e-ISSN:2582-7219

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

Volume 7, Issue 1, January 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.54

6381 907 438 6381 907 438 ijmrset@gmail.com www.ijmrset.com

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.54| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 1, January 2024 |

| DOI:10.15680/IJMRSET.2024.0701015 |

A Brief Review On Assignment Problem and Its Applications

Krisn Pratap Meena

Associate Professor, Dept. Of Mathematics, S.R.R.M. Govt. College, Nawalgarh, Rajasthan, India

ABSTRACT: Assignment problem arises in diverse situations, where one needs to determine an optimal way to assign subjects to subjects in the best possible way. With that, this paper classified assignment problems into two, which are timetabling problem and allocation problem. The timetabling problem is further classified into examination, course, and school timetabling problems, while the allocation problem is divided into student-project allocation, new student allocation, and space allocation problems. Furthermore, the constraints, which are of hard and soft constraints, involved in the said problems are briefly elaborated. In addition, this paper presents various approaches to address various types of assignment problem. Moreover, direction and potential paths of problem solving based on the latest trend of approaches are also highlighted. As such, this review summarizes and records a comprehensive survey regarding assignment problem within education domain, which enhances one's understanding concerning the varied types of assignment problems, along with various approaches that serve as solution.

KEYWORDS: assignment, student, eapplications, education, subjects

I.INTRODUCTION

Problems related to assignment arise in a range of fields, for example, healthcare, transportation, education, and sports. In fact, this is a well-studied topic in combinatorial optimization problems under optimization or operations research branches. Besides, problem regarding assignment is an important subject that has been employed to solve many problems worldwide [1]. This problem has been commonly encountered in many educational activities all over the world. Within the education domain, this review classified the assignment problem into two: timetabling problem and allocation problem. Assignment problem refers to the analysis on how to assign objects to objects in the best possible way (optimal way) [2, 3]. The two components of assignment problem are the assignments and the objective function. Te assignment signifies underlying combinatorial structure, while the objective function reflects the desires to be optimized as much as possible. Nonetheless, the question is, "how to carry out an assignment with optimal objective, and at the same time, fulfilling all the related constraints?" In order to address the question, several diverse methods have been proposed [1, 2], such as the exact method [4], the heuristics method [5], the local search-based [6], the population search-based [7], and the hybrid algorithm [8]. Te aim of looking into assignment problem is to discover an assignment among two or more sets of elements, which could minimize the total cost of all matched pairs. Relying on the specific structure of the matched sets and the cost function form, the allocation problems can be categorised into quadratic, bottleneck, linear, and multidimensional groups [9]. Hence, every assignment problem has a table or matrix. Normally, the rows are comprised of objects or people to assign, while the columns consist of the things or tasks to be assigned. Meanwhile, the numbers in the table refer to the costs related to every particular assignment. With that, this study presents a review of assignment problem within educational activities, where the problems were classified into timetabling and allocation problems. In fact, studies within this area have commonly displayed substantial progress with diverse methodologies. Te organization of this paper is given as follows: definition and the mathematical formulation of general assignment problem. Next, the types of assignment problem within the education domain, along with their approaches, are presented . In fact, this section is divided into subsections that elaborate in detail the two types of problem: (i) timetabling problem and (ii) allocation problem. Finally, the conclusion, future direction, and potential path of solution approach .[1,2,3]

The term "auction algorithm"[1] applies to several variations of a combinatorial optimization algorithm which solves assignment problems, and network optimization problems with linear and convex/nonlinear cost. An auction algorithm has been used in a business setting to determine the best prices on a set of products offered to multiple buyers. It is an iterative procedure, so the name "auction algorithm" is related to a sales auction, where multiple bids are compared to determine the best offer, with the final sales going to the highest bidders.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.54| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 1, January 2024 |

| DOI:10.15680/IJMRSET.2024.0701015 |

The original form of the auction algorithm is an iterative method to find the optimal prices and an assignment that maximizes the net benefit in a bipartite graph, the maximum weight matching problem (MWM).[2][3] This algorithm was first proposed by Dimitri Bertsekas in 1979.

