韓國 컴퓨터情報學會 論文誌
第10卷 第1號, 2005. 3.

2005-3-1-2-3

유전자 알고리즘을 이용한 혼합 네트워크에서의 Chinese Postman Problem 해법

전병현*, 강명주**, 한치근***

A Genetic Algorithm for the Chinese Postman Problem on the Mixed Networks

Byung-Hyun Jun *, Myung-Ju Kang **, Chi-Geun Han ***

요 약

Chinese Postman Problem(CPP)는 주어진 네트워크에서 모든 에지나 아크를 적어도 한번씩 경유하는 최단 경로를 찾는 문제이다. 혼합네트워크에서의 CPP(MCPP)는 기존의 CPP를 일반화시킨 문제로 현실 세계에서 많은 응용 부분들을 가지고 있으며, MCPP는 NP-Complete로 알려져 있다. 본 논문에서는 Floyd 알고리즘을 이용하여 구성된 가상 아크를 이용하여 혼합네트워크를 대칭네트워크로 변환 후 근사최적해를 탐색하는데 효율적인 유전자 알고리즘을 적용한다. 본 논문에서는 유전자 알고리즘에 적용하기 위해 경로 문자열과 에지, 아크를 구분하기 위한 문자열의 쌍으로 구성된 염색체 구조, 인코딩 및 디코딩 방법을 제안한다. 또한 보정 방법으로 Power Law 보정 방법과 Logarithmic 보정 방법을 사용하고 비교 분석하였다. 본 논문에서는 기존의 MIXED2 알고리즘과 제안된 유전자 알고리즘과의 성능 비교를 하였다. 에지가 많은 혼합 네트워크인 경우 제안된 유전자 알고리즘이 좋은 결과를 얻고, Logarithmic 보정 방법 보다 Power Law 보정 방법을 사용할 경우 좋은 결과를 얻을 수 있음을 알 수 있었다.

Abstract

Chinese Postman Problem (CPP) is a problem that finds a shortest tour traversing all edges or arcs at least once in a given network. The Chinese Postman Problem on Mixed networks (MCPP) is a practical generalization of the classical CPP and it has many real-world applications. The MCPP has been shown to be NP-complete. In this paper, we transform a mixed network into a symmetric network using virtual arcs that are shortest paths by Floyd's algorithm. With the transformed network, we propose a Genetic Algorithm (GA) that converges to a near optimal solution quickly by a multi-directional search technique. We study the chromosome structure used in the GA and it consists of a path string and an encoding string. An encoding method, a decoding method, and some genetic operators that are needed when the MCPP is solved using the proposed GA are studied. In addition, two scaling methods are used in proposed GA. We compare the performance of the GA with an existing Modified MIXED2 algorithm (Pearn et al., 1995). In the simulation results, the proposed method is better than the existing methods in case the network has many edges, the Power Law scaling method is better than the Logarithmic scaling method.

▸ Keyword : Chinese Postman Problem, CPP, MCPP, Genetic Algorithm

---

• 제1저자 : 전병현
• 접수일 : 2005.01.14, 심사완료일 : 2005.03.05
* 경희대학교 전자계산공학과 박사과정 ** 청강문화산업대학 컴퓨터소프트웨어과 교수 *** 경희대학교 전자계산공학과 교수

Ⅰ. 서 론

The Chinese Postman Problem (CPP) is to find a shortest path that starts at some node, visits each edge of the network at least once, and returns to the starting node in a given network G=(V, A) whose arcs (i, j) have an associated nonnegative weight, where V is a set of vertices and A is a set of arcs. The CPP on a directed network is reduced to a minimum cost flow problem and the CPP on an undirected network is reduced to a nonbipartite weighted matching problem. These problems are solved in polynomial time. But the CPP on a mixed network (i.e., some arcs are directed and the others are undirected) is NP-complete[1, 7].

Applications of the MCPP include: patrolling streets by police, routing of newspaper or mail delivery vehicles, routing of street sweepers, household refuse collection vehicles, fuel oil or gas delivery to households, snow plows, school buses, spraying roads with salt, and inspection of electric power lines[9,10].

