Hindawi
Journal of Mathematics
Volume 2023, Article ID 7761902, 6 pages
https://doi.org/10.1155/2023/7761902

Research Article

Computing a Many-to-Many Matching with Demands and Capacities between Two Sets Using the Hungarian Algorithm

Fatemeh Rajabi-Alni and Alireza Bagheri

Computer Engineering Department, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

Correspondence should be addressed to Alireza Bagheri; ar_bagheri@aut.ac.ir

Received 21 November 2022; Revised 4 April 2023; Accepted 15 April 2023; Published 25 April 2023

Academic Editor: Ching-Feng Wen

Copyright © 2023 Fatemeh Rajabi-Alni and Alireza Bagheri. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A many-to-many matching (MM) between two sets matches each element of one set to at least one element of the other set. A general case of the MM is the many-to-many matching with demands and capacities (MMDC) satisfying given lower and upper bounds on the number of elements matched to each element. In this article, we give a polynomial-time algorithm for finding a minimum-cost MMDC between two sets using the well-known Hungarian algorithm.

1. Introduction

Let $A = \{a_1, a_2, \ldots, a_s\}$ and $B = \{b_1, b_2, \ldots, b_t\}$ with $|A| + |B| = n$. A many-to-many matching (MM) between two sets A and B maps each element of A to at least one element of B and vice-versa. Let L be an MM between A and B. The cost of L denoted by $c(L)$ is the sum of the costs of all matched pairs $(p, q) \in L$. Given an undirected bipartite graph $G = (S \cup T, E)$, a matching in G is a subset of the edges $M \subseteq E$ such that each vertex $v \in S \cup T$ is incident to at most one edge of M. A perfect matching in G is a matching covering all vertices $v \in S \cup T$. A matched vertex is a vertex that is incident to an edge in M. A vertex that is not matched is a free vertex. Let $W(a, b)$ denote the weight of the edge $(a, b) \in E$. The weight of the matching M is the sum of the weights of all edges e in M, hence

$$W(M) = \sum_{e \in M} W(e). \tag{1}$$

The first polynomial algorithm for computing a minimum-weight perfect matching in G with $|S| = |T| = n$ is the well-known Hungarian algorithm proposed in [1], later implemented faster by the running time $O(n^3)$ in dense graphs [2] and $O(mn \log n)$ in sparse graphs [3], where $m = |E|$. Eiter and Mannila [4] solved the minimum-cost MM problem in $O(n^3)$ time by reducing it to the minimum-weight perfect matching problem in a bipartite graph. A modified Hungarian algorithm which is more complex than the approach given in [4], performing n shortest path searches using Dijkstra's algorithm by advantage of Fibonacci heaps [5], runs in $O(n^2 \log n + nm)$ time [6]. When the weights of the edges $e \in E$ are integers, a scaling algorithm was proposed by Gabow and Tarjan [7] running in $O(\sqrt{n}\, m \log(nW'))$ time, where $W' = \max(W(e))_{e \in E}$. For a more detailed discussion on the matching theory and algorithms, see [8–10]. When A and B are two sets of points in the plane, a modified Hungarian algorithm computes an MM between A and B in $O(n^2 \text{poly}(\log n))$ time due to Bandyapadhyay et al. [11]. Colannino et al. [12] proposed an $O(n \log n)$ time algorithm for the one-dimensional MM (OMM) problem, where A and B are two sets of points lying on the real line.

Without loss of generality, let $|B| \le |A|$. In a two-sided matching (TSM) between two sets A and B, each element of B must be matched to an element of A, and each element of A can be matched to at most one element of B, implying that $|A| - |B|$ elements of A remain free [13]. In a stable TSM problem, each element of one set provides preference information over the elements of the other set based on which the optimal matching is computed [14, 15]. A TSM denoted by M is a stable TSM if for any matched pair $(p, q) \in M$, p and q do not prefer to be matched to $q' \neq q$ rather than q and $p' \neq p$ rather than p, respectively [13]. Two types of stable TSM problems have been considered in the literature, multiattribute stable TSM problems in which each element of one set provides its preference over the elements of the other set using multiple attributes, and stable TSM problems with preference structures such as preference ordinals and relations [14, 15]. Although some special cases of stable TSM problems can be solved in polynomial time, most stable TSM problems are NP-hard and can be solved by optimization models such as linear and nonlinear programming models [13–15].

