

Algorithm for Fuzzy Maximum Flow Problem in Hyper-Network Setting (II)

Linli Zhu^{1*}, Yu Pan¹, Wei Gao²

¹School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China

²School of Information, Yunnan Normal University, Kunming 650500, China

*Corresponding Author:

Linli Zhu

Email: zhulinli@jsut.edu.cn

Abstract: Maximum flow problem on hypergraphs (hyper-networks) is an extension of maximum flow problem on normal graphs. In this report, we discuss a generalized fuzzy version of maximum flow problem in hyper-networks setting, and a new fuzzy maximum flow algorithm in hyper-network setting is obtained. Our algorithm mainly based on fuzzy set theory and incremental graph, and the technology of α -cut is also employed to determine the fuzzy maximum flow. The implement procedure is manifested at last.

Keywords: Hypergraph, maximum flow problem, incremental graph, α -cut.

INTRODUCTION AND NOTATIONS

There is an important component of graph theory and artificial intelligence which is Maximum flow problem of weighted graph and it shows extensive applications in various fields like: computer network, data mining, image segmentation and ontology computation [1-7]. Hyper-graph, a subset system for limited set, is one of the most general discrete structures, and is the generalization of the common graph. In terms of lots of practical problems, the usage of the concept of hyper-graph shows more effective than the concept of graph. Until now, we can see the applications of hyper-graph model in a variety of fields like: VLSI layout and electricity network topology analysis. And recently, intelligence algorithms and learning algorithms on hyper-graph and its computer applications are studied by researchers [8-17].

Let $V=\{v_1, v_2, \dots, v_m\}$ be a limited set, E be the family of subset of V , i.e., $E \subseteq 2^V$. Then $H=(V, E)$ is a hyper-graph on V . The elements of V and E are called a vertex and a hyper-edge respectively. Let $|V|$ be the order of H , $|E|$ be the scale of H . Then $|e|$ is the basic number of hyper-edge e . $r(H)=\max_j |e_j|$ is the rank of hyper-edge e , and $s(H)=\min_j |e_j|$ is the lower rank of hyper-edge e . If $|e|=k$ for each hyper-edge e of E (that is $r(H)=s(H)=k$), then H is a k -uniform hyper-graph. If $k=2$, then H is just a normal graph.

If any two hyper-edges are not contained by each other, a hyper-graph H can be called a simple hyper-graph or a sperner hyper-graph. Let $H'=(V, E')$ be a hyper-graph on V , then if $E' \subseteq E$, H' is a part-hyper-graph of H . For $S \subseteq V$, $H[S]=\{e \in E: e \subseteq S\}$ is called a sub-hyper-graph of H induced by S .

By using the set of vertices to represent the elements of V , the Hyper-graph H can be represented by graph. If $|e_j|=2$, a continuous curve which attaches to the elements of e_j is chosen to represent e_j ; If $|e_j|=1$, we use a loop which contains e_j to represent e_j ; If $|e_j| \geq 3$, we choose a simple close curve which contains all the elements of e_j to represent e_j .

In this paper, we suppose that H is a weighted hyper-graph, each edge is given a weight $w(e)$. The degree of vertex v_j in hyper-graph H is denoted as $\deg_j(H)=\sum_{e \in E} w(e)h(v, e)$, where $h(v, e)=\begin{cases} 1, & \text{if } v \in e \\ 0, & \text{if } v \notin e \end{cases}$.

Let $\delta(e) = \sum_{v \in V} h(v, e)$. Then, the normalized laplacian $L(H) \in \mathbb{R}^{m \times m}$ on hyper-graph H on hyper-graph H is defined by :

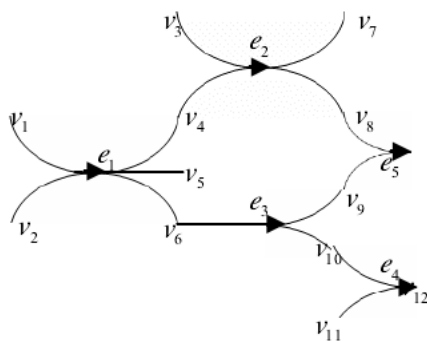
$$L_{ij}(H) = \begin{cases} -\sum_{i,j \in e} w(e) \frac{1}{\delta(e)} & i \neq j \\ \deg_j(H) & \text{otherwise} \end{cases}$$

