Iterative algorithm for lane reservation problem on transportation network

Yunfei Fang, Feng Chu, Said Mammar, Ada Che

Iterative Algorithm for Lane Reservation Problem on Transportation Network

Yunfei Fang, Feng Chu, Saïd Mammar and Ada Che

Abstract— **In this paper, we study an NP-hard lane reservation problem on transportation network. By selecting lanes to be reserved on the existing transportation network under some special situations, the transportation tasks can be accomplished on the reserved lanes with satisfying the condition of time or safety. Lane reservation strategy is a flexible and economic method for traffic management. However, reserving lanes has impact on the normal traffic because the reserved lanes can only be passed by the special tasks. It should be well considered choosing reserved lanes to minimize the total traffic impact when applying the lane reservation strategy for the transportation tasks. In this paper, an integer linear program model is formulated for the considered problem and an optimal algorithm based on the cut-and-solve method is proposed. Some new techniques are developed for the cut-and-solve method to accelerate the convergence of the proposed algorithm. Numerical computation results of 125 randomly generated instances show that the proposed algorithm is much faster than a MIP solver of commercial software CPLEX 12.1 to find optimal solutions on average computing time.**

I. INTRODUCTION

With the development of economy, high urbanization has been achieved in many countries today. For a sustainable economic development, traffic management is one of strategic issues that must be considered by government. Many transportation problems, such as design and configuration for transportation network [1], transportation planning and scheduling [2] have been drawn much attention by researchers over the last few decades. The freight transportation is an important part of the economy. It supports production, trade and consumption activities by providing safety, timely and reliable transportation of raw material or finished products. However, the increasing intensified traffic situation and saturated transportation network make the freight transportation difficult. Constructions of new transportation infrastructure are constrained by the heavy funding and long duration. Finding a flexible management that can adapt quickly to new situation is a complementary way with the construction of infrastructure.

Lane reservation strategy on the existing transportation network is such a flexible and economic option for traffic management. With this strategy, some lanes on roads are temporarily reserved for the transportation tasks. Only these transportation tasks can pass through the reserved lanes, and these tasks could be performed satisfying the conditions of time and/or safety. Moreover, lane reservation strategy can be generalized to a wide range of application, such as city public transportation, transportation of hazardous materials, design and configuration of network for automated robot-driven trucks. In fact, the lane reservation strategy has been applied successfully during the Olympic Games held in Athens and Beijing. In Paris, some lanes are reserved for public bus and taxis. Lane reservation strategy has been qualitatively studied by some researchers [3], [4].

However, reserving lanes has impact on the normal traffic, since only the special tasks can pass through the reserved lanes and congestion can be generated for other users in the transportation network. In this context, we study such a transportation problem: select lanes to be reserved on the existing transportation network for the special tasks and minimize the total traffic impact of reserving lanes on the normal traffic. We call it the *lane reservation problem on transportation network* (LRPTN). To the best of our knowledge, there are very few studies about it in the literature. Wu et al. [5] are the first to propose a mathematical model for studying lane reservation strategy. A simple heuristic algorithm is proposed for the considered problem to obtain near-optimal solutions. The LRPTN is different from the classic *vehicle routing problem* (VRP) and *facility location problem* (FLP). VRP is to minimize the total transportation cost for a fleet of vehicles to serve a set of customers. In an optimal solution of VRP, each edge can be visited only once by a vehicle. But for the LRPTN, to minimize the total traffic impact on the normal traffic, a reserved lane can be passed by several tasks in an optimal solution. In classical FLP, facilities is located on nodes. The change of the location of a facility will influence partially transportation path. For the LRPTN, lanes are selected to be reserved and the change of the reserved lanes can affect the full path of the tasks. For more detail information of VRP and FLP, please see [6], [7] and [8].

For solving transportation problems, various heuristic and exact methods have been proposed in the literature, such as metaheuristic methods [9], [10], [11], [12], methods based on Lagrangean relaxation [13], [14], hybrid methods [15], branch-and-bound [16], [17] and branch-and-cut [18]. The

This work was partially supported for the Cai Yuanpei program by the French Ministries of Foreign and European Affairs and the Higher Education and Research and the Chinese Ministry of Education, under Grant No. 24021SH.

