{subfigure}
\caption{(a) An integer weighted bipartite graph $\{G, w\}$. (b) Optimal prices $P_1$ and the subgraph $G_{cs}(P_1)$. (c) Optimal prices $P_2$ and the subgraph $G_{cs}(P_2)$.}
\end{figure}

If $P_1 = (\pi_1, p_1)$, where $\pi_1$ and $p_1$ are given by the maps $\{u_0 \mapsto -2, u_1 \mapsto 0, u_2 \mapsto 1\}$ and $\{v_0 \mapsto 3, v_1 \mapsto 1, v_2 \mapsto 0\}$ as in figure 1(b), then it is not difficult to check that

$$E_{cs}(P_1) = \{u_0v_0, u_1v_1, u_2v_1, u_2v_2\}.$$  

Also, if $P_2 = (\pi_2, p_2)$ is given as in figure 1(c), then it is not difficult to check that

$$E_{cs}(P_2) = \{u_0v_0, u_0v_1, u_1v_1, u_2v_1, u_2v_2\}$$

and therefore $G_{cs}(P_1) \neq G_{cs}(P_2)$.

It is important to remark that corollary 1.1 can be applied to any linear program in standard form to classify all its optimum primal solutions using one optimal dual solution. For instance, if we consider the transportation problem on a bipartite graph $T$ with supplies, demands, capacities, and per-unit costs. Using corollary 1.1 and one optimal dual solution, we can construct a subgraph $T_{cs}$ of $T$, such that the set of all the optimal flows of the original instance is equal to the set of all the feasible flows on the subgraph $T_{cs}$ with the same supplies, demands, and capacities. In this case there also are algorithms that find optimal dual solutions of the transportation problem. But in general, the difficult part is to find optimal dual solutions of a linear program in standard form.

### 2.2. Constructing $G_{cs}$ from $\epsilon$-optimal prices.

Several of the algorithms that solve the assignment problem, especially those based on cost scaling techniques, solve it by finding $\epsilon$-optimal solutions for a small enough $\epsilon > 0$, that guarantees the optimality of the primal solution as Proposition 2.2 implies. See for instance [8, 9, 10].

Since the subgraph $G_{cs}$ is constructed using optimal prices, in this section we provide a procedure to transform $\epsilon$-optimal solutions $M$ and $P = (\pi, p)$ with $\epsilon \leq 1/(n + 1)$ into optimal solutions $M$ and $P = (\pi, p)$ in linear $O(n)$ time. Note that the matching remains the same since it is already optimum (Proposition 2.2). This algorithm finds a value $t \in \{0, \ldots, n\}$ such that if we define

$$p(v) = [p_\epsilon(v) + t/(n + 1)] \text{ for all } v \in V \text{ and } \pi(uv) = w(uv) - p(v) \text{ for all } uv \in M,$$

then $P = (\pi, p)$ is optimal. As we will prove later, it turns out that $t$ is a good value if

$$t \neq \lceil (n + 1)([p_\epsilon(v)] - p_\epsilon(v)) \rceil \mod (n + 1) \quad \text{for all } v \in V.$$

Since $|V| = n$ and there are $n + 1$ possibilities for $t$, then there exists at least one good value for $t$. Assuming that $\epsilon \leq 1/(n + 1)$, the following algorithm shows the complete procedure.
Algorithm 2.7.
Input: $\epsilon$-optimal solutions $M$ and $P_\epsilon = (\pi_\epsilon, p_\epsilon)$.
Output: Optimal prices $P = (\pi, p)$.

```
1 Procedure get_optimal(M, P_\epsilon)
2      good(j) = true for all $j \in \{0, \ldots, n\}$;
3      good([((n+1)(p_\epsilon(v)) - p_\epsilon(v))] mod (n+1)) = false for all $v \in V$;
4      get $t$ such that good(t) = true;
5      $p(v) = [p_\epsilon(v) + t/(n+1)]$ for all $v \in V$;
6      $\pi(u) = w(uv) - p(v)$ for all $uv \in M$;
7      return $P = (\pi, p)$;
8 end
```

It is not difficult to see that all the parts in the procedure get_optimal runs in $O(n)$ time. This gives us an overall $O(n)$ time. Now, we prove that the procedure returns indeed optimal prices.

