Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi

Afyon Kocatepe University Journal of Science and Engineering

AKÜ FEMÜBİD **17** (2017) **011302** (93-100) AKU J. Sci. Eng. **17 (2017) 011302** (93-100)

DOI: 10.5578/fmbd.51761

Araştırma Makalesi / Research Article

A Hypergraph Solution to Generalized Assignment Problem and Application to Spatial Data Sets

Mehmet Ali Balcı[1]

[1] *Muğla Sıtkı Koçman Üniversitesi, Fen Fakültesi, Matematik Bölümü, Muğla.*

e-mail: mehmetalibalci@mu.edu.tr

Geliş Tarihi: 29.05.2016 ; Kabul Tarihi: 20.02.2017

Keywords
Generalized Assignment Problem; Hypergraphs; Delaunay Triangulation; Spatial Data Sets

Abstract

In the simple task assignment problem, at most one task should be assigned to each agent; this constraint is relaxed in the multiple task assignment problems. The goal of the well-known Generalized Assignment Problem is to assign tasks to agents such that the capacity of the agent does not exceed its limits as it minimizes the total cost. In this study we present a novel approach to solve the general assignment problem by using the hypergraphs. Hypergraphs can be considered as the generalization of the graphs in such way that an edge can connect any number of vertices, and can be seen as the set systems. In the hypergraph multi-assignment problem, we are looking for a cost minimizing solution to tasks assignment to the agents which are the individual hyperedges. For this purpose, we first determine the tasks as hyperedges and then obtain the vertex cover of the simple graph representation of a hypergraph. Amongst the all possible covers, we choose the cost minimizing one as the solution.

Genelleştirilmiş Atama Probleminin Hypergraf Çözümü ve Uzaysal Veri Kümelerine Uygulaması

Anahtar kelimeler
Genelleştirilmiş Atama Problemi; Hypergraflar, Delaunay Üçgenlemesi, Uzaysal Veri Kümeleri

Özet

Basit atama probleminde, her bir etkene en fazla bir iş atanmaktadır; bu kısıtlanış genelleştirilmiş atama problemlerinde gevşetilmiştir. Oldukça iyi bilinen Genelleştirilmiş Atama Probleminin hedefi etkenlere iş atarken toplam minimum maliyeti minimumlaştırırken etkenlerin kapasitelerinin limitlerini geçmemesini sağlamaktır. Bu çalışmada Genelleştirilmiş Atama Probleminin çözümü için hypergraflar kullanılarak orijinal bir yöntem verilmiştir. Hypergraflar, bir ayrıt herhangi sayıda tepeyi içerecek şekilde grafların bir genelleştirilmesi olarak ele alınabilir ve küme sistemleri olarak görülebilir. Hypergraf multi-atama probleminde, etkenleri ayrı hyperayrıtlar olarak alıp maliyeti minimize eden çözümler aranmaktadır. Bu amaçla, ilk olarak işleri hyperayrıt olarak belirleyip daha sonra hypergrafın bir basit graf gösteriminde tepe örtü kümesini elde etmekteyiz. Bütün örtü kümeleri içerisinde maliyeti minimize edeni çözüm olarak kabul ederiz.

1. Introduction

The Generalized Assignment Problem (GAP) is emerging when a set of tasks needs to be assigned to a set of agents in a way that the availability of the agent and the multiple resources consumed by him permit the assignment. Rather than the simple assignment problems, the agent consumes not unique but a variety of resources to perform the task. A task can be assigned to any number of agents, henceforth to determine the optimum many agents that minimize the cost of all tasks need to be completed becomes an important optimization problem. In other words, GAP is the problem of assigning n different tasks to m agents while the total cost of the assignment is minimized. Various types of generalized GAP such as dynamic multi-resource GAP (Shtub and Kogan 1998, Privault and Herault 1998), multi-objective GAP

(Subtil *et al.* 2010), and the equilibrium GAP (Liu *et al.* 2012) are studied by several researchers. Fisher and Jaikumar (1981) briefly showed that GAP is a NP-hard problem and many different solution methods have been developed by using branching, cutting and bounding algorithms (Avella *et al.* 2010, Fisher *et al.* 1986, Nauss 2003, Ross and Soland 1975, Savelsbergh 1997).