Y. Fang is with [1]Laboratoire IBISC, University of Evry-Val d'Essonne, 40 rue du Pelvoux, 91020 Evry Cedex, France, [2]Laboratoire LOSI, Université de Technologie de Troyes, 12 rue Marie Curie, 10010 Troyes Cedex, France yunfei.fang@iup.univ-evry.fr

F. Chu is with Laboratoire IBISC, University of Evry-Val d'Essonne, 40 rue du Pelvoux, 91020 Evry Cedex, France feng.chu@iup.univ-evry.fr

S. Mammar is with Laboratoire IBISC, University of Evry-Val d'Essonne, 40 rue du Pelvoux, 91020 Evry Cedex, France said.mammar@iup.univ-evry.fr

A. Che is with the School of Management, Northwestern Polytechnical University, Xi'an, 710072, PR China ache@nwpu.edu.cn

advantage of metaheuristic is its flexibility. For evaluating the performance of metaheuristic, it should be helped by other methods or tools such as methods based on Lagrangean relaxation. It provide a lower bound of the studied problem and useful information to construct feasible solutions. But methods based on Lagrangean relaxation is effective only for certain types of problems, such as FLP. Exact methods can obtain an optimal solution of the problem, but the computation time will increase exponentially with the size of NP-hard problem. Analysis of properties of the problem and appropriate use of methods can help solve large scale NP-hard problem [19]. Cut-and-solve exact method was introduced by Climber and Zhang in 2006 [20]. Authors claimed that it outperformed the state-of-the-art solvers for the *asymmetry traveling salesman problem* (ATSP). Yang et al. [21] applied the principe of cut-and-solve method to the *single source capacitated facility location problem* (SSCFLP) and improved results in the literature. Fang et al. [22] developed an optimal algorithm based on cut-and-solve method for solving a *lane reservation problem*.

In this paper, we study a lane reservation problem on transportation network and formulate it as an integer linear program model. New techniques of generating cuts for cut-and-solve method are developed and a cut-and-solve based algorithm is proposed for the considered problem. Computational results show that the proposed algorithm is much faster than the commercial solver CPLEX 12.1 [23].

The remainder of the paper is organized as follows. In section II, the problem is described and it is formulated as an integer linear program. Section III presents the solution approach. An optimal algorithm based on cut-and-solve method is proposed by developing new techniques of generating cuts. Computational results are reported in section IV. In section V some conclusions and related future work are discussed.

II. Problem description

The lane reservation problem we consider is as follows: select lanes to be reserved on an exist transportation network and determine the path for each task to ensure that this one can be accomplished in the path composed of reserved lanes within a deadline, while the total traffic impact of reserving these lanes on the normal traffic is minimum. To well describe the problem, some assumptions are given as follows. Firstly, there are at least two lanes on each road. Otherwise, the impact on the normal traffic of reserving the lane on the road is very great. Secondly, a reserved lane may be shared by several task paths. Since the objective of the problem is to minimize the total impact of reserving lanes, the less lanes are reserved, the less the total impact is. Thirdly, we assume that each lane in the path for any task is a reserved one so as to facilitate traffic management and safety of hazardous transportation. Fig. 1 is an illustration of the problem. There are two tasks (from node 1 to 9 and 2 to 10) to be accomplished. Lanes on arcs $(1,3),(3,6),(6,9)$ and $(2,3),(3,6),(6,10)$ are reserved for the two tasks, respectively. And the reserved lane on arc $(3,6)$ are shared by both the two tasks. The transportation network can be

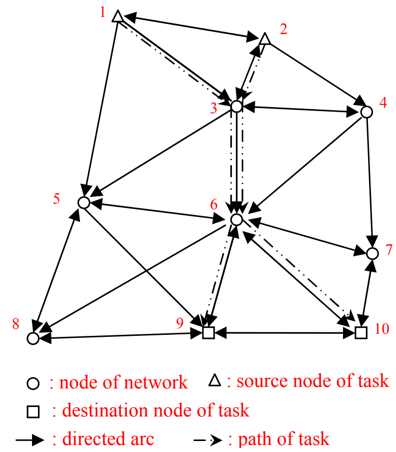

Fig. 1. Example of lane reservation strategy

considered as a directed graph $G=(V,A)$, where V is the set of nodes and A is the set of directed arcs (i,j). Let K be the set of tasks, $S=\{s_k\in V|k\in K\}$ and $D=\{d_k\in V|k\in K\}$ be the sets of source nodes and destination nodes for the tasks, respectively. p_k is the deadline for task k. t_{ij} is the travel time on a reserved lane on arc (i,j) and c_{ij} is the impact on the normal traffic of reserving a lane on arc (i,j). The binary decision variable Z_{ij} is equal to 1 if and only if a lane on arc (i,j) is reserved. The binary decision variable X_{ij}^k is equal to 1 if and only if task k passes arc (i,j). With assumptions and notations given above, the LRPTN can be formulated by the following integer linear program (P_0).

