Spatial Matching

Cheng Long, Raymond Chi-Wing Wong

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{clong, raywong}@cse.ust.hk

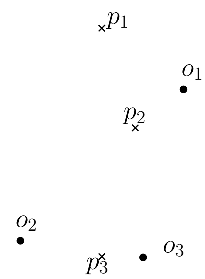
SYNONYMS

DEFINITION

A matching is a mapping from the elements of one set to the elements of another set such that each element in one set is mapped to at most one element in another set. For example, assume two sets of objects $P = \{p_1, p_2, p_3\}$ and $O = \{o_1, o_2, o_3\}$. Then, $\{(p_1, o_1), (p_2, o_2), (p_3, o_3)\}$ is a matching with three pairs, but $\{(p_1, o_1), (p_1, o_2)\}$ is not a matching since p_1 is involved in two pairs. In general, the number of possible matchings is exponential to the cardinality of P and O; e.g., if $|P| = |O| = n$, there are $n!$ matchings with n pairs. Usually, among all possible matchings, the aim is to find one that optimizes/satisfies a certain *criterion*.

Let $c(p, o)$ be the *cost* of matching $p \in P$ with $o \in O$. *Optimal matching* [12] minimizes the *sum* of the costs of all pairs. *Bottleneck matching* [7] minimizes the *maximum* cost of any pair. *Fair matching*, also known as the *stable marriage problem*, returns a matching in which the following conditions cannot hold at the same time: i) some $p \in P$ prefers some $o' \in O$ over the element o to which p is already matched (i.e., $c(p, o') < c(p, o)$) , and ii) o' also prefers p over the element p' to which o' is already matched (i.e., $c(p, o') < c(p', o')$). It has been proved that, when $|P| = |O|$, it is always possible to solve fair matching [6]. A comprehensive survey on matching problems can be found in [3].

In *spatial matching*, objects (or elements) have a *location* and the cost of matching $p \in P$ and $o \in O$ is based on the (Euclidean) distance $dist(p, o)$ between them [14, 13, 11, 4, 9]. Some examples include matching between access points and mobile devices, between emergency facilities (e.g., hospitals) and users, between parking slots and drivers etc. Even non-spatial applications may involve spatial matching based on distances; e.g., a matching between jobs and applicants, where each job/applicant is represented by a vector of attribute values.



(a) Spatial layout

$dist$	o_1	o_2	o_3
p_1	5	10	11
p_2	3	7	6
p_3	9	4	2

(b) Pairwise distances

Figure 1: A running example

HISTORICAL BACKGROUND

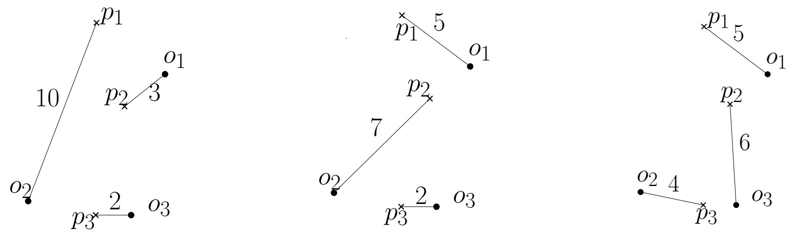
There is extensive literature on algorithms for (non-spatial) matching problems. Currently, the time complexity of the best solution is: i) for fair matching, $O(|P| \cdot |O|)$ [6], ii) for optimal matching, $O((|P| + |O|)^3)$ [3], and iii) for bottleneck matching, $O((|P| + |O|)^{2.5} \log^{0.5}(|P| + |O|))$ [5]. However, the straightforward application of these algorithms to spatial matching suffers from several drawbacks. First, most existing solutions are based on a materialized bipartite graph between P and O which has $O(|P| + |O|)$ vertices and $O(|P| \cdot |O|)$ edges. This limits their scalability since the space cost of maintaining the bipartite graph is prohibitive for large datasets. Second, the algorithms for general matching problems ignore the spatial context, which could enhance performance.

SCIENTIFIC FUNDAMENTALS

The above versions of matching problems have corresponding counterparts in the spatial domain. For illustration we use the layout of Figure 1(a) and the pairwise distances between P and O shown in Figure 1(b). *Spatial fair matching* (SFM) is the spatial version of the stable marriage problem. The matching $M_1 = \{(p_1, o_2), (p_2, o_1), (p_3, o_3)\}$ in Figure 2(a) M_1 is fair. On the other hand, $M_2 = \{(p_1, o_1), (p_2, o_2), (p_3, o_3)\}$ in Figure 2(b) is not fair because p_2 and o_1 would prefer each other to their current matches (o_2 and p_1, respectively) since $dist(p_2, o_1) < dist(p_2, o_2)$ and $dist(p_2, o_1) < dist(p_1, o_1)$. Wong et al. [14] propose four algorithms for SFM, among which the fastest has time complexity $O((|P| + |O|) \log(|P| + |O|))$.