In this study, we present a new method to solve GAP by using the edge cover of a hypergraph. Hypergraphs can be considered as the generalization of the simple graphs as their edges can involve more than two vertices. Since they include robust relationships in the model of a system, they serve as an efficient tool to model complex systems. From image processing (Yu *et al*. 2012), cybernetics (Zhang *et al.* 2014), to machine learning (Yu *et al.* 2014, Zhang *et al.* 2012), hypergraphs can be applied to several areas. Since hypergraphs provide more flexible and detailed modelling benefits rather than the simple graphs, they are also commonly used in modeling spatial data sets. Cherng and Lo (2001) showed that hypergraphs are useful to determine significant relations among the data points in data structure.

In the study of geographical information systems spatial data set play an important role. Spatial data sets can be seen as an information system where each data represented to identify the location of features. These representations usually involve the spatial coordinates of the features. However they are not only restricted to the geographic locations, the circuit chips or the point clouds of the geometric processing can also be considered as spatial data sets (Balcı *et al.* 2016). Therefore, the representation of such spatial relations can be done by the closeness relationship that is the Delaunay Triangulation of such data sets.

This paper is organized as follow: In Section 2 we give the mathematical description of the GAP and its representation by hypergraphs. We also present how to represent a spatial data set by using the hypergraphs. In Section 3, we give a method to obtain the neighborhood graph of a hypergraph. This neighborhood graph yields the solution of the GAP as a hyperedge cover which is a maximal vertex cover of the simple graph. We also give our method in a Step Algorithm form, and study the method for randomly generated spatial data sets in Section 4 in details. In Section 5, we apply our method and obtain consistent results to 112 Emergency Health Service Stations of Muğla.

2. Description of the Problem

The problem addressed in this paper is to assign tasks to the agents which have spatial coordinates. These spatial coordinates can be interpreted as the location, height, and the other geometric information about the agents. Minimizing the total cost is the objective. If the capacity of the agent exceeds its limits, the task is assigned as the spatially next available agent is involved. Soft constraints on agents in this problem are omitted.

More formally, the GAP for spatial data sets considered here can be stated as follows. A set of agents $A = \{a_1, \dots, a_n\}$ needs to be assigned a task t, where $t \in T = \{t_1, t_2, \dots, t_{max}\}$. This assignment problem is formulated as follows:

$$Minimize: \sum_{a \in A} \mathrm{c_a x_a}$$

$$Subject\ to: \sum_{a \in k} \mathrm{x_a} \geq 1 \quad for\ k \in T$$

$$x_a \in \{0,1\}.$$

where the cost matrix $[c_i]$ is $1 \times\ t_{max}$ type.

By considering the agents as vertices and tasks as hyperedges aforementioned GAP can be modelled by hypergraphs. Formally, a hypergraph H is given with the tuple (V, E). Here V is the set of vertices and E is the set of non-empty subsets of V.

One of the effective applications of Hypergraphs is to the model spatial data sets. These locations can be identified as the city locations, construction or natural features, emergency services, or even the location of transportation stops. Beside these 2 Dimensional examples, identification of the location of features may involve also the 3 Dimensional data sets which includes the spatial locations and their heights. To represent these kinds of data sets by hypergraphs, we first use the dual of the Voronoi region of the corresponding data set, which are called Delaunay Triangulation for 2D case and Delaunay Tetrahedralization for 3D case.

In the case of 2D triangulations of points $P_2 \subset \mathbb{R}^2$, the triangulation $D(P_2)$ in such way that no point is inside the circum-circle of any triangle in $D(P_2)$ is called Delaunay Triangulation. $D(P_2)$ ensures the circumcircle associated with each triangle contains no other point in its interior. By following the same idea, for a set $P_3 \subset \mathbb{R}^3$ of points in the $3d$ Euclidean space, Tetrahedralization $D(P_3)$ such that no point and edge in P_3 is inside the circum-sphere of any tetrahedron in $D(P_3)$ is called the Delaunay Tetrahedralization. For sure this idea can be extended in more general case. For a set $\mathcal{P}$ of points in the d-dimensional Euclidean space, the same triangulation procedure can be executed by using hyperspheres.