In a limited-capacity MM (LCMM) between A and B, each element $v \in A \cup B$ can be matched to a limited number of the elements, denoted by $\mathrm{Cap}(v)$. A degree-constrained subgraph (DCS) in a general graph $H = (V, E')$ is a subgraph H' in which for each vertex $v \in V$ with degree $\deg(v)$, we have ($l(v) \leq \deg(v) \leq u(v) l(v)$ and $u(v)$ denote integer bounds). The DCS problem in H was solved in $O(n'^2 \min(m' \log n', n'^2))$ time [16], where $m' = |E'|$ and $n' = |V|$. The minimum-cost LCMM problem can be reduced to finding a minimum-weight DCS in an undirected bipartite graph. A one-dimensional LCMM (OLCMM), an LCMM where A and B lie on the real line, was solved in $O(n^2)$ time [17]. A simple b-matching in H is a subset of the edges $F \subset E'$ in which there is a quota b: $V \longrightarrow Z > 0$ on the vertices $v \in V$ such that $\deg(v) \leq b(v)$ for all $v \in V$. When the edge weights are integers, the optimal simple b-matching problem in the bipartite graph $G' = (A \cup B, E'')$ was solved in the time complexity of $O(W' \sqrt{\beta'} n^2)$ [9], where $W' = \max_{e \in E''}(W(e))$ and $\beta' = \sum_{v \in A \cup B} \mathrm{Cap}(v)$.

The many-to-many matching with demands and capacities (MMDC) is a generalization of the LCMM in which each element $a_i \in A$ is matched to $\alpha_i \leq \deg(a_i) \leq \alpha_i'$ elements of B, and each element $b_j \in B$ is matched to $\beta_j \leq \deg(b_j) \leq \beta_j'$ elements of A. Observe that we can consider the minimum-cost MMDC problem a special case of the optimal DCS problem in a bipartite graph, and solve it using the algorithm proposed in [16]. In this article, we present an algorithm that computes a minimum-cost MMDC between A and B in $O(n^6)$ time using the basic Hungarian algorithm. Also, our algorithm computes an MMDC between two sets of points in the plane in $O(n^4 \mathrm{poly}(\log n^2))$ time using the modified Hungarian algorithm proposed in [11]. In fact, our algorithm imposes upper and lower bounds on the number of elements that can be matched to each element in any version of the Hungarian algorithm (see e.g., [18, 19] for further discussion and references). Note that when $t \ll s$, our algorithm runs in $O(n^3)$ time improving the previous $O(n^4)$ time algorithm. Also, our algorithm runs faster than its worst time complexity in bipartite graphs with low-range edge weights and dense graphs [19]. In Section 2, we present our algorithm and a numerical example for it.

2. Matching Algorithm

Given two sets $A = \{a_1, a_2, \ldots, a_s\}$ and $B = \{b_1, b_2, \ldots, b_t\}$, in this section, we present our algorithm for computing an MMDC between A and B. We construct a complete bipartite graph $G = (S \cup T, E)$ such that by running the Hungarian algorithm on G, the demands and capacity limitations of elements of A and B are satisfied. In the following, we explain how our complete bipartite graph $G = (S \cup T, E)$ is constructed. Note that we assume there exists at least one MMDC between A and B; this can be checked in $O(n^3)$ time [20].

We represent a set of related vertices using a rectangle, an edge with a line, and each vertex with a circle. In a complete connection between two sets, each vertex of one set is connected to all vertices of the other set. We show a complete connection using a line connecting the two corresponding sets. The Hungarian algorithm computes a perfect matching in which each vertex is incident to a unique edge. We aim to find an MMDC between two sets A and B where two or more elements may be mapped to the same element, that is, more than one element may be matched to any of the elements. Therefore, our constructed graph contains multiple copies of each element to simulate this situation. Also, the input of the Hungarian algorithm is a complete bipartite graph $G = (V \cup V', E)$ with $|V| = |V'|$, i.e., both parts of the input bipartite graph have an equal number of vertices. Therefore, we should balance two parts of our constructed bipartite graph before using the Hungarian algorithm.

