Nov. 30

International Journal of advanced studies in Computer Science and Engineering
IJASCSE Volume 4, Issue 11, 2015

# Routing Strategy based on Tarantula Mating

S Bandni
Department of B Ad., The University of Burdwan,
Burdwan, West Bengal

Gautami Matrayee
Department of B Ad, The University of Burdwan,
Burdwan, West Bengal

***Abstract*— A multi-agent and multi-criteria routing strategy with a fuzzy orientation based on Tarantula mating behavior has been proposed in this research study. The strategy as proposed in this study is applied on a traffic network. The particular interesting behavior of Tarantula mating is that the female spider sometimes eats the male spider just after their mating to satisfy immediate need for food. In order to implement the proposed routing strategy, a hierarchical structure of agents has been considered where the worker agents at the leaf level calculate shortest paths, congestion, number of intermediate nodes and so on. A master agent at the top of the hierarchy takes the final decision based on these data by PROMETHEE multi-criteria decision analysis with fuzzy concept. A traffic network has been used in order to implement the proposed Tarantula mating based routing strategy. The experimentation shows successful application of the proposed strategy.**

***Keywords*— *Multi-Agent Based Strategy; Tarantula Mating Behavior; PROMETHEE; Routing Strategy; Fuzzy Theory; Traffic Network***

## I. Introduction

This paper is based on the view that in dynamic environments, it is very difficult to decide over the entire optimum path before the journey begins, since, the data, such as, the congestion data, deadlock data, shortest path data may change over time. Thus by the time, when a particular job reaches a particular node, there is a chance that the previously determined optimum path may not be optimum anymore. Thus, instead of finding the entire point-to-point optimum path, the research study as presented in this paper chooses the next best node to route a job towards the destination by a hierarchical agent based framework and Multi-criteria decision analysis with fuzzy orientation.

The hierarchical multi-agent framework shown in this paper imitates the interesting mating behavior in routing jobs in a network. Agents are independent, autonomous, decision making components in a system. Therefore, there are various characteristics of agents which actually make them flexible and adaptive to dynamic environments. The characteristics of agents are depicted in Figure 1 [2].


Fig. 1. Characteristics of Agents

In this paper, a hierarchical multi-agent framework has been proposed. This paper also proposes the application of PROMETHEE multi-criteria decision analysis technique in order to make final decision making. MCDA techniques are the tools or aids for decision making where decision depends on more than a single criterion. Each MCDA technique has some given alternatives and a set of criteria as inputs. Depending on the various requirements of the decision maker and the type of comparison, a particular MCDA technique is applied. Although MOEA and MCDA are closely related but there is distinct difference between these two approaches. MOEA techniques provide a population of solutions called the Pareto optimal solutions as output of the method whereas MCDA technique selects an alternative among several given alternatives based on the pre-specified criteria. The values of these criteria may be optimized beforehand. There are a large number of MCDA techniques as seen in the existing literature. Some of these techniques include SMART (Simple Multi-Attribute Rating Technique), Analytic Hierarchy Process (AHP), Analytic Network Process (ANP), Aggregated Indices Randomization Method (AIRM), Data Envelop Analysis (DEA), Elimination and Choice Translating Reality

(ELECTRE), Measuring Attractiveness by a Categorical Based Evolution Technique (MACBETH), Multi-Attribute Utility Theory (MAUT), PROMETHEE, Technique for Ordering of Prioritization by Similarity to Ideal Solutions (TOPSIS), Weighted Sum Model (WSM), Weighted Product Model (WPM) and so on. This paper applies fuzzy PROMETHEE outranking multi-criteria method for final decision making purpose. The technique PROMETHE will be described while explaining the results in this paper. Since the paper also applies fuzzy concept, thus a brief introduction to fuzzy concept is provided below.

In this research study, the fuzzy concept has been applied in two places – in the calculation of the weights of the criteria in PROMETHEE multi-criteria decision analysis technique and in fuzzy Dijkstra algorithm for finding shortest paths, through the use of fuzzy number as the length of the edges in a network.