**Proposition 2.8.** Let $M$ and $P_\epsilon = (\pi_\epsilon, p_\epsilon)$ be $\epsilon$-optimal solutions with $\epsilon \leq 1/(n+1)$, then the procedure get_optimal will return optimal prices $P = (\pi, p)$.

**Proof.** First, we will prove that if $uv \in M$, then $w(uv) - p(v) \leq w(uz) - p(z)$ for all $z \in N(u)$. If $z = v$, then the inequality is clear. Thus we can assume that $z \in N(u) \setminus v$. Since $M$ and $P_\epsilon$ are $\epsilon$-optimal, then $w(uv) - p_\epsilon(v) \leq w(vz) - p_\epsilon(z) + \epsilon$. Thus,

$$
w(uv) - p(v) = w(uv) - [p_\epsilon(v) + t/(n+1)] = w(uv) + [-p_\epsilon(w) - t/(n+1)] \leq w(uz) - p_\epsilon(z) + \epsilon - t/(n+1)
$$

$$
\leq [w(uz) - p_\epsilon(z) + t/(n+1)] = w(uz) + [-p_\epsilon(z) - (t - 1)/(n+1)]
$$

In order to follow with the proof we need the following technical result:

**Claim 2.9.** Let $r \in \mathbb{R}$, $n \in \mathbb{Z}$ and $t \in \{0, \ldots, n\}$. If $t \neq \lfloor (n+1)((r - r) mod (n+1)) \rfloor$, then $[r + (t - 1)/(n+1)] = [r + t/(n+1)]$.

**Proof.** Let $R = [r]$ and $P_i = r + i/(n+1)$ for all $i \in \{0, \ldots, n+1\}$. Is not difficult to see that $R \in (P_{i-1}, P_i)$ if and only if $i = \lfloor (n+1)(R - r) \rfloor mod (n+1)$ (see Figure 2). Thus, if $t \neq \lfloor (n+1)(R - r) \rfloor mod (n+1)$, then $[P_{i-1}] = [P_i] = R - 1$ or $[P_{i-1}] = [P_i] = R$. That is, $[P_{i-1}] = [P_i]$. \hfill $\Box$

![Figure 2.](image)

The authors thank Lyle Ramshaw for having pointed out this identity.
Now, let \( t \neq \left[ (n + 1) \left(\lceil \frac{1}{2} p(z) \rceil - p(z) \right) \right] \mod (n + 1) \) for all \( z \in V \). Then, by Claim 2.9
\[
w(uv) - p(v) \leq w(uz) - \left[ p(z) + (t - 1)/(n + 1) \right] w(uz) - \left[ p(z) + t/(n + 1) \right] = w(uz) - p(z).
\]
Moreover, if we define \( \pi(u) = w(uv) - p(v) \) for each \( uv \in M \), then \( \pi(u) + p(z) \leq w(uz) \) for all \( z \in N(u) \), with equality when \( z = v \). Therefore, from Proposition 2.1 we get that \( M \) and \( P \) are optimal solutions.

### 3. Some Applications of the Subgraph \( G_{cs} \)

In this section we apply the results of section 2 to solve the weighted variants of some problems involving unweighted perfect matchings in bipartite graphs. First, we solve the problem of finding all the edges that occur in some minimum weight perfect matching. Second we solve the problem of enumerating all the minimum weight perfect matchings of a weighted bipartite graph. The unweighted version of this two problems are addressed in [15] and [13] respectively. Finally, we show how we can use the subgraph \( G_{cs} \) to solve efficiently an interesting problem which we call the reallocation problem. This problem consists of finding a minimum weight perfect matching which contains the maximum number of edges of a prescribed subset of edges.

#### 3.1. Finding all the edges that occur in some minimum weight perfect matching.

Given a weighted bipartite graph \( \{G, w\} \) we want to find the subgraph \( G_{opt} = (U \sqcup V, E_{opt}) \) on the same vertex than \( G \) and edge set given by:

\[
E_{opt} = \{ e \in M : M \text{ is a minimum weight perfect matching} \}.
\]

Note that \( E_{opt} \) is equal to the union of all the minimum weight perfect matchings. The unweighted version has been addressed by several authors, see for instance [15] [16] [17].