Hypergraph representation of a Delaunay Triangulation has been extensively studied and can be defined as follows:

Definition 2.1. Let V be a set of vertices in d-dimensional Euclidean space. The Delaunay Triangulation is a hypergraph with (V, E) with E is the set of $(d-1)$-simplices that have non-empty balls touching their vertices, such that E forms non-overlapping d-simplices.

3. Method

In this section we represent the above GAP for spatial data sets. These data sets are modelled by the hypergraphs which are obtained from the Delaunay Triangulation. The solution of the GAP is given as a vertex cover of the hypergraph. To reach this goal we first give a definition to represent a hypergraph as follows:

Definition 3.1. Let a hypergraph $H = (V, E)$ is given. The neighborhood graph G_H of H is a simple graph with the set of vertices V and the set of edges defined by the following formation rule:

$$v_i v_j \text{ is an edge in } G_H \text{ if and only if } \exists\, e_k \in E \text{ such that } v_i \in e_k \wedge v_j \in e_k.$$

To represent a neighborhood graph of a hypergraph with an adjacency matrix is another computational task to be completed. The following definitions which are studied deeply in (Balcı *et al.* 2016) are served to construct a neighborhood graph in $\mathcal{O}(|E||V|^2)$ time complexity.

Definition 3.2. Let $H = (V, E)$ be a hypergraph with $|V| = n$. The matrix $B_{e_k} = [a_{ij}]_{n\times n}$ whose entries are

$$a_{ij} = \begin{cases} 1, & \{v_i, v_j\} \subset e_k \ for\ i \neq j \\ 0, & Otherwise \end{cases}$$

is called the neighborhood matrix of the edge $e_k \in E$ (Balcı *et al.* 2016).

Definition 3.3. Let $H = (V, E)$ be a hypergraph with $|V| = n$ and $|E| = m$. The simple graph G_H with the adjacency matrix

$$A_{G_H} = B_{e_1} \oplus B_{e_1} \oplus \ldots \oplus B_{e_m}$$

where $\oplus$ is the element-wise Boolean sum of neighborhood matrices is called the neighborhood graph of the hypergraph H (Balcı *et al.* 2016).

Lemma 3.1. A vertex cover of G_H is a hyperedge cover of H.

Proof. Since the hyperedges of H involve the vertices and the vertex cover G_H, which is a set of vertices such that each edge of the G_H is incident to at least one vertex of the set, can be obtained by the neighborhood of the vertices; the adjacent vertices in G_H are equivalent to the adjacent hyperedges in H. Henceforth, by the definition of a vertex cover, a vertex cover of G_H is an edge cover of H.

The method we present in this study can be executed as the given Step Algorithm:

```
Input:    $H = (V, E)$ from the $D(\mathcal{P})$

Step 1:   Obtain the neighborhood graph $G_H$
Step 2:   Find the maximal independent set of $G_H$
Step 3:   Obtain the vertex cover as the complement of the independent set
Step 4:   Determine the set of all possible hyperedges contain the vertex cover
Step 5:   The hyperedges amongst the ones obtained in Step 4 with the minimum cost are the solutions
```

The construction of the simple graph representation is studied in (Balcı *et al.* 2016) with details. In graph theory, independent set is the set of vertices that does not include any two adjacent vertices, and a maximal independent set is an independent set that is not a subset of any other independent set. There are several approaches to approximate the maximal independent set of a graph. In this study, we approximate the maximal independent set by using the Paull-Unger Algorithm. Since this algorithm depends on the finite automata theory, it is not suitable for the large data sets. However, in practice, the numbers of agents in the spatial models are expected to be suitable for using the Paull-Unger algorithm. For instance; if the agents are considered as the emergency services and the task are as the responses, one would not to activate whole agents in the country but would do for the regional ones. To determine maximal independent set plays key role to determine vertex covers. The well-known theorem about the independent sets states that the complement of the maximal independent set in an undirected simple graph yields a vertex cover. Once vertex cover is determined, by using Lemma 3.1, it is possible to obtain the all hyperedge covers of the hypergraph model of the data set. Amongst them, the ones with the minimum cost is the solution of the Hypergraph Multi-Assignment Problem.

