A New Fuzzy Graph Model for QAP Problem

M. R. Gholamian and S. M. T. Fatemi Ghomi
Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran

Abstract—**Quadratic assignment problem (QAP) is one of the most eminent problems in facility location. This problem is known as NP-hard and so very difficult to be solved.**

In this paper, a new vision to this problem is developed by introducing graph model (i.e. tree graph model) of QAP problem with specific features. The graph model is defined with fuzzy vertices instead of fuzzy edges which is named "FuzTree" and is related to fuzzy inference engines. Hence, fuzzy inference engines can be applied for QAP problem solution based on this fuzzy graph model.

Keywords—**Quadratic assignment problem, Graphs and digraphs, Trees, Fuzzy inference engines.**

I. INTRODUCTION

Assignment of different facilities to different locations has been a problem area for the most industries. Dependence of facilities in material flow leads us to consider the problem of assigning when there is an interchange between pairs of new facilities. This problem is closely related to finite multifacility location problem and then is referred to as quadratic assignment problem (QAP), the model that was first developed by Koopmans and Backmann [1] with the following mathematical structure:

$$\text{Min} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{p=1}^{n} \sum_{q=1}^{n} c_{ipjq} V_{ip} V_{jq}$$

Subject to:

$$\sum_{i=1}^{n} V_{ip} = 1 \qquad p=1, 2\ldots n$$

$$\sum_{p=1}^{n} V_{ip} = 1 \qquad i=1, 2\ldots n$$

$$V_{ip} = 0,1 \qquad i=1,2,\ldots n \;\&\; p=1,2,\ldots,n$$

where C_{ipjq} is transportation cost when facility #i is assigned to location #p and facility #j is assigned in location #q concurrently and V_{ip} is defined as follows:

$$V_{ip} = \begin{cases} 1 & \text{if facility \#i is assigned to location \#p} \\ 0 & \text{otherwise} \end{cases}$$

In this model, the first constraint ensures that each location accepts only one facility and the last one, ensures that each facility is assigned to only one location.

Consequently, QAP structure is similar to assignment problem in constraints and differs in objective function with nonlinear (quadratic) structure. This makes the problem NP-hard with exponential time complexity which is very difficult to be solved; even for moderate size problems. There is various solution methods developed for this problem. While some researchers concentrate on exact methods [2], [3], most of the other works are performed using Meta heuristics; genetic algorithm [4], simulated annealing [5], tabu search [6], ant colony [7] and hybrid methods [8]. Meanwhile neural networks are also used as heuristic methods [9], [10].

In this paper, a new vision to this problem is suggested by developing the graph theory model of QAP problem with fuzzy vertices to demonstrate application of fuzzy inferencing system in solving such NP-hard problems.

The paper is organized as follows. In the next section the graph theory of QAP problem is introduced. It is shown that this graph is a type of directed tree. Section 3 is assigned to definition of fuzzy vertices and its application in QAP tree as new fuzzy type graph (i.e. FuzTree). The final section is devoted to concluding remarks and some recommendations for future studies.

II. GRAPH THEORY MODEL

Let suppose a QAP problem with 5 machines and 5 locations. Based on VNZ [11] method the solutions can be developed as a string, such as $[M_1, M_2, M_5, M_4, M_3]$ or simpler [1,2,5,4,3]. This is a feasible solution of QAP problem which can be demonstrated as x = [1,2,5,4,3] with total related cost f(x) resulted from objective function. Then any feasible solution of QAP problem can be defined as follows:

Definition 1: Each feasible solution of QAP problem can be demonstrated as a string x such that any element shows facility and any number of places for each facility shows location. Each string x provides related cost value on objective function f(x). So any feasible solution of QAP problem can be introduced by couple [x, f(x)] solutions.

0-7803-8519-5/04/$20.00 © 2004 IEEE

Now, suppose pairwise interchanges between two adjacent elements of (string) x; then new feasible solution of QAP problem is obtained. For example interchanging 2, 5 elements causes new feasible solution x' = [1,5,2,4,3].

Let define this new solution [x', f(x')] as neighbor of current solution [x, f(x)] as follows:

Definition 2: Neighbor of x is a solution that is obtained by any pairwise interchange in two adjacent elements of string x.

