An Integer Programming Approach for the Rural Postman Problem with Time Dependent Travel Times

Guozhen Tan and Jinghao Sun

School of Computer Science and Technology,
Dalian University of Technology, Dalian, China
gztan@dlut.edu.cn, jhsun@mail.dlut.edu.cn

Abstract. The Chinese Postman Problem is a famous and classical problem in graph theory. This paper introduces a new variant of this problem, namely Rural Postman Problem with Time Dependent Travel Times, which is motivated from scheduling with time dependent processing times. An arc-path formulation of the problem is given and strong valid inequalities are derived. A subset of constraints in this formulation has a strong combinatorial structure, which is used to define the polytope of arc-path alternation sequence. The facial structure of this polytope is investigated and facet defining inequalities are presented which may be helpful to tighten the integer programming formulation. Computational results that verifies the effect of facet defining and strong valid inequalities are presented.

Keywords: Time Dependent, Rural Postman Problem, Polyhedral Combinatorics, Valid Inequalities, Arc-Path Formulation.

1 Introduction

Let $D(V, A)$ be a directed time dependent network, where V is the vertex set, A is the arc set, and $A_R \subseteq A$ is a set of required arcs. Each arc $a_{ij} \in A$ starting at time t_i is associated with the travel time $D_{ij}(t_i)$. Let $v_0 \in V$ be the origin vertex and t_0 be the starting time. The Rural Postman Problem with Time Dependent Travel Times (RPPTDT) aims to find a minimum travel time tour starting at v_0 and at time t_0 and passing through each required arc $a_{ij} \in A_R$ at least once. When $A_R = A$, the RPPTDT problem becomes the Chinese Postman Problem with Time Dependent Travel Times, a special case of the RPPTDT.

The RPPTDT is motivated from scheduling with time dependent processing times [1] (See Sect. 5 for details). To the author's knowledge, the RPPTDT problem does not seem to have been studied before but similar timing sensitive arc routing problems have been reported, such as the Arc Routing Problem with Time Dependent Service Costs (ARPTDC) [2] and the Arc Routing with Time Windows (ARPTW) [3]. When there are no such additional timing sensitive constrains, the traditional arc routing problems (without timing constraint) can

B. Fu and D.-Z. Du (Eds.): COCOON 2011, LNCS 6842, pp. 414–431, 2011.

be solved efficiently by using the integer programming approach based on the polyhedral theory [4,5,6]. However, the polyhedral results for the timing sensitive arc routing problems cannot be easily obtained from previous works. Since presently there are two kinds of methods to solve timing sensitive arc routing prblems. The most widely used method for ARPTDC and ARPTW uses a transformation into the corresponding node routing problems. Then the latter problems can be solved by the efficient existing algorithms directly without any modification. As the travel time in the ARPTW and ARPTDC is a known constant, the transformation taking the shortest-path algorithm as their core processes is polynomially bounded [7].

At the very beginning of our research, we expect to solve the RPPTDT by transformation method. However, the RPPTDT is different from ARPTDC and ARPTW as the constant travel time assumption never holds on the time dependent network. Thus the transformation method for the RPPTDT is not algorithmic in nature known that its subproblem becomes the time dependent shortest-path problem which has been proved to be $\mathcal{NP}$-Hard [8]. Therefore, a new methodology for solving RPPTDT should be proposed, and it would be fair to say that the direct integer programming approach with polyhedral result might be the outstanding approaches. In this paper, an integer linear programming for the RPPTDT is proposed, the constraint set of which is divided into two parts. One has a strong combinatorial structure, which defines the polytope of arc-path alternation sequence (APAS), and the other closely related to the time dependent travel times. The facial structure of the APAS polytope is investigated and the facet defining inequalities are derived which may be helpful to tighten the integer programming formulation of RPPTDTs. Moreover, we also propose some further valid inequalities. Computational results verifies the effect of facet defining and strong valid inequalities.

The remainder of this paper is organized as follows. Mathematical formulation of the RPPTDT is introduced in Sect. 2. Some results on the RPPTDT polyhedron are presented in Sect. 3. Based on these polyhedral results, a cutting plane algorithm was proposed, and the computational results are reported in Sect. 4. In Sect. 5, we introduce the application of the RPPTDT to scheduling with time dependent processing times. Concluding remarks are made in the last section.

2 Problem Formulation

The RPPTDT is formulated as a mixed integer linear programming (MILP) problem whenever the travel time is a known step function $D_{ij}(t_i)$ of the starting time t_i at vertex v_i. In this way, the total period associated with each arc $(i, j) \in A$ can be divided into several time intervals. Once the time interval during which the postman starts traversing arc (i, j) is known, the travel time of arc (i, j) is a known constant. A travel time step function for arc (i, j) with three time intervals is shown in Figure 1.

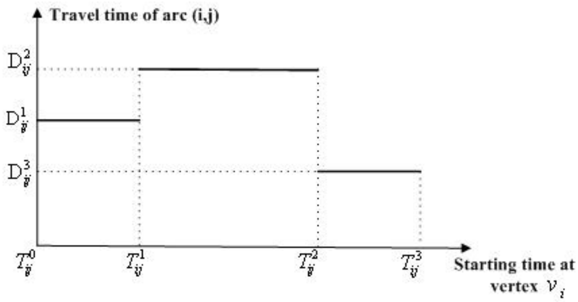

Fig. 1. Travel time step function for arc (i, j) with three time intervals

The MILP for RPPTDT is called "Arc-Path Formulation" since its main idea is to formulate the RPPTDT-tour as the alternation of arcs and paths as follow:

$$f = (v_0 - p_1(\tau_1) - a_1(t_1) - p_2(\tau_2) - \cdots - a_K(t_K) - p_{K+1}(\tau_{K+1}) - v_0)$$

where $a_k \in A_R$ starting at time $t_k(k = 1, \cdots, |A_R|)$ is the kth arc serviced in the RPPTDT-tour, and p_k $(k = 1, \cdots, |A_R| + 1)$ is the path starting at time τ_k that connects the required arcs a_{k-1} and a_k (in particular, $\tau_1 = t_0$). In the RPPTDT-tour re-formulated above, the starting time is unique for each traverse (service) of an arc in A (A_R).

