World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:4, 2011

A New Integer Programming Formulation for the Chinese Postman Problem with Time Dependent Travel Times

Jinghao Sun, Guozhen Tan, and Guangjian Hou

Abstract—The Chinese Postman Problem (CPP) is one of the classical problems in graph theory and is applicable in a wide range of fields. With the rapid development of hybrid systems and model based testing, Chinese Postman Problem with Time Dependent Travel Times (CPPTDT) becomes more realistic than the classical problems. In the literature, we have proposed the first integer programming formulation for the CPPTDT problem, namely, circuit formulation, based on which some polyhedral results are investigated and a cutting plane algorithm is also designed. However, there exists a main drawback: the circuit formulation is only available for solving the special instances with all circuits passing through the origin. Therefore, this paper proposes a new integer programming formulation for solving all the general instances of CPPTDT. Moreover, the size of the circuit formulation is too large, which is reduced dramatically here. Thus, it is possible to design more efficient algorithm for solving the CPPTDT in the future research.

Keywords—Chinese Postman Problem, Time Dependent, Integer Programming, Upper Bound Analysis.

I. Introduction

The Chinese postman problem (CPP) introduced by Meigu Guan [1] is a famous and classical problem in graph theory, which aims to find a minimum cost tour traversing each arc at least once. The problem that involves the periodic collection, delivery of goods and services are of great practice importance. Common examples of such problems include mail delivery, garbage collection, snow removal, school bus transportation and VLSI circuit design, in particular the improving of software testing [2]–[5]. Due to the CPP's wide applicability with respect to real-world problems, various extensions of the CPP have been the subject of scientific research in the past few decades [6]–[8]. The vast majority of these are conventional problems, where the timing of an intervention is insensitive.

The advancement of hybrid systems and model based testing has brought renewed interest in the subject with a new twist: time dependency of travel time. In this paper, we consider the Chinese Postman Problem with Time Dependent Travel Times (CPPTDT) (See in Section 2 for the formal definition), which is motivated from test sequence optimization based on hybrid automaton [10]. The hybrid automaton, where the delay time of transition from state s_i to s_j is a function $D_{ij}(t_i)$ of the arrival time t_i at s_i, can be easily treated as a dynamic directed network $D(V, A)$ with $D_{ij}(t_i)$ as the time dependent travel time of arc $(i, j) \in A$. As each state s_i corresponds to the vertex v_i in V, and each transition from s_i to s_j corresponds to the arc (i, j) in A, the optimal test sequence checking all transitions on the hybrid system can be equivalently cast as a minimal time Chinese postman tour that traverses all the arcs in time dependent network D.

In the literature, we have proposed an integer programming for solving CPPTDT, namely, circuit formulation [11]. However, the circuit formulation is only available for the CPPTDT problems defined on a special time dependent network with all circuits passing through the origin vertex. To address problems defined on the general time dependent networks, this paper designs a new integer programming formulation using not only circuit but also arc traversal order as decision variables. Moreover, the most crucial factor of the formulation size is the circuit number K of CPPTDT-tour, whose upper bound is proved to be $|A| - |V| + 1$ [11]. In this paper, we improve this result dramatically, such that K becomes to be in direct proportional to the maximum in/out degree of the network.

The rest of the paper is organized as follows. In Section 2, we introduce the integer programming proposed in [11]. The new formulation is presented in Section 3. Section 4 gives the improved result for circuit number K. Concluding remarks are made in the last section.

II. Circuit Formulation

The definition of CPPTDT is given at first.

***Definition** 1:* Let $D(V, A)$ be a connected digraph, where V is the set of vertices, A is the set of arcs and with each arc $(i, j) \in A$ is associated a time dependent travel time $D_{ij}(t_i)$ starting at time t_i. Given an origin $v_1 \in V$ and a starting time t_1, the Chinese Postman Problem with Time Dependent Travel Times (CPPTDT) aims to find a tour traversing each arc of A at least once such that the total travel time is minimized.