Obviously, such solutions are not unique and any interchange in string elements caused new feasible solution. For example x″ = [2,1,5,4,3] and x‴= [1,2,5,4,3] are other neighbors of x.

Now let's suppose the set of these neighbors as N(x) such that N(x) = {x' , x″ , x‴ …}. This set and its elements guide to graph concepts with the vertices of N(x) and edges which describe neighborhoods:

Definition 3: G=(V,E) is a graph model of QAP solution such that:

- Each vertex demonstrates a string solution of QAP problem: $V=\{x_1, x_2 \ldots\}$.
- Each edge, explains neighborhood of two vertices: $\{x, x'\} \in E$; $x' \in N(x)$.

Example 1: Suppose a problem with three facilities that must be located in three locations. Then based on above definitions the graph model of this problem would be (f(x) is represented as values above each vertex):

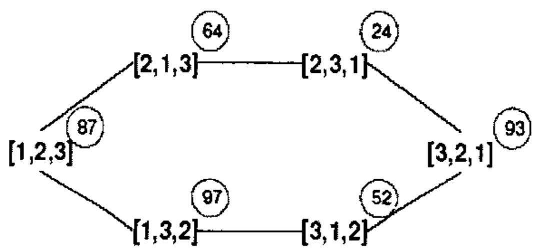

Fig. 1. Graphical representation of a QAP problem

Theorem 1: G=(V,E) is connected.

Proof: Suppose G is disconnected. Then G must be contained at least two components and one component must at least have one vertex. Based on definition 2, this vertex is related to neighbors; therefore related (neighbor) vertex must be added to this component. Each of these new vertices similarly has neighbors that their related vertices must be added in this component. Paying attention that all permutations of a sequence can be obtained by consecutive pairwise interchange of elements [12], all of vertices will be added to this component one by one and then another component becomes empty. Then G will have only one component and therefore is connected [13].□

Theorem 2: For a problem with n facilities, G=(V,E) is (n-1)-regular.

Proof: For a problem with n facilities, each vertex (string solution) is an n elements string. Based on neighborhood definition, each pairwise interchange of adjacent elements caused a new neighborhood that means an edge of the vertex. In an n elements string, (n-1) adjacent elements can be defined. Hence, each vertex has (n-1) edges and then G is (n-1)-regular. □

Example 2: Consider previous example with n = 3. Then the resulted graph is 2-regular (or specially a cycle) such as shown in Figure 1.

Corollary 2.1: G=(V,E) is a simple graph and does not contain an isolated vertex.

Corollary 2.2: For a problem with n facilities, degree of each vertex is n-1 ($\deg(v) = n-1 \ \ \forall v \in V$).

Theorem 3: Graph G=(V,E) that is related to a QAP problem with n facilities has n! vertices and $\frac{n-1}{2} \times n!$ edges.

Proof: Based on probability concepts, we can set n numbers into n places, in n! different ways. Then each QAP problem with n facilities has n! feasible solutions (x's). In fact G includes n! vertices (|V| = n!). On the other hand, G is (n-1)-regular. Then:

$$2|E| = \sum_{v \in V} \deg(v) = (n-1)\, n! \quad \Rightarrow \quad |E| = \frac{n-1}{2} \times n!\ .\ \square$$

The above theorems provide some specification of QAP graph. In addition each vertex (i.e. each feasible solution x) of this graph is evaluated with a value f(x) such as shown in Figure 1.

So G=(V,E) can be directed [14] based on these values and type of objective function (max/min) as follows:

Definition 4: D=(V,E) is a directed graph of QAP solution problem such that arrow direction of arcs is from smaller values to larger values of vertices for minimization problem and conversely from larger values to smaller values of vertices for maximization problem. If two or more vertices have equal values, their directions are optional such that no cycle is created.

Example 3: Let suppose example 1 again. Now, based on definition 4, the digraph of example 1 can be represented as follows:

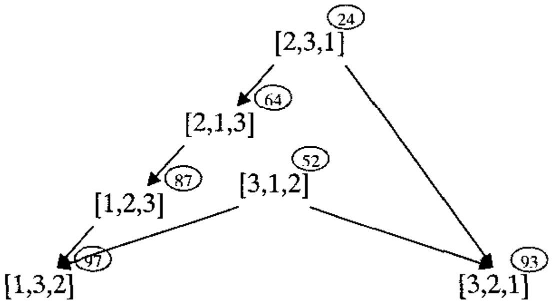

Fig. 2. Directed graph of a QAP problem

Theorem 4: Digraph D=(V,E) is acyclic.

