Computing Minimum-Cost Limited-Capacity Many-To-Many Point Matching

F. Panahi[1], A. Mohades Khorasani[2]

[1]Laboratory of Algorithms and Computational Geometry, Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran, fatemehpanahi@aut.ac.ir

[2]Laboratory of Algorithms and Computational Geometry, Department of Mathematics and Computer Science, Amirkabir University of technology, Tehran, Iran, mohades@aut.ac.ir

Abstract

Let $A = \{a_1, a_2, .., a_s\}$, and $B = \{b_1, b_2, .., b_r\}$ be two sets of points such that $s + r = n$. Also let $C_A = \{\alpha_1, \alpha_2, \dots, \alpha_s\}$ and $C_B = \{\beta_1, \beta_2, \dots, \beta_r\}$ be the capacities of points in A and B. We define minimum-cost limited-capacity many-to-many matching that matches each point $a_i \in A$ to at least one and at most α_i points in B and matches each $b_j \in B$ to at least one and at most β_j points in A, for all i, j where $0 < i \leq s$, $0 < j \leq r$, such that sum of all the matching costs is minimized. Cost of matching $a_i \in A$ *to* $b_j \in B$ is equal to the distance between a_i and b_j. We present an $O(n^3)$ algorithm to solve this problem for any dimension.

1. Introduction

Point matching problem has various applications in many fields such as pattern recognition [1], computer vision [2], music information retrieval, computational music theory. There are different types of point matching problems, depending on the application they are raised from. A minimum perfect matching between two sets maps each point exactly to one point of the other set. Many-to-one matching between A and B, maps each point of A to a point of B and each point of B to at least one point of A. Many-to-many matching matches each point of A to at least one element of B and vice-versa. Cost of a matching is the sum of all matches between points. Many-to-many matching, as a measure of similarity between two theories expressed in a logical language has been studied by Eiter and Mannila [3]. They presented an algorithm which runs in $O(n^3)$ time. J. Colannino and seven other authors provided an $O(nlogn)$ algorithm for minimum-cost many-to-many matching for the case where the points lie on a line[4]. In some applications such as resource allocation, some points may represent resources while matches represent resource allocations. In this paper we consider the case where the capacities of the resources are limited, and therefore the number of matches to each point should be limited.

2. The Proposed Algorithm

Let $A = \{a_1, a_2, .., a_s\}$, and $B = \{b_1, b_2, .., b_r\}$ be two sets of points with total cardinality *n,* and $C_A = \{\alpha_1, \alpha_2, \dots, \alpha_s\}$ and $C_B = \{\beta_1, \beta_2, \dots, \beta_r\}$ be their capacity sets. A minimum-cost limited-capacity many-to-many point matching, here called MLM matching, is a matching that matches each point $a_i \in A$ to at least one and at most α_i points in B and each $b_j \in B$ to at least one and at most β_j points in A, such that sum of the matching costs is minimized. Cost of the match of $a_i \in A$ and $b_j \in B$ is equal to the distance between them. (for all i, j which $0 < i \leq s$, $0 < j \leq r$). We assume that $\sum \alpha_i$ and $\sum \beta_j$ are at least r and s, respectively. Otherwise the matching would not exist. Here we present an algorithm to find MLM matching in $O(n^3)$ time. The idea of algorithm is reducing the problem to minimum weighted perfect-matching problem.

Let δ be a function on $A \cup B$ which maps each point of A to its nearest point in B and vice-versa. (i.e. foe each $a_i \in A$ and $b_j \in B$, $\delta(a_i)$ = Nearest point of B from a_i and $\delta(b_j)$ = Nearest point of A from b_j). Let $A' = \{a'_i | 0 < i \leq s\}$, $B' = \{b'_j | 0 < j \leq r\}$. For $a'_i \in A'$ and $b'_j \in B'$ we define a map γ on $A' \cup B'$, as $\gamma(a'_i) = \delta(a_i)$ and $\gamma(b'_j) = \delta(b_j)$. We also define B_j as $\{a' | a' \in A', \gamma(a') = b_j\}$ and A_i as $\{b' | b' \in B', \gamma(b') = a_i\}$.

