International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

Network Optimization using Multi-Agent Genetic Algorithm

Sikandar Hanif

School of Information Technology and Engineering, Tianjin University of Technology and Education, Tianjin, China

Abstract: ***This paper present an innovative technique based on multi-agent genetic algorithm for optimization of a network. We unify agent system with genetic algorithm and applied to solve multi-objective problem optimization. In this algorithm an agent illustrate a candidate results to the optimization problem. Agent lives in the grid environment and it possesses own local space called the neighborhood. In the neighborhood, an agent can compete and collaborate with other agents, to attain the purpose of gene exchanged and evolved. Agent also possesses some cognition of the surroundings and can pursue itself while expands, with the aim to adapt itself to the surroundings better and increases its viability. A new multi-agent genetic algorithm is proposed named as MAGA-NOP, in which we implement crossover operator based on neighborhood to get useable information from its neighbor and by doing this we avoid it from random recombination. We used priority based encoding mechanism to encode chromosome strings. Several networks are used to test the algorithm performance; the experimental results revealed that MAGA-NOP has a progressive performance than other algorithms.***

Keywords: Network optimization; Multi-Agent Genetic Algorithm; Optimal Path; Multi-Agent System.

1. Introduction

The optimization of network problem is to reduce the cost and delay to find optimal path from initial source node to end source node is a most famous problem in network field. Several researches focus on the shortest path algorithms, which has given arise to a different number of research techniques for different conditions and constraints [1-3]. Recently we pay attention on time dependent graphs studies [4, 5]. The foundation need for research is the potential to get multiple Pareto-optimal results in only one run. The basic motive why problem has multi objective formulation because we cannot get single unique solution that can optimize objects simultaneously. The best practical value is that algorithm that is able produce a long range of alternative result that is near to the optimal Pareto front.

Genetic algorithm basically represents a class with different methodology relying on heuristic random search technique. John H. Holland proposed it during early seventies since he has discovered application in a number of practical problems. The genetic algorithm can be seen as an evolutionary process where in a population of solutions arises beyond the sequence of generations. Mostly we used computational models for intricate system simulation activities used in engineering for performance optimization. In the area of engineering the multi objective problem appears as natural fashion. All the objective functions should be optimized once; in general all they want to compete with others and the optimization process necessary to find good optimal solution.

For the solution of multi-objective problem of optimization the foundation goals are; ü To save non-dominate result and relate with the optimal results ü For objective function we keep making progress to get the Pareto front ü For the optimal Pareto result we maintain the diversity on Pareto front ü For designer we provide large but constant number for the point selection of Pareto to take decision In past years, there is an interesting concern from the artificial intelligence community on agent-based computation methods, which are utilized to determine numerous optimization problems [6], [7], [8], [9], [10]. Liu et al. [11] give unique distributed techniques to solve constraint at is faction problems they used energy-based multi agent for the solution of 7000 queen problem. Zhong et al. [12] give a new technique by combining genetic algorithm and multi-agent system to solve the numerical optimization of the global problems. Mostly the stationary problems are solved using the agent based computation. In 1950s, to solve multi-objective optimization problem plethora of techniques developed. Some of the representative classical methods are linear programming, the weighted sum method, and the goal programming method. Furthermore, the classical routing problems have an intention to diminish the total distance or total travelling time in single objective problem. Regardless, in several applications tackling with the design and efficiently use of networks, the social complexity and economic environment requires the explicit consideration of destination functions other than cost either travel time.

In other words, it is essential to believe it that many problems in real world are multi-criteria in nature. The objective functions related to environmental impact, risk, accessibility, cost, reliability and time have relevance for choosing the best optimal route in numerous problems of optimization in a network [13]. Consequently, the study of models for shortest path is a natural beginning of network models for introducing innovative ideas, including the usage of appropriate data structures and data scaling to increase the outcomes of the algorithmic in worst cases [14]. The aptitude of multi-objective and evolutionary algorithms to determine multiple Pareto-optimal outputs in only single run has been made them attractive for solving problems with various and opposing goals. Recently, to get satisfactory solution evolutionary algorithms combined with multi-agent system to resolve combinatorial optimization and problems of constraint satisfaction. Given all such proofs, we realized that Multi-agent genetic algorithm is adequate in solving wide-scale complicated problem in similar manner to get the suitable path between nodes in network.