To introduce the new arc-path integer program, we first summarize the notations below.

Constants

n = number of vertices in V including the origin depot v_0;
m = number of arcs in A;
K = number of required arcs in A_R;
$N^+(i)$ = the set of outgoing vertices of v_i;
$N^-(i)$ = the set of incoming vertices of v_i;
H = number of time intervals associated with each arc $(i, j) \in A$;
B = a large number;
T_{ij}^h = the upper bound of the hth interval associated with arc $(i, j) \in A$ $(h = 1, \cdots, H)$– see Figure 1;
D_{ij}^h = travel time of arc (i, j) starting at v_i during the hth time interval.

Decision variables

$$x_{ij}^k = \begin{cases} 1: & \text{if the } k\text{th arc } a_k \text{ seviced in RPPTDT-tour is } (i,j) \\ 0: & \text{otherwise} \end{cases}$$

$$y_{ij}^k = \begin{cases} 1: & \text{if arc } (i,j) \text{ is contained in the } k\text{th path } p_k \text{ of RPPTDT-tour} \\ 0: & \text{otherwise} \end{cases}$$

$$\delta_{ij}^{k,h} = \begin{cases} 1: & \text{arc } (i,j) \text{ is serviced as the } k\text{th required arc } a_k \text{ during the } h\text{th interval} \\ 0: & \text{otherwise} \end{cases}$$

$$\gamma_{ij}^{k,h} = \begin{cases} 1: & \text{arc } (i,j) \text{ is travesed in the } k\text{th path } p_k \text{ during the } h\text{th interval} \\ 0: & \text{otherwise} \end{cases}$$

t_i^k = starting time of arc (i,j) if (i,j) is the kth arc serviced in RPPTDT-tour;
τ_i^k = starting time of arc (i,j) if (i,j) is contained in the kth path of RPPTDT-tour. In particular, τ_0^1 and τ_0^{K+1} are the starting time and ending time of RPPTDT-tour respectively.
With these notations the RPPTDT may be formulated as follows:

$$\min \sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{(i,j)\in A_R} D_{ij}^h \delta_{ij}^{k,h} + \sum_{k=1}^{K+1}\sum_{h=1}^{H}\sum_{(i,j)\in A} D_{ij}^h \gamma_{ij}^{k,h} \tag{1}$$

$$\text{S.t.} \sum_{k=1}^{K} x_{ij}^k = 1, \quad (i,j) \in A_R, \tag{2}$$

$$\sum_{(i,j)\in A_R} x_{ij}^k = 1, \quad k = 1, \cdots, K, \tag{3}$$

$$\sum_{j\in N^+(i)} y_{ij}^k \leq 1, \quad v_i \in V, \quad k = 1, \cdots, K, \tag{4}$$

$$\sum_{j\in N^-(i)} y_{ji}^k \leq 1, \quad v_i \in V, \quad k = 1, \cdots, K+1, \tag{5}$$

$$\sum_{j\in N^+(i)} y_{ij}^k - \sum_{j\in N^-(i)} y_{ji}^k = \sum_{j\in N^-(i)} x_{ji}^{k-1} - \sum_{j\in N^+(i)} x_{ij}^k, \quad v_i \in V, \quad k = 2, \cdots, K, \tag{6}$$

$$\sum_{j\in N^+(i)} y_{ij}^1 - \sum_{j\in N^-(i)} y_{ji}^1 = \begin{cases} 1 - \sum_{j\in N^+(i)} x_{ij}^1, & i = 0, \\ -\sum_{j\in N^+(i)} x_{ij}^1, & i \neq 0, \end{cases} \tag{7}$$

$$\sum_{j\in N^+(i)} y_{ij}^{K+1} - \sum_{j\in N^-(i)} y_{ji}^{K+1} = \begin{cases} \sum_{j\in N^-(i)} x_{ji}^K - 1, & i = 0, \\ \sum_{j\in N^-(i)} x_{ji}^K, & i \neq 0, \end{cases} \tag{8}$$

$$\tau_0^1 \geq t_0 \tag{9}$$

$$t_i^k \geq \tau_i^k, \quad v_i \in V, \quad k = 1, 2, \ldots, K, \tag{10}$$

$$\tau_i^k \geq t_i^{k-1}, \quad v_i \in V, \quad k = 2, \ldots, K+1, \tag{11}$$

$$t_j^k - t_i^k \geq D_{ij} + \delta_{ij}^{k,h} B - B \quad k = 1, 2, \ldots, K; h = 1, \cdots, H; (i,j) \in A_R; \tag{12}$$

$$\tau_j^k - \tau_i^k \geq D_{ij} + \gamma_{ij}^{k,h} B - B \quad k = 1, 2, \ldots, K+1; h = 1, \cdots, H; (i,j) \in A; \tag{13}$$

$$\sum_{h=1}^{H} \delta_{ij}^{k,h} = x_{ij}^k \quad \forall (i,j) \in A_R;\ k = 1, \cdots, K; \tag{14}$$

$$t_i^k + B(\delta_{ij}^{k,h} - 1) \leq T_{ij}^h \quad \forall (i,j) \in A_R;\ k = 1, \cdots, K;\ h = 1, \cdots, H; \tag{15}$$

$$t_i^k \geq T_{ij}^{h-1} \delta_{ij}^{k,h} \quad \forall (i,j) \in A_R;\ k = 1, \cdots, K;\ h = 1, \cdots, H; \tag{16}$$