In paper [11], we have proposed an integer programming formulation for the CPPTDT, namely, circuit formulation, the main idea of which is exhibited as follows. Suppose that all the circuits in D have a common vertex v_1, which is abbreviated to ACTO(All Circuits in D Traverse Origin v_1) for short, then each Chinese Postman tour can be formulated as a sequence of circuits $O_f = (C_1, \cdots, C_K)$ in the time dependent network D. Let the 0-1 decision variable x_{ij}^k be 1 if the kth circuit in O_f traverses arc $(i, j) \in A$, and 0 otherwise. Let t_i^k represent the starting time at vertex v_i contained in the kth circuit of

This work was supported by the National Natural Science Foundation of China (No.60873256) and National Basic Research Program of China (No.2005CB321904).

J.H. Sun, G.Z. Tan and G.J. Hou are with the School of Computer Science and Technology, Dalian University of Technology, Dalian, 116023, China. E-mail address: jhsun@mail.dlut.edu.cn.

O_f. In particular, t_1^1 and t_1^{K+1} is the starting and ending time of O_f, since the vertex v_1 denotes the origin in D. The upper bound of K has been proved to be $|A|-|V|+1$ [11]. The integer programming formulation is given below.

$$\text{Min} \quad \sum_{(i,j)\in A}\sum_{k=1}^{K} D_{ij}(t_i^k)x_{ij}^k \tag{1}$$

$$\text{s.t.} \sum_{(i,j)\in A} x_{ij}^k = \sum_{(j,i)\in A} x_{ji}^k \forall i \in V; k=1,\cdots,K; \tag{2}$$

$$\sum_{k=1}^{K} x_{ij}^k \geq 1 \qquad \forall (i,j)\in A; \tag{3}$$

$$t_i^k - t_1^k \geq 0 \qquad \forall (1,i)\in A; k=1,2,...,K; \tag{4}$$

$$t_j^k - t_i^k \geq D_{ij}(t_i^k)x_{ij}^k \qquad \forall (i,j)\in A; k=1,2,...,K. \tag{5}$$

$$t_1^{k+1} - t_i^k \geq D_{i1}(t_i^k)x_{i1}^k \quad \forall (i,1)\in A; k=1,2,...,K; \tag{6}$$

$$x_{ij}^k = \{0,1\} \qquad \forall (i,j)\in A; k=1,2,\cdots,K; \tag{7}$$

$$t_i^k \geq 0 \forall i\in V; k=1,2,\cdots,K+1; \tag{8}$$

The objective function (1) minimizes the total travel time of CPPTDT-tour. Constraint (2) ensures that all vertices must be symmetric. Constraint (3) states that each arc must be passed at least once. Constraint (4) ensures that each circuit in the tour must start at origin depot v_1. Constraint (5) computes the ending time of each arc. Constraint (6) ensures that the ending time of each circuit is larger than its immediate predecessor. If we omit constraint (6) the starting time of all the circuits will be the same t_1. Note that the Constraint (4-6) lead to a circuit sequence form of the CPPTDT-tour based on the ACTO assumption. That is, the postman can traverse the circuits in the tour one by one starting and ending at the origin v_1 (described by Constraint (4)). If the ACTO assumption is not held in D, not all the Chinese postman tour can be formulated as a circuit sequence, thus the circuit formulation is no longer right for solving CPPTDT.

This above formulation may be the first attempt to construct an integer programming directly for the timing sensitive CPPs, other examples of which are the CPP with time windows [12], [14]–[16] and CPP with time dependent service costs(CPPTDC) [13], which used to be solved by transforming them into the corresponding node routing problems until the direct circuit formulation is presented in [11]. The facial structure of Circuits Visited Orders(CVO) polytope defined by Constraint (2-3) in the above formulation is investigated and facet defining inequalities are presented as cutting planes excluding the extra circuits. However, some limitation still exists in the circuit formulation no matter how skillful it has been constructed.

