International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2015): 6.391

Multi-Agent Genetic Algorithm for One Criteria Network Routing Optimization

Kabengele Mpunga Yannick

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

Abstract: ***Network design is widely used in practice in an ever wider range of applications. The shortest path models is one of the core models of network design and as we know well in one of NP-complete Problem in network design. This paper presents a multi-agent genetic algorithm to the shortest path routing problem; this algorithm is named MAGA-Rout. Priority-based encoding and decoding have been used for encoding the chromosome (strings). In this algorithm, we design a variant of weight mapping crossover operator based on neighborhood so as to obtain useful information from its neighbors and avoid random recombination. Results after experimentation for a sample test network have been presented to demonstrate the capabilities of the proposed approach to generate a much better quality of solution (route optimality) and much higher rate of convergence than other algorithms.***

Keywords: Multi-genetic algorithm, neighborhood weight mapping crossover, shortest path, multi-agent system

1. Introduction

The Shortest Path Problem (SPP) is to find the shortest path between two nodes, from specified source node S to another specified destination node D is a well-known problem in network analysis. Shortest path algorithms have been the subject of several researches, which has given rise to a certain number of algorithms for various conditions and constraints [1-3]. Recent studies have focused on time-dependent graphs [4, 5].

Moreover, the traditional routing problem has been a single objective problem having the aim to minimize the total distance or travel time. However, in many applications dealing with the design and efficient use of networks, the complexity of the social and economic environment requires the explicit consideration of objective functions other than cost or travel time. In other words, it is necessary to take into account that many real world problems are multi-criteria in nature. The objective functions related to cost, time, accessibility, environmental impact, reliability and risk are appropriated for selecting the most satisfactory route in many network optimization problems [6]. SPP is at the heart of network optimization problems and routing is one of the most important issues that have a significant impact on the network's performance.

Being the simplest network model, shortest path can capture most significant ingredientsof network optimization problem. Even though it is relatively easy to solve a shortestpath problem, the analysis and design of efficient algorithms requires considerableingenuity. Consequently, the study of models for shortest path is a natural beginningof network models for introducing innovative ideas, including the use of appropriatedata structures and data scaling to improve algorithmic performance in the worstcases [7].

Is easy to realize that single objective optimization is degenerate case of multiobjective optimization. The ability of multiobjective evolutionary algorithms to find multiple pareto-optimal solutions in one single run have made them attractive for solving problems with multiple and conflicting objectives.Shortest path routing problem involves a classical combinatorial optimization problem arising in many designs and planning contexts. Since genetic algorithms and other evolutionary algorithms promise solutions to such complicatedproblems, they have been used successfully in variouspractical applications [12-14]. Recently, multi-agent systems have been integrated with evolutionary algorithms to solve constraint satisfaction problems and combinatorial optimization problems with satisfactory results [8-11]. Given all these evidences, we conclude that Multi-agent genetic algorithm is suitable in solving large-scale complex problem like that of looking the optimal path between two nodes in network.

In this paper, a multi-agent genetic algorithm (MAGA-Rout) is proposed to optimize routing in the network. Four genetic operators of agents are designed or re-designed to achieve the goal: neighborhood competition operator and neighborhood weight mapping crossover realize competition and cooperation among agents. Mutation and self-learning operators increase the energy of agents by knowledge.

To validate the performance of MAGA-Rout, we run it for one object and the results was compared with result of the Munemoto's algorithm, Inagaki's algorithm and result of the algorithm proposed by C. Wook and R.S. Ramakrishna. The results show that MAGA-Rout has the ability to find optimal route with high speed.

The rest of this paper is organized as follows; we review related work on shortest path problem in section 2. In Section 3 we describe the routing problem formulation; the details of MAGA-Rout are described in section 4. The experiments are performed in section 5 followed by conclusion in section 6.