Fuzzy number represents a number of whose value, we are somewhat uncertain. They is a special kind of fuzzy set whose numbers are the numbers from the real numbers. The function relating member number to its grade of membership is called a membership function and is best visualized by graphs. The membership of a number $x$ is often denoted by $\mu(x)$. Fuzzy numbers may be of almost any shape, but most frequent of them are:

- Triangular (piecewise linear)
- S-shape (piecewise quadratic)
- Normal (bell-shaped)

Fuzzy numbers may also be trapezoidal, with an interval within which the membership is 1; such numbers are called fuzzy intervals. Fuzzy intervals may have linear, s-shape or normal tails, the increasing or decreasing slopes.
In this research study, we have used the triangular fuzzy numbers in both the previously mentioned cases. A triangular fuzzy number may be represented by a triplet ($a$, $b$, $c$). Here, $\mu_A(x)$ is the fuzzy membership function given by expression (1).

$$\mu_A(x) = \begin{cases} 0 & , \quad x < a \\ \dfrac{x-a}{b-a} & , \quad a \le x \le b \\ \dfrac{c-x}{c-b} & , \quad b \le x \le c \\ 0 & , \quad x > c \end{cases} \qquad (1)$$

Where, $A$ is the Universal set.
In this research study, fuzzy number has been expressed by graded mean integration representation as provided in expression (2) which has been used in fuzzy Dijkstra's shortest path algorithm and the mean representation as given by expression (3) which has been used in calculating the fuzzy weights from the preferences for the criteria in PROMETHEE multi-criteria decision analysis technique.

$$p(A) = (a + 4 * b + c) / 6 \qquad (2)$$

$$p(A) = (a + b + c) / 3 \qquad (3)$$

## II. LITERATURE REVIEW

An agent is a computational system which is long lived, has goals, self-contained, autonomous, capable of independent decision making. Among the benchmark multi-agent technologies, GAIA [3] is a hierarchical agent-based architecture using the concepts of object-oriented analysis and design. Wooldridge et al. [3] used some concepts from FUSION [4]. GAIA is suitable for the development of the systems like ADEPT [5], ARCHON [6]. In GAIA, every agent has a role to play and they interact with each other in a certain pre-defined way which is defined in their protocols. ROADMAP [7] is another agent-based methodology which is an extension of GAIA for complex open systems. Some of the other significant technologies include PROMETHEUS [8], TROPOS [9], PASSI [10], TAPAS [11] and so on. Some of the agent based technologies as applied in manufacturing include PROSA [12], ADACOR [13], HCBA [14] and so on.

The strategy as proposed in this paper has also used multi-criteria decision analysis technique. MCDA techniques can be categorized into 1) Value Measurement Models, such as, AHP (Analytic Hierarchy Process proposed by Saaty [15], Simple Multi-Attribute Rating Technique (SMART) proposed by Edwards and Barron [16]; 2) Goal, Aspiration and Reference Level Models, such as, TOPSIS (Technique for Order Performance by Similarity to Ideal Solution); 3) Outranking Models, such as, ELECTRE [17], PROMETHEE [18], NAIADE [19]. Some of the significant research study applying both multi-agent and multi-criteria techniques include the research studies of Maione and Naso [20], Mikler et al. [21], Eltarras and Eltoweissy [22]. Research study of Gharajeh [23] may also be consulted for an application of fuzzy concept as a reference. The following section 3 describes the multi-agent strategy as proposed in this paper.

## III. THE PROPOSED STRATEGY

This paper has used a hierarchical structure of agents (Figure 2) and PROMETHEE multi-criteria outranking method with a fuzzy orientation. The leaf level of the hierarchy contains worker agents. Each of the worker agents performs a particular task. The worker agents considered in this research study are 1) shortest path agent, 2) congestion agent, 3) deadlock agent and 4) hops agent, 5) capacity (or buffer) handling agent.

The capacity handling agent is important since in order to prevent deadlock condition because of the number of vehicles at the junctions during peak hours of a day. Besides, this will also help to manage the traffic at the junctions properly. The Master agent takes the final decision from top of the hierarchy. After performing the task, each of the worker agents is killed by the master agent after taking the result of the performed task from the worker agent. Thus, the hierarchical structure does not exist after all the tasks are performed by all the worker agents. The final decision is taken by the master agent based on PROMETHEE multi-criteria decision analysis technique based on the information provided by the worker agents. The master agent gets notification after killing each of the worker agents. The idea conveyed in this research study stems from the mating incident of a type of spider called Tarantula where the female spider eats the male one after mating. The analogy of such interesting mating behavior with the idea in this research study can be described in Figure 3.