For existing methods, there are branch-and-bound algorithm using Lagrangian relaxations via integer programming and a method to convert the MCPP into a flow with gains problem via linear programming and cutting plane techniques. Because of the problem's complexity, both approaches are computationally inefficient, hence problems of great size can't be solved rapidly. So heuristic methods have been proposed to solve the MCPP approximately. In this paper, we proposed a Genetic Algorithm (GA) that converges near optimal solution quickly by multi-directional search technique.

Ⅱ. Chinese Postman Problem on Mixed Networks

2.1 Problem Definition

A mixed network is shown in the following figure 1.

Suppose that a post office is located at node F, there is a mailbox box at each arc's or edge's center, and postman gathers mail from every mailbox. Then the postman traverses every road (arcs or edges) at least once starting from F and he returns to F.

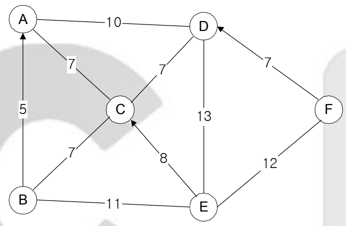

Fig 1. A Mixed network Modeling Example

Like that, the MCPP is to find a shortest path that starts at some node, visits each edge and arc of the network at least once, and returns to the starting node in a given mixed network G=(V, E, A), where E is a set of edges.

2.2 Problem Formulation

We define the following notations:

V : Set of vertices

E : Set of edges (or Undirected Edges)

A : Set of arcs (or Directed Edges)

d(i, j) : Distance of edge (i, j) ∈ E (>= 0)

c(i, j) : Distance of arc (i, j) ∈ A (>= 0)

xij : Number of passes of edge (i, j) ∈ E (>= 1)

yij : Number of passes of arc (i, j) ∈ A (>= 1)

On the mixed network G=(V, E, A), Routing cost is given as the following formulation:

$$C = \sum_{(i,j)\in E} d(i,j)x_{ij} + \sum_{(i,j)\in A} c(i,j)y_{ij}$$

The MCPP is to find a path to minimize cost C.

2.3 Existing Methods

In 1976, the MCPP has been shown to be NP-complete by Papadimitriou[1].

Christofide et al.[8] presented an integer programming formulation of the problem, and developed an exact algorithm to solve the MCPP optimally. The algorithm is essentially based on a branch-and-bound algorithm using Lagrangian relaxations. Minieka [4] presented a transformation converting the MCPP into a flow with gains problem, which allows the MCPP to be solved optimally via linear programming and cutting plane techniques. Because of the problem's complexity, as the size of the problem is growing, both approaches are computationally inefficient. So heuristic methods have been proposed to solve the MCPP approximately. Frederickson[4] and Pearn et al.[10] converted the MCPP into a symmetric ne

twork that was applied to a method solving the problem in polynomial time presented by Edmond[6], and then solved problem using a matching technique. These methods have many processes to edges, so take long time on the mixed network that has more edges than arcs.

III. A genetic algorithm for the MCPP

3.1 General Description of Genetic Algorithms

A chromosome is coded to represent the problem. Initially, the chromosomes are generated randomly as many as population size. Among them, the chromosomes more fitted with the given environment are propagated to the next generation as many as possible. As the chromosomes are evolved, dominant chromosomes are selected, decoded, and evaluated. The general genetic algorithm is described as follows:

```
Procedure : Genetic Algorithm
begin
  t = 0;
  initialize P(t);
  evaluate P(t);
  while (not termination condition) do
    t = t+1;
    select P(t) from P(t-1);
    recombine P(t);
    evaluate P(t);
  end
end
```

A variable t that represents the generation is used for a termination condition. And P(t) represents t-th population set.

3.2 Network Transformation

Existing methods[4, 10] use a minimum cost flow technique to transform the mixed network into a symmetric network. We transform the mixed network into a symmetric network using virtual arc with the lowest cost.

To make a symmetric network, we create a virtual arc which is the opposite direction of one directional arc. It has lowest cost which was solved using Floyd's algorithm in the mixed network. The virtual arc is similar to the routing path between two required edges in rural postman problem[11,12].

Figure 2 shows a simple mixed network example and table 1 is cost matrix for this network.