2. Shortest Path Related Work

The key objective of network routing problem is to find the optimal route between too given nodes in the network. Various methods have been proposed to give optimal path in the network [7, 15]. Here we reviewed the most important algorithms for solving this problem. Dijkstra provided an algorithm which solves single source shortest problem in O(n2) time, All edge must be greater than zero or equal to zero. Without worsening the run time, this algorithm can in fact compute the shortest paths from a given start point s to all other nodes [1, 7]. Bellman-Ford, provided an algorithm which can compute single source shortest paths in a weighted digraph (in this case some of the edge maybe have negative weights). Compare to Dijkstra's algorithm, which accomplishes the same problem with a lower running time, but requires edge weights to be nonnegative. Thus, Bellman-Ford is more used only when there are negative edge weights [7]. Floyd-Warshall, relaxed the constraint that all weights must be nonnegative, the algorithm provided can solve the all pairs shortest path problem in a weighted, directed graph by multiplying an adjacency-matrix representation of the graph multiple times with O(n3) time complexity [3, 7]. When the number of nodes in n and the number of edges is m, Eppstein [2] provides an algorithm that finds the k shortest paths in O(m+n log n + k) time.

Recently, serval methods received a great deal of attention to addressing the SPP, especially the GAs (and other evolutionary algorithms) because of their potential as optimization technique. In the network, they are often used to solve optimization problems in communication network optimization problems, including the effective approaches on the shortest path routing problem, multicasting routing problem, ATM bandwidth allocation problem, capacity and flow assignment problem, and the dynamic routing problem. It is noted that all these problems can be formulated as some sort of a combinatorial optimization problem [7].

In the same vein, we will mention the work of Gen et al.[16], It showed that Gas can solve the shortest path problem on a simple undirected graph with static weights. It would be better to mention that, the goal of the exercise was not to compute with conventional algorithm, but to show an encoding scheme that might be extendable to more difficult problems for which no-known algorithm exists. Since then many genetic algorithms have been developed, notably a genetic algorithm for shortest path routing problem and the sizing of populations by Ahn and al.[13].

The present study is based on the combination of multi-agent systems and genetic algorithms to solve the shortest path problem. The proposed approach is general enough to allow for many practical applications.

3. Problem Formulation

Given a network represented by a directed graph $G = (V, A)$ comprise a finite set of nodes $V = \{1,2,\dots,n\}$ and a collection of ordered pairs of nodes $A = \{(i,j),(k,l),\dots,(u,v)\}$ from V, called the edge set A. Any edge (u, v) represents a connection from node u to node v. cis a nonnegative value corresponding to the travel's cost from node u to node v. A path is a sequence of nodes $(v_1, v_2, \dots, v_{n-1}, v_n)$, where each node is distinct and $(v_{i-1}, v_i) \in A$, $1 < i \leq n$ [17]. Figure 1 shows an example of a directed graph with weighted edges. In the Fig.1,(1,2,3,4) is a path from node 1 to node 4. The shortest path problem between two nodes can be defined as the lowest cost path between these two nodes.

The shortest path problem is to find minimum cost z between to specific nodes, source node and destination node; here is the formulation in the form of integer programming [7]:

$$\min z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij} \qquad (1)$$

Where: nnumber of nodes; c_{ij} transmission cost of arc (i, j) and x_{ij} the link on an arc$(i, j) \in A$.

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

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

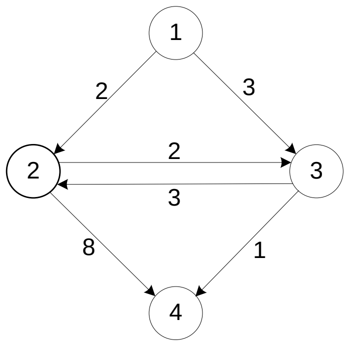


Figure 1: Example of directed graph

4. MAGA-Rout

4.1 Agent for shortest path problem

An agent is a computational mechanism that exhibits a high degree of autonomy, performing actions in its environment based on information (sensors, feedback) received from the environment for specific purpose. A multi-agent environment is one in which there is more than one agent, where they interact with one another, and further, where there are constraints on that environment such that agents may not at any given time know everything about the world that other agents know (including the internal states of the other agents themselves) [8, 19].