$$\sum_{h=1}^{H} \gamma_{ij}^{k,h} = y_{ij}^{k} \quad \forall (i,j) \in A;\ k = 1, \cdots, K+1; \tag{17}$$

$$\tau_i^k + B(\gamma_{ij}^{k,h} - 1) \leq T_{ij}^h \quad \forall (i,j) \in A;\ k = 1, \cdots, K+1;\ h = 1, \cdots, H; \tag{18}$$

$$\tau_i^k \geq T_{ij}^{h-1} \gamma_{ij}^{k,h} \quad \forall (i,j) \in A;\ k = 1, \cdots, K+1;\ h = 1, \cdots, H. \tag{19}$$

$$x_{ij}^k = \{0,1\}, \quad (i,j) \in A_R, \quad k = 1, \cdots, K, \tag{20}$$

$$y_{ij}^k = \{0,1\}, \quad (i,j) \in A, \quad k = 1, \cdots, K+1, \tag{21}$$

The objective function (1) minimizes the total travel time of RPPTDT-tour. Constraint (2) ensures that all required arcs must be serviced exactly once. Constraint (3) states that there is only one kth arc serviced in RPPTDT-tour ($k = 1, \cdots, K$). Constraints (4) and (5) ensure that each vertex in the kth path p_k should be passed at most once ($k = 1, \cdots, K+1$). In order to explain Constraints (6-8), we first divide the RPPTDT-tour f into three parts: $\omega_1 = (v_0 - p_1(\tau_1))$, $\omega_2 = a_1(t_1) - p_2(\tau_2) - \cdots - p_K(\tau_K) - a_K(t_K)$ and $\omega_3 = (p_{|A|+1}(\tau_{|A|+1}) - v_0)$. Then, Constraint (6) ensures that each internal vertex v_i in the middle part ω_2 must meet one of the following four conditions: 1) If v_i is the terminus of a_{k-1} in ω_2, but not the origin of a_k, then the out-degree of v_i in path p_k is one more than its in-degree. 2) Else if v_i is the origin of a_k, but not the terminus of a_{k-1}, then the out-degree of v_i in path p_k is one less than its in-degree. 3) Else if v_i is neither the origin of a_k nor the terminus of a_{k-1}, then the out-degree of v_i in path p_k is equal to its in-degree. 4) Else if the $k-1$th and kth required arcs are adjacent with the common vertex v_i, then the kth path in f would be void, and the out-degree of v_i contained in p_k is equal to its in-degree.

Constraint (7) describes the following conditions for any vertex v_i in the first part ω_1. 1) For $i = 0$, if v_0 is the origin of the first arc serviced in f, then the out-degree of v_0 in path p_1 is equal to its in-degree. Otherwise the out-degree of v_0 in path p_1 is one more than its in-degree. 2) For any $i \neq 0$, if v_i is the internal vertex of the first path in f, then the out-degree of v_i in path p_1 is equal to its in-degree. Otherwise the out-degree of v_i in path p_1 is one less than its in-degree.

For the last part ω_3, Constraint (8) can also be proposed for similar reason.

In constraint (9), t_0 is a lower bound of the earliest starting time of the RPPTDT-tour. Constraints (10) and (11) indicate the alternation sequence of arcs and paths in terms of time. Constraint (12) and (13) calculate the travel time of arc (i,j) when the postman services arc (i,j) as the kth required arc or traverses arc (i,j) in the kth path starting at vertex v_i during the hth time interval. Constraint (14) ensures that if the postman services arc (i,j) as the kth required arc, then it must depart from v_i within one time interval. If the departure time t_i^k of v_i belongs to the hth time interval of (i,j), then it is necessary to check whether t_i^k is in the range from the lower bound T_{ij}^{h-1} to the upper bound T_{ij}^h, which are guaranteed by constraints (15) and (16) separately. Constraints (17-19) play a similar role for the arc (i,j) that is traversed in the kth path of the Chinese tour.

3 Results on RPPTDT Polyhedron

In this section, we summarize some polyhedral results for the RPPTDT, based on which our cutting plane algorithm is developed.

According to the formulation in Sect. 2, every RPPTDT-tour in D can be considered as an arc-path alternation sequence. We can associate with this sequence an integer vector $f = (y_1, x_1, \cdots, x_K, y_{K+1})$, where x_k and y_k are also integer vectors, denoted as $x_k = (x_{ij}^k : (i,j) \in A_R) \in \mathbb{B}^K$ and $y_k = (y_{ij}^k : (i,j) \in A) \in \mathbb{B}^m$ respectively. It is evident that the incidence vector f of each arc-path sequence satisfies Constraints (2-8) and, conversely, each feasible solution of inequality system (2-8) is the incidence vector of an arc-path sequence. Therefore, we will use the Constraints (2-8) which have a strong combinatorial structure to define the polytope of APAS, $P_{APAS}(D)$, and denote by $\mathscr{F}$ the set of the feasible solutions for the APAS polytope $P_{APAS}(D)$:

$$\mathscr{F} = \{f \in \mathbb{B}^{K^2+mK+m} : f \texttt{ satisfies Constraints (2-8)}\}$$

This representation yields to the definition of $P_{APAS}(D)$ as the convex hull of all the feasible solution for the APAS polytope: $P_{APAS}(D) = conv(\mathscr{F})$. The polyhedral results of $P_{CA}(D)$ are listed as follows.

Theorem 1. *$Dim(P_{APAS}(D)) = K^2 + (m - n - 2)(K + 1) + 4$ if $m - n \geq 1$.*

Theorem 1 exhibits the dimension of the APAS polytope, the details about the proof of which can be found in Appendix A.