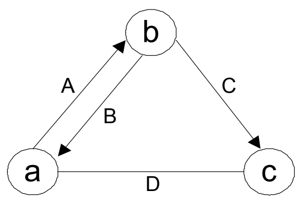

Fig 2. Mixed network example for transformation

Table 1. Cost matrix table

From \ To	a	b	c
a	0	5	5
b	3	0	7
c	5	∞	0

The example network requires arc (c, b) to be a symmetric network. Therefore, we must make a virtual arc (c, b). Table 2 is virtual arc table for this example network.

Table 2. Virtual arc table

Nstart	*Nend*	*Pnode_list*	*Cost*
c	b	a	10

3.3 Coding

A genetic algorithm for the MCPP requires a chromosome structure that represents a solution. The chromosome is composed of a path string (PS) and an encoding string (ES).

A path in the problem should pass all edges or arcs at least once on the given network. Hence, an element of PS represents an edge or an arc on the given mixed network and the order of the elements in the string is the same as the visiting order of the edges or arcs.

An encoding string is required to distinguish an edge and an arc and to represent an edge's direction simultaneously. An edge element has 0 or 1 which denotes forward direction and opposite direction respectively. And an arc element has 2 in ES. An edge's direction is decided by cost matrix. Forward direction is path that appears upper triangular part of a distance matrix such as <Table 1> In (figure 2) example, path that goes node **a** to **c** is forward direction for edge (a, c)

The sizes of a path string and an encoding string are |A| + |E|.

(Figure 3) illustrates a chromosome composed of PS and ES for (figure 2) example.

Path String	A	B	C	D
Encoding String	2	2	2	1

|*A*| + |*E*|

Fig 3. Example Coding String

Initial population is composed of chromosomes that have a pair of elements chromosome, randomly positioned.

3.4 Decoding

To evaluate a coded chromosome, it must be decoded. The decoding method is to translate a value in a path string's element into two nodes and to arrange the order of path string's nodes.

Fig 4 illustrates that the path string's element is translated into two nodes.

P_1	P_2	P_3	....	P_n
N_{11} ⋮ N_{12}	N_{21} ⋮ N_{22}	N_{31} ⋮ N_{32}	....	N_{n1} ⋮ N_{n2}

Fig 4. Decoding of a path string

Suppose that a path P is P1→P2→...→Pn→P1, and each Pk, k=1,...,n may be an arc or edge. If Pk is composed of two nodes (Nk1, Nk2) which denotes an arc or edge (i, j), then start node Nk1 and end node Nk2 is defined as follows:

$$\begin{cases} N_{k1} = i, N_{k2} = j & \text{if } ES_k = 0, 2 \\ N_{k1} = j, N_{k2} = i & \text{if } ES_k = 1 \end{cases}$$

By checking the value of ESk, we can decide the order of nodes in the path.

Fig 5 illustrates decoding for the figure 3 coding string.

PS	A		B		C		D	
ES	2		2		2		1	
node	a	b	b	a	b	c	c	a

Fig 5. Decoding for the figure 3 coding string

3.5 Evaluation and Fitness

After decoding the structure of a chromosome, we compute the cost of the chromosome using the order of the nodes.

The MCPP is to find a shortest path that starts at some node, visits each edge and arc of the network at least once, and returns to the starting node. So, we can define the evaluation function of the i-th chromosome as follows:

$$C_i = \sum_{k=1}^{n-1} \{ d(N_{k1}, N_{k2}) + d(N_{k2}, N_{(k+1)1}) \} + \{ d(N_{n1}, N_{n2}) + d(N_{n2}, N_{11}) \}$$

The evaluation function Ci is composed of cost of the path Pk, cost between path Pk and Pk+1, and cost of returning to the starting node.

For the figure 5 decoding example, evaluation path is A→B→C→D. In details, start/end node is a. Evaluation path is A(**a**→**b**)→B(**b**→**a**)→C(**b**→**c**)→D(**c**→**a**), that is A(**a**→**b**)→B(**b**→**a**)→(**a**→**b**)→C(**b**→**c**)→D(**c**→**a**). Path (**a**, **b**) is additional to build a way from node a to node b between arc B and arc C.