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-R, 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 of the shortest path problem that is under consideration; while the value of its energy is define by the fitness function in Eq. (16).

Definition 2. All agents stay in a lattice environment. Every agent has his fixed position in lattice environment and can interact only with its neighbors. Figure 2 shows us the environment for the agent with Lat_{size} xLat_{size} agents.

For an agent $Lat_{i,j}$ located at i^{th} line j^{th} row in the lattice, $i, j = 1,2,\dots,Lat_{size}$, then the Neighbors$_{i,j}$ of $Lat_{i,j}$ can be define as follows (17) and can be better visualizedin Figure3.

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

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

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

4.2 Genetic representation and initialization of agents

How to encode a solution of the shortest path problem in the network is the key issue for GAs. We must to consider these critical issues carefully when designing a new non-binary application string coding so as to design an effective GA chromosome. In MAGA-Rout, we use the priority-based encoding and decoding proposed by Linand Gen in Ref. [20]. 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 constructing a path among candidate.

Figure 4 illustrates 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 mostexisting genetic operators,thirdly,anypath has a corresponding encoding and fourthly,is searching in all solution space.

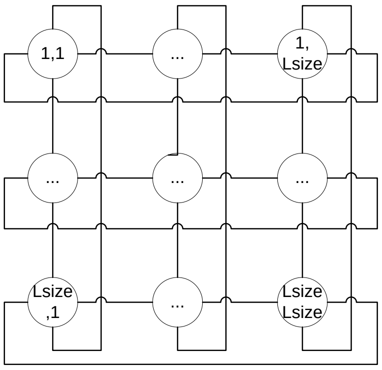


Figure 2: Lattice environment

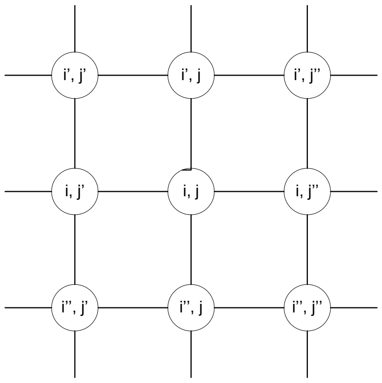


Figure 3: Agent's neighborhood

However it presented the disadvantage that n-to-1 mapping may occur for the encoding at some case. The random initialization is applied for this work.

4.3 Genetic operators of agents

MAGA-Rout has four genetic operators: neighborhood competition operator and neighborhood weight mapping crossover realize competition and cooperation among agents.Mutation and self-learning operators increase the energy of agents by knowledge. Given an agent $Lat_{i,j} = (n_1, n_2, \dots, n_n)$, we define maximum agent $Max_{i,j}$ in the neighborhood of $Lat_{i,j}$ as follow:

$Max_{i,j} = (m_1, m_2, \dots, m_n) \in Neighbors_{i,j}$ and $\forall Agent \in Neighbors_{i,j}$, $Energy(Agent) \leq Energy(Max_{i,j})$. Below the details of the four operators:

Neighborhood competition operator: In all agent lattice, this operator leads to competition between an agent $Lat_{i,j} = (n_1, n_2, \dots, n_n)$and the agent $Max_{i,j} = (m_1, m_2, \dots, m_n)$in its local environment.If an agent is winner that means its energy is more thanmaximum energy in its neighbors then it can live and will be left untouched. Otherwise is loser, it must die and a new agent that generated by strategy described below will occupy his lattice-point.

The generation of the new agent $New_{i,j} = (e_1, e_2, \dots, e_n)$is determined by (1)

$$e_k = \begin{cases} m_k\left(\mathrm{m_k} + \mathrm{U}(-1,1) * (\mathrm{m_k} - \mathrm{n_k})\right) \leq n_k \\ n_k \qquad\qquad\qquad \text{else}, \quad \mathrm{k} = 1, \dots, \mathrm{n} \end{cases} \quad (4)$$

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 revaluated.