Let $S \cup T$ be a bipartition of G with $S = (\cup_{i=1}^{s} A_i) \cup (\cup_{i=1}^{s} A_i') \cup (\cup_{j=1}^{t} X_j) \cup (\cup_{j=1}^{t} W_j)$, and $T = (\cup_{i=1}^{s} Bset_i) \cup Y$ (see Figure 1). Let $A_i = \{a_{i1}, \ldots, a_{i\alpha_i}\}$ be the set of α_i copies of the element a_i for $1 \leq i \leq s$. Note that each element $a_i \in A$ has a limited capacity, i.e., it can be matched to at most a given number of elements of B. Thus, each a_i is copied $\alpha' i - \alpha_i$ times to constitute the set $A_i' = \{a_{i1}', \ldots, a_{i(\alpha' i - \alpha_i)}'\}$ for $1 \leq i \leq s$. Each set $B\mathrm{set}_i = \{b_{1i}, \ldots, b_{ti}\}$ is a copy of the set $B = \{b_1, b_2, \ldots, b_t\}$. Assume that all vertices b_{ji} with $1 \leq i \leq s$constitute the set B_j. In fact, each set B_j includes s copies of b_j for $1 \leq j \leq t$. We use the set $W_j = \{w_{j1}, \ldots, w_{j(s-\beta_j')}\}$ to limit the number of elements matched to $b_j \in B$. The set $Y = \{y_1, y_2, \ldots, y_z\}$ compensates our bipartite graph G (Y balances S and T). Also, the sets $X_j = \{x_{j1}, \ldots, x_{j(\beta_j' - \beta_j)}\}$ for $1 \leq j \leq t$ guarantee that the output MMDC is a minimum-cost MMDC.

The vertices of the sets A_i, $B\mathrm{set}_i$, and A_i' for $1 \leq i \leq s$ are called main vertices, since they are copies of elements of $A \cup B$. On the other hand, the vertices of the sets Y, X_j, and W_j for $1 \leq j \leq t$ are called dummy vertices. All edges (p, q) whose both end vertices are main vertices, i.e., $p, q \in A_i \cup A_i' \cup B\mathrm{set}_i$ for $1 \leq i \leq s$, are called main edges.

Each set A_i is completely connected to the set $B\mathrm{set}_i$. This complete connection is shown using a line connecting the two corresponding rectangles of A_i and $B\mathrm{set}_i$. Note that $W(a_{ik}, b_{ji}) = \delta(a_i, b_j)$ for $1 \leq k \leq \alpha_i$, where $\delta(a_i, b_j) \in \mathbb{R}$ is the cost of matching a_i to b_j. Each set A_i guarantees that the element $a_i \in A$ is matched to at least α_i elements of B. Each set A_i' is completely connected to the set $B\mathrm{set}_i$, where $W(a_{ik}', b_{ji})$ is equal to $\delta(a_i, b_j)$ for $1 \leq k \leq (\alpha' i - \alpha_i)$. The sets A_i' guarantee that each element $b_j \in B$ is matched to at least

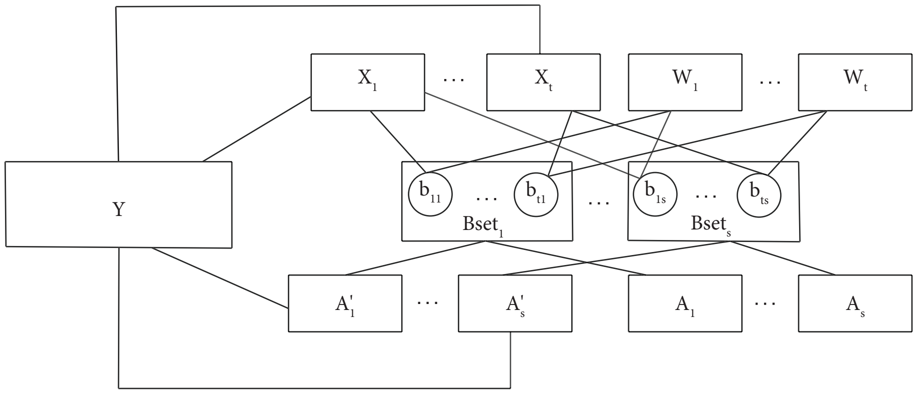

Figure 1: The constructed complete bipartite graph G by our algorithm.

β_j elements of A. Moreover, each set A'_i assures that a_i is matched to at most α'_i elements.

Let $\gamma = \min(\delta(a_i, b_j))_{1\le i\le s, 1\le j\le t}$. Given an arbitrary real value γ' with $\gamma' < \gamma$, there is a complete connection with γ' weighted edges between B_j and X_j for $1 \le j \le t$. Also, there is a zero weighted complete connection between B_j and W_j.