The main limitation is that the ACTO assumption is too strict for many practical time dependent networks. In order to deal with all the general CPPTDT instances, a new integer programming will be proposed in Section 3. Another disadvantage is the "large" number of the circuits number K. According to the above formulation, the large number of K does not affect the result of CPPTDT but may cause additional computation time. An upper bound of K is given as $|A|-|V|+1$ in [11], which limits the solvable instances to small or medium sized ones. In Section 4, this upper bound of K is improved to $\texttt{MAX}_{i\in V}d^{+/-}(i)$, where $d^{+/-}$ denotes the out/in degree of vertex v_i. It is easy to show that $\texttt{MAX}_{i\in V}d^{+/-}(i) \ll m-n+1$ for sparse graph.

III. New Integer Programming for CPPTDT

Motivated by circuit formulation, the Chinese postman tour can be seen as a merge of several circuits which can cover all the arcs in the network. For example, in the network G(see in Fig.1(a)), there exists a Chinese postman tour P as shown in Fig.1(b). It is easy to show that P can be obtained by merging the circuits in set $\mathscr{C}=\{C_1,\cdots,C_6\}$ (see in Fig.1(c)), and there are two kinds of circuit in set $\mathscr{C}$. One with an uninterrupted traversal in the tour is called integrated circuit such as C_3, C_5 and C_6 as shown in Fig.1(b). While the other is non-integrated, during the traversal of which the postman can traverse and complete other circuits. Take the non-integrated circuit C_4 as an example, the traversal of C_4 is interrupted by traversing circuits C_5, C_6 and C_3 successively as shown in Fig.1(b).

It's easy to prove that all the circuits in any Chinese postman tour are integrated ones if the CPPTW instance is defined on the special networks with all circuits passing through the origin. This instance can be solved successfully by circuit formulation. However, if there is a non-integrated circuit in the Chinese postman tour, then there must exists at least one circuit without passing through the origin in the CPPTW instance. Thus, circuit formulation cannot deal with such general instance whose feasible Chinese postman tour contains non-integrated circuits. To formulate these general instances, we propose a circuit variable to express the circuits in Chinese postman tour and use an interrupt variable to formulate the interrupted traversal of non-integrated circuit. Before introducing the new formulation, some notations used in the formulation is first given as follows:

Constants and Time Dependent Travel Time Function:

V: the number of vertices;
A: the number of arcs;
K: the number of circuits covering the G;
$D_{ij}(t)$: the travel time of arc (i,j) starting at time t;

Variables:

circuit variable x_{ij}^p :

$$x_{ij}^p = \begin{cases} 1: & \text{if arc } (i,j) \text{ is in circuit } C_p \\ 0: & \text{else} \end{cases}$$

interrupt variable s_{ij}^{pq}:

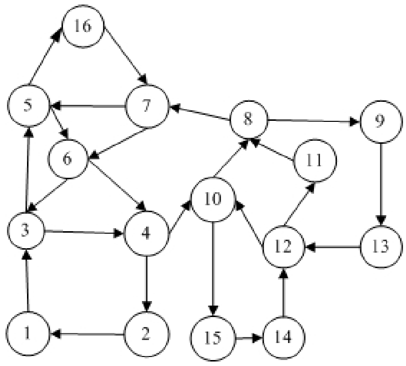
$$s_{ij}^{pq} = \begin{cases} 1: & \text{if } (i,j) \text{ is traversed and } v_i \in C_p,\ v_j \in C_q \\ 0: & \text{else} \end{cases}$$

t_i^p: starting time of arc (i,j) traversed in circuit C_p. In particular, let t_1^0 be the starting of the CPPTDT-tour. With the above notations the CPPTDT may be formulated as follows:

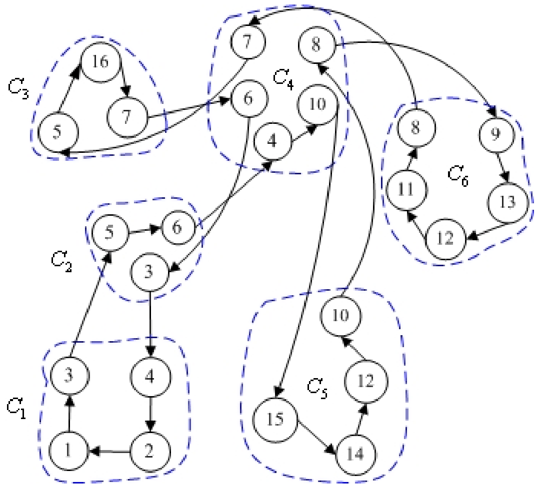
$$\text{Min} \quad \sum_{(i,j)\in A}\sum_{p=1}^{K} D_{ij}(t_i^p)x_{ij}^p \tag{9}$$

$$\text{s.t.} \quad \sum_{(i,j)\in A} x_{ij}^p = \sum_{(l,i)\in A} x_{li}^p \quad \forall v_i \in V; p=1,\cdots,K \tag{10}$$

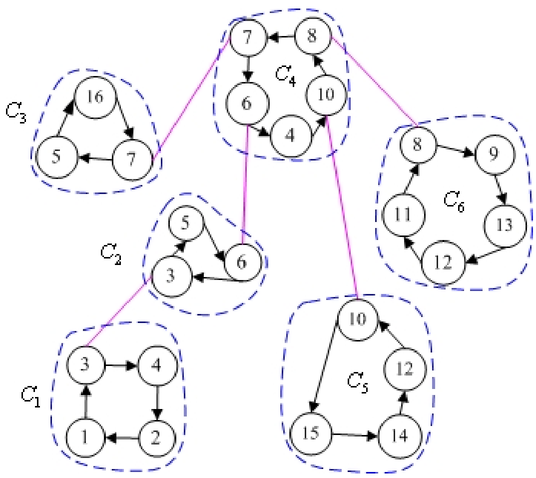
$$\sum_{p=1}^{K} x_{ij}^p \geq 1 \quad \forall (i,j)\in A \tag{11}$$



(a) the directed network G with the origin v_1



(b) a feasible Chinese postman tour P in the network G



(c) the circuit set $\mathscr{C}$ which covers all the arcs in the network G

Fig. 1. The Chinese postman tour can be obtained by merging circuits that cover all the arcs

$$\sum_{q=1}^{K}\sum_{(i,j)\in A} s_{ij}^{pq} = \sum_{(l,i)\in A} x_{li}^{p} \quad \forall v_i \in V; p = 1,\cdots,K \tag{12}$$

$$\sum_{p=1}^{K}\sum_{(i,j)\in A} s_{ij}^{pq} = \sum_{(j,l)\in A} x_{jl}^{q} \quad \forall v_j \in V; q = 1,\cdots,K \tag{13}$$

$$\sum_{q=1}^{K}\sum_{(1,j)\in A} s_{1j}^{0q} = 1 \tag{14}$$

$$t_j^p - t_1^0 \geq D_{1j}(t_1^0)s_{1j}^{0q} \forall (1,j) \in A; q = 1,\cdots,K \tag{15}$$

$$t_j^q - t_i^p \geq D_{ij}(t_i^p)s_{ij}^{pq} \forall (i,j) \in A, i \neq 1; p,q = 1,\cdots,K \tag{16}$$

The objective function (9) minimizes the total travel time of CPPTDT-tour. Constraint (10) ensures that all vertices must be symmetric. Constraint (11) states that each arc must be passed at least once. Constraint (12) describes the traversal of non-integrated circuits in the CPPTDT-tour. That is, once the vertex v_i is visited, then the arc (i,j) should be traversed subsequently, where vertices v_i and v_j are contained in circuits C_p and C_q respectively. Therefore, C_p is a non-integrated circuit in the CPPTDT-tour as $p \neq q$. Constraint (13) which plays a similar role as Constraint (12) ensures that the predecessor arc (l,i) should be traversed before visiting the vertex v_i, where the vertex v_i is contained in circuit C_p. Constraint (14) means that the postman must start from the origin v_1. Constraint (15) computes the travel time of the first arc in the CPPTDT-tour and other arcs' travel times are computed by constraint (16).