In this paper, a multi-agent genetic algorithm (MAGA-NOP) is proposed to optimize the network. Four genetic operators of agents are designed or redesigned to attain the goal: neighborhood competition parameter and neighborhood weight mapping crossover realize contention and collaboration among agents. Mutation and self-learning parameter enhance the strength of agents by information.

2. Related Work

The key objective of network optimization problem is to avail the feasible route between two given nodes in the network. Different methods are proposed to give optimal path in the network [14, 15]. Here we reviewed the important algorithms for the solution of this problem. Dijkstra presenting a method which gives solution for unique source shortest problem in O (n square) time; all edge needs to be more than zero either zero. We can execute the shortest paths given from start node to all different nodes without worsening run time [1, 14]. Hence, Bellman-Ford delivered an algorithm which can compute one source optimal paths in weighted digraph (for these case few edges may retain negative weights). Compare to Dijkstra's algorithm, which accomplishes the similar problem that's running time is low, but it required positive edge weights. Hence, Bellman-Ford is mostly used when weights of edge is negative [14]. Floyd-War shall, relaxed the constraint that all weights must be nonnegative, the algorithm provided can solve all the shortest route problem by multiply representation of adjacency-matrix in directed, weighted graph multiple time with O(n3) time complexity [3, 14].

Eppstein [2] gives algorithm that can able to provide the k optimal routes in O (m+ n log n + k) time when the n is nodes and m is edges. Recently, several methods catch attention towards addressing the SPP, especially the GAs (and other evolutionary algorithms) because of their capability as optimization technique. In the network, they usually used to get the solution of optimization problems in communication network optimization problems, including the effective approaches on dynamic routing problem, multicasting routing problem. These types of problems can formed as combinatorial optimization [14]. In a similarly way, we will stated the work of Gen et al. [16]; it demonstrated that it can solves the shortest path problem using an ordinary undirected graph assigning static weights. It will be better by indicating that, the purpose of the operation was not to calculate using conventional algorithm, but to represent an encoding scheme that perhaps is extendible to numerous challenging problems for which undetermined algorithm exists. Since then allot of genetic algorithms have been created, specifically a genetic algorithm for solution of routing problem for shortest path and for sizing populations by Ahn and al. [17]. The current study is basis on the association of multi-agent systems and genetic algorithms for network optimization to find optimal path. The proposed algorithm can be imposed on real world applications.

3. Problem Formulation

A constant set of nodes V={1,2..,n} and a pairs ordered of nodes collection A={(i, j),(k, l),(m, n),..,(u, v)} from v is called the set A edge when a given network is presented by graph G=(V, A).An edge(u, v) show that u and v have a connection that contains a positive value represent cost from u to v. Path is a chain of nodes from v1 to Vn where every node is different. Figure 1 shows an example of a directed graph with weighted edges. In the Fig.1, (1, 2, 3, 4) is path between 1 and node 4.

The key objective of optimal problem is to find minimum cost z between given nodes. Mathematical representation is given below

$$min\ z = \sum_{i=1} n \ \sum_{j=1} n \ cij \ xij \qquad (1)$$

Where: *n* number of nodes; *cij* transmission cost of arc *(i, j)* and *x ij* the link on an arc *i, j belong* to A.