Let $H=(V, E)$ be a fixed, directed and weighted hyper-graph with n vertices which express a hyper-network. Directed hyper-graph models can represent relationships among elements there in a lot of projects like large super-network research, database systems research, timing research, circuit design research and so on. According to its good application background, the directed hyper-graph theory has been a subject which develops rapidly in the field of graph theory.

Particularly, a directed hyper-graph is a hyper-graph where each hyper-edge is divided into two sets: $e=(X, Y)$ with $X \cap Y = \emptyset$ and X, Y can be the empty set. Here, X is called a tail point set and Y is called a head point set and they are denoted by $T(e)$ and $H(e)$ respectively. Similar to the undirected hyper-graph, we can define the hyper-road, hyper-path, hyper-cycle in the directed hyper-graph in directed hyper networks.

A $\{-1,0,1\}$ incidence matrix is introduced to represent the directed hyper-graph. The j -th column expresses the j -th vertex v_j and i -th row expresses the i -th hyper-edge $e_i : [a_{ij}]_{m \times n} = \begin{cases} -1, & v_i \in T(e_j) \\ 1, & v_i \in H(e_j) \\ 0, & \text{otherwise} \end{cases}$.

What is followed below is an example of the directed hyper-graph and its incidence matrix:



	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}
e_1	-1	-1	0	1	1	1	0	0	0	0	0	0
e_2	0	0	-1	-1	0	0	1	1	0	0	0	0
e_3	0	0	0	0	0	-1	0	0	-1	1	0	0
e_4	0	0	0	0	0	0	0	0	0	-1	-1	1
e_5	0	0	0	0	0	0	0	-1	-1	0	0	0

Actually, some uncertain factors which cannot be expressed by fixed functions or parameters exist in many hyper-networks applications. Hence, the theory shows extensive applications in networks and hyper-networks [18-22]. In this paper, the fuzzy maximum flow problem in hyper-networks is considered in detail. Then the new optimization model is presented by means of the fuzzy set theory and incremental graph.

SETTING

In this section, we introduce several famous concepts of fuzzy theory. A trapezoidal fuzzy number is denoted as a $\tilde{a} = (m_1, m_2, \delta, \beta)$, and it has a membership function defined by

$$\mu_{\alpha}(x) = \begin{cases} 0, & \text{if } x \leq m_1 - \delta \\ \frac{x - m_1 + \delta}{\delta}, & \text{if } m_1 - \delta < x < m_1 \\ 1, & \text{if } m_1 \leq x \leq m_2 \\ \frac{m_2 + \beta - x}{\delta}, & \text{if } m_2 < x < m_2 + \beta \\ 0, & \text{if } x \geq m_2 + \beta \end{cases},$$

where m_1 is left extremities of the modal value, m_2 is the right extremities of the modal value, δ is the left spread, and β is the right spread. The terms $m_1 - \delta$ and $m_2 + \beta$ expressed the lower and upper bounds respectively. Modal value is the largest value $x \in [m_1 - \delta, m_2 + \beta]$ of the membership function.

Let A be a fuzzy set, then the α -cut of A denoted by A_{α} is a collection consisting of those members of the generalized X whose membership values outnumber the threshold level α , $A_{\alpha} = \{x | A(x) \geq \alpha\}$.

In rich applications of maximum flow problem in hyper-network setting, the hyper-arcs capacities only have the upper bounds. Then, the capacities of the hyper-arcs discussed in such situations, possess their lower bounds $m_1 - \delta$ and their inferior extremities of the modal interval m_1 equal zero. Therefore, these capacities are denoted as trapezoidal fuzzy numbers with $m_1 - \delta = m_1 = 0$.

