韓國 컴퓨터情報學會 論文誌
第10卷 第5號, 2005. 11.

2005-10-5-2-9

MOX 교차 연산자를 이용한 Rural Postman Problem with Time Windows 해법

강 명 주*

A Genetic Algorithm using A Modified Order Exchange Crossover for Rural Postman Problem with Time Windows

Myung-Ju Kang*

요 약

본 논문에서는 유전자 알고리즘을 이용한 Rural Postman Problem with Time Windows(RPPTW) 해법을 위해 유전자 알고리즘에 사용되는 교차 연산자를 제안하고, 기존의 교차 연산자와 비교한다. RPPTW는 다중목적 최적화 문제로서, Rural Postman Problem(RPP)에 서비스 시간 제한을 위한 시간 윈도우(Time Windows)를 두고 제한된 시간 내에 서비스를 받을 수 있도록 구성된 문제이다. 따라서, RPPTW는 주어진 시간 내에 서비스를 받으면서 최소 비용으로 라우팅을 하는 다중 목적 최적화 문제이다.

다중 목적 최적화 문제인 RPPTW를 해결하기 위해서는 Pareto-optimal 집합을 구해야 한다. Pareto-optimal 집합은 각 목적값들의 우수성을 비교할 수 없는 집합이다. 본 논문에서는 12개의 임의로 생성된 문제들에 대해 3개의 교차 연산자를 사용하여 실험을 하여 그 결과를 비교하였다. 본 논문에서 사용된 교차 연산자들은 PMX(Partially Matched Exchange), OX(Order Exchange), 그리고 본 논문에서 제안한 MOX(Modified Order Exchange)이다. 각 문제들에 대한 실험 결과를 통해서 RPPTW를 위한 교차 연산자 중에 본 논문에서 제안한 MOX 방법이 효율적임를 알 수 있었다.

Abstract

This paper describes a genetic algorithm and compares three crossover operators for Rural Postman Problem with Time Windows (RPPTW). The RPPTW which is a multiobjective optimization problem, is an extension of Rural Postman Problem(RPP) in which some service places (located at edge) require service time windows that consist of earliest time and latest time. Hence, RPPTW is a multiobject optimization problem that has minimal routing cost being serviced within the given time at each service place.

To solve the RPPTW which is a multiobjective optimization problem, we obtain a Pareto-optimal set that the superiority of each objective can not be compared. This paper performs experiments using three crossovers for 12 randomly generated test problems and compares the results. The crossovers using in this paper are Partially Matched Exchange(PMX), Order Exchange(OX), and Modified Order Exchange(MOX) which is proposed in this paper. For each test problem, the results show the efficacy of MOX method for RPPTW.

▸ Keyword : MOX, 유전자알고리즘(genetic algorithm), Rural Postman Problem with Time Windows

---

• 제1저자 : 강명주
• 접수일 : 2005.10.10, 심사완료일 : 2005.10.29
* 청강문화산업대학 컴퓨터소프트웨어과

Ⅰ. Introduction

The Rural Postman Problem with Time Windows (RPPTW) is considered in this paper. The RPPTW is an extension of the Rural Postman Problem(RPP) in which some service places (located at edge) require service time windows that consist of earliest time and latest time[1]. The RPPTW is defined as follows. If a service man arrives at a service place beforre the earliest time, he must wait until the service place is ready for the service and the cost of traveling for the traveling service man increases. Also, it a service man arrives at a service place after the latest time, some cost penalty would be given to the service man from the service place. Hence, service man would like to arrive at the service place within the given time windows in order to reduce his total traveling cost and total penalty. So, the RPPTW is a multiobjective optimization problem.

A studty of multiobjective optimization using genetic algorithms was proposed by Rosenberg in 1960's and Schaffer tried Vector Evaluated Genetic Algorithm program in 1984[2]. In recent, the studies of multiobjective optimization have been applied to shortest path problem on acyclic network[3], analysis for water quality[4], vehicle routing problem[5, 6] and so on.