The ideas of the auction algorithm and ε-scaling[1] are also central in preflow-push algorithms for single commodity linear network flow problems. In fact the preflow-push algorithm for max-flow can be derived by applying the original 1979 auction algorithm to the max flow problem after reformulation as an assignment problem. Moreover, the preflow-push algorithm for the linear minimum cost flow problem is mathematically equivalent to the ε-relaxation method, which is obtained by applying the original auction algorithm after the problem is reformulated as an equivalent assignment problem.[4]

A later variation of the auction algorithm that solves shortest path problems was introduced by Bertsekas in 1991.[5] It is a simple algorithm for finding shortest paths in a directed graph. In the single origin/single destination case, the auction algorithm maintains a single path starting at the origin, which is then extended or contracted by a single node at each iteration. Simultaneously, at most one dual variable will be adjusted at each iteration, in order to either improve or maintain the value of a dual function. In the case of multiple origins, the auction algorithm is well-suited for parallel computation.[5] The algorithm is closely related to auction algorithms for other network flow problems.[5] According to computational experiments, the auction algorithm is generally inferior to other state-of-the-art algorithms for the all destinations shortest path problem, but is very fast for problems with few destinations (substantially more than one and substantially less than the total number of nodes); see the article by Bertsekas, Pallottino, and Scutella, Polynomial Auction Algorithms for Shortest Paths.

Auction algorithms for shortest hyperpath problems have been defined by De Leone and Pretolani in 1998. This is also a parallel auction algorithm for weighted bipartite matching, described by E. Jason Riedy in 2004[4,5,6]

The weapon target assignment problem (WTA) is a class of combinatorial optimization problems present in the fields of optimization and operations research. It consists of finding an optimal assignment of a set of weapons of various types to a set of targets in order to maximize the total expected damage done to the opponent.

The basic problem is as follows:

> There are a number of weapons and a number of targets. The weapons are of type . There are available weapons of type . Similarly, there are targets, each with a value of . Any of the weapons can be assigned to any target. Each weapon type has a certain probability of destroying each target.

Notice that as opposed to the classic assignment problem or the generalized assignment problem, more than one agent (i.e., weapon) can be assigned to each task (i.e., target) and not all targets are required to have weapons assigned. Thus, we see that the WTA allows one to formulate optimal assignment problems wherein tasks require cooperation among agents. Additionally, it provides the ability to model probabilistic completion of tasks in addition to costs.

Both static and dynamic versions of WTA can be considered. In the static case, the weapons are assigned to targets once. The dynamic case involves many rounds of assignment where the state of the system after each exchange of fire (round) is considered in the next round. While the majority of work has been done on the static WTA problem, recently the dynamic WTA problem has received more attention.

In spite of the name, there are nonmilitary applications of the WTA. The main one is to search for a lost object or person by heterogeneous assets such as dogs, aircraft, walkers, etc. The problem is to assign the assets to a partition of the space in which the object is located to minimize the probability of not finding the object. The "value" of each element of the partition is the probability that the object is located there.[7,8,9]

II.DISCUSSION

The assignment problem is a fundamental combinatorial optimization problem. In its most general form, the problem is as follows:

> The problem instance has a number of agents and a number of tasks. Any agent can be assigned to perform any task, incurring some cost that may vary depending on the agent-task assignment. It is required to perform as many tasks as possible by assigning at most one agent to each task and at most one task to each agent, in such a way that the total cost of the assignment is minimized.

Alternatively, describing the problem using graph theory:

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.54| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 1, January 2024 |

| DOI:10.15680/IJMRSET.2024.0701015 |

The assignment problem consists of finding, in a weighted bipartite graph, a matching of a given size, in which the sum of weights of the edges is minimum.