The routing strategy considered in this research study finds the next optimum neighboring node through agent based technique, instead of finding the entire source-to-destination path. Thus the worker agents and then the master agent will start functioning whenever there will be a need to route a job to the next optimum neighboring node and whenever a new job enters the system. The master agent invokes and initiates the actions of the worker agents, just like the female spider chooses a male spider for mating. The worker agent, after performed their tasks, return the results to the master agent, just like the male spider transfers the genetic material to the female spider during mating. The master agent kills the worker agents after receiving the results from the worker agents, just like the female spider kills and eats the male spider after mating. The master agent gets the notification of the killing of the worker agents, just like the female spider takes the male spider as food. The various functions as performed by various worker agents and the master are described in the following subsections.

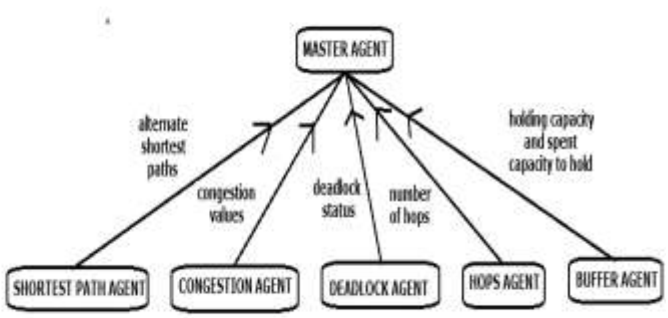


Fig. 2. Hierarchy of Agents

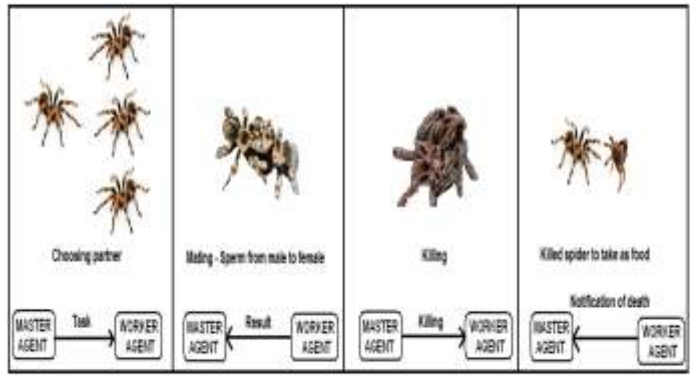


Fig. 3. Analogy with Tarantula Mating Behavior

### A. Shortest Path Agent

The shortest path agent finds the fuzzy shoretst path towards destination node from the each of the neighbors of the current node. The fuzzy shortest path is determined by the fuzzy Dijkstra's algorithm following the research study of Deng et al. [24]. The algorithm is depicted in the Figure 4. Here, *perm[]* represents Permanent node; *v[]* holds the distance to each node from current node. In this algorithm, the edge lengths are triangular fuzzy numbers from which the fuzzified edge lengths are calculated from the fuzzy numbers by using expression (4) below.

$$v_{ij} = \frac{\mathbf{Pr}_{ij}}{\sum_{j=1}^{C} \mathbf{Pr}_{ij}} \quad (4)$$

Where, $v_{ij}$ is the normalized value of the preference and an entry in the *i-th* row and *j-th* column in the matrix containing normalized values for the *i-th* decision maker and the *j-th* criterion; $\mathrm{Pr}_{ij}$ is the respective original preference value delivered by *i-th* decision maker, for the *j-th* criterion.

```
Input: Set of edges (E), Set of vertex (V), d[], source node (s), destination node (d)
Output: Shortest Path from s to d of length contained in v[d]

1.  Initialize:
    For each edge e ∈ E
       Set Fuzzy Number W_e = (a_e + 4*b_e + c_e) / 6
    End For
    Set d[s] ← 0 and d[i] ← ∞, ∀ i, i ≠ s
    Set permanent node p ← s
2.  For each neighbor j ∈ neighbor(p)
        Set v[j] ← minimum (d[j], d[p]+c_p,j)
        Set pred[j] ← p
    End For
3.  Find minimum among non-permanent node:
    Set min ← ∞
    For each vertex v ∈ V
       If v is not permanent node Then
          If d[v] < min Then
             Set min ← d[v]
             Set j' ← v
          End If
       End If
    End For
4.  Set next permanent node p ← j'
5.  Repeat steps 2 – 4 until all the nodes become permanent or the destination is
    reached.
```