IV. Improved Upper Bound of K

In the previous section, we know that before solving the integer programming, we must give a sufficiently large value of K. However, the circuit number K bounded to $|A|-|V|+1$ in [11] leads to a much lager size of circuit formulation, which might have implications on the solvability of the CPPTDT problem. In order to reduce the computational time, the new formulation redefines K as the iteration number, and each iteration includes several circuits which do not contain the same vertices. Obviously, each circuit can be seen as one iteration in the former formulation present in [11]. Thus, the value of K defined here should be much less than the one in the former circuit formulation.

The main idea is motivated by the augmentation problem presented by Meigu Guan [1], who proposed the Chinese postman problem at first time, and formulated it as a problem which aims to determine a minimum cost augmentation of the graph, i.e., a least-cost set of arcs that will make the graph unicursal. We call the arcs in the least-cost arc set as augmentation arcs, and denote the circuit which is constructed by the augmentation arcs as augmentation circuit. In order to analyze the upper bound of K, we first give three lemmas which are closely related to the augmentation graph, and then, the upper bound will be proved in Theorem 1.

***Lemma** 1:* Each circuit C_i constructed by the augmentation arcs needs not to be one segment in the optimal CPPTDT-tour if waiting is allowed.

Proof: It is easy to show that the augmentation graph $\bar{D}$ constructed by adding the augmentation arc set into D is also Eulerian if we remove the circuit C_i from $\bar{D}$. However, the augmentation arcs in C_i are needed sometimes to link several unconnected segments of the CPPTDT-tour. Suppose that the optimal CPPTDT-tour O_f contains C_i as its one segment. Let C_i start and end at vertex v_i, and the travel time of C_i be t_{Ci}. Because all the arcs in C_i are traversed in the other segments of the CPPTDT-tour, we can wait at vertex v_i for t_{Ci} instead of traversing circuit C_i. Thus a new CPPTDT-tour O'_f without circuit C_i is constructed, whose ending time equals to O_f's, which is contrary to the assumption. ■

***Lemma** 2:* The total degree of vertex v_i and vertex v_j in the augmentation graph $\bar{D}$ is at least $3k$ if there are k augmentation circuits contained the two above vertices in $\bar{D}$ denoted as $C_1, \cdots, C_k$.

Proof: According to Lemma 1, each augmentation circuit C_l should be traversed in the CPPTDT-tour as two segments at least $(l = 1, \cdots, k)$. Let the paths P_{ij} and P_{ji} contained in C_l be the two segments, without loss of generality, assume that the path P_{ij} is traversed before P_{ji}. Then there must exist some arcs (s, i) and (i, r') associated with vertex v_i and some arcs (j, r) and (s', j) associated with vertex v_j in $\bar{D}$, which are not contained in each C_l $(l = 1, \cdots, k)$, such that $(s, i) - P_{ij} - (j, r) - \cdots - (s', j) - P_{ji} - (i, s')$ denotes one segment of the CPPTDT-tour. That is, it needs at least one out-arc and one in-arc associated with either v_i or v_j, which are not contained in each circuit C_l $(l = 1, \cdots, k)$, to ensure that an augmentation circuit can be traversed as two segments in the CPPTDT-tour. So there must exist at least $3k$ arcs associated with either v_i or v_j if the number of augmentation circuits contained the above two vertices is k. ■

According to Lemma 1 and 2, it is easy to prove Lemma 3.

***Lemma** 3:* If there is no augmentation circuit in the augmentation graph $\bar{D}$, then one can augment at most $\frac{\bar{d}}{2}$ circuits contained vertex v_i, where $\bar{d}_i$ is the total degree of v_i.

***Theorem** 1:* The upper bound of K in new formulation is $2\texttt{MAX}_{i\in V}(d_i^+, d_i^-) + \sum_{i\in V} |d_i^+ - d_i^-|$, where d_i^+ and d_i^- are the out-degree and in-degree of vertex v_i in D respectively.