Next we will introduce a valid inequality in the APAS polytope. Note that the path indicated by each sub-vector $y_k = (y_{ij}^k : (i,j) \in A)$ of y would rather be named as "subgraph" than "path". That is because the extra circuit is not forbidden by Constraints (2-8). When the $k-1$th and kth serviced arcs of the RPPTDT-tour are adjacent with a common vertex v_s, the vector y_k must meet one of the following three cases: 1) the vector y_k is void; 2) the arcs of a circuit that passes though v_s is contained in y_k; 3) the arcs of a circuit that does not pass though v_s is contained in y_k. It is easy to see that the "subgraph" indicated by y_k is not always a path, but it sometimes contains circuits. The RPPTDT-tour will not be optimal if the associated y_k contains circuit discussed in cases (2) and (3). In particular, the RPPTDT-tour taking circuit described in case (3) as y_k will not be feasible, because this circuit that is disjoint with the current tour will be never traversed by the postman. The circuit described in case (3) can be called isolated circuit which can be avoided by the following valid inequality.

$$\sum_{(s,t)\in\omega_{jl}} y_{st}^k \geq \sum_{(l,r)\in N_l^+} y_{lr}^k - (1 - \sum_{(i,j)\in N_j^-} x_{ij}^{k-1}) \quad j,l \in V; \quad k = 2, 3, \ldots, K+1 \tag{22}$$

where ω_{jl} is the minimum cut set with the source vertex v_j and sink vertex v_l in D. When the $k-1$th arc is (i,j) in D, the inequality in (22) degenerates into $\sum_{(s,t)\in\omega_{jl}} y_{st}^k \geq \sum_{(l,r)\in N_l^+} y_{lr}^k$. If an isolated circuit containing any $(l,r) \in D$

is added into the kth path of the RPPTDT-tour, then the right side of the inequality is equal to 1 while the left side is equal to 0 as there is no path from source vertex v_j to sink vertex v_l. Thus the isolated circuits can be forbidden by the inequalities (22), which defines facet of APAS polytope below.

Theorem 2. *For arbitrary two vertices $v_j, v_l \in V$ and the arc index k, the inequalities (22) defines a facet of $P_{APAS}(D)$, if the out-degree or the in-degree of each vertex in D is more than 2, and the out-degree or the in-degree of v_l is at most 3.*

Details about the proof of Theorem 2 can be found in Appendix B.

To check the effect of the facet defining inequalities (22), we have designed an LP-based heuristic cutting plane algorithm that adds inequalities (22) as cutting planes. Our computational experiments in Sect. 4 show that inequalities (22) can provide a better lower bound. Moreover, as our next theorem shows, inequalities (22) possesses another nice property.

Theorem 3. *The separation problem for inequalities (22) can be solved in polynomial time.*

Proof. The separation problem for inequalities (22) is to determine whether a given (rational) vector $f^* \in \mathbb{Q}^{K^2+mK+m}$ satisfies inequalities (22), which can be solved as follows. Firstly, for each $v_j, v_l \in V$ and $k = 2, \cdots, K+1$, one can calculate the right-side of inequalities (22) $\sum_{(l,r)\in N_l^+} y_{lr}^k - (1 - \sum_{(i,j)\in N_j^-} x_{ij}^{k-1})$ by direct substitution, the result value of which is denoted by u_{jl}^k. Then, we define the weight $w_{st}^k := y_{st}^k$ for each arc $(s,t) \in A$. Since, for any minimum cut set ω_{jl}, f^* satisfies (22) $\Leftrightarrow \sum_{\omega_{jl}} w_{ij}^k \geq u_{jl}^k$, one can see that the separation problem for the inequalities (22) reduces to the problem of determining a minimum cut of D with respect to the nonnegative weight function w. Padberg and Rao (1982) have shown that the latter is polynomially solvable. Hence the theorem is proved. □

Moreover, as the timing constraints involve a "big" number B that generally results in weak LP relaxations, strong timing constraints should also be derived, the forms of which are listed as follows.

Successors' inequalities:

$$\delta_{ij}^{k,h} + \sum_{(j,l)\in A} \sum_{p\in A_{ijl}^h} \gamma_{jl}^{k+1,p} \leq 1 \quad \forall (i,j) \in A_R; k = 1, 2, \cdots, K; h = 1, 2, \cdots, H; \tag{23}$$

$$\gamma_{ij}^{k,h} + \sum_{(j,l)\in A} \sum_{p\in A_{ijl}^h} (\gamma_{jl}^{k,p} + \delta_{jl}^{k,p}) \leq 1 \quad \forall (i,j) \in A; k = 1, 2, \cdots, K; h = 1, 2, \cdots, H; \tag{24}$$

Predecessors' inequalities:

$$\sum_{(i,j)\in A} \sum_{h\in B_{ijl}^p} \gamma_{ij}^{k,h} + \delta_{jl}^{k+1,p} \leq 1 \quad \forall (j,l) \in A_R; k = 1, 2, \cdots, K; h = 1, 2, \cdots, H; \tag{25}$$

$$\sum_{(i,j)\in A}\sum_{h\in B^p_{ijl}}(\delta^{k-1,h}_{ij}+\gamma^{k,h}_{ij})+\gamma^{k+1,p}_{jl}\le 1 \quad \forall (j,l)\in A; k=2,\cdots,K; h=1,2,\cdots,H; \tag{26}$$

where

$$A^h_{ijl}=\{p\,|T^p_{jl}<T^{h-1}_{ij}+D^h_{ij}\quad or \quad T^{p-1}_{jl}>T^h_{ij}+D^h_{ij}+\mathtt{DIFF}\}$$

$$B^p_{ijl}=\{h\,|T^p_{jl}<T^{h-1}_{ij}+D^h_{ij}\quad or \quad T^{p-1}_{jl}>T^h_{ij}+D^h_{ij}+\mathtt{DIFF}\}$$

and

$$DIFF=max\{0,D^{p-1}_{jl}-D^p_{jl}\}$$

Successor constraints (23) and (24) ensure that if the postman services or traverses arc (i,j) starting at v_i during the hth time interval, then for any successor arc (j,l) of (i,j), it is impossible to traverse or service (j,l) during the time intervals in A^h_{ijl} which includes a term DIFF and has been introduced by Malandraki [9]. The inclusion of the term DIFF allows the travel time step function on (j,l) to behave as if it was a piecewise linear continuous function when the travel time in period p is less than that in the preceding period $p-1$, as discussed in [9]. Similarly, the predecessor (25) and (26) constraints restricts the time intervals of all the predecessors of each arc in the Chinese tour.