$$(P_0)\ \min \sum_{(i,j)\in A} c_{ij}Z_{ij} \tag{1}$$

$$s.t. \sum_{j:(s_k,j)\in A} X_{s_kj}^k = 1, \forall k\in K, s_k\in S \tag{2}$$

$$\sum_{i:(i,d_k)\in A} X_{id_k}^k = 1, \forall k\in K, d_k\in D \tag{3}$$

$$\sum_{j:(j,i)\in A} X_{ji}^k = \sum_{j:(i,j)\in A} X_{ij}^k, \forall k\in K, \forall i\neq s_k, d_k \tag{4}$$

$$\sum_{(i,j)\in A} t_{ij}X_{ij}^k \le p_k, \forall k\in K \tag{5}$$

$$X_{ij}^k \le Z_{ij}, \forall (i,j)\in A, \forall k\in K \tag{6}$$

$$X_{ij}^k \in \{0,1\}, \forall (i,j)\in A, \forall k\in K \tag{7}$$

$$Z_{ij} \in \{0,1\}, \forall (i,j)\in A \tag{8}$$

The objective function (1) is to minimize the total impact of reserving lanes on the normal traffic. Constraints (2) (resp. (3)) represent that there is only one path for each task k starting from (resp. arriving at) the source node s_k (resp. destination node d_k). Constraints (4) ensure the flow conservation. Constraints (5) assure that the total travel time for task k does not exceed the deadline p_k. Constraints (6) guarantee that any lane in the path for task k is reserved.

It is not hard to see that the LRPTN is NP-hard. If all the tasks have the same source node and $p_k=+\infty$ for any $k\in$

K, then the reduced lane reservation problem corresponds to the *steiner tree problem in a directed graph* (STDG), which is known to be NP-hard [24]. So the LRPTN is NP-hard.

III. Solution approach

In this section, a cut-and-solve based algorithm is proposed to solve the LRPTN optimally. The cut-and-solve method [20], which was introduced by Climer and Zhang for solving ATSP, can be explained as follows. At each iteration of the cut-and-solve method, a *piercing cut* (PC) is generated and it cuts the solution space of the current problem into two subspaces, which correspond to a sparse problem (SP) and a remaining problem (RP). The solution space of the SP is relatively sparse, so it can be solved optimally easily. The SP is a subproblem of the original problem so its optimal value is an upper bound of the original problem. The *best upper bound of the original problem* $(UB_{\min})$ is updated if necessary. Then a lower bound of the RP is obtained by solving its linear relaxation problem. If this lower bound is greater than or equal to the $UB_{\min}$, the optimal value of the RP cannot be smaller than the $UB_{\min}$. Hence the $UB_{\min}$ is the global optimal value and the iteration is terminated. Otherwise, the RP is set to the new current problem for the next iteration. More details can be seen in [20].

To solve the LRPTN, a pre-processing is used to tighten the relaxed problem of P_0. Some new techniques of generating piercing cut are developed for the cut-and-solve method. The solution approach is described in details below.

A. The pre-processing for P_0

Let $dis(i,j)$ denote the shortest travel time from i to j in a reserved path. Define set E_1 as follows

$$E_1 = \{X^k_{s_k j} \mid t_{s_k j} + dis(j, d_k) > p_k, \forall k \in K\}, \quad (9)$$

where (s_k, j) is an arc outgoing from the source node s_k for task k. Then task k must not pass the arc (s_k, j) in the optimal solution, because by the definition of E_1 the sum of the travel time on arc (s_k, j) and the shortest travel time from j to d_k in a reserved path is greater than the deadline p_k. For a similar case, set E_2 is defined as follows

$$E_2 = \{X^k_{id_k} \mid dis(s_k, i) + t_{id_k} > p_k, \forall k \in K\}, \quad (10)$$

where (i, d_k) is an arc incoming into the destination node d_k for task k.