The multiobjective optimization problems are difficult to obtain the optimal solution. Single objective optimization problems have a single optimal point, whereas multiobjective optimization problems have a set of optimal points known as the Pareto-optimal set. Each point in the Pareto-optimal set is optimal in the sense that a component of the cost vector is nondominated by at least one of the remaining components. The Pareto-optimal set is defined as follows[7]:

Definition 1. Inferiority
A vector $u = (u_1, \ldots, u_n)$ is said to be inferiority to $v = (v_1, \ldots, v_n)$ iff u is partially less than v, i.e.,
$\forall i = 1, \ldots, n,\ u_i \le v_i \land \exists i = 1, \ldots, n : u_i < v_i$.

Definition 2. Superiority
A vector $u = (u_1, \ldots, u_n)$ is said to be superior to $v = (v_1, \ldots, v_n)$ iff u is superior to v, i.e.,
$\forall i = 1, \ldots, n,\ u_i \ge v_i \land \exists i = 1, \ldots, n : u_i > v_i$.

Definition 3. Non-inferiority
Vector $u = (u_1, \ldots, u_n)$ and $v = (v_1, \ldots, v_n)$ are said to be non-inferior to one another if u is neither inferior nor superior to v.

That is, Each element in the Pareto-optimal set constitutes a non-inferior solution to the multiobjective optimization problem.

This paper describes a genetic algorithm for the RPPTW and compares the performances of crossovers. In such RPPTW as Traveling Salesman Problem (TSP), the order of strings in each chromosome is important. Hence, we use partially matched exchange (PMX) proposed by Goldberg and Lingle[2,8], order exchange(OX) proposed by Davis[2,8], and modified order exchange(MOX) proposed in this paper.

Ⅱ. Multiobjective Optimization with Genetic Algorithms

In recent, Genetic Algorithms (GAs) emerged as one of the most effective and robust search algorithms. In a single objective optimization and search, the desired end result is a single solution, whereas, in a multiobjective optimization, the goal is to find a set of solutions distributed all along the Pareto set of the different objective functions[7]. As GAs always work with a population of solutions while progressing from one generation to the other, rather than a single solution at a time, they are particularly attractive for multiobjective optimization which deals with a set of Pareto solutions rather than a single solution. Once the final Pareto set is found, we can choose a suitable solution from this set according to our purpose. In this paper, to solve multiobjective optimization problem, following two objectives are considered to obtain the Pareto-optimal set.

- *Minimizing the total routing cost.*
- *Minimizing the total penalty.*

III. Rural Postman Problem with Time Windows and Objective Function

RPP is to find the shortest traveling path that passes a set of edges of a given graph at least once. (Figure 1) shows a traveling path of RPP. In this (figure 1), *a-a′, b-b′, c-c′, d-d′*, and *e-e′* are the edges in E′(⊆ E) that must be passed at least once in the path. *a′~b, b′~c, c′~d*, and *d′~e* are the paths that should be decided in order to find the shortest traveling path.

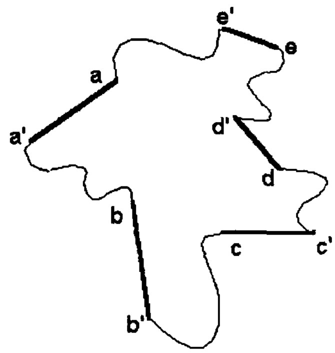

Figure 1. The traveling path

RPPTW is an extension of the RPP in which some service places (located at edge) require service time windows that consist of earliest and latest time[1]. If a service man arrives at a service place before the earliest time, he must wait until the service place is ready for the service and the cost of traveling for the traveling service man increases. Also, if a service man arrives at a service place after the latest time, some cost penalty would be given to the service man from the service place. Hence, service man would like to arrive at the service place within the given time windows in order to reduce his total traveling cost and total penalty.

The followings are parameters, an objective function, and a fitness function for RPPTW.

Parameters:

- $d_{e_i^2, e_{i+1}^1}$: Cost calculated by Dijkstra algorithm from the second node of the i^{th} edge (∈E′) of a tour to the first node of the $(i+1)^{st}$ edge of the tour (∈E′), where e_i = (e_i^1, e_i^2) and $e_i^1, e_i^2 \in V$.
- c_{e_i} : Cost in the ith edge of the tour, where $e_i \in E'$.
- a_i : Arrival time at the service place in the i^{th} edge of the tour.
- p_i : Penalty at the service place in the i^{th} edge of the tour.
- TW_i : Time Windows at the service place in the i^{th} edge of the tour that consist of followings:

$$TW_i = \begin{cases} e_i & - \text{ealiest time} \\ l_i & - \text{latest time} \end{cases} \quad \text{.......................} (1)$$

- C : Total routing cost
- P : Total penalty
- F(C,P) : Objective function

Objective function :

-*min F(C,P) = (C×P)^m*, where m is a non-negative integer.