By analyzing the fuzzy hyper-arc capacity and trapezoidal membership function, if the hyper-flow in this hyper-arc $\leq m_2$, its degree of satisfaction is 1. When the flow is between m_2 and $m_2 + \beta$, its satisfaction degree is between 1 and zero. That is to say, it is partially meeting the restriction of capacity of that hyper-arc. From this point of view, we implemented a heuristic, relied on α -cuts [23-28]. This heuristic alternates this fuzzy problem into a crisp problem, through which the experts (decision maker) can select the minimum satisfaction degree of the last solution.

ALGORITHM

In this section, we first present the problem and notations of the maximum flow problem, the fuzzy maximum flow problem in hyper-networks setting from mathematical point of view, and the whole implementation.

Consider a directed flow hyper- network $G=(V, E, C)$, where V implies the finite set of vertices, which is denoted by the number $\{1, 2, \dots, n\}$. E expresses the set of the directed hyper-edge, and each directed hyper-edge e is denoted by an ordered pair $(G(e), G(e))$, where $e \in E$. C represents the set of directed hyper-edge capacities. In the fuzzy maximum flow problem in hyper-networks setting, every directed hyper-edge e has a nonnegative, independent, fuzzy flow capacity ξ_e with the membership functions μ_e . Then, for each pair of vertices (v_i, v_j) , we use $\xi_{ij} = \sum_{\{v_i, v_j\} \subseteq e \in E} \xi_e$

to denote its fuzzy flow capacity associated with certain membership functions μ .

In what follows, flow representation is employed by: $x = \{x_{ij} = \sum_{\{v_i, v_j\} \subseteq e \in E} x_e\}$

where x_e denotes the flow of directed hyper-edge (hyper-arc) e . The flow is called a feasible flow in hyper-networks setting if the following two conditions are established:

- (1) For each vertex, the outgoing flow and incoming flow must meet the following balance conditions.

$$\begin{cases} \sum_{\{v_i, v_j\} \subset e \in E} x_{1j} - \sum_{\{v_j, v_i\} \subset e \in E} x_{j1} = f \\ \sum_{\{v_i, v_j\} \subset e \in E} x_{ij} - \sum_{\{v_j, v_i\} \subset e \in E} x_{ji} = 0, \quad 2 \leq i \leq n-1 \\ \sum_{\{v_n, v_j\} \subset e \in E} x_{nj} - \sum_{\{v_j, v_n\} \subset e \in E} x_{jn} = -f \\ \forall e \in E \end{cases}$$

in which f denotes the flow of the hyper-network G .

(2) The flow at each directed hyper-edge must be satisfied by the capacity constraint. Clearly,

- (i) $x_{ij} \geq 0$ (non-negative flow);
- (ii) $x_{ij} \leq \xi_{ij}$ (capacity restrictions);

Having fixed a hyper-path between s and t in a hyper-graph G , the maximum flow can be sent of s to t for the hyper-path p , meeting the following restrictions: $\min\{\xi_{ij} : (i, j) \in e \in p\}$.

Therefore, the hyper-arc of p is called saturated if it has the smallest capacity.

Thus, class of problem can be determined if we use the two different tricks: the method of the incremental hyper-graph and of the minimum cuts. The method of the incremental hyper-graph searches all hyper-paths between the source vertex and destination vertex, and it analyzes the capacities of each hyper-arc of these hyper-paths and deliveries the possible maximum flow. The second technology consists of two parts, V_1 and V_2 , from the vertices where the source vertex s , belongs to the V_1 and the destination vertex t belongs to V_2 . This method searches all the possible cuts of the hyper-graph. The smallest cut capacity is the maximum flow, where the cut capacity is the addition of the hyper-arcs capacities that have the source vertex in V_1 and the destination vertex in V_2 .

In our algorithm, the experts (decision makers) should decide the flow which needs possible large but at the same time can't infringe excessively the hyper-arcs capacities limitations. Then, the decision maker needs to determine the membership function for the flow:

$$\mu_s(y) = \begin{cases} 1, & \text{if } y > y_0 \\ L(y_1, y_0; y), & \text{if } y_1 \leq y \leq y_0 \\ 0, & \text{if } y < y_1 \end{cases}$$

where y_0 is the minimum ideal flow; y_1 is the acceptable minimum flow. $L(y_1, y_0; y)$ is a fixed linear function with $L(y_1, y_0; y_0) = 1$ and $L(y_1, y_0; y_1) = 0$. Thus, the degree of the goal realization at the flow value y is determined by the membership function value $\mu_s(y)$.