Spatial optimal matching (SOM) refers to the optimal matching problem in the spatial context. The three matchings M_1, M_2 and M_3 of Figure 2 have total costs (i.e., sum of distances of all pairs) 15, 14 and 15, respectively. It can be verified that among all matchings with 3 pairs, M_2 has the smallest total cost, and is the solution of this SOM. U et al. [13] model SOM as a *minimum-cost flow* problem and solve it by using R-trees and an adaptation of the *successive shortest path algorithm* [2].

Spatial bottleneck matching (SBM) is the spatial version of the bottleneck matching



(a) M_1 (SFM) (b) M_2 (SOM) (c) M_3 (SBM)

Figure 2: Spatial matching problems

problem. The maximum cost of any pair in M_1, M_2 and M_3 is 10, 7 and 6, respectively. It can be verified that among all matchings with 3 pairs, M_3 has the lowest maximum cost and constitutes the solution of this SBM. Efrat et al. [4] extend the *Threshold* algorithm [3], originally designed for general bottleneck matching, by utilizing geometrical properties. *Swap-Chain* [9] generates an optimal solution of SBM by repeatedly re-matching pairs so that the maximum cost decreases, until no such adjustment is possible.

KEY APPLICATIONS

Spatial matching is useful in the following applications: emergency facility allocation (e.g., between fire stations/hospitals/police stations and users), profile matching (e.g., between jobs and applicants where both jobs and applicants are represented by vectors/multi-dimensional points), parking slot matching (e.g., between parking slots and drivers) and wireless networks (e.g., between access points and mobile devices).

FUTURE DIRECTIONS*

Currently, most existing studies assume a *static* environment where the objects have fixed locations. One interesting direction refers to spatial matching problems in *dynamic* environments, where the objects move and the goal is to continuously monitor the matching changes (i.e., similar to [11] for SOM). Another direction concerns spatial matching problems with alternative criteria (e.g., *rank-maximal matching* [8], *pareto-optimal matching* [1] and *leximin-optimal matching* [10]).

DATA SETS*

The datasets used by [14] (for SFM) and by [9] (for SBM) can be found at the R-tree portal (http://www.rtreeportal.org/).

URL to CODE*

The code for SFM used by [14] can be found at http://www.cse.ust.hk/~raywong/code/quota.zip. The code for SBM used by [9] can be found at http://www.cse.ust.hk/~raywong/code/spm-mm.zip.

CROSS REFERENCES

NEAREST NEIGHBOR QUERY
R-TREE

RECOMMENDED READING*

References

[1] D. Abraham, K. Cechlarova, D. Manlove, and K. Mehlhorn. Pareto optimality in house allocation problems. *Lecture notes in computer science*, 3341:3–15, 2004.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[3] R. E. Burkard, M. Dell'Amico, and S. Martello. *Assignment problems*. Society for Industrial Mathematics, 2009.

[4] A. Efrat, A. Itai, and M. J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, 2001.

[5] H. N. Gabow and R. E. Tarjan. Algorithms for two bottleneck optimization problems. *Journal of Algorithms*, 9(3):411–417, 1988.

[6] D. Gale and L. Shapley. College admissions and the stability of marriage. *Amer. Math. Monthly*, 69:9–15, 1962.

[7] O. Gross. The bottleneck assignment problem. *The Rand Corporation*, 1959.

[8] R. W. Irving, T. Kavitha, K. Mehlhorn, D. Michail, and K. Paluch. Rank-maximal matchings. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 68–75. Society for Industrial and Applied Mathematics, 2004.

[9] C. Long, R. C.-W. Wong, P. S. Yu, and M. Jiang. On optimal worst-case matching. In *SIGMOD*, 2013.

[10] K. Mehlhorn. Assigning papers to referees. *Automata, Languages and Programming*, pages 1–2, 2009.

[11] K. Mouratidis and N. Mamoulis. Continuous spatial assignment of moving users. *The VLDB Journal*, 19(2):141–160, 2010.

[12] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[13] L. H. U, M. L. Yiu, K. Mouratidis, and N. Mamoulis. Capacity constrained assignment in spatial databases. In *SIGMOD*, 2008.

[14] R. C.-W. Wong, Y. Tao, A. W.-C. Fu, and X. Xiao. On efficient spatial matching. In *VLDB*, 2007.