Given optimal prices \( P = (\pi, p) \), Theorem 2.5 gives us a way to compute the subgraph \( G_{opt} \). First, we construct \( G_{cs}(P) \) and then we obtain \( E_{opt} \) by finding all the edges that belong to at least one perfect matching of \( G_{cs}(P) \). This can be done using the algorithm given in [15]. The following proposition summarizes one of the main results of [15].

**Proposition 3.1.** [15] Given a feasible bipartite graph \( G = (U \sqcup V, E) \). All the edges that occur in some perfect matching of \( G \) can be found in \( O(m) \) time.

Assuming that the bipartite graph is feasible, the following algorithm finds its subgraph \( G_{opt} \).

**Algorithm 3.2.**

Input: An integer weighted bipartite graph \( \{G, w\} \) and optimal dual prices \( P \).
Output: The subgraph \( G_{opt} \) of \( G \).

1. Procedure \( \text{get}_G_{opt}(G, w, P) \)
2. Construct the subgraph \( G_{cs}(P) = (U \sqcup V, E_{cs}) \);
3. Using [15] Algorithm 2 get \( E_{opt} = \{ e \in M \mid M \text{ is a perfect matching of } G_{cs}(P) \} \);
4. Return \( G_{opt} = (U \sqcup V, E_{opt}) \);
5. end

**Proposition 3.3.** Given a weighted bipartite graph \( \{G, w\} \), the procedure \( \text{get}_G_{opt} \) returns \( G_{opt} \).

Proof. Note that that the edge set returned by the procedure \( \text{get}_G_{opt} \) is equal to \( \bigcup \{ M : M \in \mathcal{M}(G_{cs}(P)) \} \). But, by Theorem 2.5 \( \mathcal{M}(G_{cs}(P)) = \mathcal{M}(G, w) \). Then procedure \( \text{get}_G_{opt} \) returns the edge set \( \bigcup \{ M : M \in \mathcal{M}(G, w) \} \), which is equivalent to \( E_{opt} \). \( \square \)
Theorem 3.4. Given a weighted bipartite graph \( \{G, w\} \) and optimal dual prices \( P \), all the edges of \( G \) that occur in some minimum weight perfect matching can be found in \( O(m) \) time.

Proof. By Propositions 2.3 and 3.1 \( G_{cs}(P) \) can be obtained in \( O(m) \) time and \( E_{opt} \) can be constructed in \( O(|E_{cs}(P)|) \) time. Therefore algorithm 3.2 works on \( O(m) \) overall time. \( \square \)

3.2. Enumerating all the minimum weight perfect matchings. In this section we give an algorithm to enumerate all the minimum weight perfect matchings of \( \{G, w\} \). Given \( G_{cs}(P) \) for some optimal dual prices \( P \) we can enumerate all the perfect matchings of \( G_{cs}(P) \) using some of the algorithms contained in [13, 14]. For instance, in [13] we find the following result:

Theorem 3.5. [13, Theorem 1]. Perfect matchings in a bipartite graph \( G = (V, E) \) can be enumerated in \( O(|E|\sqrt{|V|}) \) preprocessing time and \( O(\log |V|) \) time per perfect matching.

Assuming that the bipartite graph is feasible, the following algorithm describes how to enumerate all the minimum weight perfect matchings.

Algorithm 3.6.

Input: An integer weighted bipartite graph \( \{G, w\} \) and optimal dual prices \( P \).

Output: The enumeration of all the minimum weight perfect matchings of \( \{G, w\} \).

Procedure enumerate\_MW\_Per\_Mat\((G, w, P)\)

1. Construct the subgraph \( G_{cs}(P) = (U \sqcup V, E_{cs}) \);
2. Using [13] Algorithm in page 369 enumerate all the perfect matchings of \( G_{cs}(P) \);
3. return;
4. end

Proposition 3.7. The procedure enumerate\_MW\_Per\_Mat enumerates all the minimum weight perfect matchings of a weighted bipartite graph \( \{G, w\} \).