In Table 3.1, we present the computational times of our method with Bees Algorithm (Özbakır et al, 2010), Tabu Search Algorithm (Dıaz ve Fernández, 2001), and Differential-Evolution Algorithm (). Test case is type gap-b which is obtained from the OR-library (Int Kyn. 2). Bee, Tabu, DE, HG are the abbreviations for Bee's Algorithm, Tabu Search Algorithm, Differential Evolution Algorithm, and Hypergraph solution that we present in this study, respectively.

		Methods			
m	**n**	**BEE**	**TABU**	**DE**	**HG**
5	100	5.97	95.8	30.56	0.40
5	200	45.99	97.2	50.89	1.32
10	100	0.36	160	60.25	2.22
10	200	315.04	339.1	102.4	31.67
20	100	1.32	389.8	212.5	51.91
20	200	28.65	465.4	350.5	404.68

Table 4.1. Computational Experience for Gapb type GAP.

It can straightforwardly be seen from Table 4.1 the algorithm we present in this study is computationally efficient for the relatively small sized problems.

4. Examples

In this section we give the interpretation of our solution for 2 and 3 dimensional spatial data sets. First scenario is agents do not exceed their capacity for the given task. Second is for the case of the exceeding the capacities, that is the next available agent is included the task. The next available agent may be determined by the nearest agent to the task in the means of Hausdorff distance. However, to keep the initial topology between the agents, we consider the nearest agents amongst the element of Delaunay Triangulation which has an adjacent $(d-1)$-simplices.

4.1.1 2D Spatial Data Sets

Let us consider coordinates of randomly generated 20 agents in $\mathbb{R}^2$ as in Figure 4.1.1.

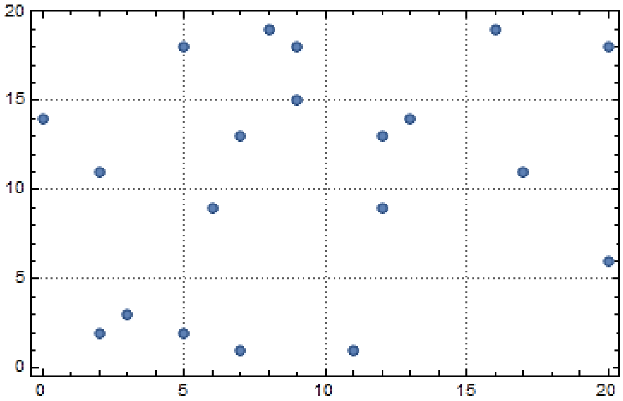

Figure 4.1.1. Randomly generated 2 Dimensional spatial data points

The neighborhood graph of this hypergraph is given in Figure 4.2.2 with the agents numbered from 1 to 20. For the sake of simplicity let us consider $c_i = i$.

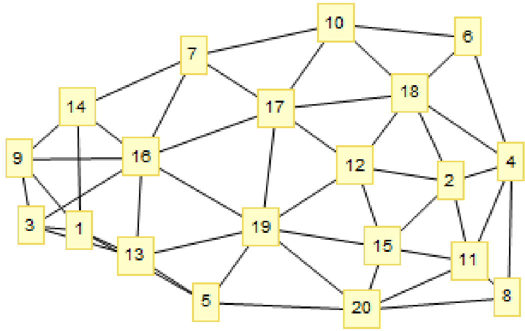

Figure 4.1.2. Representation of the hypergraph in simple form that is obtained from the randomly generated spatial data points.

The vertex cover set of G_H is

$$\left\{ \begin{matrix} \{a_1, a_4, a_7, a_9, a_{10}, a_{11}, a_{12}, a_{13}, \\ a_{15}, a_{16}, a_{18}, a_{19}, a_{20}\}, \\ \{a_1, a_2, a_4, a_7, a_9, a_{10}, a_{11}, a_{12}, a_{13} \\ a_{16}, a_{18}, a_{19}, a_{20}\} \end{matrix} \right\}.$$

As a result, the tasks with minimum cost can be obtained as

$$e_1, e_2, e_3, e_5, e_{10}, e_{12}, e_{14}, e_{15}, e_{16}, e_{17}, e_{24},$$

which is an hyperedge cover set of H.