Subject to :

$$C = \sum_{i=1}^{n} (c_{e_i} + d_{e_i^2, e_{i+1}^1}) \text{ and } P = \sum_{i=1}^{n} p_i \quad (2)$$

where,

$$p_i = \begin{cases} 0, & e_i \leqq a_i \leqq l_i \\ e_i - a_i, & a_i < e_i \\ a_i - l_i, & a_i > l_i \end{cases}$$

and n is the size of E′, and if $i=n$, let $i+1 = 1$ (we assume that the tour starts at edge 1 and ends of edge 1).

Fitness function :

$$F^* = \frac{1}{F(C,P)} \quad \cdots\cdots (3)$$

IV. Genetic Algorithm

4.1 The Structure of Chromosome

An undirected graph G = (V, E) comprises a set V of n vertices, $\{v_i\}$, a set E ⊆ V × V of edges connecting vertices in V and a subset E′ (⊆ E) that is a set that must be passed at least once.

In this paper, the chromosome consists of two kinds of strings. One is for describing the visiting order of the edge in E′ and the other is for a set of binary codes (0 or 1) that indicate the decoding information. For example, assume that E′ = {1,2,3,4,5}, where 1, 2, 3, 4 and 5 denote edges (*a*, *a*′), (*b*, *b*′), (*c*, *c*′), (*d*, *d*′), and (*e*, *e*′), respectively, and 0 and 1 denote directions of the edges. If the decoding information of an element is 1, the direction of the tour is reverse. For example, assume that the following describes the structure of chromosome.

Edge (E′) Information	1	3	2	4	5
Decoding Information	0	1	0	1	0

The 0 of edge 1 means that in the tour we travel from *a* to *a*′, and the 1 of edge 3 denotes a path from node *c*′ to *c*, because the decoding information is 1.

4.2 Modified OX

Crossover is an operator that exchanges some strings in two selected chromosomes appropriately and a pair of new chromosomes are produced.

The order of strings are important in our problem. Hence, we use PMX proposed by Goldberg and Lingle[2,8], OX proposed by Davis[2,8], and MOX which has been modified from the OX.

This paper will describe MOX method only which we propose. Both PMX and OX were described in [2,8].

MOX builds an offspring by choosing a subsequence of a tour from one parent and preserving the order and position of as many strings as possible from the other parent. Hence, the children can inherit larger characters from the parents than the other methods (PMX, OX) and the possibility of premature convergence can be reduced. A subsequence of a routing is selected by choosing two random cut points, which serve as boundaries for reordering operations.

For example, the two parents with two cut points marked by '|',

p1 = (012|345|6789)
and
p2 = (987|654|3210)

would produce offspring in the following way. First, the segments between cut points are copied into offspring:

o1 = (***|345|****)
and
o2 = (***|654|****).

Next, we remove string 3, 4, and 5, which are already in the first offspring from the sequence of the strings in the second parent. And we get

9-8-7-6-2-1-0.

This sequence is place in the '*' positions of the first offspring in order:

o1 = (987|345|6210).

Similarly we get the other offsping:

o2 = (012|654|3789).

4.3 Mutations

In this paper, three mutation methods are applied. The first is that a decoding information of edges is flipped at the selected point marked by '-'(mutation1). The second is that two selected points marked by '-' are swapped (mutation2). The last is that the substring between two cut points marked by '|' along the length of the chromosome is inversed (mutatiion3). These are :

mutation1. Reverse
p = (1001110011)
o = (1001010011)

mutation2. Reciprocal exchange
p = (9876543210)
o = (9876143250)

mutation3. Inversion
p = (987|6543|210)
o = (987|3456|210).

Here, the reverse method is applied to the decoding information, and both the reciprocal exchange and the inversion are applied to the edge information in the chromosome.

V. Experimental Results

The GA was programmed in MSC++ version 6.0 and tested on an IBM PC Pentium IV for 12 randomly generated problems which was generated by the same method as [1].