Proof. It follows because by Theorem 2.5 \( M(G_{cs}(P)) = M(G, w) \). \( \square \)

Theorem 3.8. Given a weighted bipartite graph \( \{G, w\} \) and optimal dual prices \( P \), all its minimum weight perfect matchings can be enumerated in \( O(m + |E_{cs}(P)|\sqrt{n} + |M(G, w)|\log n) \) time.

Proof. By Proposition 2.4 the subgraph \( G_{cs}(P) \) can be obtained in \( O(m) \) time. Also, by Theorem 3.5 we can enumerate all the perfect matchings of \( G_{cs}(P) \) in \( O(|E_{cs}(P)|\sqrt{n} + M(G, w)\log n) \) time. Therefore, Algorithm 3.6 runs in \( O(m + |E_{cs}(P)|\sqrt{n} + |M(G, w)|\log n) \) time. \( \square \)

Note that \( G_{cs}(P) \) can have a lot less edges than \( G \) and therefore \( G \) can have a very small number of minimum weight perfect matchings in such a way that \( O(m + |E_{cs}(P)|\sqrt{n} + M(G, w)\log n) = O(m) \).

3.3. The preallocation problem. The preallocation problem can be stated as follows: Given a weighted bipartite graph \( \{G, w\} \) and a subset of edges \( E_p \subseteq E(G) \), we want to find a minimum weight perfect matching \( M \) of \( \{G, w\} \) such that \( |M \cap E_p| \) is maximum. In other words, there is no other minimum weight perfect matching of \( \{G, w\} \) that has more edges of \( E_p \) than \( M \). We can see the set \( E_p \) as a set of preferences. Note that there can be several minimum weight perfect matchings that respect a maximum number of preferences.

This problem can be easily solved with the help of the subgraph \( G_{cs} \). Given the subgraph \( G_{cs}(P) \) for some optimal dual prices \( P \), we define a weight function \( w_p : E(G_{cs}(P)) \to \{0, 1\} \) over the edges of
\( G_{cs}(P) \), as follows:
\[
wp(e) = \begin{cases} 
0 & \text{if } e \in E_p, \\
1 & \text{if } e \notin E_p.
\end{cases}
\]
The algorithm can be derived from the following proposition.

**Proposition 3.9.** If \( M \) is a minimum weight perfect matching of \( \{ G_{cs}(P), wp \} \), then \( M \) is a minimum weight perfect matching of \( \{ G, w \} \) such that \( |M \cap E_p| \) is maximum.

**Proof.** Since \( P \) are optimal dual prices, then from Theorem 2.5 follows that \( M \) is a minimum weight perfect matching of \( \{ G, w \} \). Also, since \( M(G_{cs}(P)) = M(G, w) \), then \( M \) maximizes \( |M \cap E_p| \) over the elements of \( M(G, w) \) if and only if it maximizes \( |M \cap E_p| \) over the elements of \( M(G_{cs}(P)) \).

Now, assume that \( M' \) is a perfect matching of \( G_{cs}(P) \) containing more edges of \( E_p \) than \( M \). Since \( M' \) has more edges with zero weight than \( M \), then \( w(M') < w(M) \) on \( \{ G_{cs}(P), wp \} \), a contradiction to the optimality of \( M \). Therefore \( M \) maximizes \( |M \cap E_p| \) over the elements of \( M(G_{cs}(P)) \). \( \square \)

Assuming that the bipartite graph is feasible, the following algorithm shows how to solve the preallocation problem.

**Algorithm 3.10.**
*Input:* An integer weighted bipartite graph \( \{ G, w \} \), optimal dual prices \( P \) and a subset of edges \( E_p \).
*Output:* A minimum weight perfect matching that contains a maximum number of edges of \( E_p \).