In the pre-processing, sets E_1 and E_2 are firstly defined and then all the variables in E_1 and E_2 are set to 0. By this pre-processing, the linear relaxation problem of P_0 is tightened but no feasible solutions of P_0 are removed. So in the following steps of the algorithm, a new integer program P_1 is solved instead of P_0. The P_1 is defined as follows.

$$(P_1) \ \min \sum_{(i,j)\in A} c_{ij} Z_{ij}$$
$$s.t. \text{constraints } (2) - (8)$$
$$X^k_{ij} = 0, \forall X^k_{ij} \in E_1 \cup E_2 \quad (11)$$

B. New techniques of generating piercing cut

For the cut-and-solve method, the appropriate PC is very important. Let SS_{sp} be the solution space of the SP. SS_{sp} should be relatively sparse, so that the SP can be solved easily. Meanwhile, SS_{sp} should be large enough to contain feasible solution(s) of the original problem, otherwise the $UB_{\min}$ cannot be improved. In the following, new techniques of generating PC are developed.

1) Definition of piercing cut, sparse problem and remaining problem: [20] defined a set U composed of all the variables with reduced cost greater than a given number. Because all the variables are binary, either the sum of the variables in U is equal to 0 or it is greater than or equal to 1. Then the PC is defined as the sum of the variables in U is greater than or equal to 1. The solution space of the current problem is cut into the sparse space (with the constraint that the sum of the variables in U is equal to 0) and the remaining space (with the constraint that the sum of the variables in U is greater than or equal to 1) by this PC.

For the LRPTN, tasks paths are chosen on the reserved lanes. And the objective function is only related with Z_{ij}. Z_{ij} is "more" decisive. In addition, the number of Z_{ij} is much less than that of X^k_{ij}. Because of these reasons, we define the set U by considering only Z_{ij}, not all the variables. Let U_l, PC_l, SP_l, and RP_l $(l \geq 1)$ denote the set, the piercing cut, the sparse problem, and the remaining problem in l-th iteration, respectively. U_l is defined as follows

$$U_l = \{Z_{ij} \mid \text{reduced cost of } Z_{ij} > a_l, \forall (i,j) \in A\}, \quad (12)$$

where a_l is a given number. The value for a_l is dependent on the distribution of reduced cost. We solve the linear relaxation problem of the current problem and obtain the reduced cost of each variable at each iteration. Then we select n variables Z_{ij} with largest reduced cost, a_l is set to the minimum reduced cost among these n variables. Once we obtain U_l, the PC_l is defined as follows

$$(PC_l) \ \sum_{Z_{ij}\in U_l} Z_{ij} \geq 1. \quad (13)$$

By the cut-and-solve method, the current problem at l-th $(l > 1)$ iteration is defined as RP_{l-1} (for $l = 1$, the current problem is P_1). Then the SP_l is defined as follows

$$(SP_l) \ \min \sum_{(i,j)\in A} c_{ij} Z_{ij}$$
$$s.t. \text{ constraints } (2) - (8) \text{ and } (11)$$
$$\sum_{Z_{ij}\in U_t} Z_{ij} \geq 1, \quad t = 1, \ldots, l-1 \quad (14)$$
$$\sum_{Z_{ij}\in U_l} Z_{ij} = 0. \quad (15)$$

And RP_l is defined as follows

$$(RP_l) \ \min \sum_{(i,j)\in A} c_{ij} Z_{ij}$$
$$s.t. \text{ constraints } (2) - (8), (11), (13) \text{ and } (14).$$

For the first iteration $l = 1$, there is no (14) in SP_1 and RP_1. It is not hard to see that SP_l and RP_l are RP_{l-1} with additional constraints (15) and (13), respectively.

2) Improved piercing cut: When the problem size becomes large, the PC above is not very effective. To make some improvement, we firstly give the following theorem.

Theorem 1: For $l \geq 2$, if

$$U_1 \supseteq \cdots \supseteq U_{l-1} \supseteq U_l, \tag{16}$$

holds, then SP'_l is equal to SP_l and RP'_l is equal to RP_l, where SP'_l is

$$(SP'_l) \ \min \sum_{(i,j)\in A} c_{ij} Z_{ij}$$
$$s.t. \text{ constraints } (2)-(8), (11) \text{ and } (15)$$
$$\sum_{Z_{ij} \in U_{l-1} \setminus U_l} Z_{ij} \geq 1, \tag{17}$$

and RP'_l is

$$(RP'_l) \ \min \sum_{(i,j)\in A} c_{ij} Z_{ij}$$
$$s.t. \text{ constraints } (2)-(8), (11) \text{ and } (13).$$

Proof: To prove SP'_l is equal to SP_l, we just have to prove that (14) is equal to (17). If (14) is true, then we have $\sum_{Z_{ij}\in U_{l-1}} Z_{ij} \geq 1$. Meanwhile, $\sum_{Z_{ij}\in U_{l-1}} Z_{ij} = \sum_{Z_{ij}\in U_{l-1}\setminus U_l} Z_{ij} + \sum_{Z_{ij}\in U_l} Z_{ij}$, because $U_{l-1} \supseteq U_l$. Since $\sum_{Z_{ij}\in U_l} Z_{ij} = 0$ by (15), so we have $\sum_{Z_{ij}\in U_{l-1}\setminus U_l} Z_{ij} \geq 1$, i.e. (17) is true.