Fig. 4. Fuzzy Dijkstra's Shortest Path Algorithm

```
For each neighbor i
        Record number of jobs on arc (src, i)
End For
For each neighbor i
        Find the nextnode n of i on SP(i)
        Record number of jobs on arc (i, n)
End For
```

Fig. 5. Congestion Finding Algorithm

```
For each neighbor n1
    For each neighbor n2
        If n1 != n2 Then
          If neighbor[n1] != neighbor[n2] Then
              Safe[n1] = 1
              Safe[n2] = 1
          End If
        End If
End For
```

Fig. 6. Deadlock Finding Algorithm

```
For each neighbor i
   Find shortest path SP(i) to destination d from i
   Set count[i] ← 0
   For each node from n on SP(i)
       Set count[i] ← count[i] + 1
   End
End For
```

Fig. 7. Hops Algorithm

### B. Congestion Agent

The congestion agent checks for the congestion of the edges from the current node to each of the neighboring nodes and from each of the neighboring nodes to their immediate neighbors on their shortest paths towards destination (Figure 5). Although the congestion can be represented by more than one factor, but in this research study, congestion is represented by the number of jobs travelling on a particular edge.

### C. Deadlock Agent

The deadlock agent (Figure 6) checks whether the neighbors of the current node faces any immediate cyclic path. Let the current node is *c* and the neighbors of c are *x*, *y* and *z*. If the immediate neighbors of two or more neighbor of the neighbors *x*, *y*, *z* be same, then the algorithm marks those neighbors (*x* and/or *y* and/or *z*) as unsafe, otherwise they are safe. It can easily be realized that, in dynamic environment where the number of jobs on each path vary continuously, it will be less significant to find a cyclic deadlock throughout the entire network. Thus instead of finding the cyclic deadlock in the entire network, it will better to find such an immediate cycle. The algorithm endeavors to avoid cyclic path since in such path, there is more chance of facing a collision.

### D. Hops Agent

The hops agent finds the number hops or intermediate nodes on a path towards destination (Figure 7). Thus for each of the shortest path from the neighboring nodes towards destination, there is a particular number representing the number of intermediate nodes on the way. The target is to choose that particular neighbor as the better node which will have least number of hops since, the greater the number of hops, greater is the chance of facing more congestion, more deadlock, more blockage at the nodes due to loaded buffers.

### E. Capacity Handling Agent

Each of the nodes in a network is assumed to have fixed capacity to station the number of vehicles. The capacity handling agent provides two types of data on each of the immediate neighbors – 1) the total capacity of each of the immediate neighbors and 2) the capacity spent at each of the immediate neighboring node.

### F. Master Agent

The master agent takes the final decision based on the information provided by the worker agents (Figure 8). The shortest path agent provides the alternate path through neighbor of the current node. Thus the number of alternate paths equals the number of immediate neighbors of the current node. The congestion agent provides the congestion data in terms of the number of jobs travelling on the edge between the current node and each of the neighboring nodes and the number of jobs travelling on the edge between the immediate neighbors of the current node and the neighbors of the immediate neighbors. The deadlock agent provides the boolean values indicating whether the immediate neighbors of the current node are safe. The hops agent delivers the number of hops or intermediate nodes between the current node and the destination node on each of the alternate paths through the neighbors. The capacity handling agent provides the total capacity of each of the immediate neighbors and the capacity spent at each of those immediate neighbors. Based on the above data and information, the master agent takes decision using a multi-criteria decision analysis technique known as PROMETHEE by selecting best neigbor to which the job may be routed next.

```
1.  Input a network with fuzzy edge values
2.  Input source and destination nodes s and d respectively
3.  Find neighbor N of s
4.  For each neighbor n ⊂ N
        Find shortest path p from n to d
        Store p in PATH
    End For
5.  For each neighbor n ⊂ N
        Find path p ⊂ PATH FROM N
        Find number of jobs on edge (s, n)
        Find number of jobs on edge (n, i), i: neighbor of n on p
        Check whether there is any deadlock cycle with n
        Calculate number of hops from n on p
    End For
6.  Apply PROMETHEE to find the best neighbor n
```