Neighborhood weight mapping crossover: From the weight mapping crossover propose by Lin and Gen [20]; what can be considered as an extension of one-cut point crossover for permutation 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 remap the right segment based on the weight of other parent of right segment.

In this paper, we have designed a new weight mapping crossover operator based on neighborhood. An agent $Lat_{i,j} = (n_1, n_2, \dots, n_n)$ on the lattice will only cross with $Max_{i,j} = (m_1, m_2, \dots, m_n)$ so as to obtain useful information from its neighbors and avoid random recombination. In order to protect good patterns, we will not change $Max_{i,j} = (m_1, m_2, \dots, m_n)$. $\forall Lat_{i,j} = (n_1, n_2, \dots, n_n)$, if $U(0,1) < P_c$and $Energy(Lat_{i,j}) < Energy(Max_{i,j})$,we will perform the Neighborhood weight mapping crossover between the agent $Lat_{i,j}$ and $Max_{i,j}$then the newly generated agent will replace $Lat_{i,j}$. At the end of this operation, the energy of each agent is revaluated

Mutation operator: Performing insertion mutation operator [7] for each agent $Lat_{i,j}$, if $U(0,1) < P_m$.

Self-learning operator: In order to reduce the computational cost, the self-learning operator is only performed on the current best agent in each generation. $SlBest_t$ is generated by conducting neighbor-based mutation operator [7] on $CBest_t$. If the energy of agent find after performing self-learning operator is not more than the energy of current best agent, it is not considered. And the $SlBest_t$ are tanking the characteristics of $CBest_t$.

4.3 Implementation on MAGA-Rout

In this section we give the details about the framework of the proposed algorithm in Figure 5. First of all, neighborhood competition operator is performing on each agent to leave out from the lattice agents with lower energy. In order to explore the search space and maintain the diversity of the population, then, neighborhood weight mapping crossover and mutation operator are performing with probabilities P_c and P_m. After all, self-learning operator is only performed on the best agent in current generation because to avoid so much cost computational, but it played an important role in improving the performance of MAGA-Rout. According to the nature of multi-agent system [19], the agent in MAGA-R has incomplete information about the entire system. It only uses the information of its neighbors or itself, and the algorithm does not have any system global control.

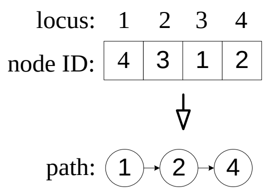


Figure 4: Priority-based chromosome and its decoded path

5. Experimental Results

In this section, we study the effectiveness of our approach. We compare the results obtained by MAGA-Rout with result of the Munemoto's [12] algorithm, Inagaki's [21] algorithm and result of the algorithm proposed by C. Wook and R.S. Ramakrishna [13] at the end each solution is compared with Dijkstra's SP [1] solution as it provided a reference point. MAGA-Rout is running on Core(TM) i5-4200U processor (1.60GHz 2.30 GHz). We show that our algorithm successfully finds the optimal route in the network and it is competitive with other approaches.

The MAGA-Rout algorithm has been written in C language, as regards MAGA-Rout, we employed a trial-and-error procedure and then selected the parameter values giving good results for all simulation data sets. Thus, we set P_c 0.6, P_m 0.4, the population size was 100 this means the agent lattice dimension was 10 and number of generations 100.

The simulation studies has been done in weighted network topology (with 20 nodes) as shows by in Figure 6.

For comparison purposes, which is based on the performance, the population size of the algorithm which has been compared was taken to be the same as the number of nodes in the network while for MAGA-Rout because of his nature, the agent lattice dimension has been set to 5 this means population size was 25, and the fitness function was adapted.