If the numbers of agents and tasks are equal, then the problem is called balanced assignment. Otherwise, it is called unbalanced assignment.[1] If the total cost of the assignment for all tasks is equal to the sum of the costs for each agent (or the sum of the costs for each task, which is the same thing in this case), then the problem is called linear assignment. Commonly, when speaking of the assignment problem without any additional qualification, then the linear balanced assignment problem is meant.

Examples

Suppose that a taxi firm has three taxis (the agents) available, and three customers (the tasks) wishing to be picked up as soon as possible. The firm prides itself on speedy pickups, so for each taxi the "cost" of picking up a particular customer will depend on the time taken for the taxi to reach the pickup point. This is a balanced assignment problem. Its solution is whichever combination of taxis and customers results in the least total cost.

Now, suppose that there are four taxis available, but still only three customers. This is an unbalanced assignment problem. One way to solve it is to invent a fourth dummy task, perhaps called "sitting still doing nothing", with a cost of 0 for the taxi assigned to it. This reduces the problem to a balanced assignment problem, which can then be solved in the usual way and still give the best solution to the problem.

Similar adjustments can be done in order to allow more tasks than agents, tasks to which multiple agents must be assigned (for instance, a group of more customers than will fit in one taxi), or maximizing profit rather than minimizing cost.[10,11,12]

The (sequential) auction algorithms for the shortest path problem have been the subject of experiments which have been reported in technical papers.[7] Experiments clearly show that the auction algorithm is inferior to the state-of-the-art shortest-path algorithms for finding the optimal solution of single-origin to all-destinations problems.[7]

Although with the auction algorithm the total benefit is monotonically increasing with each iteration, in the Hungarian algorithm (from Kuhn, 1955; Munkres, 1957) the total benefit strictly increases with each iteration.

The auction algorithm of Bertsekas for finding shortest paths within a directed graph is reputed to perform very well on random graphs and on problems with few destinations.[5]

The quadratic assignment problem (QAP) is one of the fundamental combinatorial optimization problems in the branch of optimization or operations research in mathematics, from the category of the facilities location problems first introduced by Koopmans and Beckmann.[1]

The problem models the following real-life problem:

There are a set of n facilities and a set of n locations. For each pair of locations, a distance is specified and for each pair of facilities a weight or flow is specified (e.g., the amount of supplies transported between the two facilities). The problem is to assign all facilities to different locations with the goal of minimizing the sum of the distances multiplied by the corresponding flows.

Intuitively, the cost function encourages facilities with high flows between each other to be placed close together.

The problem statement resembles that of the assignment problem, except that the cost function is expressed in terms of quadratic inequalities, hence the name.

In graph theory and combinatorial optimization, a closure of a directed graph is a set of vertices C, such that no edges leave C. The closure problem is the task of finding the maximum-weight or minimum-weight closure in a vertex-weighted directed graph.[1][2] It may be solved in polynomial time using a reduction to the maximum flow problem. It may be used to model various application problems of choosing an optimal subset of tasks to perform, with dependencies between pairs of tasks, one example being in open pit mining.

III.RESULTS

A naive solution for the assignment problem is to check all the assignments and calculate the cost of each one. This may be very inefficient since, with n agents and n tasks, there are n! (factorial of n) different assignments.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.54| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 1, January 2024 |

| DOI:10.15680/IJMRSET.2024.0701015 |

Another naive solution is to greedily assign the pair with the smallest cost first, and remove the vertices; then, among the remaining vertices, assign the pair with the smallest cost; and so on. This algorithm may yield a non-optimal solution. For example, suppose there are two tasks and two agents with costs as follows:

- Alice: Task 1 = 1, Task 2 = 2.
- George: Task 1 = 5, Task 2 = 8.

The greedy algorithm would assign Task 1 to Alice and Task 2 to George, for a total cost of 9; but the reverse assignment has a total cost of 7.