$$\cdot\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = \begin{cases} 1 & (i = 1) \\ 0 & (i = 2,3, \dots, n-1) \\ -1 & (i = n) \end{cases} \qquad (2)$$

$$x_{ij} = 0 \text{ or } 1 \quad \forall\, i, j \qquad (3)$$

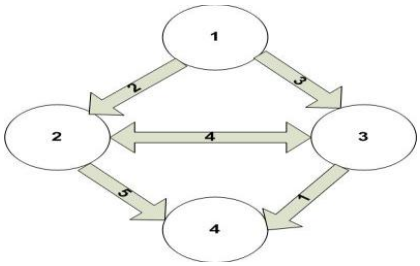

Figure 1: Direceted Graph

4. Algorithm Description

a) Agents For Optimal Path

Genetic algorithm is particularly suitable to solve multi-objective optimization problems, because they deal simultaneously with a set of possible solutions. A lot of Pareto optimal solutions can be obtained by a single simulation run of genetic algorithm, instead of having to perform a series of separate runs as in the case of the traditional mathematical programming techniques. Additionally, genetic algorithm does not need derivative and continuity of the objective functions, so that it is easy to apply to practice problem.

An agent can be defined as a arithmetically process that manifest high degree of sovereignty, by conducting actions in atmosphere based on knowledge acquired by the environment for special use. In multi agent atmosphere there are more than one agent lives, so they can communicate with one another and moreover where the atmosphere constraints so the agent doesn't know each and everything at any specific given time while the other agents know (including the internal states of the other agents themselves) [8,18].

These show that the meaning of an agent is different for different problems and should be designed according to the problems under consideration. In MAGA-NOP, an agent is defined as a path between two nodes, source node and destination node in the network.

Definition 1: An agent is a candidate solution for the optimal path problem that is under analysis; so its energy value can be defined by the fitness function.

Definition 2: All agents represents in a lattice like atmosphere. Each agent has its own fixed place in this lattice like atmosphere and they can only interchange information with neighbors / contiguous. In Figure 2 we can see agents Latsize x Latsize environment. For an agent $Lati,j$ place at ith line jth row in the lattice, $i, j = 1,2, \dots , Latsize$, then the Contiguousi,j of $Lati,j$ can be define as follows (17) and can be better visualized in Figure 4.

$$Neighbors_{i,j} = \{Lat_{i',j}, Lat_{i,j'}, Lat_{i,j''}, Lat_{i'',j}\} \quad (17)$$

$$i' = \begin{cases} i-1 & i \neq 1 \\ Lat_{size} & i = 1 \end{cases}, j' = \begin{cases} j-1 & j \neq 1 \\ Lat_{size} & j = 1 \end{cases}$$

$$i'' = \begin{cases} i+1 & i \neq Lat_{size} \\ 1 & i = Lat_{size} \end{cases}, j'' = \begin{cases} j+1 & j \neq Lat_{size} \\ 1 & j = Lat_{size} \end{cases}$$