Proof: Suppose, D is cyclic and for any $[a_0, f(a_0)]$, $[a_1, f(a_1)]$, ..., $[a_n, f(a_n)] \in V$, there is a cycle that starts at a_0 and also terminates at a_0 in the shortest length; then based on the above definition we will have:

$f(a_0) \leq f(a_1) \leq \ldots \leq f(a_n) \leq f(a_0)$.

But this relation is correct only in equal conditions such that:

$f(a_0) = f(a_1) = \ldots = f(a_n) = f(a_0)$

which is a cycle of equal values and so is incorrect based on definition 4.□

Such as shown in Figure 2, there are some pendant vertices in this digraph, which are the best and the worst solutions of QAP problem.

Theorem 5: Digraph D=(V,E) has at least two pendant vertices.

Proof: It has been proved in mathematical programming that any quadratic programming model is optimized in at least one global solution [15]; although the model may include some local solutions aside with this global solution. Based on definition 4, the vertices related to these solution points are defined without any vertices arriving to them; since they are minimum vertices. In fact, V is a global / local solution if and only if $deg^-(V) = 0$; although $deg^+(V) \neq 0$. Now, suppose the opposite situation, such that for a digraph D=(V,E) with n vertices $a_1, a_2, \ldots, a_n \in V$, we have $deg^-(a_i) \neq 0$ $\forall i$. Then because of existence a trail between each two a_i and a_j vertices, all vertices are related to each other. This is a circuit which is incorrect based on previous theorem.

Similar result can be extended to the worst case solutions where for any worst vertex v, $deg^+(v) = 0$ and $deg^-(v) \neq 0$ and then such v vertices can be known as other pendant vertices of QAP graph. Therefore, digraph D=(V,E) has at least two pendant vertices related to the best and worst QAP solutions.□

Based on the above theorems, these digraphs are very similar to tree graphs with roots of general/local solutions and leaf of the worst case solutions.

Theorem 6: Graph D=(V,E) is a directed tree.

Proof: D=(V,E) is connected because its underlying graph G=(V,E) is connected based on theorem 1. On the other hand D=(V,E) is acyclic based on theorem 5. So D=(V,E) is a directed tree (i.e. T=(V,E)).□

Corollary 7.1: T=(V,E) related to QAP problem is spanning tree.

Corollary 7.2: If QAP problem has only one global solution, then its related tree is rooted tree and if some local solutions also exist, its related tree is multi rooted tree.

The rooted trees are used in various applications such as computer algorithms, coding systems, telecommunications and so on, but perhaps one of the most important one, is successful applications such graphs in artificial intelligence [16]. In rule base inference engines, in order to obtain a problem solution, rules are activated based on initial facts and inductive results of other rules. This inductive procedure is repeated inasmuch as the root solution is obtained. The procedure is quite similar of finding rooted order of spanning tree in graph theory. Even similar search algorithms such as Breath-first search (BFS) and Depth-first search (DFS) [17] and so on, are used in two cases.

In order to reach this similarity, let suppose graph interpretation of "If...Then" rules such that the antecedents and consequents of the rules are defined as vertices which are related to each other using rule definition. Hence a knowledge base can be represented as rooted tree of the rules. Now in problem solution process, by activating an antecedent of a rule, its consequent is also activated which caused itself (or with any other results) activating the antecedents of the other rules. In fact a vertex is activated if all related arriving vertices are activated. This is the application of graph theory in artificial intelligence; specially in rule-based reasoning.

Similarly QAP tree T=(V,E) can be also explained as "If...Then" rules. In fact, knowledge base of QAP solutions can be developed as a rule base. But this rule base is not accessible; since the rules describe the neighborhood of the solutions. In addition, with assumption of existence such rule base, this system is very difficult to be used. This is the problem which is solved in the next section by offering new definitions.