Fig. 8. Algorithm for Master Agent

## IV. RESULTS AND DISCUSSION

The multi-agent and multi-criteria approach as proposed in this research study has been applied on a network example (Figure 9) with edge lengths represented by triangular fuzzy numbers. The experimentation has been performed in C# of Visual Studio .Net 2008 in a dual core PC with 2 GB memory. The worker agents have been implemented by using threads which run in parallel through thread synchronization. The relevant details are shown in Figure 10.

Next, a total of 4 decision makers is assumed and they all assign their own preference to the sever criteria. The seven criteria are – 1) Path length; 2) Number of jobs travelling on the edge from current node (node 3) to immediate neighbors (nodes 4, 5). The respective edges in this example are: 3-4, and 3-5; 3) Number of jobs travelling on the edge between the immediate neighbors (nodes 4, 5) and their neighbors on their respective shortest paths; 4) deadlock status of each of the neighbors; 5) Number of intermediate nodes or hops; 6) the total capacity at each of the immediate neighbors; 7) the capacity spent at each of the immediate neighbors.

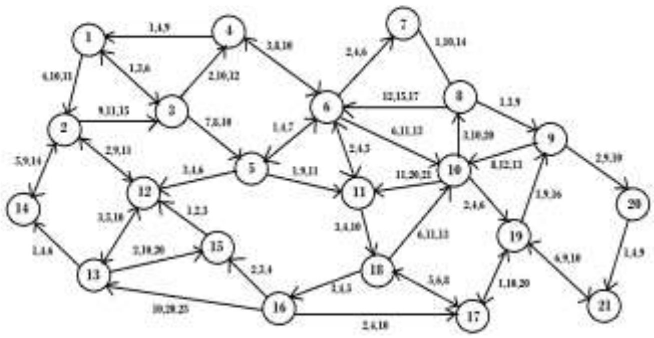

Fig. 9. Network Diagram

**Source Node: 3; Destination Node: 10**
**Number of neighbors of Source Node 3 ⇨ 4, 5**
**Alternate Paths through neighbors**

| Alternates | Paths |
|---|---|
| A1 | 3→4→6→10<br>3→4→6→7→8→9→10 |
| A2 | 3→5→6→10<br>3→5→6→7→8→9→10<br>3→5→11→6→10<br>3→5→11→6→7→8→9→10 |

Fig. 10. Relevant Data on Network

Figure 11 shows the preferences as provided by the 4 decision makers (DM), the fuzzy weights as calculated for the above seven criteria. For calculating the fuzzy weights, first the preferences for each DM are converted to probability values. Thus there will be 4 probability values from each DM under each criterion. Then the minimum, intermediate and maximum values are found out from each of the 4-valued set for each criterion. These three calculated numbers form the fuzzy number for each criterion and then the fuzzified value of the criterion is calculated following expression (3).

**Preference for 7 Criteria from 4 Decision Makers**

| Decision Maker | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
|---|---|---|---|---|---|---|---|
| 1 | 3 | 6 | 4 | 5 | 7 | 2 | 1 |
| 2 | 5 | 2 | 6 | 1 | 4 | 3 | 7 |
| 3 | 1 | 3 | 6 | 7 | 2 | 4 | 5 |
| 4 | 7 | 2 | 6 | 4 | 1 | 5 | 3 |

**Weights of 7 Criteria**

| | |
|---|---|
| C1 | 0.1547619 |
| C2 | 0.1309524 |
| C3 | 0.1190476 |
| C4 | 0.1428571 |
| C5 | 0.1190476 |
| C6 | 0.1309524 |
| C7 | 0.1309524 |