In the case of limit exceeding, the same set of agents yield the neighborhood graph that is given in Figure 4.1.3.

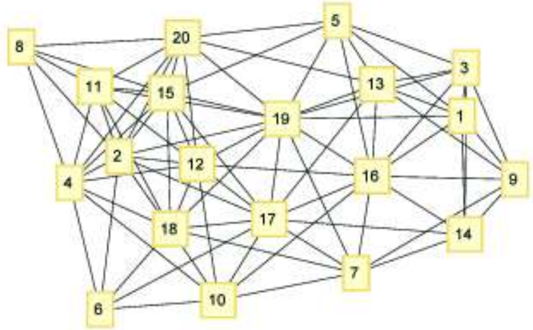

Figure 4.1.3. Neighborhood graph representation of the hypergraph when the limits of the agents exceed and the next available agent is included to the task.

In this case, the vertex cover set is

$$a_2, a_3, a_4, a_5, a_9, a_{10}, a_{11}, a_{13}, \\ a_{14}, a_{15}, a_{16}, a_{17}, a_{18}, a_{19}, a_{20}$$

and corresponding hyperedge cover with minimum cost is

$$e'_1, e'_2, e'_3, e'_4, e'_9, e'_{15}, e'_{17}, e'_{22}, e'_{25}$$

where $E = \{e'_i \ : \ |e'_i| = 4, \ 1 \leq \ i \leq \ 39\}$.

4.1.2 3D Spatial Data Sets

Now, consider for the coordinates of randomly generated 20 agents in $\mathbb{R}^3$ as in Figure 4.2.1.

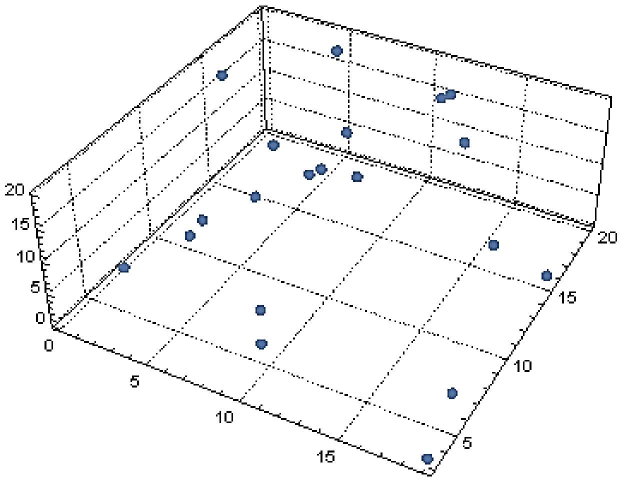

Figure 4.2.1. Randomly generated 3 Dimensional spatial data points

The Delaunay Tetrahedralization of the generated points is given in Figure 4.2.2.

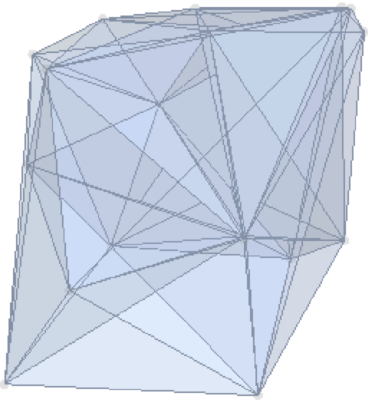

Figure 4.2.2. Delaunay Tetrahedralization that yields the hypergraph with 179 hyperedges.

The neighborhood graph G'_H is in Figure 4.2.3 and the vertex covers are

$$\left\{\begin{array}{c} \{b_1, b_2, b_3, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, \\ b_{13}, b_{14}, b_{15}, b_{16}, b_{19}\}, \\ \{b_1, b_2, b_3, b_5, b_7, b_{10}, b_{11}, b_{12}, b_{13}, \\ b_{14}, b_{15}, b_{16}, b_{18}, b_{19}, b_{20}\} \end{array}\right\}$$