As presented in above section, for each capacity limitation $x_{ij}^y \leq \xi_{ij}$ ($x_{ij}^y = \{x_{ij} : (i, j) \in A, x_{ij} \geq 0\}$) a satisfaction function is associated:

$$\mu_{ij}(x_y) = \begin{cases} 1, & \text{if } x_{ij}^y > \xi_{ij} \\ \bar{L}(\xi_{ij}, \bar{\xi}_{ij}; x_{ij}^y), & \text{if } \xi_{ij} \leq x_{ij}^y \leq \bar{\xi}_{ij} \\ 0, & \text{if } x_{ij}^y < \bar{\xi}_{ij} \end{cases}$$

where $\mu_{ij}(x_y)$ is the satisfaction degree of the fuzzy capacity restriction in view of the hyper-arc flow x_{ij} (here, ξ_{ij} and $\bar{\xi}_{ij}$ are defined by the fuzzy hyper-arc capacity).

Hence, in terms of the concept of decision making in fuzzy restrictions the minimum flow problem in hyper-networks setting reduces to get the flow x_y maximizing the membership function $\mu_D: \mu_D(x_y) = \mu_C(x_y) \wedge \mu_S(x_y)$,

$$\mu_C(x_y) = \bigwedge_{\{i,j\} \subseteq e \in E} \mu_{ij}(x_{ij})$$

where $\mu_C(x_y)$ is the degree of simultaneous satisfaction of all hyper-arc capacity constraints in the hyper-network by means of the flow x_y . The flow maximizing the membership function μ_D is called the maximal flow in the situation so that it maximizes the flow value. The flow with the largest value in the collection of the maximal flows is then called optimal.

Let V be the set of vertex; $\bar{\alpha}$ be the minimum degree of satisfaction which is decided by experts; H be the partition number of the interval $[\bar{\alpha}, 1]$; h be the scale of subinterval $[\alpha_k, \alpha_{k+1}]$; $(\tilde{c}_{ij})_\alpha = \sum_{\{i,j\} \subseteq e \in E} (\tilde{c}_e)_\alpha$ be the capacity value of the hyper-arc e contains $\{i,j\}$ with satisfaction degree α ; (t, s) be the artificial hyper-arc that joins the destination vertex t to source vertex s ; $(x_{ij})_\alpha = \sum_{\{i,j\} \subseteq e \in E} (x_e)_\alpha$ be the flow in the hyper-arc contain $\{i,j\}$ with membership α ; T be the hyper-circle obtained by the union between the hyper-path (between two vertices s and t) and the artificial hyper-arc (t, s) ; Γ be the hyper-cycle yielded between the hyper-chain (between the vertices s and t) and the artificial hyper-arc (t, s) ; Γ^+ be the hyper-arcs of Γ with the uniform direction of (t, s) ; Γ^- be the hyper-arcs of Γ with the opposite direction of (t, s) .

Now, we present the main algorithm in our paper. This algorithm is relied on the following implements: the first procedure is attributed to the number of α -cuts partitions and the minimum degree of membership. And the procedure A2 uses a heuristic by means of α -cuts. The initial flow is manifested in procedure A3. The procedure A4 applies an algorithm to search a hyper-path between the vertex s and vertex t . The procedure A5 shows the probability of an augmenting hyper-chain. At last, the algorithm is finished in procedure A6.

Fuzzy Maximum Flow Algorithm A in Hyper-network Setting

A1: Initialization (attribution of the minimum satisfaction degree and the number of partition for α -cuts). Fixed H and $\bar{\alpha}$.