1. Procedure \texttt{preallocation}(\( G, w, P, E_p \))
2. Construct the subgraph \( G_{cs}(P) = (U \sqcup V, E_{cs}) \);
3. Construct the weight function \( wp \), as given above;
4. Get a minimum weight perfect matching \( M \) of \( \{ G_{cs}(P), wp \} \);
5. return \( M \);
6. end

**Proposition 3.11.** Algorithm 3.10 returns an optimum matching which maximizes \( |M \cap E_p| \).

**Proof.** It follows directly from Proposition 3.9. \( \square \)

**Theorem 3.12.** Given a weighted bipartite graph \( \{ G, w \} \), optimal dual prices \( P \) and a subset of edges \( E_p \). A minimum weight perfect matching that maximizes \( |M \cap E_p| \), can be found in \( O(\sqrt{n} m \log n) \) time.

**Proof.** By Proposition 2.3 the subgraph \( G_{cs}(P) \) can be obtained in \( O(m) \) time. Also, the weight function \( wp \) can be constructed in \( O(m) \). Finally, by the algorithm given in [9, Section 2.1] we have that the minimum weight perfect matching of \( \{ G_{cs}(P), wp \} \) can be obtained in \( O(\sqrt{n} m \log n) \) time. Giving a \( O(\sqrt{n} m \log n) \) total time. \( \square \)

4. Strategies for unfeasible bipartite graphs

In this section we deal with the case when \( G \) does not have necessarily a perfect matching, and we are interested in minimum weight maximum cardinality matchings. A simple way to compute \( G_{cs} \) in this case, is by constructing a feasible weighted bipartite graph \( \{ G', w' \} \), such that a minimum weight perfect matching of \( \{ G', w' \} \) induces a minimum weight maximum cardinality matching of \( \{ G, w \} \) and vice versa. Then we solve the problem that we are addressing in \( \{ G', w' \} \) and translate the solution obtained to a solution on the original weighted bipartite graph \( \{ G, w \} \).
We will give three graph transformations that satisfy the above conditions. Let \( \{G, w\} \) be an integer weighted bipartite graph with \( U = \{u_1, \ldots, u_n\} \), \( V = \{v_1, \ldots, v_s\} \), \( n \geq s \), \( m = |E| \), and \( W = \max_{e \in E} \{|w(e)|\} \).

4.1. **The first doubling transformation.** Let \( G_d = (U_d \sqcup V_d, E_d) \) with \( U_d = U \sqcup V' \), \( V_d = V \sqcup U' \), and \( E_d = E \cup E_f \cup U_d \cup E_V \), where \( U' = \{u_1', \ldots, u_n'\} \) is a copy of \( U \), \( V' = \{v_1', \ldots, v_s'\} \) is a copy of \( V \), \( E_f = \{v_ju_i' \mid u_i, v_j \in E\} \) is a flipped copy of \( E \), \( E_U = \{u_iu_i' \mid u_i \in U\} \), and \( E_V = \{v_jv_j' \mid v_j \in V\} \), see figure 3. Also, let \( w_d : E' \to \mathbb{Z} \) be given by:

\[
\begin{align*}
\text{for all } e & \in E, \\
\text{for all } e & = v_ju_i' \in E_f, \\
2sW & \text{ for all } e \in E_U \cup E_V.
\end{align*}
\]

Note that \( E_U \cup E_V \) is a perfect matching of \( G_d \). The following result gives us a correspondence between the optimum matchings of \( \{G_d, w_d\} \) and \( \{G, w\} \).

**Proposition 4.1.** Let \( \{G, w\} \) be an integer weighted bipartite graph. If \( N \) is a minimum weight perfect matching of \( \{G_d, w_d\} \), then \( M = N \cap E \) is an optimum matching of \( \{G, w\} \).

**Proof.** Given a matching \( A \) of \( G \), let \( \text{flip}(A) = \{u'v' \mid uv \in A\} \). If \( M_f = N \cap E_f \), then \( M \) and \( M_f \) have the same cardinality because \( M \) and \( \text{flip}(M_f) \) cover the same vertices of \( G \). Now, assume that \( M \) is not an optimum matching, that is, there exist a matching \( M^* \) such that \( |M| < |M^*| \) or \( w(M) > w(M^*) \). Let \( N' = M^* \cup \text{flip}(M^*) \cup I \), where \( I = \{zz' \mid z \in V(G) \text{ is not cover by } M^*\} \). Since we can assume without loss of generalization that \( w(M) = \min(w(M), w(M_f)) \), then

\[
\begin{align*}
w(N) & \geq (n + s - 2|M|)2sW + 2w(M) > (n + s - 2|M^*|)2sW + 2w(M^*) = w(N');
\end{align*}
\]

a contradiction to the fact that \( N \) is of minimum weight.