4 Computational Results

This section summarizes the computational results obtained with a cutting plane algorithm using the facet defining inequalities (22) and valid inequalities (23-26). The algorithm is a standard cutting plane method, where an initial linear program with constraints (2-21) is set up that is solved to optimality. One checks whether the optimum solution x^* represents a Chinese tour and if not tries to find valid inequalities (22) and (23-26) that are violated by x^*. These cutting planes are added to the current LP and the process is repeated. When the current LP solution is fractional and violates no inequalities (22) and (23-26), apply a heuristic that uses the fractional LP solution x^* to construct an approximate Chinese tour, calculate an upper bound and stop.

This cutting plane algorithm algorithm coded in C++ using Microsoft Visual C++ 6.0 Environment and LINGO library was run on a PC with a Pentium processor at 2.2GHz and 1G RAM. Several randomly generated instances were used for our computational study.

Let $D=(V,A)$ be a randomly generated network with vertex set V and arc set A. Suppose $|V|=n$, then the vertices of V are indexed by the integers $1,2,\cdots,n$. Let the period associated with each arc $(i,j)\in A$ be divided into H time intervals, then the travel time associated with each arc $(i,j)\in A$ is treated as a step function with H time intervals for our computational study. We generate time dependent travel time randomly by the step function generation procedure described in Procedure 4.1.

Procedure 4.1

1. Let the randomly generated network be $D(V, A)$, and then randomly distribute these $|V|$ vertices of D in the $[0, 100]^2$ square.
2. For each arc $(i, j) \in A$, randomly generate an integer d that is proportional to the Euclidean distance between its two incident vertices. Denote by D^1_{ij} the travel time of the earliest time interval associated with arc (i, j). Set $D^1_{ij} = d$.
3. Generate the duration of each hth interval randomly$(h = 1, \cdots, H)$.
4. For each $h = 2, \cdots, H$, and each arc $(i, j) \in A$, randomly generate an integer Δd in a fluctuation interval $[-r, r]$, where r is an arbitrary but fixed positive integer. Set $D^h_{ij} := D^1_{ij} + \Delta d$, and ensure that the duration of the hth time interval is not smaller than its travel time D^h_{ij}.

The algorithm is tested on two sets of randomly generated instances. In all instances, the number of vertices ranges from 10 to 25, and the number of arcs ranges from 20 to 50; the travel time is treated as the step function with 3 and 4 time intervals, and the percentage R of required arcs ranges from 10% to 30%. In table 1 we indicate the characteristics of the three instance sets in terms of the number of time intervals H and percentage R.

Table 1. Characteristics of the two instance sets

Inst	H	R
A1-1/A1-4	3	10%
A2-1/A2-4	3	20%
A3-1/A3-4	3	30%
B1-1/B1-4	4	10%
B2-1/B2-4	4	20%
B3-1/B3-4	4	30%

Computational results obtained by the cutting plane algorithm for all 42 test instances are summarized in Table 2 and Table 3. For each instance, $|V|$ is the number of vertex, and $|A|$ is the number of arcs. LP is the value of the linear programming relaxation of the RPPTDT formulation described in Sect. 2. LB_1 is the lower bound obtained by the cutting planes algorithm only using the valid inequalities (23-26), and LB_2 is the lower bound obtained by adding both facet defining and valid inequalities (22) and (23-26). UB is the value of the best feasible solution obtained, i.e., the best upper bound. The values of LP, LB_1 and LB_2 are rounded up to the closest integer. Columns G_0, G_1 and G_2 display relative gaps between the best feasible solution (UB) and lower bounds LP, LB_1 and LB_2, computed as $G_0 = ((UB - LP)/LP) \times 100$, $G_1 = ((UB - LB_1)/LB_1) \times 100$ and $G_2 = ((UB - LB_2)/LB_2) \times 100$.

Results presented on Table 2 and Table 3 indicate that the proposed algorithm can solve instances up to 25 vertices and 50 arcs. The relative gap G_2 is 3.16%

Table 2. Computational result of the first group of instances

Inst.	$\|V\|$	$\|A\|$	LP	LB_1	LB_2	UB	$G_0(\%)$	$G_1(\%)$	$G_2(\%)$
A1-1	10	20	230.67	231.88	301.44	308	33.52	32.83	2.18
A1-2	13	26	275.83	277.56	311.00	311	12.75	12.05	0.00
A1-3	15	30	260.47	260.47	298.30	300	15.18	15.18	0.57
A1-4	17	34	302.78	305.70	364.00	377	24.51	23.32	3.57
A1-5	20	40	244.25	255.36	300.00	310	26.92	21.40	3.33
A1-6	22	44	261.36	266.33	341.67	357	36.59	34.04	4.49
A1-7	25	50	331.52	331.52	370.00	378	14.02	14.02	2.16
A2-1	10	20	260.00	260.00	541.27	541	108.08	108.08	0.00
A2-2	13	26	342.75	348.00	516.73	530	54.63	52.30	2.71
A2-3	15	30	297.44	297.57	577.12	591	98.70	98.61	2.43
A2-4	17	34	275.61	277.84	533.00	561	103.55	101.91	5.25
A2-5	20	40	279.59	261.00	488.37	518	85.27	98.47	6.07
A2-6	22	44	300.18	304.90	574.00	577	92.22	89.24	0.52
A2-7	25	50	372.56	377.00	583.00	604	62.12	60.21	3.60
A3-1	10	20	351.00	351.77	601.00	601	71.23	70.85	0.00
A3-2	13	26	329.64	330.00	593.38	620	88.08	87.88	4.49
A3-3	15	30	366.67	367.21	607.31	617	68.27	68.12	1.60
A3-4	17	34	351.47	355.67	591.56	613	74.41	72.64	3.62
A3-5	20	40	376.08	378.00	613.07	627	66.72	65.87	2.27
A3-6	22	44	388.57	390.91	605.73	622	60.07	59.48	2.69
A3-7	25	50	350.51	351.00	637.35	657	87.44	87.18	3.08