Fig. 11. Preference Function Values and Weights

| Path Lengths | Number of jobs from source node to destination | Number of jobs from neighbors to neighbors of neighbor | Deadlock status | Number of hops | Capacity to station the vehicles | Spent capacity |
|---|---|---|---|---|---|---|
| 19 | 2 | 4 | 1 | 2 | 26 | 17 |
| 15 | 3 | 1 | 1 | 2 | 26 | 20 |

Fig. 12. Data Values from Agents

| Alternatives | $\phi^+$ | $\phi^-$ | $\phi$ |
|---|---|---|---|
| A1 | -0.4523808 | 0.4523808 | -0.9047616 |
| A2 | 0.4523808 | -0.4523808 | 0.9047616 |

Fig. 13. Outranking Flows

Next, the values as obtained from the seven agents are shown in Figure 12. The preference index and the outranking flows are calculated following expressions (5), (6), (7) and (8). The value of Φ is calculated by expression (9) (Figure 13). Since higher the value of Φ, higher is the preference of the alternative, thus the ranked alternatives in the descending order are: A2 → A1, A2 being the highest ranked alternative and thus the next best neighbor is 5.

$$\pi(a,b) = \sum_{j=1}^{C} W_j P_j(a,b) \quad (5)$$

$$\pi(b,a) = \sum_{j=1}^{C} W_j P_j(b,a) \quad (6)$$

$$\phi^+(a) = \frac{1}{(n-1)} \sum_{x \in A} \pi(a,x) \quad (7)$$

$$\phi^-(a) = \frac{1}{(n-1)} \sum_{x \in A} \pi(x,a) \quad (8)$$

$$\phi(a) = \phi^+(a) - \phi^-(a) \quad (9)$$

Where, $W_j$ is the weight of the $j^{th}$ criterion, $P_j(a,b)$ is the preference function value for preference of a alternative *a* over alternative *b*, $\pi(a,b)$ is the preference index, $\phi^+(a)$ is the positive outranking flow, for alternative *a*, $\phi^-(a)$ is the negative outranking flow for alternative *a*.

## V. CONCLUSION

This paper proposes a routing strategy for manufacturing networks where, instead of establishing a point-to-point connection between source and destination nodes, a job or message is routed to the next optimal neighboring node. A hierarchical multi-criteria multi-agent based system is considered with a master agent and several worker agents for the proposed routing strategy. The number of worker agents is same as the number of criteria considered for decision making of the master agent. In this paper, a total of seven criteria have been considered to determine the next optimum node to route a particular job. These criteria are – 1) shortest path length between the each of the immediate neighbors and the final destination; 2) the number of jobs on route between the current node and each of the immediate neighbors; 3) the number of jobs between each of the immediate neighbors and the neighbors of the immediate neighbors; 4) the deadlock status involving the current node; 5) the number of intermediate nodes (hops); 6) the total capacity to station vehicles at each of the immediate neighbors; 7) the capacity spent at each of the immediate neighbors. The consideration for capacity is required for proper management of traffic at the junctions, especially during peak hours of a day. The master agent takes all these inputs from the worker agents and selects the best immediate neighbor using a multi-criteria outranking method known as PROMETHEE. The entire idea is based on the mating behavior of a species of spider known as Tarantula. The female Tarantula sometimes eats the male Tarantula just after mating to satisfy the intermediate need for food or for any genetic reason. Specific example has been considered to implement the proposed strategy.

As the future scope of this paper, the strategy can be applied to various fields of study for any problem which can be represented by a network. Thus primarily, this strategy may be applied to manufacturing scheduling and routing problems, supply chain networks, traffic networks, medical diagnosis and many more real-life problems. The number and type of criteria will vary accordingly.

## REFERENCES

[1] S. Bandyopadhyay, and R. Bhattacharya, "Finding optimum neighbor for routing based on multi-criteria, multi-agent and fuzzy approach", Journal of Intelligent Manufacturing, vol. 26(1), pp. 25-42, 2015.

[2] S. Bandyopadhyay, and R. Bhattacharya, "Discrete and Continuous Simulation: Theory and Practice", CRC Press, Florida, 2014.s

[3] M. Wooldridge, N.R. Jennings, and D. Kinny, "The Gaia methodology for agent-oriented analysis and design", Autonomous Agents and Multi-Agent Systems, vol. 3(3), 285–312, 2000.