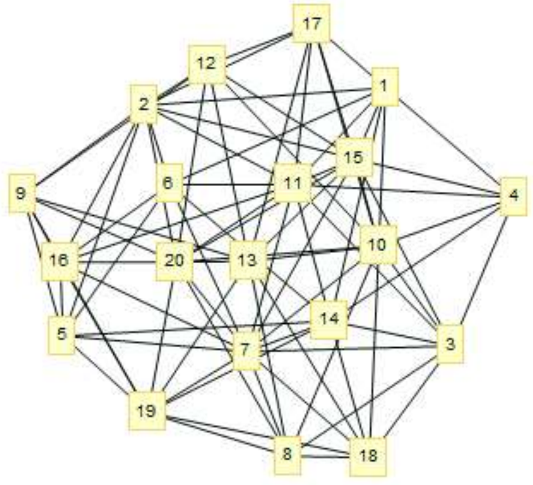

Figure 4.2.3. Representation of the hypergraph in simple form that emerging from the Delaunay Tetrahedralization of the randomly generated 3D spatial data points.

The corresponding hyperedges with minimum costs are

$$\{16, 19, 5, 9\} \quad \{3, 18, 14, 7\} \quad \{11, 15, 3, 7\}$$
$$\{13, 9, 20, 12\} \quad \{4, 15, 1, 10\} \quad \{6, 16, 5, 2\}.$$

Now let us consider the case of agents limits exceed, hence we include the nearest agents to the tasks and obtain 86 hyperedges. The same set of agents yield the neighborhood graph that is given in Figure 4.2.4.

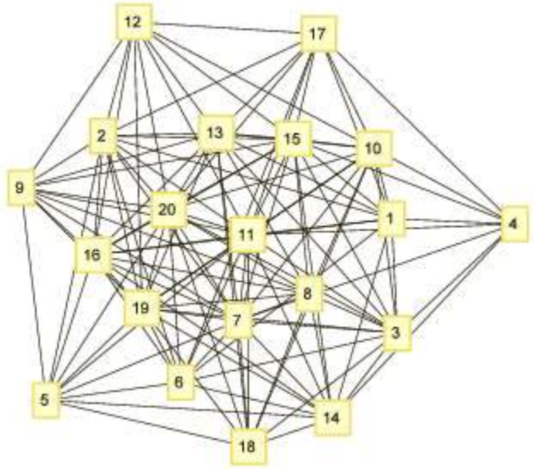

Figure 4.2.4. Neighborhood graph representation of the hypergraph of the 3D spatial data set when the limits of the agents exceed and the next available agent is included to the task.

The vertex covers of this neighborhood graph are

$$\left\{\begin{array}{c} \{b'_1, b'_2, b'_3, b'_5, b'_7, b'_9, b'_{10}, b'_{11}, b'_{13}, \\ b'_{14}, b'_{15}, b'_{16}, b'_{17}, b'_{19}\}, \\ \{b'_2, b'_3, b'_4, b'_6, b'_7, b'_9, b'_{10}, b'_{11}, b'_{13}, \\ b'_{14}, b'_{15}, b'_{16}, b'_{17}, b'_{18}, b'_{19}, b'_{20}\} \end{array}\right\}.$$

The corresponding hyperedges with minimum costs are

$$\{9, 13, 16, 19, 20\} \quad \{9, 12, 13, 19, 20\} \quad \{5, 7, 9, 16, 19\}$$
$$\{2, 9, 16, 19, 20\} \quad \{3, 7, 11, 14, 18\} \quad \{3, 7, 11, 14, 15\}$$
$$\{3, 4, 7, 11, 15\} \quad \{3, 7, 10, 11, 15\} \quad \{3, 6, 7, 11, 14\}$$

5. A Local Application

In this Section, we apply our method to the data sets of Turkish Republic, Ministry of Health, 112 Emergency Health Service Stations of Muğla. The spatial coordinates include the latitude and the longitude of each station for the 2D spatial data sets and the altitude for the 3D spatial data sets. Each coordinates are obtained by using Google Earth and marked as in Figure 5.1.1.

The cost of each hyperedge that emerge from the Delaunay Triangulation is calculated as

$$c_i = \sum_{k=1}^{l} \frac{1}{3d_k + 2p_k + t_k},$$

where d_k, p_k, t_k are the numbers of doctors, paramedics, health technician of k-th station that is in the i-th hyperedge, and l is the cardinality of the hyperedges. The updated numbers can be found in (Int Kyn. 1).