III. FUZZY VERTICES

Now let get new vision. In the previous section, each vertex is activated in true or false, based on classical modus ponens. In fact, each vertex can be evaluated with only two values; zero or one. But this evaluation can be expanded to the all of values in zero and one interval. Then each vertex can be evaluated based on "degree of compatibility" instead of quite correctness/incorrectness crisply. This is a new concept oriented from generalized modus ponens, such as shown in the following figure.

A	A′
$\underline{A \rightarrow B}$	$\underline{A \rightarrow B}$
B	B′
Classical Modus Ponens	Generalized Modus Ponens

Fig. 3. Classical versus generalized modus Ponens

Such as shown in Figure 3, in classical modus ponens complete fitness is needed in order to activate a vertex (rule). But in generalized modus ponens, degree of similarity (i.e. A′ for A) activates vertex (rule) and gives results with degree of truthness (i.e. B′ instead of B). This is the same fuzzy logic concept [18] which can be used in definition of directed tree graphs:

Definition 5: V is fuzzy vertex if its truthness value is evaluated based on degree of truthness instead of crisp truthness or falseness.

Definition 6: T_f=(V,E) is *FuzTree* graph if the tree contains at least one fuzzy vertex.

In global vision, the truthness degree of a vertex is not necessarily restricted to zero-one intervals and each other interval is accepted based on universe of discourse of the problem [19]. Specially in QAP problem the estimations of worst case and best case are good candidates to constitute such interval.

Now using above definition, the QAP graph T=(V,E) can be suggested as direct FuzTree graph T_f=(V,E) such that each vertex is evaluated in the range of estimating worst and best solutions.

Similar to the previous section, which direct spanning trees are related to rule bases, direct FuzTree graphs can be related to fuzzy rule bases. Such as fuzzy inference engines that input facts affect antecedence of the rules as a matter of degree of truthness and then they activate consequent parts fuzzily, vertices are also activated as a matter of truthness and activate other related vertices fuzzily which are aggregated in rooted vertices. Unlike classical rule bases, fuzzy rule bases have more robust structure. Fuzzy inference engines are program intensive systems which use parallel processing in numerical approaches. This is the reason of simple application of such systems.

This similarity is also right for T_f=(V,E) related to QAP problem. In fact the knowledge base of QAP solutions can be defined fuzzily. Unlike classical knowledge, this knowledge is accessible and can be simply extracted from the set of previous data or from the experts. Meanwhile the designed rules can be clustered to provide rapid and effective rule base.

So, in order to solve QAP NP-hard problem, an intelligent system (fuzzy rule-based reasoning system) can be used instead of classical mathematical programming solution methods. It is sufficient to generate suitable estimation data set and then fuzzy rules can be extracted using knowledge acquisition process. Specially neural networks can be used in this process; since these computational intelligences are very powerful in knowledge acquisition [20]. Along with this process, rules can be extracted from the experts by defining a suitable set of linguistic values and variables [19] in a range of universe of discourse. Then an expert can develop fuzzy rules according to these linguistic terms. In addition fuzzy rule base will not be great. In the case of abundant rules, the rules can be clustered to provide powerful and effective rule base.

IV. CONCLUDING REMARKS

This paper is oriented on new vision to QAP solution methods. Since QAP is NP-hard problem, there are various exact and heuristic methods developed for this problem.

But in this paper, another vision to QAP problem is suggested with explanation of QAP solutions as vertices of a graph theory model (i.e. G=(V,E)). It is shown that this graph can be directed based on related cost values (i.e. f(x)). In addition, the digraph D=(V,E) can be concluded to a directed tree T=(V,E) with special specifications (i.e. rooted tree).

It is also shown that rooted tree is the same graph representation of rule bases in intelligent systems and hence QAP rooted trees can be interpreted as rule base of QAP solutions. But this knowledge base is dummy which is not accessible. This problem is solved using new definition of fuzzy vertices and FuzTree T_f=(V,E). QAP FuzTree can be interpreted as fuzzy rule base of QAP solutions. This is the rule base that can be extracted from suitable data set of solutions in the well-designed knowledge acquisition process using simple multilayer feedforward neural networks. Meanwhile, the rules can be directly extracted from the experts. Since the fuzzy rule bases are numerical approaches with parallel processing, their computational costs are not much and the system uses less experience hardware. Besides, the rules can be clustered to provide stronger rule base.