for all the 42 instances on average. It is easy to show that the best upper bound UB obtained by the cutting plane heuristic is always very close to the value of the new linear relaxation bound LB_2. Meanwhile, we also find that the changes of percentage R might affect gap $G2$. For the 21 instances with $R = 10\%$, the relative gap $G2$ attains 2.42% on average, and it becomes 3.23% and 3.82% on average for those instances with $R = 20\%$ and $R = 30\%$ respectively.

The lower bound LB_1 and LB_2 obtained by adding cutting planes improves substantially the linear relaxation bound LP of the original formulation for all 42 instances. By comparing columns G_0, G_1 and G_2, one can appreciate how much our valid and facet defining inequalities have contributed to the relative gaps. As shown in Table 2 and Table 3, gap G_1 is equal to G_0 approximately on average, and gap G_2 is 66.84% smaller than G_0 on average. It is evident that the facet defining inequalities (22) are the most effective, followed by the successor and predecessor inequalities from Sect. 3.

5 Application in Scheduling with Time Dependent Processing Times

The RPPTDT problem can be applied to solving the problem of scheduling with time dependent processing times [1]. In the classical scheduling theory processing

Table 3. Computational result of the second group of instances

Inst.	$\|V\|$	$\|A\|$	LP	LB_1	LB_2	UB	$G_0(\%)$	$G_1(\%)$	$G_2(\%)$
B1-1	10	20	174.23	176.21	261.00	261	49.80	48.12	0.00
B1-2	13	26	152.67	155.96	277.44	290	89.95	85.95	4.53
B1-3	15	30	145.00	145.00	266.57	274	88.97	88.97	2.79
B1-4	17	34	114.01	116.00	197.00	206	80.69	77.59	4.57
B1-5	20	40	201.29	217.40	290.67	295	46.55	35.69	1.49
B1-6	22	44	157.99	157.99	247.56	258	63.30	63.30	4.22
B1-7	25	50	162.35	163.00	270.00	270	66.31	65.64	0.00
B2-1	10	20	211.75	212.98	402.00	422	99.29	99.06	4.98
B2-2	13	26	194.36	196.64	374.70	408	109.92	108.16	8.89
B2-3	15	30	203.00	203.00	427.04	441	117.24	117.24	3.27
B2-4	17	34	176.49	178.00	370.51	384	117.58	115.73	3.64
B2-5	20	40	293.18	296.74	507.00	513	74.98	72.88	1.18
B2-6	22	44	300.70	301.00	501.09	521	73.26	73.09	3.97
B2-7	25	50	281.60	281.65	483.35	517	83.59	83.56	6.96
B3-1	10	20	299.07	301.72	573.88	600	100.62	98.86	4.55
B3-2	13	26	319.33	320.00	588.67	612	91.65	91.25	3.96
B3-3	15	30	341.00	341.00	577.31	607	78.01	78.01	5.14
B3-4	17	34	375.60	377.00	607.03	632	68.26	67.64	4.11
B3-5	20	40	358.06	358.06	612.38	623	73.99	73.99	1.73
B3-6	22	44	400.01	400.01	621.09	648	62.00	62.00	4.33
B3-7	25	50	390.15	395.00	607.78	630	61.48	59.49	3.66

times of jobs are constant; however, there are many situations where the processing time depends on the starting time of the job. For example, in scheduling of steel rolling mills where the temperature of an ingot, while waiting in a buffer between the furnace and the rolling machines, has dropped below a certain temperature, then the ingot needs to be reheated to bring it up to the temperature required for rolling. In such situations the processing time of a job may be a function of its starting time.

Furthermore, we formulate the machine operation process as a directed graph $D(V, A)$, where each state s_i of machine corresponds to a vertex v_i in V, and each transition from machine states s_i to s_j corresponds to an arc (i, j) in A. From this point of view, the process of machining a job J_k can be seen as a sequence of state transitions on the machine. Let the starting state and ending state of this sequence be s_t and s_e respectively, and then we can add a new arc (t, e) into the network D to express machining job J_k, which is associated with a time dependent processing time. Note that the starting state of the current job may not be the same as the ending state of its immediate predecessor. It means that machine should reach the starting state of J_{k+1} through several extra transitions starting from the ending state of J_k during the process of machining jobs. Thus, the aim of this scheduling problem is to find a feasible job process strategy, such that the processing times of all the jobs and the extra times of additional transitions are minimal, which can be equivalently cast as the Rural Postman problem defined on the time dependent network D where an arc associated with each job should be serviced at least once.

6 Conclusions

We proposed a linear integer programming formulation, namely, arc-path formulation, based on arc-path decision variables for the RPPTDT. A subset of the constraints in this formulation has a strong combinatorial structure which defines the polytope of arc-path alternation sequence (APAS) in the RPPTDT-tour. A polyhedral investigation of the APAS yielded results on the dimension and facet defining inequalities of the APAS polytope. Moreover, two families of strong valid inequalities are also derived. Further research in this direction will be helpful to strengthen the integer programming formulation of the RPPTDT.

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.2005CB321904. These supports are gratefully acknowledged.