For the source node and the destination node considered, it is seen that the path computed by the algorithm proposed by C. Wook and R.S. Ramakrishna and MAGA-R coincides with that found by Dijkstra's algorithm. The latter is the reference for optimal shortest path. **Figure 7** compares the speed of convergence, it is seen that MAGA-Rout exhibits the fastest rate of convergence because the number of generation up to convergence is the smallest. The algorithm converging through smaller generations has better convergence performance.

Table 1: Comparison results for the paths found by each algorithm

Algorithm proposed by	*Optimal path find*	*Total path costs*
Inagaki	1-4-9-14-14-20	234
Munemoto	1-2-3-8-14-20	187
C. Wook and R.S. Ramakrishna	1-3-8-14-20	142
MAGA-R	1-3-8-14-20	142

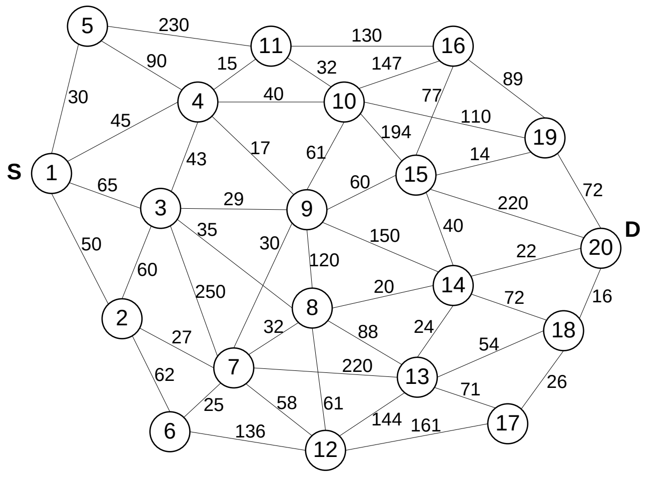
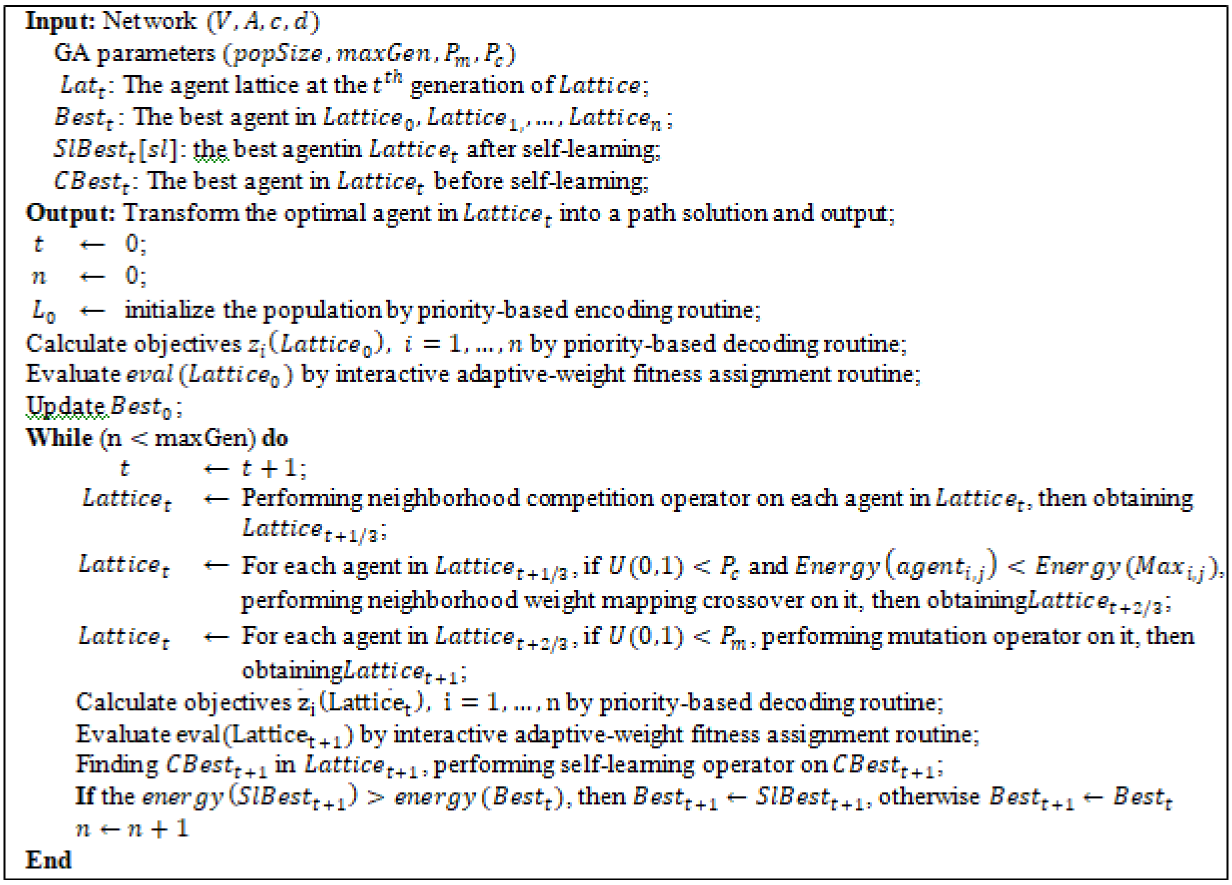