If $\sum_{Z_{ij}\in U_{l-1}\setminus U_l} Z_{ij} \geq 1$, i.e. (17) is true. Because $U_1 \supseteq \cdots \supseteq U_{l-1} \supseteq U_{l-1}\setminus U_l$, then $\sum_{Z_{ij}\in U_t} Z_{ij} = \sum_{Z_{ij}\in U_t\setminus(U_{l-1}\setminus U_l)} Z_{ij} + \sum_{Z_{ij}\in U_{l-1}\setminus U_l} Z_{ij} \geq \sum_{Z_{ij}\in U_{l-1}\setminus U_l} Z_{ij}$, $t = 1, \ldots, l-1$. So we have $\sum_{Z_{ij}\in U_t} Z_{ij} \geq 1$, $t = 1, \ldots, l-1$, i.e. (14) is true. So (14) is equal to (17), and SP'_l is equal to SP_l.

All the constraints are the same both in RP'_l and RP_l except that there is no (14) in RP'_l. To prove RP'_l is equal to RP_l, we just have to prove that (14) is redundant in RP'_l. By (13) we have $\sum_{Z_{ij}\in U_l} Z_{ij} \geq 1$. Because $U_1 \supseteq \cdots \supseteq U_{l-1} \supseteq U_l$, then $\sum_{Z_{ij}\in U_t} Z_{ij} = \sum_{Z_{ij}\in U_t\setminus U_l} Z_{ij} + \sum_{Z_{ij}\in U_l} Z_{ij} \geq \sum_{Z_{ij}\in U_l} Z_{ij} \geq 1, t = 1, \ldots, l-1$, (14) is true. So RP'_l is equal to RP_l. ■

There are $l-1$ equalities in (14) for SP_l, but only one in (17) for SP'_l. In addition, the equalities in (14) for RP_l are totally removed for RP'_l. SP'_l and RP'_l have less constraints than SP_l and RP_l, respectively, it will be more efficient to solve SP'_l and RP'_l instead of SP_l and RP_l.

As explained above, the set U'_l ($l \geq 1$) used in the proposed algorithm is defined as follows

$$U'_l = \{Z_{ij} \mid Z_{ij} \in U_l \cap U_{l-1}, \forall (i,j) \in A\}, \tag{18}$$

where $U_0 = \{Z_{ij} \mid \forall (i,j) \in A\}$. The piercing cut PC'_l is defined as follows

$$(P'_l) \ \sum_{Z_{ij}\in U'_l} Z_{ij} \geq 1. \tag{19}$$

SP''_l is defined as follows

$$(SP''_l) \ \min \sum_{(i,j)\in A} c_{ij} Z_{ij}$$
$$s.t. \text{ constraints } (2)-(8) \text{ and } (11)$$
$$\sum_{Z_{ij}\in U'_l} Z_{ij} = 0 \tag{20}$$
$$\sum_{Z_{ij}\in U'_{l-1}\setminus U'_l} Z_{ij} \geq 1. \tag{21}$$

And RP''_l is defined as follows

$$(RP''_l) \ \min \sum_{(i,j)\in A} c_{ij} Z_{ij}$$
$$s.t. \text{ constraints } (2)-(8), (11) \text{ and } (19).$$

For $l = 1$, there is no (21) in the SP''_1.

The overall algorithm is presented as follows.

Algorithm 1

1) Do the pre-processing for P_0 and obtain P_1.
2) Initialize $UB_{\min} := +\infty$, $l := 0$ and the current problem as P_1.
3) Solve the linear relaxation problem of the current problem and obtained reduced cost of each variable.
4) **do**
   a) Set $l := l+1$, define U'_l by (18) and PC'_l by (19).
   b) Use PC'_l to cut the solution space of the current problem and obtained the SP''_l and the RP''_l.
   c) Solve the SP''_l and obtain its optimal value UB_l if exists, update the $UB_{\min}$ if necessary.
   d) Solve the linear relaxation problem of the RP''_l and obtain its lower bound LB_l and reduced cost of each variable.
   e) Set the current problem as RP''_l.

   while $LB_l < UB_{\min}$
5) return $UB_{\min}$ as the global optimal value, algorithm is terminated.