Fortunately, there are many algorithms for finding the optimal assignment in time polynomial in n. The assignment problem is a special case of the transportation problem, which is a special case of the minimum cost flow problem, which in turn is a special case of a linear program. While it is possible to solve any of these problems using the simplex algorithm, each specialization has a smaller solution space and thus more efficient algorithms designed to take advantage of its special structure.

In the balanced assignment problem, both parts of the bipartite graph have the same number of vertices, denoted by n.

One of the first polynomial-time algorithms for balanced assignment was the Hungarian algorithm. It is a global algorithm – it is based on improving a matching along augmenting paths (alternating paths between unmatched vertices). Its run-time complexity, when using Fibonacci heaps, is ,[2] where m is a number of edges. This is currently the fastest run-time of a strongly polynomial algorithm for this problem. If all weights are integers, then the run-time can be improved to , but the resulting algorithm is only weakly-polynomial.[3] If the weights are integers, and all weights are at most C (where C>1 is some integer), then the problem can be solved in weakly-polynomial time in a method called weight scaling.[4][5][6]

In addition to the global methods, there are local methods which are based on finding local updates (rather than full augmenting paths). These methods have worse asymptotic runtime guarantees, but they often work better in practice. These algorithms are called auction algorithms, push-relabel algorithms, or preflow-push algorithms. Some of these algorithms were shown to be equivalent.[7]

Some of the local methods assume that the graph admits a perfect matching; if this is not the case, then some of these methods might run forever.[1]:3 A simple technical way to solve this problem is to extend the input graph to a complete bipartite graph, by adding artificial edges with very large weights. These weights should exceed the weights of all existing matchings, to prevent appearance of artificial edges in the possible solution.

As shown by Mulmuley, Vazirani and Vazirani,[8] the problem of minimum weight perfect matching is converted to finding minors in the adjacency matrix of a graph. Using the isolation lemma, a minimum weight perfect matching in a graph can be found with probability at least ½. For a graph with n vertices, it requires time.

Other approaches for the assignment problem exist and are reviewed by Duan and Pettie[9] (see Table II). Their work proposes an approximation algorithm for the assignment problem (and the more general maximum weight matching problem), which runs in linear time for any fixed error bound.

When phrased as a graph theory problem, the assignment problem can be extended from bipartite graphs to arbitrary graphs. The corresponding problem, of finding a matching in a weighted graph where the sum of weights is maximized, is called the maximum weight matching problem.[13,14,15]

Another generalization of the assignment problem is extending the number of sets to be matched from two to many. So that rather than matching agents to tasks, the problem is extended to matching agents to tasks to time intervals to locations. This results in Multidimensional assignment problem (MAP).

In applied mathematics, the maximum generalized assignment problem is a problem in combinatorial optimization. This problem is a generalization of the assignment problem in which both tasks and agents have a size. Moreover, the size of each task might vary from one agent to the other.

This problem in its most general form is as follows: There are a number of agents and a number of tasks. Any agent can be assigned to perform any task, incurring some cost and profit that may vary depending on the agent-task assignment. Moreover, each agent has a budget and the sum of the costs of tasks assigned to it cannot exceed this budget. It is required to find an assignment in which all agents do not exceed their budget and total profit of the assignment is maximized.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.54| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 1, January 2024 |

| DOI:10.15680/IJMRSET.2024.0701015 |

In the special case in which all the agents' budgets and all tasks' costs are equal to 1, this problem reduces to the assignment problem. When the costs and profits of all tasks do not vary between different agents, this problem reduces to the multiple knapsack problem. If there is a single agent, then, this problem reduces to the knapsack problem.