The compensator set $Y = \{y_1, y_2, \ldots, y_z\}$ with $z = |Y| = \sum_{i=1}^{s}\alpha'_i - \sum_{j=1}^{t}\beta_j$ is inserted in G as follows. Note that we assume $\sum_{i=1}^{s}\alpha'_i \ge \sum_{j=1}^{t}\beta_j$ (and $\sum_{j=1}^{t}\beta'_j \ge \sum_{i=1}^{s}\alpha_i$), since otherwise, there would not exist any MMDC between A and B. Let $\eta = \max(\delta(a_i, b_j))_{1\le i\le s, 1\le j\le t}$. Each set A'_i is completely connected to Y with η' weighted edges, where η' is an arbitrary real value with $\eta' > \eta$. Thus, the priority of each set A'_i is the vertices of $B\text{set}_i$. We also have a complete connection between X_j and Y whose edge weights equal a real value γ''. Note that the priority of the vertices of X_j is the vertices of B_j, thus $\gamma'' > \gamma'$.

Notice moreover that, since for each vertex $b_{ji} \in B_j$, the vertices of X_j should have priority over the vertices of A'_i, we have

$$W(b_{ji}, x_{jk}) + W(a'_{il}, y_{k'}) < W(b_{ji}, a'_{il}) + W(x_{jk}, y_{k'}), \qquad (2)$$

for all $1 \le k \le (\beta'_j - \beta_j)$, $1 \le k' \le z$, and $1 \le l \le (\alpha' i - \alpha_i)$. Thus,

$$\gamma' + \eta' < \delta(b_j, a_i) + \gamma'', \qquad (3)$$

for all $1 \le i \le s$ and $1 \le j \le t$. Therefore,

$$\begin{aligned} &\max\left(\gamma' + \eta' - \delta(a_i, b_j)\right)_{1\le i\le s, 1\le j\le t} < \gamma'', \text{and} \\ &\gamma' + \eta' - \min\left(\delta(a_i, b_j)\right)_{1\le i\le s, 1\le j\le t} < \gamma''. \end{aligned} \qquad (4)$$

Therefore, we have

$$\gamma' + \eta' - \gamma < \gamma''. \qquad (5)$$

Observe that $\gamma' + \eta' - \gamma > \gamma'$, since otherwise, we would have $\gamma' + \eta' - \gamma \le \gamma'$, and thus, $\eta' \le \gamma$. This is a contradiction.

Observe that we have

$$\begin{aligned} |S| &= \left|\bigcup_{i=1}^{s} A_i\right| + \left|\bigcup_{i=1}^{s} A'_i\right| + \left|\bigcup_{j=1}^{t} X_j\right| + \left|\bigcup_{j=1}^{t} W_j\right| \\ &= \sum_{i=1}^{s}\alpha'_i + (s * t) - \sum_{j=1}^{t}\beta_j, \text{and} \\ |T| &= \left|\bigcup_{i=1}^{s} B\text{set}_i\right| + |Y| = (s * t) + \sum_{i=1}^{s}\alpha'_i - \sum_{j=1}^{t}\beta_j. \end{aligned} \qquad (6)$$

Note that G must be a complete bipartite graph. Thus, if there does not exist an edge between any two vertices $p \in S$ and $q \in T$, we assume that an infinite weighted edge connects p and q, i.e., $W(p, q) = \infty$.

We claim that from a minimum-weight perfect matching in $G = (S \cup T, E)$ denoted by M, we can get a minimum-cost MMDC between A and B. Let $\text{Main}(M)$ be the set of the main edges of M. Let L denote a minimum-cost MMDC between A and B. In the following, we prove that the weight of $\text{Main}(M)$, i.e., $W(\text{Main}(M))$, is equal to the cost of L, i.e., $c(L)$.

Lemma 1. $W(\text{Main}(M)) \le c(L)$.

Proof. We get from L a perfect matching M' in our complete bipartite graph G such that the weight of $\text{Main}(M')$ equals the cost of L, i.e., $W(\text{Main}(M')) = c(L)$.