IV. Computational results

In this section, the performance of the proposed algorithm is compared with a commercial solver CPLEX 12.1. The algorithm is implemented in Visual C++ and run on a PC with 3.00GHz CPU and 4.00GB RAM. We use a LP and MIP solver CPLEX in default setting to solve the linear relaxation problem of the remaining problem and the sparse problem, respectively.

The random instances are generated in the following way. The network is generated by Waxman's random network model [25]. The nodes of the network are randomly distributed in a rectangle. The probability of arc existence between nodes i and j is given by

$$P(i,j) = \alpha \exp\left(\frac{-d(i,j)}{\beta L}\right), \tag{22}$$

where $d(i,j)$ is the Euclidean distance from i to j, and L is the maximum distance of all pairs of nodes. α and β are parameters in $(0,1]$. Large value of α results in high average node degree, while large value of β yields a high ratio of long arcs to short ones. The travel time on a reserved lane on (i, j) is set to $t_{ij} = r_{ij} d(i, j)$, where r_{ij} is randomly generated in $[0.5, 0.8]$. The deadline p_k is equal to $b_k \cdot dis(s_k, d_k)$, where $dis(s_k, d_k)$ is the shortest travel time from s_k to d_k in a reserved path and b_k is randomly generated in $[1, \sqrt{2}]$. The impact of reserving lanes on the normal traffic c_{ij} is difficult to evaluate. [5] defined it by $d(i, j)/(M_{ij} - 1)$, where M_{ij} is the number of lanes on road (i, j) and is randomly and uniformly generated in [2, 4].

The problem instances are divided in 25 sets of 5 each. The average node degree of each instance is 7 to denote a sparse network [26]. Let LP_0 and LP_1 denote the lower bound of P_0 and P_1 obtained by linear relaxation, respectively. With the notations given in Table I, the computational results are summarized in Table II and Fig. 2.

In Table II the Gap between the lower bound of P_0 and P_1 varies from 0.85% to 3.86% and the average value of Gap for all instances is 3.03%, which shows that the pre-processing tightened the lower bound of P_0. With the increase of $|K|$, the Gap becomes larger. Take instances sets 5 and 25 for example, both have 150 nodes but 10 and 30 tasks, respectively. The Gap is 2.05% for S5 while 3.78% for S25. The computational time by Algorithm *1* is less than that by CPLEX for all the instances sets. And Algorithm *1* takes 1.47-3.93 times less computation time than that by CPLEX. Our algorithm is 2.45 times faster than CPLEX in terms of an average computation time. In addition, CPLEX takes much computation time when $|K|$ increases. The computation time by CPLEX is 4058.54s for S25 with 30 tasks, while Algorithm *1* takes only 1543.34s for S25.

Because of length of the paper, we give the comparison of computational time by Algorithm *1* and CPLEX in Fig. 2 (a) and (b) corresponding to the instances with 10 and 30 tasks in Table II. Both in Fig. 2 (a) and (b), the computational time by CPLEX grows much quickly with number of nodes, while for Algorithm *1*, the computation time grows slowly. Moreover, we observe that the computation time increases much sharply with the number of tasks for both CPLEX and Algorithm *1*. In Fig. 2 (a), both of CPLEX and Algorithm *1* take no more than 20s for the instances with 10 tasks. While in Fig. 2 (b), the computation time by both of CPLEX and Algorithm *1* increases quickly to several hundred seconds or even more for instances with 30 tasks. However, Algorithm *1* is more effective for instances with 30 tasks than with 10 tasks. For sets S21-S25, which are the instances with 30 tasks, Algorithm *1* is 2.35% times faster than CPLEX on average computational time. While for S1-S5 with 10 tasks, Algorithm *1* is 2.02% times faster than CPLEX on average computational time.