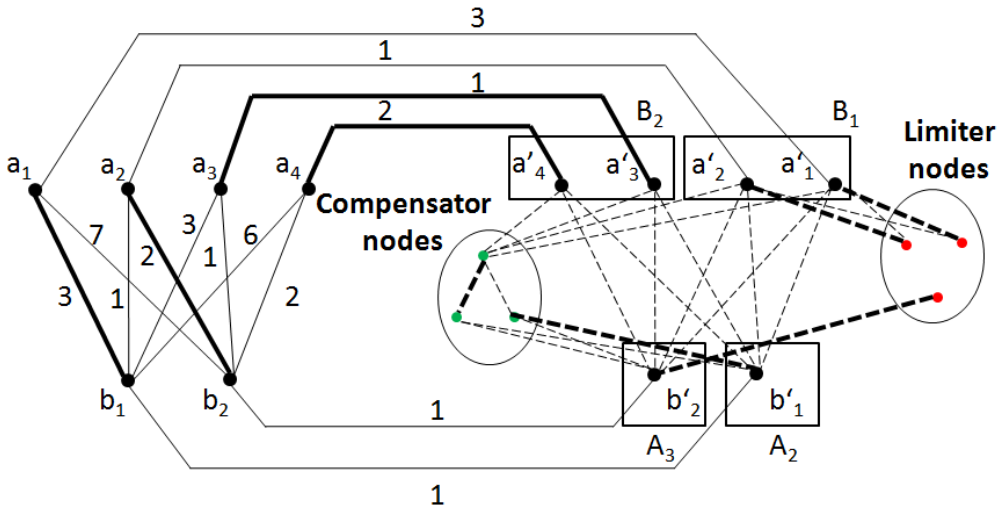

Figure 1. The graph G for capacity sets C_A={2,3,1,2} and C_B={1,3} (dash-lines are zero-weighted edges)

Having the above definitions, we define two complete bipartite graphs as follows:
$G_1 = \{V_1, E_1, w_1\}, V_1 = A \cup B$,
$E_1 = \{(a_i, b_j) | a_i \in A, b_j \in B\}$, $w_1(e_{ij}) = distance(a_i, b_j)$ where $e_{ij} = (a_i, b_j)$.
$G_2 = \{V_2, E_2, w_2\}, V_1 = A' \cup B'$,
$E_1 = \{(a'_i, b'_j) | a'_i \in A', b'_j \in B'\}$, $w_2(e_{ij}) = 0$ where $e_{ij} = (a'_i, b'_j)$.
The next step is to build a graph $G = \{V, E, w\}$. The graph G includes all the edges and nodes with the same weights in G_1 and G_2. G will also have n edges, here called cut edges, between nodes of G_1 and G_2 which connect each a_i in A (and b_j in B) to its corresponding node in A' (B') . The weight of these edges will be $w((a_i, a'_i)) = \delta(a_i)$ and $w((b_j, b'_j)) = \delta(b_j)$. There are two other groups of nodes in G which are called Limiter nodes and Compensator nodes. For each set A_i $(and\ B_j)$, if $|A_i| \geq \alpha_i$ ($|B_j| \geq \beta_j$) there will be $|A_i|$-α_i+1 ($|B_j|$-β_j+1) Limiter nodes, each of which, is connected to all the nodes in A_i (B_j) with a zero-weighted edge.
For every Limiter node, which is connected to A_i (B_j) there is a Compensator node connected to all nodes of A'(B') with a zero-weighted edge. Let com(A') be the set of all compensator nodes connected to A' and com(B') be the set of all the compensator nodes connected to B'. Every node in com(A') is connected to all the nodes in com(B') using a zero-weighted edge (Figure 1).
Since total number of limiter nodes and compensator nodes is $O(n)$, the number of nodes in G is $O(n)$. So the minimum weighted perfect-matching in G could be found in $O(n^3)$ time [5]. We claim that MLM matching of A and B can be reached from the minimum perfect-matching in $O(n^3)$. The matches in MLM matching are the edges in G_1 which appear in the minimum perfect-matching of G plus the matches $[x, \delta(x)]$ where (x, x') is a cut-edge appeared in the minimum perfect-matching.
The MLM matching that resulted from the above procedure will have the same cost as the original perfect matching. We are also sure that no capacity constraint of any node is violated. That is because for any node a_i Limiter nodes connected to A_i will make sure that no more than $\alpha_i - 1$ cut edges connected to A_i will appear in the perfect matching. Similarly, that is true for nodes in B.
It can also be proved that for any MLM matching there is a corresponding perfect matching in graph G built through the above procedure. The details of the algorithm to do such a conversion are given in the full paper.
In this way, MLM matching can be computed in $O(n^3)$ time.

3. References

[1] Buss, S. R., Yianilos, P. N.: A bipartite matching approach to approximate string comparison and search. Technical report, NEC Research Institute, Princeton, New Jersey (1995)
[2] Demirci, M.F., Shokoufandeh, A.,Keselman, Y., retzner, L., Dickinson, S.: Object recognition as many-to-many feature matching. Int. J. Comput. Visi. 69(2), 203–222 (2006)
[3] Eiter, T., Mannila, H.: Distance measures for point sets and their computation. Acta Inf.34(2), 109–133 (1997)
[4] Justin Colannino, Mirela Damian, Ferran Hurtado, Stefan Langerman, Henk Meijer, Suneet Ramaswami, Diane Souvaine, and Godfried Toussaint, Efficient Many-To-Many Point Matching in One Dimension, Graphs and Combinatorics, (2007)
[5] K. Mehlhorn. Graph Algorithms and NP-Completeness, volume 2 of Data Structures and algorithms. Springer, (1984)