Let us denote by MWPM the Minimum Weight Perfect Matching problem and by OM the Optimum Matching problem. Now, we will show that both problems are reducible.

**Proposition 4.2.** The MWPM and OM problems are mutually reducible.

**Proof.** First, the reduction from MWPM to OM is trivial, because a perfect matching is of maximum cardinality. On the other hand, the reduction from OM to MWPM is obtained by Proposition 4.1. □
The graph $G_d$ has the disadvantage that the weights of the edges $E_V \cup E_V$ increase with $s$. Therefore it can produce overflow errors for the computer data types on relatively small graphs. In order to avoid this disadvantage we present other two transformations, which also give us a reduction between OM and MWPM. However, this two transformations need to assume that $G$ has a matching covering the small side $V$.

4.2. The second doubling transformation. To the knowledge of the authors, this is a new variant of the previous transformation. Let $G_s = G_d \setminus E_V$ and $w_s : E' \rightarrow \mathbb{Z}$ be given by:

$$w_s(e) = \begin{cases} w(e) & \text{for all } e \in E \\ w(u_i v_j) & \text{for all } e = v'_i u'_i \in E_f \\ k & \text{for all } e \in E_U, \end{cases}$$

where $k \in \mathbb{Z}$ is any constant.

In this case we cannot guarantee that $G_s$ has a perfect matching. However, $G_s$ has a perfect matching if and only if $G$ has a matching that covers all the vertices of the small side $V$. The following result gives us a correspondence between the optimum matchings of $\{G_s, w_s\}$ and $\{G, w\}$.

**Proposition 4.3.** Let $\{G, w\}$ be an integer weighted bipartite graph. If $N$ is a minimum weight perfect matching of $\{G_s, w_s\}$, then $M = N \cap E$ is an optimum matching of $\{G, w\}$ which covers $V$.

**Proof.** First, since $N$ is perfect, $M$ must cover $V$. The optimality of $M$ follows from the fact that the cost of any perfect matching $N$ of $G_s$ is $w(N) = 2w(N \cap E) + (n - s)k$ and $(n - s)k$ is constant. □

This transformation has the advantage that the maximum weight of the edges is not increased. However, when $|U| - |V|$ is small enough ($G$ is almost balanced), we need to solve an instance $G_s$ with the double of edges. In order to avoid this disadvantage we present a last transformation.

4.3. The artificial vertices transformation. The last construction consists in balancing $G$ by adding $|U| - |V|$ “artificial” vertices to $V$. More precisely, let $G_a = (U \sqcup V_a, E_a)$ where $V_a = V \cup \{v_{s+1}, \ldots, v_n\}$, $D = \{u_i v_j \mid u_i \in U \text{ and } v_j \in V_a \setminus V\}$, and $E_a = E \cup D$; see figure 3. Also, let $w_a : E' \rightarrow \mathbb{Z}$ given by

$$w_a(e) = \begin{cases} w(u_i v_j) & \text{if } e \in E \\ k & \text{if } e \in D, \end{cases}$$

where $k \in \mathbb{Z}$ is any constant.

Clearly $G_a$ has a perfect matching if and only if $G$ has a matching that covers $V$.

**Proposition 4.4.** Let $\{G, w\}$ be an integer weighted bipartite graph. If $N$ is a minimum weight perfect matching of $\{G_a, w_a\}$, then $M = N \cap E$ is an optimum matching of $\{G, w\}$ which covers $V$.

**Proof.** Follows by similar arguments of those in Proposition 4.3. □

This transformation has the advantage that the number of edges of $G_a$ is only increased a little when $|U| - |V|$ is sufficiently small. In the counterpart, the density of $G_a$ is increased considerably when $|U| - |V|$ is big.