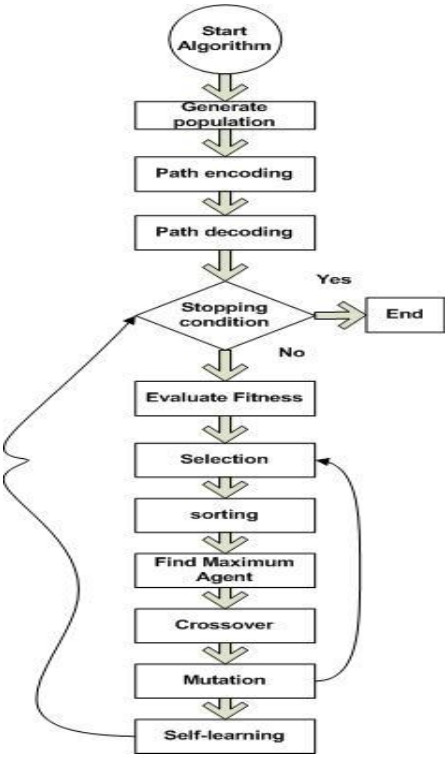

Figure 2: Flow Chart for Proposed Algorithm

b) Population Initialization and Agent Representation

Random initialization and heuristic initialization are two general initialization methods acting that satisfy the boundary and/or scheme restraint to the problem. However the mean fitness for the initialization of heuristic is relatively high so that it may help the genetic algorithmic program to uncovering a solvent faster. At the similar time, for large weighing machine problem like routing optimization, the heuristic approach may just explore a small character of the solution blank makes it's difficult to uncovering a optimal global solution Usually the encryption process is designed depending on the chromosome nature for generating the initial population. Therefore, this research uses random initialization that we can create the first population. In increase, the rejection strategy is used to grasp chromosome representation constraints. However it presented the disadvantage that n-to-one mapping may appear for the encoding at some case. The random initialization is applied for this work. How to encode a solution of the optimal path problem in the network is the basic issue for GAs. We need to consider these crucial issues carefully when building a new non-binary application string coding so as to design an effective GA chromosome. In MAGA-NOP, we use the priority-based encoding and decoding proposed by Lin and Gen in Ref. [19]. In this representation method, the node ID is represented by the position of a gene and the value of the node ID is used to represent the priority of the node for building a path among candidate.

The encoding method and decoding of the path in the graph in Figure 1.We start by finding a node for the position next to source node 1, node 2 and 3 are eligible for the position, which is fixed according to adjacent relation among nodes. Respectively, the priorities values of them are 3 and 1. Node 2 is chosen and is put into the path because of highest priority value. The possible nodes next to node 2 are node 3 and node 4. Because of his highest priority value, node 4 is put into the path. Then we form the set of nodes available for next position and select the one with the highest priority among them. Repeat these until we obtain a complete path. (1-2-4).We use priority-based encoding and decoding because is offering many advantages, first, any permutation of the encoding produces a path, secondly, it suitable for most existing genetic operators, thirdly, any path has a corresponding encoding and fourthly, is searching in all solution space.

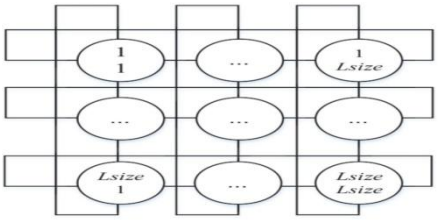

Figure 3: Agent Lattice Environment

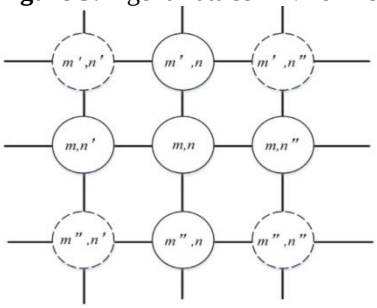

Figure 4: Agent Neighbors

c) Selection Operator

In accordance to attain limited resources in the environment, agents compete against each other. Only the individuals with good adaptability can be chosen to survive, others will be eliminated. Each time three individual are selected using roulette wheel and then the best two individual are selected.

d) Genetic Parameter of Agent

MAGA-NOP has four genetic parameters: neighborhood competition operator and neighborhood weight mapping crossover, they yield competition and collaborate among agents. Agent's energy by knowledge increased by self-learning and mutation techniques. . Given an agent $Lati,$ =

($n1$, $n2$, … , nn), we define maximum agent $Maxi,j$ in the neighborhood of $Lati,j$ as follow: $Maxi,$ = $m1$, $m2$, … , mn ∈ $Neighborsi,j$ and ∀$Agent$ ∈ $Neighborsi,j$, $Energy(Agent)$ ≤ $Energy$ $Maxi,j$. Below is the brief explanation of four operators:

1) Neighborhood competition Operator

In all agent lattice environment, this operator takes to competition between an agent $Lati,$= ($n1$, $n2$, … , nn)and the agent $Maxi,j$ = $m1$, $m2$, … , mn in its local environment. If an agent is winner that means its energy is more than maximum energy in its neighbors then it can live and will be left untouched. Otherwise agent is loser, so it must die and a new agent that generated by plan described below will occupy his lattice-point.

This strategy is derived from that proposed by Zhong et al.[8] and was taken over by Xiaoying P. et al.[9], is a kind of heuristic crossover, is in favor of reserving some information of a loser. The dead agent perhaps still has useful information, so the new agent generated by both the agent L and agent Max. At the end of this operation, the energy of each agent is reappraisal.