⟨Table 1⟩ describes the problems applied to GA. In GA, the size of population is 100, and we evolve the population for 100 generations. The selection scheme in this paper is roulette wheel method according to fitness function. Each crossover (MOX, OX, and PMX) rate was 0.6, the mutation1 (Reverse) rate was 0.05, the mutation2 (Reciprocal Exchange) rate was 0.04, and the mutation3 (Inversion) rate was 0.03. A Pareto-optimal set of each generation is maintained and the final Pareto-optimal set of the algorithm is the solution of a problem.

Table 1. Test problems

Test Problem	|V|	E'|
1	20	17
2	30	13
3	40	21
4	50	16
5	20	14
6	30	14
7	40	14
8	50	38
9	20	19
10	30	12
11	40	15
12	50	47

The results are shown in ⟨Table 2⟩ and (Figure 2 ~Figure 7). ⟨Table 2⟩ describes the size of Pareto-optimal set obtained by GA according to each crossover operator for 12 test problems and (Figure 2~Figure 7) describe the results of comparison of crossover operators (PMX, OX, and MOX). The results show that the MOX method is more efficient than the existing PMX and OX operators for 10 test problems except problem 1 and 8. This is because the MOX can preserve the order and position of as many strings as possible from parents and the chromosomes in the current generation can inherit larger characters from the chromosomes in the old genertation than the existing crossover methods. Hence, in our GA using MOX, the possibility of premature convergence is reduced.

Table 2. The experiment results

Test Problem	Pareto-optimal set		
	PMX	OX	MOX
1	19	5	20
2	53	32	30
3	9	7	16
4	8	8	13
5	43	20	70
6	11	7	11
7	21	16	31
8	4	4	8
9	54	21	58
10	22	13	27
11	19	12	44
12	6	11	8