The fitness value indicates that how much a chromosome has dominant characteristics. A chromosome that has the largest fitness value is the nearest optimal solution.

The fitness value and the solution are in inverse proportion to each other, so the fitness function of the i-th chromosome is defined as follows:

$$F(C_i) = \frac{1}{C_i}$$

To generate next population, we use Roulette Wheel method and select chromosomes from current population according to the fitness function values.

3.6 Scaling

Difference between a dominant chromosome and a recessive chromosome causes to determine the characteristics of the next generation and affects convergence speed toward a good solution.

If the difference is too big, it causes premature convergence that chromosomes converge a local minimum and can't evolve. To the opposite case, it causes a genetic drift phenomena that chromosomes can't converge to a solution. To solve these problems, a scaling method can be used.

We used the Power Law scaling method ($f'_k = f^a_k$) and the Logarithmic scaling method ($f'_k = \log_{10}(f_k)$) where f^a_k is a power of f_k. In the Power Law scaling method, a is a variable according to the problem. The difference between the scaled dominant chromosome and the scaled recessive chromosome increases or decreases according to value of a. If a is closed to 0, then the difference is closed to 0. If a is greater than 1, then the difference is bigger. In this paper, a is greater than or equal to 2 and less than or equal to 5.

3.7 Genetic Operations

Genetic operations play a part in propagating dominant characteristics to the next population as much as possible. A crossover operation and mutation operations are used.

To solve the MCPP, we used Partially Matched Crossover (PMX) that has been proposed by Goldberg and Lingle[2]. We used exchange, reverse, and inverse operation as mutation operators. The inverse operation changes edge's encoding string value of a chromosome's element, that is 0 to 1, 1 to 0.

Ⅳ. Computational Results

4.1 Simulation Environment

We compared an existing method with the proposed genetic algorithm. Also we compared a genetic algorithm using the Power Law Scaling with a genetic algorithm using the Logarithmic Scaling. Table 3 shows the test problems.

Table 3. Test Problems

Problem	Number of nodes	Number of arcs	Number of edges
1	6	3	6
2	9	6	6
3	6	2	6
4	9	8	8
5	9	11	6
6	11	10	16
7	31	21	18
8	16	4	19
9	17	4	21
10	18	3	25

We compared GAs with the Modified MIXED2 algorithm that is a heuristic search method and has been proposed in 1995[6].

In the GAs, population size is 100, number of generation is 1000, and crossover operation rate is between 0.7 and 0.9, each mutation operation rate is 0.0 or 0.3, and a is between 2.0 and 5.0.

4.2 Comparison with the Modified MIXED2

In table 4, the performance of Modified MIXED2 (MM2) and a proposed genetic algorithm using Power Law Scaling(GA-PLS) are compared. The result contains a minimum cost for each problem and a minimum time to find the minimum cost.

Table 4. Comparison with the Modified MIXED2

Problem	MM2		GA-PLS	
	Cost	Time(sec)	Cost	Time(sec)
1	45	0.571	45	0.1
2	96	0.139	96	0.23
3	81	0.35	64	0.1
4	82	1.78	80	0.24
5	117	1.312	118	0.41
6	228	4.176	222	2.194
7	178	12.488	175	10.5
8	64	9.117	62	3.255
9	58	20.611	59	3.107
10	55	24.606	55	9.23

In the problems 8, 9, 10, a proposed genetic algorithm is faster than the Modified MIXED2. These problems have much more edges than arcs. Note that the Modified MIXED2 has heavy edge processing (i.e. transformation of edges into arcs, minimum cost flow, etc.). But a proposed genetic algorithm is independent on the numbers of edges or arcs.

In the real world, two-way roads are more common than one-way roads. So a proposed genetic algorithm is better than the existing method such as the Modified MIXED2.

4.3 Comparison with GAs

In the genetic algorithms, obtained optimal solutions are dependent on scaling functions of the fitness value.

In table 3, the performance of the Power Law scaling function and the Logarithmic scaling function on the MCPP is described.