2) Crossover Operator

The crossover operation reflects the collaborative behavior of agents. The agent who lives in the environment will collaborate with others in the same neighborhood, to improve their own energy. From the weight mapping crossover proposed by Lin and Gen [18]; what can be considered as an addition of one-cut point crossover for transmutation representation. In this one-cut point crossover, two chromosomes (parents) would choose a random-cut point and generate the offspring by using a segment of its own parent to the left of the cut point, then reshape the right segment based on the weight of other parent of right segment.

In this we use two-cut point crossover instead of one cut-point crossover, two chromosomes (parents) will choose two random-cut points and generate the offspring by using a segment of its own parent between the middle of the cut points, then reshape the cut segment based on the weight of other parent of the cut segment. We also use another parameter as adaptive crossover probability for mixing parameter.

In this paper, we have designed a new weight mapping crossover operator based on neighborhood. An agent $Lati,$ = ($n1$, $n2$, … , nn) on the lattice will only cross with $Maxi,j$ = $m1$, $m2$, … , mn so as to obtain useful information from its neighbors and avoid random recombination. In order to protect good patterns, we will not change $Maxi,$= $m1$, $m2$, … , mn . ∀$Lati,j$ = $n1$, $n2$, … , nn , if U 0,1 < Pc and $Energy$ $Lati,j$ < $Energy(Maxi,j$),we will perform the Neighborhood weight mapping crossover between the agent $Lati,j$ and $Maxi,j$ then the newly generated agent will replace $Lati,j$. At the end of this operation, the energy of each agent is revaluated.

3) Mutation Operator

Agent possesses some environment knowledge, and it can make use of the knowledge to learn for improving energy itself. For mutation we will use inverse transformation method. We generate adaptive mutation probability. If P < Pm then we create two random points. If point > point 1 then we do reverse transformation. This is a very simple mutation operator. Select two random points (i.e.; positions 2 through 5) and reverse the genes between them. If the mutation rate is not in [0, 1], the mutation rate is repaired into the [0, 1] according to the repair rule.

0 **1 2 3 4** 5 6 7
Becomes
0 **4 3 2 1** 5 6 7

4) Self-learning Parameter

We will find the maximum value from the adjacent three. We set the self learning probability 0.3. If the agent energy found after using self-learning operator is not more than the current best agent energy, it will not be taken into consideration. And the $SlBestt$ is taking the proprieties of $CBestt$.

5. Experimental Results

In this section, we study the effectiveness of our approach. We compare the results obtained by MAGA-NOP with result of the NSGA-II algorithm. MAGA-NOP is running on Core(TM) M5-5Y10c processor (0.80GHz 1.00 GHz) with 4GB RAM. We show that our algorithm successfully finds the optimal route in the network and it is competitive with other approaches. The MAGA-NOP algorithm has been written in C language, as regards MAGA-NOP, we employed a trial-and-error procedure and then selected the parameter values giving good results for all simulation data sets. Thus, we set Pc 0.6, Pm 0.4, the population size was 100 this means the agent lattice dimension was 10 and number of generations 300.The simulation studies have been done in weighted network topology (with 20) as shows by in Figure 5.