A2: Using α -cuts in the capacities. For all α execute the A3, A4 and A5, where

$$\alpha = \bar{\alpha} + ih \ (h = \frac{1 - \bar{\alpha}}{H - 1}, i = 0, \dots, H-1) \text{ and } (\tilde{c}_{ij})_\alpha = \sum_{\{i,j\} \subseteq e \in E} (\tilde{c}_e)_\alpha.$$

A3: Attribution of the initial flow. For each pair of vertices (i, j) , $(x_{ij})_\alpha \rightarrow 0$.

A4: Determination of the hyper-paths between vertices s and t . Repeat this A4 until we can't find any hyper-path between s and t . Specifically, search a hyper-path p between the vertices s and t , do: determine the hyper-cycle T ; contain the flow f in the hyper-cycle; $f = \min_{\{i,j\} \subseteq T} [(\tilde{c}_{ij})_\alpha - (x_{ij})_\alpha]$; at last, we delete the saturated hyper-arcs.

A5: Determination of a hyper-chain. If there exist a chain between vertices s and t , do: determine $\delta = \min\{\delta_1, \delta_2\}$ where

$$\delta_1 = \min_{(i,j) \subseteq \Gamma^+} \sum_{\{i,j\} \subseteq e \in E} [(\tilde{c}_e)_\alpha - (x_e)_\alpha], \delta_2 = \min_{(i,j) \subseteq \Gamma^-} \sum_{\{i,j\} \subseteq e \in E} (x_e)_\alpha. \text{ If } \delta > 0, \text{ do:}$$

$$\forall (i, j) \in \Gamma^+ \Rightarrow \sum_{\{i,j\} \subseteq e \in E} (x_e)_\alpha \leftarrow \sum_{\{i,j\} \subseteq e \in E} (x_e)_\alpha + \delta$$

$$\forall (i, j) \in \Gamma^- \Rightarrow \sum_{\{i,j\} \subseteq e \in E} (x_e)_\alpha \leftarrow \sum_{\{i,j\} \subseteq e \in E} (x_e)_\alpha - \delta$$

A6: End

CONCLUSION

In this report, we discuss the fuzzy maximum flow problem in hyper-networks setting. Our algorithm is designed based on fuzzy set theory and incremental graph. The result achieved in our paper illustrates the promising application prospects for algorithms using hypergraph model.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

ACKNOWLEDGMENT

We thank the reviewers for their constructive comments in improving the quality of this paper. We also would like to thank the anonymous referees for providing us with constructive comments and suggestions. We wish to acknowledge the National Natural Science Foundation of China (61142007) and the general program of Jiangsu University of Technology in China (KYY14013).