Let p_i be the number of the elements $b_j \in B$ with $1 \le j \le t$ matched to $a_i \in A$ in L. It is obvious that $\alpha_i \le p_i \le \alpha'_i$. Firstly, for each pairing (a_i, b_j) with $1 \le j \le t$ in L, we select the edge of G connecting b_{ji} to an arbitrary free vertex $a_{ik} \in A_i$ with $1 \le k \le \alpha_i$, and add to M' until there does not exist any free vertex in A_i. Then, depending on the value of p_i, two cases arise as follows:

(i) either $p_i = \alpha_i$. In this situation, we select the η' weighted edges of G connecting each $a'_{ik} \in A'_i$ to an arbitrary free vertex of Y for $1 \le k \le (\alpha'_i - \alpha_i)$ and add to M'.

(ii) or $p_i > \alpha_i$. In this case, we match $p_i - \alpha_i$ number of vertices of A_i' to the vertices of $B\text{set}_i$ as follows. For each of the remaining $p_i - \alpha_i$ pairings (a_i, b_j) with $1 \le j \le t$ in L that have no equivalent edges in M', we add the edge of G connecting b_{ji} to an arbitrary free vertex a_{ik}' with $1 \le k \le (\alpha_i' - \alpha_i)$ to M'. Then, if there exist free vertices in A_i' (since $p_i < \alpha_i'$), for each of them, we select an edge of G connecting it to an arbitrary free vertex of Y and add to M'.

Then, we add the edges of G connecting each $w_{jk} \in W_j$ to an arbitrary free vertex of B_j for $1 \le k \le (s - \beta_j')$ to M'. Now, the vertices of X_j are matched to the vertices of B_j, unless no free vertices remain in B_j. Thus, we first add the edges connecting the vertices of X_j pairwise to the remaining free vertices of B_j. Then, we add the edges connecting the free vertices of X_j, if exist, to the free vertices of Y pairwise to M'.

Observe that M' is a perfect matching, since each vertex of G is incident with exactly one edge in M'. For each $(a_i, b_j) \in L$, there exists an edge with equal weight in $\text{Main}(M')$, thus $W(\text{Main}(M')) = c(L)$. □

Lemma 2. *Let M be a minimum-weight perfect matching in G. Then, for any perfect matching in G denoted by M'', we have*

$$W(\text{Main}(M)) \le W\left(\text{Main}\left(M''\right)\right). \tag{7}$$

Proof. Observe that we have

$$W(M) = W(\text{Main}(M)) + W(M \backslash \text{Main}(M)). \tag{8}$$

Note that

$$|L| = \max\left(\sum_{i=1}^{s} \alpha_i, \sum_{j=1}^{t} \beta_j\right). \tag{9}$$

Observe that the set $M'' \backslash \text{Main}(M'')$ contains

(i) the zero weighted edges connecting the vertices of W_j to B_j for $1 \le j \le t$, with a total number of $s * t - \sum_{j=1}^{t} \beta_j'$,

(ii) the γ' weighted edges connecting $\sum_{j=1}^{t} \beta_j' - |L|$ number of vertices of X_j to B_j for $1 \le j \le t$,

(iii) the γ'' weighted edges connecting $|L| - \sum_{j=1}^{t} \beta_j$ number of vertices of X_j to Y for $1 \le j \le t$,

(iv) the η' weighted edges connecting $\sum_{i=1}^{s} a_i' - |L|$ number of vertices of A_i' to Y for $1 \le i \le s$.

Thus,

$$W\left(M''/\text{Main}\left(M''\right)\right) = \left(\sum_{j=1}^{t} \beta_j' - |L|\right) * \gamma' + \left(|L| - \sum_{j=1}^{t} \beta_j\right) * \gamma'' + \left(\sum_{i=1}^{s} \alpha_i' - |L|\right) * \eta'. \tag{10}$$

So, we have

$$W(M \backslash \text{Main}(M)) = W\left(M'' \backslash \text{Main}\left(M''\right)\right). \tag{11}$$

Note that M is a minimum-weight perfect matching in G, thus

$$\begin{aligned} W(M) &\le W\left(M''\right), \text{and} \\ W(M \backslash \text{Main}(M)) + W(\text{Main}(M)) &\le W\left(M'' \backslash \text{Main}\left(M''\right)\right) + W\left(\text{Main}\left(M''\right)\right). \end{aligned} \tag{12}$$

Therefore, we have

$$W(\text{Main}(M)) \le W\left(\text{Main}\left(M''\right)\right). \tag{13}$$

From the above lemma, we have $W(\text{Main}(M)) \le W(\text{Main}(M'))$. Notice that $W(\text{Main}(M')) = c(L)$, thus

$$W(\text{Main}(M)) \le c(L). \tag{14}$$

□

Lemma 3. $c(L) \le W(\text{Main}(M))$.