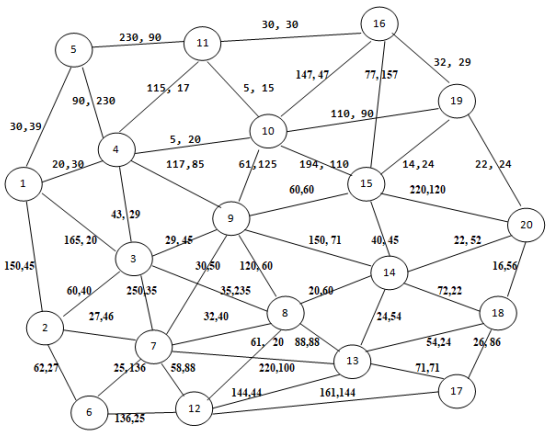

Figure 5: Network with 20 Nodes

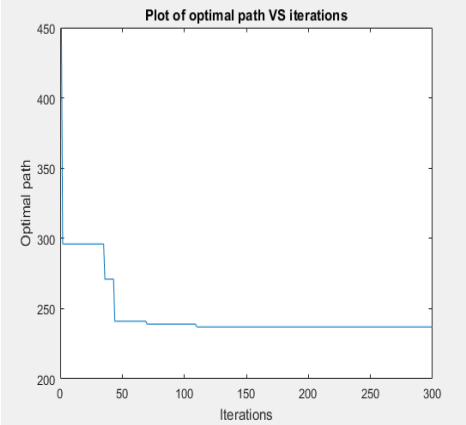

Figure 6: Optimal Path using ACO

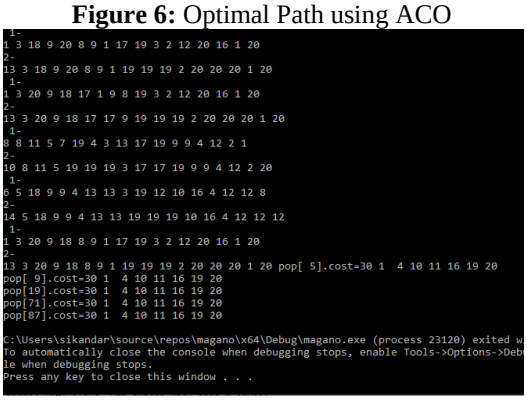

Figure 7: Optimal Path using MAGA-NOP

Table 1: Comparison of GA, ACO and MAGA-NOP

Algorithm	Node	Parameters	Path	Total cost
Ant colony optimization	20	iteration = 300	1-4-9-15-19-20	233
Genetic Algorithm	20	Pop_size = 100; max gen = 300	1-3-9-15-14-20	316
MAGA-NOP	20	Pop_size = 100; max gen = 300	1-4-10-11-16-19-20	114

Table 2: Comparison of two objectives

Algorithm	Parameters	Path	Total cost	Total Delay
NSGA-II	Pop_size = 100; max gen = 300	1-4-10-19-20	157	164
MAGA-NOP	Pop_size = 100; max gen = 300	1-4-10-11-16-19-20	114	148

Table 1 represents the result get for one objective using same number of parameter for each algorithm.

Table 2 represents the result for two objectives that is optimization of cost and delay.

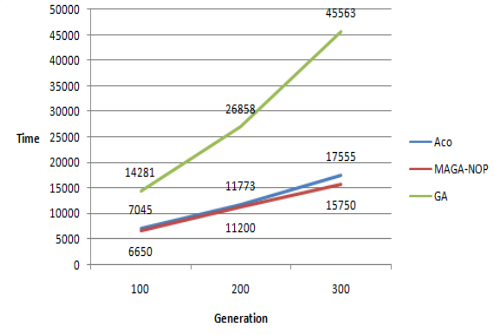

Figure 8: Time vs Generations

Figure 8 show convergence of each algorithm. We compare computation time for each algorithm for different number of generations. MAGA-NOP take less time of computation and it is more convergent toward optimal solution in less time.

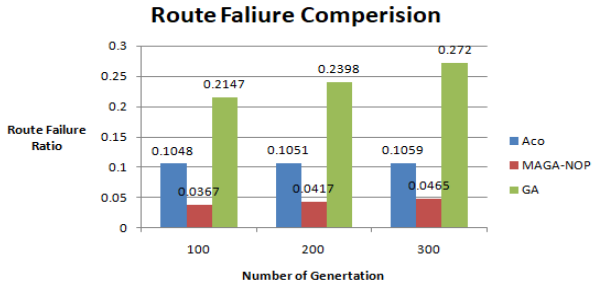

Figure 9: Route Failure Ratio

First the quality of solution for each algorithm is investigated. The route optimality is defined as the percentage of time that the algorithm needs to find the optimal path or shortest path. The route failure is the inverse of the route optimality. It is asymptotically the probability that the computed route is not optimal, because it is the relative frequency of the route failure.