REFERENCES

1. Hafezalkotob A, Ahmad AM; Cooperative maximum-flow problem under uncertainty in logistic networks. *Applied Mathematics and Computation*, 2015; 250: 593-604.
2. SullivanKM, SmithJC; Exact algorithms for solving a euclidean maximum flow network interdiction problem. *Networks*, 2014; 64(2): 109-124.
3. Han SW, Peng ZX, Wang SQ; The maximum flow problem of uncertain network. *Information Sciences*, 2014; 265: 167-175.
4. Amankwah H, Larsson T, Textorius B; A maximum flow formulation of a multi-period open-pit mining problem. *Operational Research*, 2014; 14(1): 1-10.
5. Gamst M; A local search heuristic for the multi-commodity k-splittable maximum flow problem. *Optimization Letters*, 2014; (8)3: 919-937.
6. Dressler D, Strehler M; Polynomial-time algorithms for special cases of the maximum confluent flow problem. *Discrete Applied Mathematics*, 2014;163: 142-154.
7. Illes T, Molnar-Szipai R; On strongly polynomial variants of the MBU-simplex algorithm for a maximum flow problem with non-zero lower bounds. *Optimization*, 2014; 63(1): 39-47.
8. Zhou D, Huang J, Scholkopf B; Beyond pair-wise classification and clustering using hypergraphs, Canada: University of Waterloo, 2005.
9. Zhou D, Huang J, Scholkopf B; Learning with hypergraphs, clustering, classification, and embedding, Proceedings of 20th Annual Conference on Neural Information Processing Systems, 2006, Vancouver / Whistler, Canada: IEEE, 2006; 1601-1608.
10. Liang S, Ji S, Ye J; Hypergraph spectral learning for multi-label classification, Proceeding of the 14th ACM SIGKDD international conference on Know ledge discovery and data mining. Las Vegas, Nevada, USA: ACM, 2008; 668- 676.
11. Chen G, Zhang J, Wang F, Zhang C, Gao Y; Efficient multi-label classification with hypergraph regularization. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on 2009; 1658-1665. IEEE.
12. Yan L, Li JS, Gao W; Notes on vertex classification problem in hypergraph setting. *Information Technology Journal*, 2013; 12(15): 3309-3314.
13. Zhu LL, Gao W; Analysis of classification algorithm on hypergraph. *The Open Cybernetics & Systemics Journal*, 2014; 8: 122-127.
14. Zhu LL, Gao Y, Gao W; Arriving and convert times in hyper-networks. *Scholars Journal of Physics, Mathematics and Statistics*, 2014; 1(2):74-80.
15. Gao W, Liang L; Ontology concept similarity computation based on regularization framework of hypergraph. *Microelectronics and computer*, 2011; 28(5): 15-17.
16. Tisza D, Olah A, Levendovszky J; Novel algorithms for quadratic programming by using hypergraph representations. *Wireless Personal Communications*, 2014; 77(3): 2305-2339.
17. Kostyanoi DS, Mokryakov AV, Tsurkov VI; Hypergraph recovery algorithms from a given vector of vertex degrees. *Journal of Computer and Systems Sciences International*, 2014; 53(4): 511-516.
18. Wang J, Wang CH, Chen CLP; The bounded capacity of fuzzy neural networks (FNNs) via a new fully connected neural fuzzy inference system (F-Confis) with its applications. *IEEE Transactions on Fuzzy Systems*, 2014; 22(6): 1373-1386.
19. Turan HH, Kasap N, Savran H; Using heuristic algorithms for capacity leasing and task allocation issues in telecommunication networks under fuzzy quality of service constraints. *International Journal of Systems Science*, 2014; 45 (3): 307-324.
20. Aminnayeri M, Biukaghazadeh P; Robust clustering approach for the shortest path problem on finite capacity fuzzy queuing network. *International Journal of Advanced Manufacturing Technology*, 2012; 61(5-8): 745-755.

21. Soudmand S, Ghatte M, Hashemi SM; SA-IP method for congestion pricing based on level of service in urban network under fuzzy conditions. *International Journal of Civil Engineering*, 2013; 11(4A): 281-291.
22. Havens TC, Bezdek JC, Leckie C, Ramamohanarao K, Palaniswami M; A soft modularity function for detecting fuzzy communities in social networks. *Fuzzy Systems, IEEE Transactions on*, 2013; 21(6):1170-5.
23. Yin S, Zhao X, Wang W, Gong M; Efficient multilevel image segmentation through fuzzy entropy maximization and graph cut optimization. *Pattern Recognition*, 2014; 47(9):2894-907.
24. Mockor J; Isomorphisms and functors of fuzzy sets and cut systems. *Soft Computing*, 2014; 18(7): 1237-1245.
25. Lee G, Jun KS, Chung ES; Robust spatial flood vulnerability assessment for Han River using fuzzy TOPSIS with α -cut level set. *Expert Systems With Applications*, 2014; 41(2): 644-654.
26. Torkjazi M, Fazlollahtabar H, Mahdavi I; Optimizing an unconstrained multi objective model using a utility based fuzzy probabilistic α -cut method. *Journal of Intelligent & Fuzzy Systems*, 2014; 26(6): 2927-2936.
27. Martinetti D, Janis V, Montes S; Cuts of intuitionistic fuzzy sets respecting fuzzy connectives. *Information Sciences*, 2013; 232: 267-275.
28. Ayag Z, Ozdemir RG; Evaluating machine tool alternatives through modified TOPSIS and α -cut based fuzzy ANP. *International Journal of Production Economics*, 2012; 140(2): 630-636.