Proof: The upper bound of K can be obtained as the maximum number n_C of circuits contained the same vertex. According to Lemma 3, the maximum number of circuits contained the same vertex v_i equals to $\frac{\bar{d}}{2}$ in the final augmentation graph $\bar{D}$. Now we construct $\bar{D}$ to maximize n_C. First, augment $\sum_{i\in V} \frac{1}{2}|d_i^+ - d_i^-|$ paths to balance the degree of each vertex v_i in D and then an augmentation Euler graph D' without augmentation circuit is obtained. At the worst case, all the augmentation paths pass the vertex with the maximum out-degree or in-degree in D denoted as $\texttt{MAX}_{i\in V}(d_i^+, d_i^-)$, then the maximum out-degree or in-degree of each vertex in augmentation Euler graph D' is at most $\texttt{MAX}_{i\in V}(d_i^+, d_i^-) + \sum_{i\in V} \frac{1}{2}|d_i^+ - d_i^-|$. Second, augment circuits in D' such that each of them can be traversed in the CPPTDT-tour as several segments according to Lemma 1, then the final augmentation Euler graph $\bar{D}$ is constructed at the worst case. According to Lemma 3, the maximum number of circuits containing the same vertex augmented in the second step is $(\texttt{MAX}_{i\in V}(d_i^+, d_i^-) + \sum_{i\in V} \frac{1}{2}|d_i^+ - d_i^-|)$. Thus the maximum number of circuits contained common vertex is $n_C = 2\texttt{MAX}_{i\in V}(d_i^+, d_i^-) + \sum_{i\in V} |d_i^+ - d_i^-|$, which is also the upper bound of K. ■

According to Theorem 1, the value of K is bounded by the maximum degree of the network. It is easy to prove that $2\texttt{MAX}_{i\in V}(d_i^+, d_i^-) + \sum_{i\in V} |d_i^+ - d_i^-| \ll m - n + 1$ for sparse network.

V. Computational Instance and Conclusion

In this section, we give an example to verify the correctness of the new formulation. Such a network $G(V, A)$ with seven vertices and eight arcs associated with time phases is shown in Fig.2. For each arc $(i, j) \in A$, the associated time phase $[E_{ij}, L_{ij}]$ expresses that the travel time of (i, j) is D_{ij} if the starting time at vertex v_i belongs in $[E_{ij}, L_{ij}]$, and 1000 otherwise.

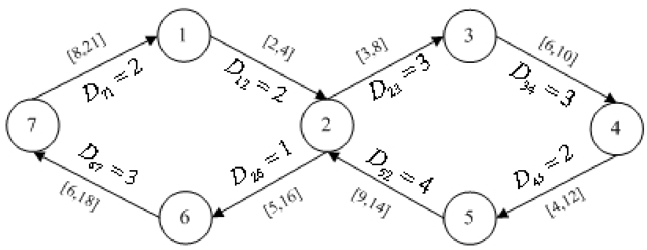


Fig. 2. a CPPTDT instance for testing the new formulation

According to Theorem 1, the upper bound of K in this instance is equal to 2. By using LINGO, nonzero variables in the result are shown as follows:

$$(x_{12}^1,\ x_{26}^1,\ x_{67}^1,\ x_{71}^1,\ x_{23}^2,\ x_{34}^2,\ x_{45}^2,\ x_{52}^2) = (1, \cdots, 1)$$
$$(S_{12}^{01}, S_{23}^{12}, S_{34}^{22}, S_{45}^{22}, S_{52}^{22}, S_{26}^{21}, S_{67}^{11}, S_{71}^{11}) = (1, \cdots, 1)$$
$$t_1^0 = 0, t_2^1 = 2, t_3^2 = 5, t_4^2 = 8$$
$$t_5^2 = 10, t_2^2 = 14, t_6^1 = 15, t_7^1 = 18, t_1^1 = 20.$$

The value of the variable set $\{x_{ij}^k|(i, j) \in A, k = 1, \cdots, K\}$ shows that the optimal CPPTDT-tour is constructed by merging two circuits $C_1 = (1-2-6-7-1)$ and $C_2 = (2-3-4-5-2)$. According to value of every variable S_{ij}^{pq}, we can obtain the trace of the optimal CPPTDT-tour. $S_{23}^{12} = 1$ exhibits that circuit C_1 is non-integrated circuit since its traversal is interrupted at vertex v_2. $S_{26}^{21} = 1$ means that the postman return to traverse circuit C_1 after the completion of C_2. By examining the value of variable t_i^p, it is shown that the optimal CPPTDT-tour P of this instance is $1-2-3-4-5-6-7-1$, and its total travel time is 20.