We can use any of these transformations to solve the problem of finding all the edges that occur in any optimum maximum cardinality matching of a weighted bipartite graph $\{G, w\}$. We only need to use a transformation $\{G', w'\}$ and find all the edges $E'_{opt}$ that occur in any minimum weight perfect matching of the transformation. And the solution to the original instance will be given by $E_{opt} = E'_{opt} \cap E$.

Something similar works for the preallocation problem where we want to maximize the preferences.
over the optimum maximum cardinality matchings. However, this does not work for efficiently solve
the problem of enumerating all the optimum maximum cardinality matchings. Because the same
matching can be enumerated a non constant number of times. For instance, consider a complete
bipartite graph with weights \( w(e) = 1 \) for all \( e \in E \). This is due to the fact that the relation between
the optimum matchings of these transformations and the original instance is not a one-to-one relation.

References

[1] R. Burkard, M. Dell’Amico, and S. Martello, Assignment problems, Society for Industrial and Applied Mathematics
(SIAM), Philadelphia, PA, 2009.
[2] L. Lovász and M.D. Plummer, Matching theory, AMS Chelsea Publishing, Providence, RI, 2009.
[3] D. Bertsimas and J.N. Tsitsiklis, Introduction to linear optimization, Athena Scientific, Belmont, MA, 1997.
[4] A. Schrijver, Combinatorial optimization: Polyhedra and efficiency, Algorithms and Combinatorics 24, Springer-Verlag, Berlin, 2003.
[5] H.W. Kuhn, The hungarian method for the assignment problem, Naval Research Logistics Quart. 2, 83–97, 1955.
[6] H.N. Gabow, Scaling algorithms for network problems, Journal of computer & system sciences 31, 148–168, 1985.
[7] E.A. Dinic and M.A. Kronrod, An algorithm for the solution of the assignment problem, Soviet Math. Dokl. Vol 10, 1324–1326, 1969.
[8] D.P. Bertsekas, Auction algorithms for network flow problems: A tutorial introduction, Computational optimization
and applications 1, 7–66, 1992.
[9] H.N. Gabow and R.E. Tarjan, Faster scaling algorithms for network problems, SIAM J. Computation Vol. 18, No.
5, 1013–1036, 1989.
[10] A.V. Goldberg and R. Kennedy, An efficient cost scaling algorithm for the assignment problem, Math. Programming
71, 153–177, 1995.
[11] J. Hopcroft and R.M. Karp, An \( n^{5/2} \) algorithm for maximum matchings in bipartite graphs, SIAM J. Comput. 2,
225–231, 1973.
[12] H. Halt, N. Blum, K. Mehlhorn and M. Paul, Computing a maximum cardinality matching in a bipartite graph in
\( \text{time } O(n^{1.5} \sqrt{m/\log n}) \), Information Processing Letters 37, 237–240, 1991.
[13] T. Uno, A Fast Algorithm for Enumerating Bipartite Perfect Matchings, Lecture Notes in Computer Science,
Springer Verlag, Vol. 2223, 367–379, 2001.
[14] K. Fukuda and T. Matsui, Finding All the Perfect Matchings in Bipartite Graphs, Appl. Math. Lett, Vol. 7, No. 1,
15–18, 1994.
[15] T. Tassa, Finding All Maximally-Matchable Edges in a Bipartite Graph, Theoretical Computer Science 423, 50–58,
2012.
[16] M.C. Costa, Persistency in maximum cardinality bipartite matchings, Oper. Res. Lett. 15,143–149, 1994.
[17] J.C. Régin, A filtering algorithm for constraints of difference in CSPs, Proceedings of the 12th National Conference
on Artificial Intelligence (AAAI), 362–367, 1994.

Departamento de Matemáticas, Centro de Investigación y de Estudios Avanzados del IPN, Apartado
Postal 14–740, 07000 México City, D.F.

E-mail address, Carlos E. Valencia: cvalencia@math.cinvestav.edu.mx, cvalencia75@gmail.com
E-mail address, Marcos C. Vargas: esocram@gmail.com

1Corresponding author