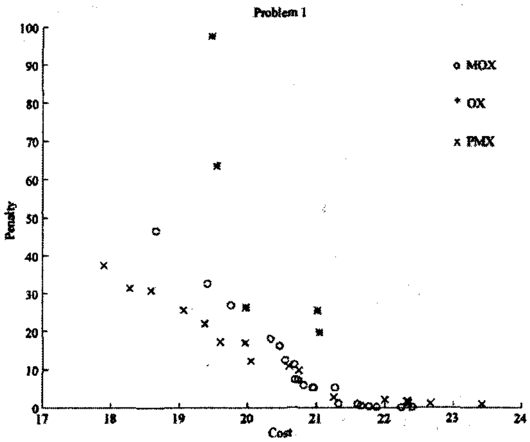

Figure 2. The experimental result of problem 1

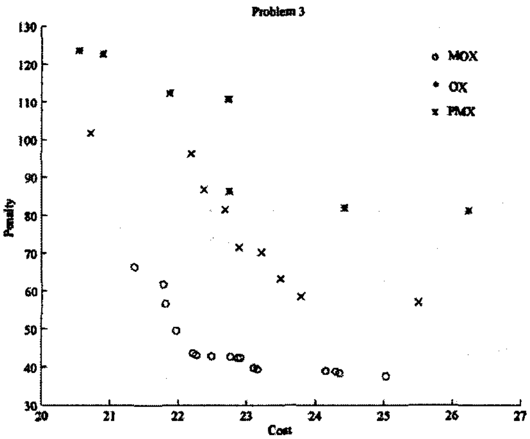

Figure 3. The experimental result of problem 3

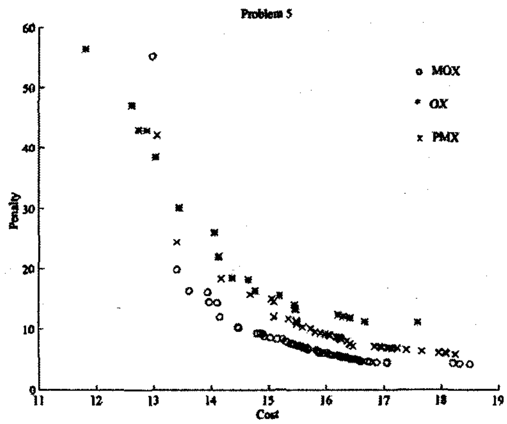

Figure 4. The experimental result of problem 5

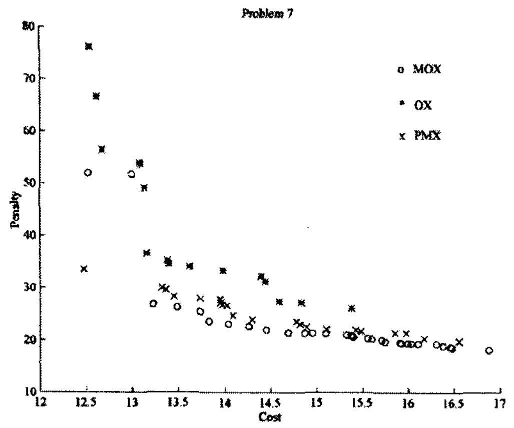

Figure 5. The experimental result of problem 7

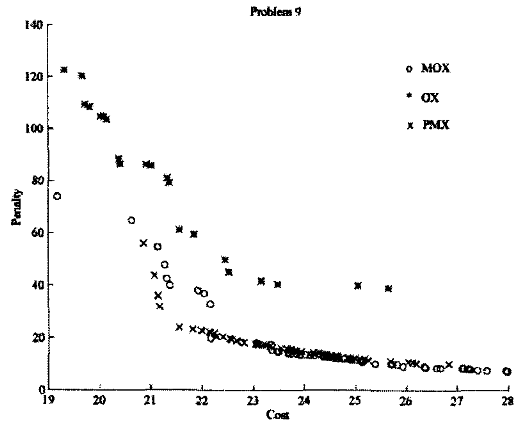

Figure 6. The experimental result of problem 9

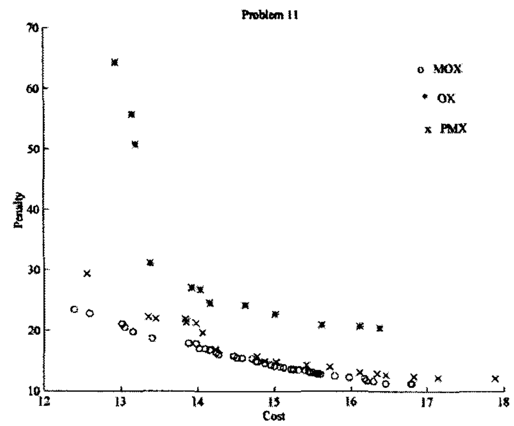

Figure 7. The experimental result of problem 11

Ⅵ. Conclusions

This paper introduces a genetic algorithm and compares three crossovers (PMX, OX, and MOX) for RPPTW. According to the experimental results, we can know clearly that the proposed MOX crossover method produces more and better Pareto-optimal solutions than the existing PMX and OX methods for our test problems. The comparison made on the basis of the number of Pareto-optimal solutions describes that the proposed MOX method produces more in number and better results.

References

[1] Kang, M. J. and Han, C. G., 1996, A Genetic Algorithms for the Rural Postman Problem with Time Windows, Proceedings of 20th International Conference on Computer & Industrial Engineering, pp. 321-324.

[2] Goldberg, D. E., 1989, GENETIC ALGORITHMS in Search, Optimization & Machine Learning, Addison Wesley.

[3] Azevedo, J. A. & Martins, E. Q. V., 1991, An Aogirithm for the Multibojective Shortest Path Problem on Acyclic Networks, Investigacao Operacional, Vol.11, No. 1.

[4] Makowski, M., Somlyody, L. & Watkins, D., Multiple, 1995, Criteria Analysis for Regional Water Quality Management: The Nitra River Case, IIASA.

[5] Lenstra, J.K. and Rinooy Kan, A.H.G., 1976, On General Routing Problems, Network, Vol.6, pp.273-280.
[6] Thangiah, S. R., 1995, Vehicle Routing with Time Windows using Genetic Algorithms, Practical Handbook of GENETIC vol. 2, ALGORITHMS, CRC Press.
[7] Fonseca, C. M. and Fleming, P. J., 1993, Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization, Proceedings of the Fifth International Conference.
[8] Michalewicz, Z., 1994, Genetic Algorithms + Data Structures = Evolution Programs, 2nd rev. and extended ed. Springer-Verlag, Berlin, Heidelberg.

저 자 소 개

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, 유비쿼터스