TABLE I

NOTATIONS OF THE RESULTS

$\|V\|$	number of nodes of the network
$\|K\|$	number of tasks
Gap	$100 \times (LP_1 - LP_0)/LP_0$
T_a	computation time by Algorithm *1* in seconds
T_{cp}	computation time by CPLEX in seconds
T_{cp}/T_a	ratio of computation time by CPLEX and Algorithm *1*

TABLE II

COMPARISON OF ALGORITHM *1* WITH CPLEX.

set	$\|V\|$	$\|K\|$	Gap(%)	T_{cp}(s)	T_a(s)	T_{cp}/T_a
S1	110	10	0.85	4.57	2.37	1.93
S2	120	10	2.26	6.01	2.45	2.45
S3	130	10	2.97	8.18	5.05	1.62
S4	140	10	2.08	14.54	4.36	3.34
S5	150	10	2.05	12.35	8.40	1.47
S6	110	15	2.24	45.05	11.45	3.93
S7	120	15	1.04	90.49	45.80	1.98
S8	130	15	2.11	151.46	52.02	2.91
S9	140	15	3.43	178.30	64.09	2.78
S10	150	15	2.56	274.81	114.81	2.39
S11	110	20	3.34	240.56	111.62	2.16
S11	120	20	3.05	417.17	166.41	2.51
S13	130	20	2.89	1075.65	438.96	2.45
S14	140	20	3.65	1370.31	551.62	2.48
S15	150	20	3.65	1648.54	591.96	2.78
S16	110	25	2.73	958.79	349.40	2.74
S17	120	25	2.84	1014.07	422.98	2.40
S18	130	25	3.47	1689.28	671.37	2.51
S19	140	25	3.31	2014.52	765.84	2.63
S20	150	25	3.81	2202.74	840.97	2.62
S21	110	30	3.25	1418.11	613.34	2.31
S22	120	30	3.62	2038.60	672.55	3.03
S23	130	30	3.10	2660.95	1631.65	1.63
S24	140	30	3.86	3038.63	1172.22	2.59
S25	150	30	3.78	4058.54	1543.43	2.63
Average			3.03	1065.29	434.22	2.45

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper, we studied a lane reservation problem on transportation network. The problem is to choose reserved lanes to minimize the total impact of reserving lanes while the tasks could be completed within the deadlines. The considered problem is NP-difficult. For solving the problem, an integer linear program model was formulated and an optimal algorithm based on cut-and-solve method was proposed. A pre-processing was done to accelerate the convergence of the algorithm by tighten the linear relaxation of the original problem. In addition, new techniques of generating piercing cut were developed for our problem. The computational results showed that the proposed algorithm outperform the MIP solver of commercial software CPLEX 12.1.

B. Future Works

The study of lane reservation problem in this paper is the first part of our project. In the future, more transportation problems, such as dynamic transportation network design for automated trucks, and more practical transportation factors will be considered in our work. In addition, a further study of more advanced techniques in cut-and-solve method will be developed in future work.

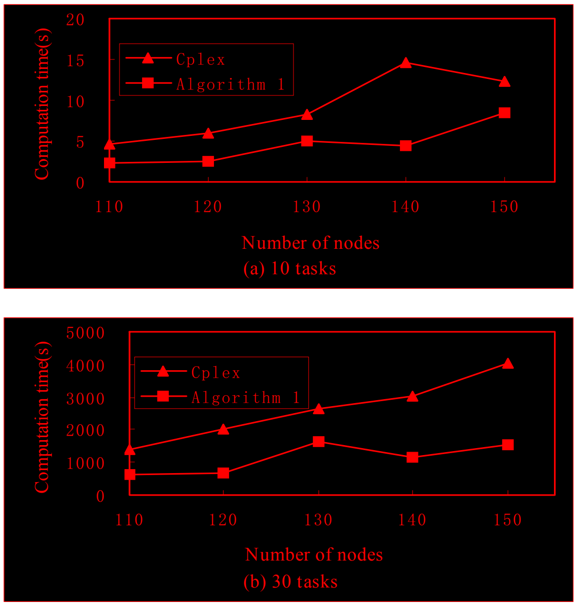

Fig. 2. Computation time of Algorithm *1* and CPLEX. (a) instances with 10 tasks (b) instances with 30 tasks

VI. ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers for their insightful remarks and suggestions on improving the quality of the paper.

REFERENCES

[1] S. Melkote and M. S. Daskin, "An integrated model of facility location and transportation network design," *Transportation Research Part A: Policy and Practice*, vol. 35, no. 6, pp. 515–538, July 2001.

[2] L. Equi, G. Gallo, S. Marziale, and A. Weintraub, "A combined transportation and scheduling problem," *European Journal of Operational Research*, vol. 97, no. 1, pp. 94–104, Feb. 1997.