Obviously, this new concept differs from traditional concepts of fuzzy graphs which the graphs are fuzzified on the edges. So, the comparison of strengths and weaknesses of each method and their applications in various scopes (specially in artificial intelligence), can be known as suitable recommendation for future studies.

As another recommendation FuzTrees can be used in other NP-hard problems such as traveling salesman problem (TSP), job shop and flow shop scheduling, knapsack problem and so on. On the other hand, fuzzy inference engines can be suggested to solve some NP-hard graph problems such as graph coloring, graph traversal, graph partitioning, planarity and matching with suitable changes.

REFERENCES

[1] T. Koopmans, and M. Beckmann, "Assignment problems and the location of economic activities", Econometrica, vol. 25, pp. 53-76, 1957.

[2] K. Anstreicher, "Recent advances in the solution of quadratic assignment problem", Math. Program., vol. 97, no. 1-2, ser. B, pp. 27-42, 2003.

[3] K. Anstreicher, N. Brixius, J. P. Goux, and J. Linderoth, "Solving quadratic assignment problems on computational grids", Math. Program, vol. 91, no. 3, ser. B, pp. 563-588, 2003.

[4] A. Misevicius, "Genetic algorithm hybridized with ruin and recreate procedure: application to the quadratic assignment problem", Knowledge-Based Systems, vol. 16, no. 5-6, pp. 261-268, 2003.

[5] P. Tian, J. Ma, and D.M. Zhang, "Application of the simulated annealing algorithm to the combinatorial optimization problem with permutation property: An investigation of generation mechanism", European Journal of Operational Research, vol. 118, no. 1, pp. 81-94, 1999.

[6] W. C. Chiang, and C. Chiang, "Intelligent local search strategies for solving facility layout problems with the quadratic assignment problem formulation", European Journal of Operational Research, vol. 106, no. 2-3, pp. 457-488, 1998.

[7] E. G. Talbi, O. Roux, C. Fonlupt, and D. Robillard, "Parallel ant colonies for the quadratic assignment problem", Future Generation Computer Systems, vol. 17, no. 4, pp. 441-449, 2001.

[8] J. Balakrishnan, C.H. Cheng, and K.F. Wong, "FACOPT: a user friendly FACility layout OPTimization system", Computers and Operations Research, vol. 30, no. 11, pp. 1625-1641, 2003.

[9] K. Tsuchiya, S. Bharitkar, and Y. Takefhji, "A neural network approach to facility layout problems", European Journal of Operational Research, vol. 89, pp. 556-563, 1996.

[10] M. Hasegawa, T. Ikeguchi, K. Aihara, and K. Itoh, "A novel chaotic search for quadratic assignment problems", European Journal of Operational Research, vol. 139, no. 3, pp. 543-556, 2002.

[11] S. Heragu, *Facilities design*, Boston: PSW Publishing, 1997.

[12] K.R. Baker, *Elements of sequencing and scheduling*, New York: John Wiley and Sons, 1995.

[13] R. P. Grimaldi, *Discrete and combinatorial mathematics*, 5th Ed, Pearson Education, 2003.

[14] G. Chartrand, and L. Cesniak, *Graphs & digraphs*, 3rd Ed, Belmont: Wadsworth & Brooks/Cole, 1996.

[15] D. G. Luenberger, *Linear and nonlinear programming*, 2nd Ed. Boston: Kluwer, 2003.

[16] G. Luger, *Artificial intelligence: structures and strategies for complex problem solving*, 4th Ed, New York: Addison Wesley, 2002.

[17] M. O. Albertson, and J. P. Hutchinson, *Discrete mathematics with algorithms*, New York: John Wiley and Sons, 1988.

[18] J. Durkin, and J. Durkin, *Expert systems: design and development*, New Jersey: Prentice Hall, 1998.

[19] H. J. Zimmmermann, *Fuzzy set theory and its applications*, 3rd Ed, Boston: Kluwer, 1996.

[20] E. Turban, J.E. Aronson, and T.P. Liang, *Decision support systems and intelligent systems*, 7th Ed, New Jersey: Prentice Hall, 2004.