References

1. Sundararaghavan, P.S., Kunnathur, A.S.: Single machine scheduling with start time dependent processing times: Some solvable cases. European Journal of Operational Research 78, 394–403 (1994)
2. Tagmouti, M., Gendreau, M., Potvin, J.Y.: Arc routing problems with time-dependent service costs. European Journal of Operational Research 181, 30–39 (2007)
3. Mullaseril, P.A.: Capacitated rural postman problem with time windows and split delivery, Ph.D Thesis. University of Arizona (1996)
4. Ghiani, G., Laporte, G.: A branch-and-cut algorithm for the Undirected Rural Postman Problem. Mathematical Programming 87, 467–481 (2000)
5. Corberan, A., Romero, A., Sanchis, J.M.: The mixed general routing polyhedron. Mathematical Programming 96, 103–137 (2003)
6. Corberan, A., Mejia, G., Sanchis, J.M.: New Results on the Mixed General Routing Problem. Operations Research 52, 363–376 (2005)
7. Longo, H., Aragäo, M.P., Uchoa, E.: Solving capacitated arc routing problems using a transformation to the CVRP. Computers and Operations Research 33, 1823–1837 (2006)
8. Orda, A., Rom, R.: Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. Journal of ACM 37, 607–625 (1990)
9. Malandraki, C., Daskin, M.S.: Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms. Transportation Science 26, 185–200 (1992)
10. Bang, J., Gutin, G.: Digraphs: Theory, Algorithms and Applications. Springer, London (2001)

Appendix A: Dimension of the APAS Polytope

In order to prove the dimension of $P_{APAS}(D)$, we first introduce an approach to construct the affinely independent RPPTDT-tour in $\mathscr{F}$ in Sect. 6.1, then analyze the dimension of $P_{APAS}(D)$ in Sect. 6.2.

6.1 Affinely Independent RPPTDT-Tours in APAS Polytope

To show the facial structure of the APAS polytope $P_{APAS}(D)$, this section exhibits an approach to find affinely independent RPPTDT-tours in $P_{APAS}(D)$. According to the definitions of RPPTDT-tour set $\mathscr{F}$ and arc array set $\mathscr{X}$, the set of affinely independent RPPTDT-tours in $\mathscr{F}$ can be obtained by constructing the linearly independent arc arrays in $\mathscr{X}$. It is evident that the incidence vectors in $\mathscr{X}$ correspond to the full array of K arcs in A_R. Without loss of generality, let an arbitrary arc array be $a_K, a_{K-1} \cdots, a_1$, then its incidence vector $x \in \mathscr{X}$ can be written as $(x_1^K, x_2^{K-1}, \cdots, x_K^1)$, where $x_k^l = (x_{ij}^k : (i,j) \in A_R \wedge$ `if` $a_{ij} = a_l$`,` $x_{ij}^k = 1$`; otherwise,` $x_{ij}^k = 0)$ indicates that the kth arc serviced in the RPPTDT-tour is a_l. Obviously, the cardinality of $\mathscr{X}$ is $K!$, while the number of linearly independent arc arrays in $\mathscr{X}$ is much less. See Theorem 4 for more details.

Theorem 4. *There are $(K-1)^2+1$ linearly independent points in $\mathscr{X}$.*

Proof. When $K=1$, there is only one required arc in A_R. It is easy to see that the arc array in $\mathscr{X}$ is unique, which by itself is linearly independent.

When $K=l$, assume that there are $(l-1)^2+1$ linearly independent arc arrays in $\mathscr{X}$, and we will denote by $\mathscr{X}^l$ the set of these linearly independent arc arrays. Then we will prove that the number of linearly independent arc arrays in $\mathscr{X}$ is l^2+1 for $K=l+1$.

When $K=l+1$, we will construct the linearly independent arc array set $\mathscr{X}^{l+1}$ as follows.

Firstly, we will construct the first part of linearly independent arc arrays in $\mathscr{X}^{l+1}$. For each arc array $x \in \mathscr{X}^l$, written as $(x_1^i, \cdots, x_l^j)$ $(i,j \le l)$, we can append the incidence vector associated with arc a_{l+1} to x, then obtain a new arc array $x' = (x_1^i, \cdots, x_l^j, x_{l+1}^{l+1})$. Therefore, we can construct $(l-1)^2+1$ arc arrays taking a_{l+1} as their $l+1$th arc, the set of which is denoted as $\mathscr{X}_{l+1}^{l+1}$. It is evident that these arc arrays in $\mathscr{X}_{l+1}^{l+1}$ are linearly independent since the elements in set $\mathscr{X}^l$ are linearly independent.

Secondly, denote by x^i the arc array in $\mathscr{X}_{l+1}^{l+1}$ taking a_i as its lth arc, and other l linearly independent arc arrays can be obtained as follows. For each ith arc array $x^i = (x_1^j, \cdots, x_l^i, x_{l+1}^{l+1})$ in set $\{x^1, \cdots, x^l\}$, exchange the lth sub-vector with the $l+1$th sub-vectors, then a new arc array is obtained as $(x_1^j, \cdots, x_l^{l+1}, x_{l+1}^i)$. Thus, we can construct l arc arrays, the set of which is denoted by $\mathscr{X}_l^{l+1}$. It is easy to see that the last arc of the ith arc array in $\mathscr{X}_l^{l+1}$ is a_i. Therefore, the incidence matrix of $\mathscr{X}_l^{l+1}$ is non-singular since these l arcs $a_1, \cdots, a_l$ are linearly independent. Furthermore, note that the lth arc of each arc array $x \in \mathscr{X}_l^{l+1}$ is a_{l+1}, then the incidence matrix of $\mathscr{X}_{l+1}^{l+1}$ and $\mathscr{X}_l^{l+1}$ is also non-singular because of the linearly independence of the $l+1$ arcs $a_1, \cdots, a_{l+1}$.