[3] N. Ravi, S. Smaldone, L. Iftode, and M. Gerla, "Lane reservation for highways (position paper)," in *Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC'97)*, Seattle, Washington, USA, Sept. 2007, pp. 795–800.

[4] E. Zagorianakos, "Athens 2004 olympic games transportation plan: a missed opportunity for strategic environmental assessment," *Journal of Transport Geography*, vol. 12, no. 2, pp. 115–125, June 2004.

[5] Y. Wu, C. Chu, F. Chu, and N. Wu, "Heuristic for lane reservation problem in time constrained transportation," in *Proc. IEEE International Conference on Automation Science and Engineering*, Bangalore, India, Aug. 2009, pp. 543–548.

[6] Z. Drezner and H. W. Hamacher, Eds., *Facility Location: Applications and Theory*. New York: Springer, 2002.

[7] G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 3, pp. 345–358, June 1992.

[8] P. Toth and D. Vigo, Eds., *The Vehicle Routing Problem*. Philadelphia, USA: Society for Industrial and Applied Mathematics, 2002.

[9] P. Badeau, F. Guertin, M. Gendreau, J.-Y. Potvin, and E. Taillard, "A parallel tabu search heuristic for the vehicle routing problem with time windows," *Transportation Research Part C: Emerging Technologies*, vol. 5, no. 2, pp. 109–122, Apr. 1997.

[10] F. Chu, N. Labadi, and C. Prins, "A scatter search for the periodic capacitated arc routing problem," *European Journal of Operational Research*, vol. 169, no. 2, pp. 586–605, Mar. 2006.

[11] M. Gendreau, F. Guertin, J.-Y. Potvin, and R. Séguin, "Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries," *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 3, pp. 157–174, June 2006.

[12] D. Naso, M. Surico, B. Turchiano, and U. Kaymak, "Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete," *European Journal of Operational Research*, vol. 177, no. 3, pp. 2069–2099, Mar. 2007.

[13] J. E. Beasley, "Lagrangean heuristics for location problems," *European Journal of Operational Research*, vol. 65, no. 3, pp. 383–399, Mar. 1993.

[14] Z. Zhu, F. Chu, and L. Sun, "The capacitated plant location problem with customers and suppliers matching," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 3, pp. 469–480, May 2010.

[15] J. Li, F. Chu, and C. Prins, "Lower and upper bounds for a capacitated plant location problem with multicommodity flow," *Computer & operations research*, vol. 36, no. 11, pp. 3019–3030, Nov. 2009.

[16] R. Radharamanan and L. Choi, "A branch and bound algorithm for the travelling salesman and the transportation routing problems," *Computers & Industrial Engineering*, vol. 11, no. 1-4, pp. 236–240, 1986.

[17] M. Sun, "The transportation problem with exclusionary side constraints and two branch-and-bound algorithms," *European Journal of Operational Research*, vol. 140, no. 3, pp. 629–647, Aug. 2002.

[18] Éric Duchenne, G. Laporte, and F. Semet, "Branch-and-cut algorithms for the undirected m-peripatetic salesman problem," *European Journal of Operational Research*, vol. 162, no. 3, pp. 700–712, May 2005.

[19] Y. Yu, H. Chen, and F. Chu, "A new model and hybrid approach for large scale inventory routing problems," *European Journal of Operational Research*, vol. 189, no. 3, pp. 1022–1040, Sept. 2008.

[20] S. Climer and W. Zhang, "Cut-and-solve: An iterative search strategy for combinatorial optimization problems," *Artificial Intelligence*, vol. 170, no. 8-9, pp. 714–738, June 2006.

[21] Z. Yang, F. Chu, and H. Chen, "A cut-and-solve approach for the single-source capacitated facility location problem based on fenchel cutting plane," *European Journal of Operational Research*, submitted for publication.

[22] Y. Fang, F. Chu, S. Mamar, and A. Che, "An optimal algorithm for solving lane reservation problem on transportation network," *Transportation Research Part C: Emerging Technologies*, submitted for publication.

[23] *IBM ILOG CPLEX 12.1*, 2009.

[24] R. T. Wong, "A dual ascent approach for steiner tree problems on a directed graph," *Mathematical Programming*, vol. 28, no. 3, pp. 271–287, 1984.

[25] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

[26] E. Chávez, N. Mitton, and H. Tejeda, "Routing in wireless networks with position trees," in *Proceedings of the 6th international conference on AD-HOC, mobile and wireless networks*, Morelia, Mexico, Sept. 2007, pp. 32–45.