The Hungarian method is a combinatorial optimization algorithm that solves the assignment problem in polynomial time and which anticipated later primal–dual methods. It was developed and published in 1955 by Harold Kuhn, who gave it the name "Hungarian method" because the algorithm was largely based on the earlier works of two Hungarian mathematicians, Dénes Kőnig and Jenő Egerváry.[1][2] However, in 2006 it was discovered that Carl Gustav Jacobi had solved the assignment problem in the 19th century, and the solution had been published posthumously in 1890 in Latin.[3]

James Munkres reviewed the algorithm in 1957 and observed that it is (strongly) polynomial.[4] Since then the algorithm has been known also as the Kuhn–Munkres algorithm or Munkres assignment algorithm. The time complexity of the original algorithm was , however Edmonds and Karp, and independently Tomizawa, noticed that it can be modified to achieve an running time.[5][6] One of the most popular

The problem is NP-hard, so there is no known algorithm for solving this problem in polynomial time, and even small instances may require long computation time. It was also proven that the problem does not have an approximation algorithm running in polynomial time for any (constant) factor, unless P = NP.[2] The travelling salesman problem (TSP) may be seen as a special case of QAP if one assumes that the flows connect all facilities only along a single ring, all flows have the same non-zero (constant) value and all distances are equal to the respective distances of the TSP instance. Many other problems of standard combinatorial optimization problems may be written in this form.
The maximum-weight closure of a given graph G is the same as the complement of the minimum-weight closure on the transpose graph of G, so the two problems are equivalent in computational complexity. If two vertices of the graph belong to the same strongly connected component, they must behave the same as each other with respect to all closures: it is not possible for a closure to contain one vertex without containing the other. For this reason, the input graph to a closure problem may be replaced by its condensation, in which every strongly connected component is replaced by a single vertex. The condensation is always a directed acyclic graph.

As Picard (1976) showed,[2][3] a maximum-weight closure may be obtained from G by solving a maximum flow problem on a graph H constructed from G by adding to it two additional vertices s and t. For each vertex v with positive weight in G, the augmented graph H contains an edge from s to v with capacity equal to the weight of v, and for each vertex v with negative weight in G, the augmented graph H contains an edge from v to t whose capacity is the negation of the weight of v. All of the edges in G are given infinite capacity in H.[1]

A minimum cut separating s from t in this graph cannot have any edges of G passing in the forward direction across the cut: a cut with such an edge would have infinite capacity and would not be minimum. Therefore, the set of vertices on the same side of the cut as s automatically forms a closure C. The capacity of the cut equals the weight of all positive-weight vertices minus the weight of the vertices in C, which is minimized when the weight of C is maximized. By the max-flow min-cut theorem, a minimum cut, and the optimal closure derived from it, can be found by solving a maximum flow problem.[1]

VI.CONCLUSION

In combinatorial optimization, a field within mathematics, the linear bottleneck assignment problem (LBAP) is similar to the linear assignment problem.[1]

In plain words the problem is stated as follows:

> There are a number of agents and a number of tasks. Any agent can be assigned to perform any task, incurring some cost that may vary depending on the agent-task assignment. It is required to perform all tasks by assigning exactly one agent to each task in such a way that the maximum cost among the individual assignments is minimized.

The term "bottleneck" is explained by a common type of application of the problem, where the cost is the duration of the task performed by an agent. In this setting the "maximum cost" is "maximum duration", which is the bottleneck for the schedule of the overall job, to be minimized. In addition to the original plant location formulation, QAP is a mathematical model for the problem of placement of interconnected electronic components onto a printed circuit board or on a microchip, which is part of the place and route stage of computer aided design in the electronics industry.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.54| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 1, January 2024 |

| DOI:10.15680/IJMRSET.2024.0701015 |

In mathematics, the quadratic bottleneck assignment problem (QBAP) is one of the fundamental combinatorial optimization problems in the branch of optimization or operations research, from the category of the facilities location problems.[1]

It is related to the quadratic assignment problem in the same way as the linear bottleneck assignment problem is related to the linear assignment problem, the "sum" is replaced with "max" in the objective function.