To obtain more accurate results, we discard the hyperedge of the related triangle with the longest edges.

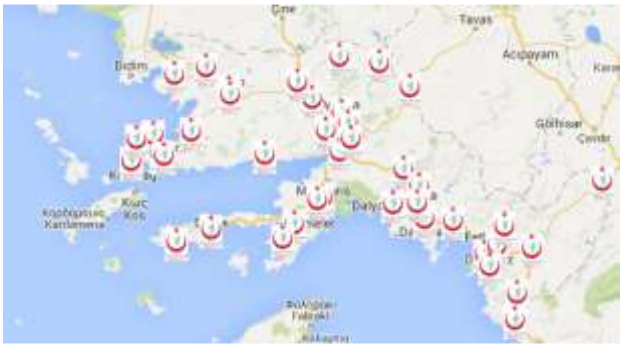

Figure 5.1.1. The map of 112 Emergency Health Service Stations of Muğla. The more detailed map can be seen in (Int Kyn. 1).

The corresponding neighborhood graph of 2D and 3D data sets are shown in Figure 5.1.2 and Figure 5.1.3, respectively.

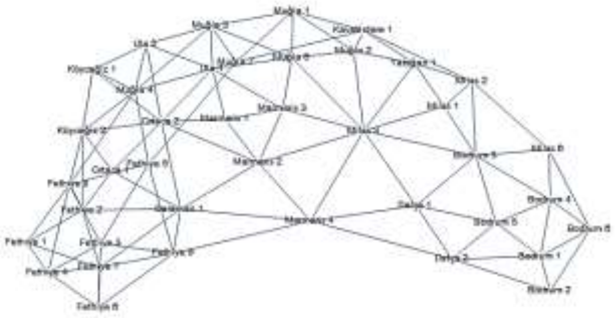

Figure 5.1.2. The neighborhood graph of 2D data sets that obtained from the latitude-longitude coordinates of 112 Emergency Health Service Stations.

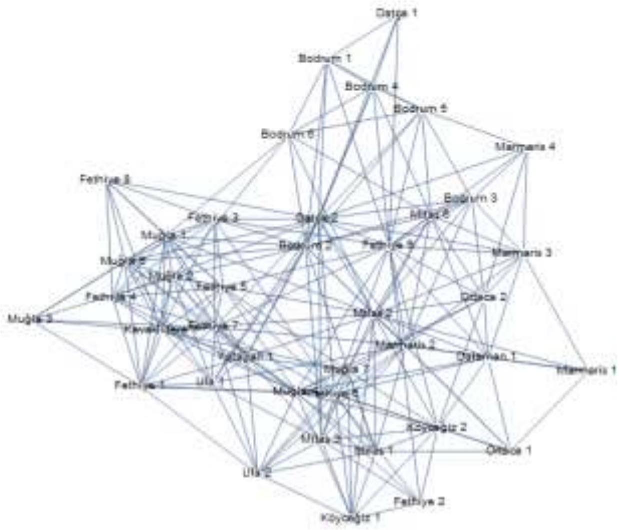

Figure 5.1.3. The neighborhood graph of 3D data sets that obtained from the latitude-longitude-altitude coordinates of 112 Emergency Health Service Stations.

As a result, the hyperedge cover of 2D data sets involves the stations

Fethiye 1, Fethiye 5, Fethiye 6, Dalaman 1, Köyceğiz 2, Marmaris 1, Datça 1, Ula 2, Muğla 6, Kavaklıdere 1, Milas 1, Milas 6, Bodrum 1,

and the hyperedge cover of 3D data sets involves the stations

Fethiye 2, Fethiye 3, Marmaris 1, Marmaris 4, Datça 1, Ula 1, Muğla 3, Milas 1, Milas 6

which are with the minimum cost.

6. Conclusions

We have presented an inventive method which solves GAP by using the hypergraph theory in this paper.

The corresponding hypergraph is basically obtained from the triangulation of the data set. Its hyperedges involve the interrelation of n-D data points. Henceforth, the method is also effective when a new dimension or a new agent is added to the tasks.