[4] D. Coleman, P. Arnold, S. C. Bodoff, H. Dollin Gilchrist, F. Hayes, and P. Jeremaes, "Object-Oriented Development: The FUSION Method", Prentice Hall International, Hemel Hempstead, England, 1994.

[5] N. R. Jennings, P. Faratin, M. J. Johnson, T. J. Norman, P. O'Brien, and M. E Wiegand Wiegand, "Agent-based business process management", International Journal of Cooperative Information Systems, vol. 5(2-3), 105–130, 1996.

[6] N. R. Jennings, "Using ARCHON to develop real-world DAI applications", IEEE Expert, vol. 11(6), 64–70, 1997.

[7] T. Juan, A. Pearce, and L. Sterling, "ROADMAP: extending the Gaia methodology for complex open systems", In M. Gini, T. Ishida, C. Castelfranchi, W.L. Johnson (Eds.), Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002), ACM Press, pp. 3–10.

[8] L. Padgham, and M. Winikoff, "Developing Intelligent Agent Systems - A Practical Guide", John Wiley & Sons, ISBN 0-470-86120-7, 2004.

[9] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini, "Tropos: An agent-oriented software development methodology", Journal of Autonomous Agents and Software Development Methodologies, vol. 8, 203–236, 2004.

[10] P. Burrafato, and M. Cossentino, "Designing a multi-agent solution for a bookstore with the PASSI methodology", In Proceedings of the Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002), Toronto, 2002.

[11] J. Holmgren, P. Davidsson, J. A. Persson, and L. Ramstedt, "TAPAS: A multi-agent-based model for simulation of transport chains", Simulation Modelling Practice and Theory, vol. 23, 1–18, 2012.

[12] H. V. Brussel, J. Wyns, P. Valckenaers, and L. Bongaerts, "Reference architecture for holonic manufacturing systems: PROSA", Computers in Industry, vol. 37(3), 255-274, 1998.

[13] P Leitao, A. Colombo, and F. Restivo, "ADACOR: a collaborative production automation and control architecture", IEEE Intelligent Systems, vol. 20(1), 58–66, 2005.

[14] J. Chirn, and D. McFarlane, "A component-based approach to the holonic control of a robot assembly cell", In Proceedings of the IEEE 17th International Conference on Robotics and Automation, ICRA, 2000.

[15] T. L. Saaty, "The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation", McGraw-Hill, New York, USA, 1980.

[16] W. Edwards, and F. H. Barron, "SMARTS and SMARTER: improved simple methods for multiattribute utility measurement", Organizational Behavior and Human Decision Processes, vol. 3, 306–325, 1994.

[17] B. Roy, "The outranking approach and the foundations of ELECTRE methods", In Bana e Costa, C.A. (Ed.), Readings in Multiple Criteria Decision Aid. Springer-Verlag, Berlin, German, pp. 155–183, 1990.

[18] J. P. Brans, and Ph. Vincke, "PROMETHEE. A new family of outranking methods in MCDM", Management Science, vol. 6, 647–656, 1985.

[19] G. Munda, "Multicriteria Evaluation in a Fuzzy Environment—Theory and Applications in Ecological Economics", Physica-Verlag, Heidelberg, Germany, 1995.

[20] G. Maione, and D. Naso, "A soft computing approach for task contracting in multiagent manufacturing control", Computers in Industry, vol. 52(3), 199-219, 2003.

[21] Armin R. Mikler, V. Honavar, and J. S. K. Wong, "Autonomous agents for coordinated distributed parameterized heuristic routing in large dynamic communication networks", The Journal of Systems and Software, vol. 56, 231-246, 2001.

[22] R. Eltarras, and M. Eltoweissy, "Associative routing for wireless sensor networks", Computer Communications, 34(18), 2612-2173, 2011.

[23] M. S. Gharajeh, "Determining the state of the sendor nodes based on fuzzy theory in WSNs", International Journal of Computers, Communications & Control, vol. 9(4), 419-429, 2014.

[24] Y. Deng, Y. Chen, Y. Zhang, and S. Mahadevan, "Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment", Applied Soft Computing, vol. 12(3), 1231–1237, 2012.