This paper modified the circuit formulation of the Chinese Postman Problem with Time Dependent Travel Times(CPPTDT), and the new formulation can solve all the instances of CPPTDT. In addition, the circuit number K bounded to $|A| - |V| + 1$ in [11] which is too large and may cause additional computational time, is improved here to the maximum in/out degree of the network that is far less than $|A| - |V| + 1$. A computational instance is given finally to verify the correctness of the new formulation.

Acknowledgments

This research was performed at Dalian University of Technology and was supported by a National Natural Science Foundation of China No.60873256 and National Basic Research Program of China No.2005CB 321904. These supports are gratefully acknowledged. The authors would like to thank...

REFERENCES

[1] M. K. Guan, Graphic programming using odd or even points, *Chinese Mathematics*, vol.1, pp.273-277, 1962.
[2] A. V. Aho, A. T. Dahbura, D. Lee, M. U. Uyar, An Optimisation technique for protocol conformance test generation based on UIO sequences and rural Chinese postman tours, *IEEE transactions on communications* vol.39, pp.1604-1615, 1991.
[3] R. Lai, A survey of communication protocol testing, *Journal of Systems and Software* vol.62, pp.21-46, 2002.
[4] M. Yannakakis, Testing, optimization, and games, In *Proceedings of the Nineteenth Annual IEEE Symposium on Logic In Computer Science*, LICS, pp.78-88, 2004.
[5] C. Chi, R, Hao, Test Generation for Interaction Detection in Feature-Rich Communication Systems, *Computer Networks* vol.51, pp.426-438, 2007.
[6] H. A. Eiselt, M. Gendreau, G. Laporte, Arc routing problems, Part I: The Chinese postman problem, *Operations Research*, vol.43, pp.231-242, 1995.
[7] H. A. Eiselt, M. Gendreau, G. Laporte, Arc routing problems, Part II: The Chinese postman problem, *Operations Research*, vol.43, pp.399-414, 1995.
[8] M. Dror, *Arc routing: Theory, solutions and applications*, Kluwer Academic Publishers. Boston, 2000.
[9] A. Hessel, K. G. Larsen, B. Nielsen, P. Pettersson, A. Skou, Time-Optimal Test Cases for Real-Time Systems, In *3rd International Workshop on Formal approaches to Testing of Software* (FATES 2003), Montreal, Quebec, Canada, October 2003.
[10] P. R. Springintveld, F. Vaandrager, Testing timed automata, *Theoretical computer science* vol.254, pp.225-257, 2001.
[11] G. Z. Tan, J. H. Sun, J. X. Wang, Chinese Postman Problem with Time Dependent Travel Times: Polyhedron Results, *Discrete Optimization*, 2010, Summited to be published
[12] P. A. Mullaseril, Capacitated rural postman problem with time windows and split delivery, Ph.d thesis. University of Arizona, 1996.
[13] M. Tagmouti, M. Gendreau, J. Y. Potvin, Arc routing problems with time-dependent service costs, *European Journal of Operational Research*, vol.181, pp.30–39, 2007.
[14] W. L. Pearn, A. Assad, B. L. Golden, Transforming arc routing into node routing problems, *Computers and Operations Research*, vol.14, no.4, pp.285-288, 1987.
[15] G. Laporte, Modeling and solving several classes of arc routing problems as traveling salesman problems, *Computers and Operations Research* vol.24, pp.1057-1061, 1997.
[16] H. Longo, M. P. Aragao, E. Uchoa, Solving capacitated arc routing problems using a transformation to the CVRP, *Computers and Operations Research*, vol.33, pp.1823-1837, 2006.