Proof. From $\text{Main}(M)$, we get an MMDC between A and B denoted by L' whose cost is equal to the weight of $\text{Main}(M)$, i.e., $W(\text{Main}(M)) = c(L')$. For each edge $m \in M$, we add the pairing (a_i, b_j) to L' if $m = (a_{ik}, b_{ji})$ or $m = (a_{ik}', b_{ji})$, and no pairing otherwise.

For each $a_i \in A$, there exists the set A_i in G with α_i vertices, which are connected only to one set, i.e., $B\text{set}_i$, with finite edge weights. Thus, the vertices of each A_i are matched to some vertices of $B\text{set}_i$, i.e., b_{ji} with $1 \le j \le t$. Hence, in L', each $a_i \in A$ is matched to at least α_i elements of B (and thus, the demand of a_i is satisfied) for $1 \le i \le s$. In G, there exist α_i plus $\alpha_i' - \alpha_i$ copies of each element a_i, that is the vertices of A_i plus the vertices of A_i'. Thus, in L', the number of elements matched to each $a_i \in A$ is at most α_i'.

Consider the sets B_j with $1 \le j \le t$. Recall that $B_j = \{b_{ji}\}_{1 \le i \le s}$ and the vertices of W_j are connected to B_j by zero weighted edges. W_j is connected only to B_j (with finite edge weights), thus the vertices of W_j are matched to $s - \beta_j'$ vertices of B_j, and β_j' vertices of B_j remain free. Suppose that k number of β_j' free vertices in B_j are matched to some vertices of the sets A_i for $1 \le i \le s$, then the remaining $\beta_j' - k$

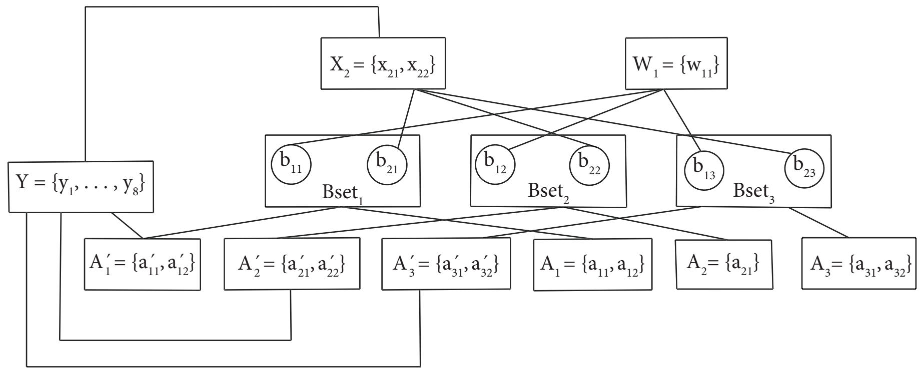

Figure 2: An example for our constructed complete bipartite graph.

vertices of B_j should be matched to other vertices. We discuss two cases, depending on the value of k.

(i) if $k < \beta_j$, then $(\beta'_j - k) > (\beta'_j - \beta_j)$. In this case, the vertices of X_j are matched to $\beta'_j - \beta_j$ number of the remaining $\beta'_j - k$ vertices of B_j. We have

$$\begin{aligned} \left(\beta'_j - k\right) - \left(\beta'_j - \beta_j\right) &= \beta'_j - k - \beta'_j + \beta_j \\ &= \beta_j - k > 0, \end{aligned} \tag{15}$$

thus the remaining$\beta_j - k$ vertices of B_j are matched to some vertices of the sets A'_i for $1 \leq i \leq s$. Note that k and $\beta_j - k$vertices of B_j are matched to some vertices of the sets A_i and A'_i, respectively, for $1 \leq i \leq s$. The demand of the element b_j is satisfied, since

$$\beta_j - k + k = \beta_j. \tag{16}$$

(ii) if $k > \beta_j$, then $(\beta'_j - k) < (\beta'_j - \beta_j)$. Thus, all the remaining $\beta'_j - k$ vertices of B_j are matched to the vertices of X_j.