Figure 6: Undirected graph with 20 nodes and 48 edges



```
Input: Network $(V, A, c, d)$
    GA parameters $(popSize, maxGen, P_m, P_c)$
    $Lat_t$: The agent lattice at the $t^{th}$ generation of $Lattice$;
    $Best_t$: The best agent in $Lattice_0, Lattice_1, \dots, Lattice_n$;
    $SlBest_t[sl]$: the best agentin $Lattice_t$ after self-learning;
    $CBest_t$: The best agent in $Lattice_t$ before self-learning;
Output: Transform the optimal agent in $Lattice_t$ into a path solution and output;
 $t \leftarrow 0$;
 $n \leftarrow 0$;
 $L_0 \leftarrow$ initialize the population by priority-based encoding routine;
Calculate objectives $z_i(Lattice_0)$, $i = 1, \dots, n$ by priority-based decoding routine;
Evaluate $eval(Lattice_0)$ by interactive adaptive-weight fitness assignment routine;
Update $Best_0$;
While (n < maxGen) do
        $t \leftarrow t+1$;
    $Lattice_t \leftarrow$ Performing neighborhood competition operator on each agent in $Lattice_t$, then obtaining
        $Lattice_{t+1/3}$;
    $Lattice_t \leftarrow$ For each agent in $Lattice_{t+1/3}$, if $U(0,1) < P_c$ and $Energy(agent_{i,j}) < Energy(Max_{i,j})$,
        performing neighborhood weight mapping crossover on it, then obtaining $Lattice_{t+2/3}$;
    $Lattice_t \leftarrow$ For each agent in $Lattice_{t+2/3}$, if $U(0,1) < P_m$, performing mutation operator on it, then
        obtaining $Lattice_{t+1}$;
    Calculate objectives $z_i(Lattice_t)$, $i = 1, \dots, n$ by priority-based decoding routine;
    Evaluate $eval(Lattice_{t+1})$ by interactive adaptive-weight fitness assignment routine;
    Finding $CBest_{t+1}$ in $Lattice_{t+1}$, performing self-learning operator on $CBest_{t+1}$;
    If the $energy(SlBest_{t+1}) > energy(Best_t)$, then $Best_{t+1} \leftarrow SlBest_{t+1}$, otherwise $Best_{t+1} \leftarrow Best_t$
    $n \leftarrow n+1$
End
```