The problem models the following real-life problem:

> There are a set of n facilities and a set of n locations. For each pair of locations, a distance is specified and for each pair of facilities a weight or flow is specified (e.g., the amount of supplies transported between the two facilities). The problem is to assign all facilities to different locations with the goal of minimizing the maximum of the distances multiplied by the corresponding flows.[16]

REFERENCES

1. Dimitri P. Bertsekas. "A distributed algorithm for the assignment problem", original paper, 1979.
2. ^ M.G. Resende, P.M. Pardalos. "Handbook of optimization in telecommunications", 2006
3. ^ M. Bayati, D. Shah, M. Sharma. "A Simpler Max-Product Maximum Weight Matching Algorithm and the Auction Algorithm", 2006, webpage PDF: MIT-bpmwm-PDF Archived 2017-09-21 at the Wayback Machine.
4. ^ Dimitri P. Bertsekas. "Distributed Relaxation Algorithms for Linear Network Flow Problems," Proc. of 25th IEEE CDC, Athens, Greece, 1986, pp. 2101-2106, online from IEEEXplore [1]
5. ^ Jump up to:[a b c d] Dimitri P. Bertsekas. "An auction algorithm for shortest paths", SIAM Journal on Optimization, 1:425—447, 1991,PSU-bertsekas91auction
6. ^ "The Parallel Auction Algorithm for Weighted Bipartite Matching", E. Jason Riedy, UC Berkeley, February 2004, [2].
7. ^ Jump up to:[a b] Larsen, Jesper; Pedersen, Ib (1999). "Experiments with the auction algorithm for the shortest path problem". Nordic Journal of Computing. 6 (4): 403–42. ISSN 1236-6064., see also A note on the practical performance of the auction algorithm for the shortest path Archived 2011-06-05 at the Wayback Machine (1997) by the first author.
8. Lyle Ramshaw, Robert E. Tarjan (2012). "On minimum-cost assignments in unbalanced bipartite graphs" (PDF). HP research labs.
9. ^ Fredman, Michael L.; Tarjan, Robert Endre (1987-07-01). "Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms". J. ACM. 34 (3): 596–615. doi:10.1145/28869.28874. ISSN 0004-5411. S2CID 7904683.
10. ^ Thorup, Mikkel (2004-11-01). "Integer priority queues with decrease key in constant time and the single source shortest paths problem". Journal of Computer and System Sciences. Special Issue on STOC 2003. 69 (3): 330–353. doi:10.1016/j.jcss.2004.04.003. ISSN 0022-0000.
11. ^ Gabow, H.; Tarjan, R. (1989-10-01). "Faster Scaling Algorithms for Network Problems". SIAM Journal on Computing. 18 (5): 1013–1036. doi:10.1137/0218069. ISSN 0097-5397.
12. ^ Goldberg, A.; Kennedy, R. (1997-11-01). "Global Price Updates Help". SIAM Journal on Discrete Mathematics. 10 (4): 551–572. doi:10.1137/S0895480194281185. ISSN 0895-4801.
13. ^ Orlin, James B.; Ahuja, Ravindra K. (1992-02-01). "New scaling algorithms for the assignment and minimum mean cycle problems". Mathematical Programming. 54 (1–3): 41–56. doi:10.1007/BF01586040. ISSN 0025-5610. S2CID 18213947.
14. ^ Alfaro, Carlos A.; Perez, Sergio L.; Valencia, Carlos E.; Vargas, Marcos C. (2022-06-01). "The assignment problem revisited". Optimization Letters. 16 (5): 1531–1548. doi:10.1007/s11590-021-01791-4. ISSN 1862-4480. S2CID 238644205.
15. ^ Mulmuley, Ketan; Vazirani, Umesh; Vazirani, Vijay (1987). "Matching is as easy as matrix inversion". Combinatorica. 7 (1): 105–113. doi:10.1007/BF02579206. S2CID 47370049.
16. ^ Duan, Ran; Pettie, Seth (2014-01-01). "Linear-Time Approximation for Maximum Weight Matching" (PDF). Journal of the ACM. 61: 1–23. doi:10.1145/2529989. S2CID 207208641.