The cost of L' is equal to the weight of $\text{Main}(M)$, i.e., $c(L') = W(\text{Main}(M))$, since for each edge of $\text{Main}(M)$, we add a pairing with equal cost to L'. L is a minimum-cost MMDC between A and B, thus $c(L) \leq c(L')$. Therefore,

$$c(L) \leq W(\text{Main}(M)). \tag{17}$$

□

Theorem 1. *Let M be a minimum-weight perfect matching in G, and let L be a minimum-cost MMDC between A and B. Then, $W(\text{Main}(M)) = c(L)$.*

Proof. From Lemmas 1 and 3, we have $W(\text{Main}(M)) \leq c(L)$ and $W(\text{Main}(M)) \geq c(L)$, respectively. Thus, $W(\text{Main}(M)) = c(L)$.

Recall that the time complexity of the Hungarian algorithm is $O(n^3)$, where the number of vertices of the input graph is $O(n)$. Our complete bipartite graph has $O(n^2)$ vertices (by construction), so our algorithm takes $O(n^6)$ time. Observe that when $t \ll s$, we have $s * t \approx n = O(n)$ (for example, when a very large number of distributed access points must be matched to a much smaller number of mobile stations [21], or limited resources are matched to a massive number of targets [22]), and thus, our algorithm runs in $O(n^3)$ time. Moreover, in bipartite graphs with low-range edge weights, our algorithm runs well due to [19]

As an example, consider two sets $A = \{a_1, a_2, a_3\}$ and $B = \{b_1, b_2\}$ with

$$\begin{aligned} \{\alpha_1, \alpha_2, \alpha_3\} &= \{2, 1, 2\}, \{\alpha'_1, \alpha'_2, \alpha'_3\} \\ &= \{4, 3, 4\}, \{\beta_1, \beta_2\} \\ &= \{2, 1\}, \{\beta'_1, \beta'_2\} \\ &= \{2, 3\}. \end{aligned} \tag{18}$$

Then, we have

$$\begin{aligned} A_1 &= \{a_{11}, a_{12}\}, \\ A_2 &= \{a_{21}\}, \\ A_3 &= \{a_{31}, a_{32}\}, \\ A'_1 &= \{a'_{11}, a'_{12}\}, \\ A'_2 &= \{a'_{21}, a'_{22}\}, \\ A'_3 &= \{a'_{31}, a'_{32}\}, \text{and} \\ \text{Bset}_1 &= \{b_{11}, b_{21}\}, \\ \text{Bset}_2 &= \{b_{12}, b_{22}\}, \\ \text{Bset}_3 &= \{b_{13}, b_{23}\}. \end{aligned} \tag{19}$$

Also, we have

$$\begin{aligned} X_1 &= \varnothing, X_2 = \{x_{21}, x_{22}\}, \\ W_1 &= \{w_{11}\}, W_2 = \varnothing, Y = \{y_1, \ldots, y_8\}. \end{aligned} \tag{20}$$

Our constructed complete bipartite graph for the above example is shown in Figure 2. □

3. Conclusion

In this article, we presented an algorithm for finding a minimum-cost MMDC between two sets A and B, where each element of A (resp. B) must be matched to at least and at most given numbers of elements of B (resp. A). We constructed a complete bipartite graph $G = (S \cup T, E)$ with $|S| = |T| = O(n^2)$, where $|A| + |B| = n$. Then, we used the Hungarian algorithm with the input G to find a minimum-cost MMDC between A and B in $O(n^6)$ time. Our constructed complete bipartite graph G can be used as the input for the modified Hungarian algorithm proposed in [11] for computing a minimum-cost MMDC between A and B in $O(n^4\text{poly}(\log n^2))$ time when A and B are points in the plane. Observe that when $t \ll s$, running the Hungarian algorithm (and the modified Hungarian algorithm proposed in [11]) on G takes $O(n^3)$ (and $n^2\text{poly}(\log n)$) time. It is expected that the computational complexity of the MMDC problem will be reduced by exploiting the geometric information. One could study the two-dimensional MMDC as a future work, where A and B are points on the plane. We could also consider the case in which A and B are points on two perpendicular lines. The online MMDC is another open problem where the elements of $A \cup B$ arrive online.

Data Availability

No data were used to support this study.

Disclosure

The preprint of this article is available in [23].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[2] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart, and Winston, New York, NY, USA, 1976.