Figure 5: Pseudo-code of the MAGA-R algorithm

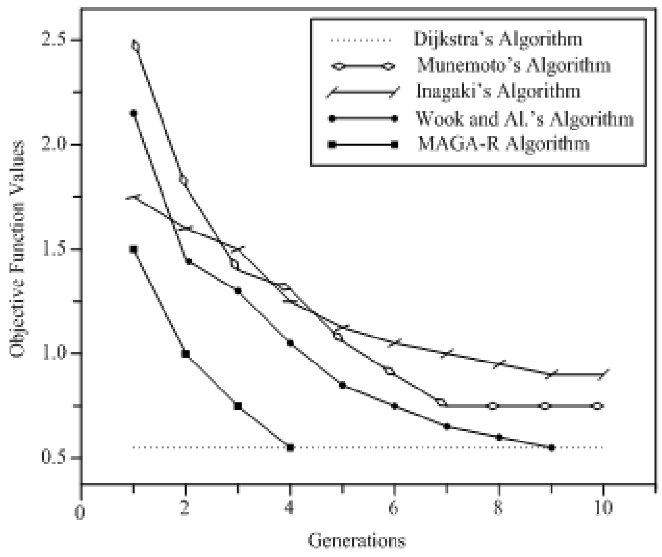


Figure 7: Convergence property of each algorithm

6. Conclusion

This paper proposes the algorithm named as MAGA-Rout to optimize the routing in the network. The experiments in single optimization problem show MAGA-Rout has the large ability to find optimal solutions with a higher convergence rate. It is taking advantages of the combination of evolutionary computing and multi-agent system to outperform the existing routing algorithm. In addition, it is confirmed that MAGA-Rout can continue to find an optimal path in the networks with different number of nodes and edges. However this performance largely depends to the quality and number of chromosomes at initialization step.

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

References

[1] E.W. Dijkstra, A note on two papers in connection with graphs, Numeriske Mathematics 1 (1959) 269-271.
[2] D. Eppstein, Finding the k shortest paths, SIAM Journal on Computing 28 (2) (1998) 653-674.
[3] R.W. Floyd, Algorithm 97: 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.
[6] J.C.N. Climaco, J.M.F Cravereirinha and M.B. Pascoal, A bicriterion approach for routing problems in multimedia networks, Networks, 41(4), 206-220.
[7] M. Gen, R. Cheng and L. Lin, Network models and optimization—Multiobjective genetic algorithm approach, Springer 2008

[8] W. Zhong, J. liu, M. Xue and L. Jiao, Global numerical optimization using multi-agent genetic algorithm, ICCIMA 2003.
[9] X. Pan, L. Jiao and F. Liu, An improved multi-agent genetic algorithm for numerical optimization, Springer Science+Business Media B.V. 2010.
[10] Z. Li and J. Liu, Multi-agent genetic algorithm for community detection in complex networks, PhysicaA 449 (2016) 336-347.
[11] J. Liu, W. Zhong and L. Jiao, A multiagent evolutionary algorithm for combinatorial optimization problems, IEEE Trans. Syst. Man Cybern. B 40 (1) (2010) 229-240.
[12] M. Munemoto, Y. Takai, and Y. Sato, “A migration scheme for the genetic adaptive routing algorithm,” in Proc. IEEE Int. Conf. Systems, Man, and Cybernetics, 1998, pp. 2774–2779.
[13] 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.
[14] C. Chitra and P. Subbaraj, A nondominated sorting genetic algorithm for shortest Path routing problem, IJCIE, Vol:4, No:8, 2010
[15] 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.
[16] 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.
[17] X. Junming, A first course in graph theory, Science press, Beijing, 2015
[18] Lo, C. & Chang, W. (2000). A multiobjective hybrid genetic algorithm for the capacitated multipoint network design problem. IEEE Transaction on Systems, Man and Cybernestic, 30(3).
[19] L. Panait and S. Luke, Cooperative Multi-Agent Learning: The State of the Art, Autonomous Agents and Multi-Agent Systems, 11, 387-434,2005
[20] 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.
[21] J. Inagaki, M. Haseyama, and H. Kitajima, “A genetic algorithm for determining multiple routes and its applications,” in Proc. IEEE Int. Symp. Circuits and Systems, 1999, pp. 137–140.