Table 3. Comparison with GAs

Problem	Power Law Scaling		Logarithmic Scaling	
	Cost	Time(sec)	Cost	Time(sec)
1	45	0.1	45	2.624
2	96	0.231	17	7.16
3	81	0.1	70	0.11
4	82	0.24	107	2.147
5	117	0.41	158	3.316
6	228	2.194	328	5.138
7	178	10.5	446	7.271
8	64	3.255	101	1.499
9	58	3.107	95	0.8
10	55	9.23	91	1.389

The Logarithmic scaling method is worse than the Power Law scaling method. Because of the log function's characteristic in scaling function $f'_k = \log(f_k)$, the small difference between dominant and recessive chromosomes causes the genetic drift phenomena.

Ⅴ. Conclusion

In this paper, we presented an efficient genetic algorithm to solve the MCPP. We studied a simple network transformation method using virtual arc, and chromosome structure used in GA that consists of a path string and an encoding string. An encoding method and a decoding method are studied, and we used some genetic operators such as PMX as crossover operator, and exchange, reverse, and inverse operation as mutation operators.

10 test problems which has 6-31 nodes, 2-21 arcs, 6-25 edges are generated for computational experiments. We compare the performance of the GA with an existing Modified MIXED2 algorithm for the same problems. It has been observed that the proposed GA is better than the Modified MIXED2 algorithm and, especially, the GA is robust for the problems that have many edges. For the scaling methods, we found that Power Law Scaling is better than Logarithmic Scaling in the proposed GA.

References

[1] C. H. Papadimitriou, 1976, On the Complexity of Edge Traversing, ACM, 23:544-554.

[2] D. E. Goldburg and R. Lingle, 1985, Alleles, loci, and the Traveling Salesman Problem, Proceedings of an International Conference on Genetic Algorithms and Their Applications, 145-159.

[3] D. E. Goldburg, 1989, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company.

[4] E. Minieka, 1979, The Chinese Postman Problem for Mixed Networks, Management Science, 25:643-648.

[5] G. N. Fredericson, 1979, Approximation Algorithms for some Postman Problems, ACM, 26:538-554.

[6] J. Edmonds and E. Johnson, 1973, Matching Euler Tours and the Chinese Postman, Math. Program., 5:88-124.

[7] M. R. Garey and D. S. Johnson, 1979, Computers and Interactability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company.

[8] N. Chistofides, E. Benavent, V. Campos, A. Corbran, and E. Mota, 1983, An Optimal Method for the Mixed Postman Problem, Proceedings of the 11th IFIP conference, Copenhagen, Denmark :641-649.

[9] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, 1993, Network Flows, Theory, Algorithms, and Applications, Prentice Hall, 740-744.

[10] W. L. Pearn and C. M. Liu, 1995, Algorithms for the Chinese Postman Problem on Mixed Networks, Computer & Operations Research, 22-5:479-489.

[11] 강명주, "Rural Postman Problem 해법을 위한 향상된 Simulated Annealing 알고리즘", 한국컴퓨터정보학회 논문지, 제6권, 1호, 2001. 3.

[12] 강명주, "Rural Postman Problem 해법을 위한 Iterative Local Search 알고리즘", 한국컴퓨터정보학회 논문지, 제7권, 1호, 2002. 3.

저 자 소 개

전 병 헌
1996년 경희대학교 전자계산공학과 학사
1998년 경희대학교 전자계산공학과 석사
1998년~현재 경희대학교 전자계산공학과 박사과정
<관심분야> 알고리즘, 계산 이론, 임베디드 시스템, 메타휴리스틱 알고리즘, 웹서비스

강 명 주
1993년 경희대학교 전자계산공학과 학사
1995년 경희대학교 전자계산공학과 석사
1998년 경희대학교 전자계산공학과 박사
2000년~현재 청강문화산업대학 컴퓨터소프트웨어과 교수
<관심분야> 네트워크 디자인, 메타휴리스틱 알고리즘, 게임알고리즘

한 치 근
1983년 서울대학교 산업공학과 학사
1985년 서울대학교 산업공학과 석사
1988년 미국 Pennsylvania 주립대학 Computer Science 석사
1991년 미국 Pennsylvania 주립대학 Computer Science 박사
1992년~현재 경희대학교 전자계산공학과 교수
<관심분야> 계산 이론, 알고리즘, 수치 해석, 병렬 처리