[3] Z. Galil, S. Micali, and H. N. Gabow, "Priority queues with variable priority and an O(EVlogV) algorithm for finding a maximal weighted matching in general graphs," in *Proceedings of the 22nd Annual IEEE Symposium on Foundations of Computer Science (FOCS'82)*, Chicago, IL, USA, November 1982.

[4] T. B. Eiter and H. Mannila, "Distance measures for point sets and their computation," *Acta Informatica*, vol. 34, no. 2, pp. 109–133, 1997.

[5] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *Journal of the ACM*, vol. 34, no. 3, pp. 596–615, 1987.

[6] J. Keijsper and R. Pendavingh, "An efficient algorithm for minimum-weight bibranching," *Journal of Combinatorial Theory - Series B*, vol. 73, no. 2, pp. 130–145, 1998.

[7] H. N. Gabow and R. E. Tarjan, "Faster scaling algorithms for network problems," *SIAM Journal on Computing*, vol. 18, no. 5, pp. 1013–1036, 1989.

[8] M. Imanparast and S. N. Hashemi, "A linear time randomized approximation algorithm for Euclidean matching," *The Journal of Supercomputing*, vol. 75, no. 5, pp. 2648–2664, 2019.

[9] C. C. Huang and T. Kavitha, "New algorithms for maximum weight matching and a decomposition theorem," *Mathematics of Operations Research*, vol. 42, no. 2, pp. 411–426, 2017.

[10] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*, Society for Industrial and Applied Mathematics, Philadelphia, PL, USA, 2012.

[11] S. Bandyapadhyay, A. Maheshwari, and M. Smid, "Exact and approximation algorithms for many-to-many point matching in the plane," in *Proceedings of the In the 32nd International Symposium on Algorithms and Computation (ISAAC 2021)*, Fukuoka, Japan, December 2021..

[12] J. Colannino, M. Damian, F. Hurtado et al., "Efficient many-to-many point matching in one dimension," *Graphs and Combinatorics*, vol. 23, no. S1, pp. 169–178, 2007.

[13] Z. Zhang, X. Kou, I. Palomares, W. Yu, and J. Gao, "Stable two-sided matching decision making with incomplete fuzzy preference relations: a disappointment theory based approach," *Applied Soft Computing*, vol. 84, Article ID 105730, 2019.

[14] Z. Zhang, J. Gao, Y. Gao, and W. Yu, "Two-sided matching decision making with multi-granular hesitant fuzzy linguistic term sets and incomplete criteria weight information," *Expert Systems with Applications*, vol. 168, Article ID 114311, 2021.

[15] Z. Zhang, X. Kou, W. Yu, and Y. Gao, "Consistency improvement for fuzzy preference relations with self-confidence: an application in two-sided matching decision making," *Journal of the Operational Research Society*, vol. 72, no. 8, pp. 1914–1927, 2021.

[16] H. N. Gabow, "An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems," in *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, New York, NY, USA, December 1983.

[17] F. Rajabi-Alni and A. Bagheri, "An $O(n^2)$ algorithm for the limited-capacity many-to-many point matching in one dimension," *Algorithmica*, vol. 76, no. 2, pp. 381–400, 2016.

[18] A. Gupta and L. Ying, "Algorithms for finding maximum matchings in bipartite graphs," Technical report, IBM T. J. Watson Research Center, Ossining, NY, USA, 1999.

[19] P. A. Lopes, S. S. Yadav, A. Ilic, and S. K. Patra, "Fast block distributed CUDA implementation of the Hungarian algorithm," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 50–62, 2019.

[20] G. Liu and B. Zhu, "Some problems on factorizations with constraints in bipartite graphs," *Discrete Applied Mathematics*, vol. 128, no. 2-3, pp. 421–434, 2003.

[21] X. Liu, X. Di, J. Li et al., "Allocating limited resources to protect a massive number of targets using a game theoretic model," *Mathematical Problems in Engineering*, vol. 2019, Article ID 5475341, 16 pages, 2019.

[22] S. Buzzi, C. D'Andrea, A. Zappone, and C. D'Elia, "User-centric 5G cellular networks: resource allocation and comparison with the cell-free massive MIMO approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 1250–1264, 2020.

[23] F. Rajabi-Alni and A. Bagheri, "Computing a many-to-many matching with demands and capacities between two sets using the Hungarian algorithm," 2022, https://doi.org/10.48550/arXiv.2211.01612.