The hyperedge cover which is congruent to the maximal vertex cover of its neighborhood graph becomes solution for the purposed GAP. One of the greatest advantage of the method is it can be extended to any finite dimensional data sets by using the generalized triangulations. We also applied our method to a real data set of emergency service stations and obtained the consistent results. These results can be re-modeled respect to the updated data.

References

Avella, P., Boccia, M., Vasilyev, I., 2010. A computational study of exact knapsack separation for the generalized assignment problem. Computational Optimization and Applications, 45(3), 543—555.

Balcı, M. A., Atmaca, S. P., Akgüller, Ö., 2016. Hyperpath Centers. In Advanced Computational Methods for Knowledge Engineering, Vol 473, 129—137.

Cherng, J.S., Lo, M.J., 2001. A hypergraph based clustering algorithm for spatial data sets, Data Mining, ICDM 2001, Proceedings IEEE International Conference on, pp. 83—90. IEEE.

Dıaz, J. A., Fernández, E., 2001. A tabu search heuristic for the generalized assignment problem. European Journal of Operational Research, 132(1), 22—38.

Fisher, M. L., Jaikumar, R., 1981. A generalized assignment heuristic for vehicle routing, Networks, 11(2), 109—124.

Fisher, M. L., Jaikumar, R., Van Wassenhove, L. N., 1986. A multiplier adjustment method for the generalized assignment problem. Management Science, 32(9), 1095—1103.

Liu, L., Mu, H., Song, Y., Luo, H., Li, X., Wu, F., 2012. The equilibrium generalized assignment problem and genetic algorithm. Applied Mathematics and Computation, 218(11), 6526—6535.

Nauss, R. M., 2003. Solving the generalized assignment problem: an optimizing and heuristic approach. INFORMS Journal on Computing, 15(3), 249—266.

Özbakir, L., Baykasoğlu, A., Tapkan, P. 2010. Bees algorithm for generalized assignment problem. Applied Mathematics and Computation, 215(11) 3782—3795.

Privault, C., Herault, L., 1998. Solving a real world assignment problem with a metaheuristic. Journal of Heuristics, 4(4), 383—398.

Ross, G. T., Soland, R. M., 1975. A branch and bound algorithm for the generalized assignment problem. Mathematical programming, 8(1), 91—103.

Savelsbergh, M., 1997. A branch-and-price algorithm for the generalized assignment problem. Operations Research, 45(6), 831—841.

Sethanan, K., Pitakaso, R., 2016. Improved differential evolution algorithms for solving generalized assignment problem. Expert Systems with Applications, 45, 450—459.

Shtub, A., Kogan, K., 1998. Capacity planning by the dynamic multi-resource generalized assignment problem (DMRGAP). European Journal of Operational Research, 105(1), 91—99.

Subtil, R.F., Carrano, E. G., Souza, M .J., Takahashi, R. H., 2010. Using an enhanced integer NSGA-II for solving the multi objective generalized assignment problem. In Proceedings of the 2010 IEEE congress on evolutionary computation(CEC), 1—7, IEEE.

Yu, J., Tao, D., Wang, M., 2012. Adaptive hypergraph learning and its application in image classification, IEEE Transactions on Image Processing,, 21(7), 3262—3272.

Yu, J., Tao, D., Li, J., Cheng, J., 2014. Semantic preserving distance metric learning and applications, Information Sciences, 281, 674—686.

Zhang, H., Yu, J., Wang, M., Liu, Y., 2012. Semi-supervised distance metric learning based on local linear regression for data clustering. Neurocomputing, 93, 100—105.

Zhang, L., Gao, Y., Hong, C., Feng, Y., Zhu, J., Cai, D., 2014. Feature correlation hypergraph: exploiting high-order potentials for multimodal recognition. Cybernetics, IEEE Transactions on Cybernetics, 44(8), 1408—1419.

İnternet kaynakları

1-http://www.muglasm.gov.tr/sayfa/62/acil-saglik-hizmetleri-istasyonlari, (29.05.2016)

2-
http://people.brunel.ac.uk/~mastjjb/jeb/orlib/gapinfo.html, (16.01.2017)