6. Conclusion

This paper proposes the algorithm named as MAGA-NOP to optimization of a network. The observations in Multi-optimization problem show that MAGA-NOP has a vast ability to find optimal solutions in the network. Because it is getting advantages of the combination of evolutionary algorithm and multi-agent system to exceed the existing routing algorithm. Although, we can also able to find optimal path in the network using different number of nodes and edges by MAGA-NOP algorithm. Initialization step is responsible for performance because mostly it depends on the quality and number of chromosomes.

In MAGA-NOP, a series of operators are designed or redesigned, neighborhood competition operator, neighborhood weight mapping crossover, reverse mutation and self-learning operator to realize the multi-agent system and genetic algorithm behaviors. In our future work we plan to redesign MAGA-NOP considering the dynamic nature of network.

References

[1] E.W. Dijkstra, A note on two papers in connection with graphs, Numeriske Mathematics 1 (1959) 269271'.
[2] D. Eppstein, Finding the k shortest paths, SIAM Journal on Computing 28 (2) (1998) 653-674.
[3] R.W. Floyd, Algorithm97: Shortest paths, Communications of the ACM 5 (1962) 345
[4] L. Fu, Real-time vehicle routing and scheduling in dynamic and stochastic networks, Ph.D. Thesis at the University of Alberta, 1996. **[5]**A. Orda, R. Rom, Distributed shortest-path protocols for time-dependent networks, Distributed Computing 10 (1) (1996) 49-62
[5] J. Ferber, Multi-Agent Systems: An Introduction to Distributed Artificial Intelligenc, New York: Addison-Wesley, 1999

[6] N. R. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agent research and development," Autonomous Agents and Multi-Agent Systems Journal, vol. 6, no. 4, pp. 317-331, 1998
[7] J. Liu, "Autonomous Agents and Multi-Agent Systems," Explorations in Learning Self-Organization, and Adaptive Computation, Singapore: World Scientific, 2001.
[8] J. Liu, Y. Y. Tang, and Y. C. Cao, "An evolutionary autonomous agents approach to image feature extraction," IEEE Trans. on Evol. Comput., vol. 1, pp. 141-158, Feb. 1997
[9] Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Englewood Cliffs, NJ: Prentice-Hall, 1995.
[10] J. Liu, H. Jing, and Y. Y. Tang, "Multi-agent oriented constraint satisfaction," Artif. Intell., vol. 136, no. 1, pp. 101-144, 2002
[11] W. C. Zhong, J. Liu, M. Z. Xue, and L. C. Jiao, "A multiagent genetic algorithm for global numerical optimization," IEEE Trans. on System, Man, and Cybernetics-Part B., vol. 34, no. 2, pp. 1128-1141, 2004
[12] J.C.N. Climaco, J.M.F Cravereirinha and M.B. Pascoal, A bicriterion approach for routing problemsin multimedia networks, Networks, 41(4), 206-220.
[13] M. Gen, R. Cheng and L. Lin, Network models and optimization—Multiobjective genetic algorithm approach, Springer 2008
[14] C. Davies and P. Lingras, Genetic algorithms for rerouting shortest paths in dynamic and stochastic networks, European journal of Operation Research 144 (2003) 27-38
[15] M. Gen, R. Cheng, D. Wang, Genetic algorithms for solving shortest path problems, in: Proceedings of 1997 IEEE International Conference on Evolutionary Computing , 1997, pp. 401-406.
[16] C.W. Ahn and R.S. Ramakrishna, A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations, IEEE Trans. On Evo. Comp., Vol. 6, No. 6, December 2002.
[17] L. Panait and S. Luke, Cooperative Multi-Agent Learning: The State of the Art, Autonomous Agents and Multi-Agent Systems, 11, 387-434, 2005
[18] Lin, L. & Gen, M. (2007). Bicriteria network design problem using interactive adaptive weight GA and priority-based encoding method. IEEE Transactions on Evolutionary Computation in Reviewing.