Finally, we need to find the last $l-1$ linearly independent arc arrays of $\mathscr{X}^{l+1}$, the ith arc array of which is constructed as follow $(i = 1, \cdots, l-1)$. For a given arc array $x = (x_1^p, \cdots, x_i^i, \cdots, x_l^q, x_{l+1}^{l+1})$ in $\mathscr{X}_{l+1}^{l+1}$ whose ith arc is a_i, exchange the ith and the $l+1$th sub-vectors, then a new arc array is constructed as $(x_1^p, \cdots, x_i^{l+1}, \cdots, x_l^q, x_{l+1}^i)$. Note that each ith arc array constructed above is marked with a_{l+1} as its ith arc, it is evident that the incidence matrix of $\mathscr{X}^{l+1}$ is non-singular, hence, the l^2+1 incidence vectors in $\mathscr{X}^{l+1}$ are linearly independent. $\square$

According to the above theorem, it is easy to prove the following theorem.

Theorem 5. *There are at least* $(K-1)^2+1$ *affinely independent RPPTDT-tours in* $\mathscr{F}$.

Denote by $\mathscr{X}^*$ the set of linearly independent arc arrays analyzed in Theorem 4, it is easy to show that each arc array corresponds to at least one RPPTDT-tour in $\mathscr{F}$. Thus, let the set of affinely independent RPPTDT-tours induced from $\mathscr{X}^*$ be $\mathscr{F}^*$, we can easily concludes that its cardinality is at least $(K-1)^2+1$.

6.2 Proof of the Dimension of $P_{APAS}(D)$

In this section, we will show the upper bound and the lower bound of the dimension of $P_{APAS}(D)$. Let (A_1^E, b_1^E) and (A_2^E, b_2^E) be the coefficient matrices associated with the equalities system (2),(3) and equalities system (6-8) respectively. As we know, the number of equalities in Constraints (2) and (3) is $2K$, and there are $(K+1)n$ equalities in Constraints (6-8). The ranks of these two coefficient matrices are calculated in the following lemmas.

Lemma 1. *Rank*$(A_1^E, b_1^E) = 2K-1$.

Proof. (A_1^E, b_1^E) is a matrix with $2K$ rows and K^2+1 columns, as follow:

$$\begin{pmatrix} 1 & & & 1 & & & & 1 & & & 1 \\ & \ddots & & & \ddots & & \dots & & \ddots & & \vdots \\ & & 1 & & & 1 & & & & 1 & 1 \\ 1 & \dots & 1 & & & & & & & & 1 \\ & & & 1 & \dots & 1 & & & & & 1 \\ & & & & & & \ddots & & & & \\ & & & & & & & 1 & \dots & 1 & 1 \end{pmatrix}_{2K \times (K^2+1)} \tag{27}$$

We transform it with elementary row transformation. Firstly, subtract each elements in the first K rows from the ones in the $K+1$th row and obtain the following matrix:

$$\begin{pmatrix} 1 & & & 1 & & & & 1 & & & 1 \\ & \ddots & & & \ddots & & \dots & & \ddots & & \vdots \\ & & 1 & & & 1 & & & & 1 & 1 \\ & & & -1 & \dots & -1 & \dots & -1 & \dots & -1 & -K+1 \\ & & & 1 & \dots & 1 & & & \dots & & 1 \\ & & & & & & \ddots & & & & \vdots \\ & & & & & & & 1 & \dots & 1 & 1 \end{pmatrix}_{2K\times(K^2+1)} \tag{28}$$

Then, add the elements of the $K+2$th row to the ones of the $K+1$th row, and the elements of the $K+3$th row to the ones of the $K+2$th row, and so on, until the elements of the $2K$th row to the ones of the $2K-1$th row, as follow:

$$\begin{pmatrix} 1 & & & 1 & & & & & 1 & & & 1 \\ & \ddots & & & \ddots & & \dots & \dots & & \ddots & & \vdots \\ & & 1 & & & 1 & & & & & 1 & 1 \\ & & & -1 & \dots & -1 & \dots & \dots & -1 & \dots & -1 & -K+1 \\ & & & & & -1 & -1 & \dots & -1 & \dots & -1 & -K+2 \\ & & & & & & & \ddots & & & & \vdots \\ & & & & & & & & -1 & \dots & -1 & 1 \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & \dots & 0 & \dots & 0 & 0 \end{pmatrix}_{2K\times(K^2+1)} \tag{29}$$

According to (29), we know the elements of the last row are all zero, therefore $Rank(A_1^E, b_1^E) = 2K-1$. □

Lemma 2. *Rank*$((A_2^E, b_2^E))$=$Kn+n-1$.

Proof. (A_2^E, b_2^E) is a matrix with $(K+1)n$ rows and $(K+1)m+m^2+1$ columns, as follow:

$$\begin{pmatrix} a\ c & & & & & & f \\ & b\ a\ c & & & & & 0 \\ & & b\ a\ c & & & & 0 \\ & & & \ddots & & & \vdots \\ & & & & b\ a\ c & & 0 \\ & & & & & b\ a\ c & 0 \\ & & & & & & b\ a\ e \end{pmatrix}_{(Kn+n)\times[(K+1)m+m^2+1]} \tag{30}$$

Note that the above matrix is a partitioned matrix, where a is the incidence matrix of D with n rows and m columns, b and c are the in-arc and out-arc incidence matrix of D with n rows and m columns respectively, $e = (1\ 0\ 0 \dots\ 0)^T$

is a matrix with n rows and 1 column, and $f = (-1\ 0\ 0\ \ldots\ 0)^T$ is a matrix with n rows and 1 column, too. Obviously, $b + c = a$.

We transform it with elementary row transformation. Firstly, add the elements of the second row to the ones of the first row, and the elements of the third row to the ones of the second row, and so on, until the elements of the $(K+1)n$th row to the ones of the $(K+1)n - 1$th row, as follow:

$$\begin{pmatrix} a\ c & & & f \\ a\ a\ a\ c & & & f \\ a\ a\ a\ a\ a\ c & & & f \\ & \ddots & & \vdots \\ a\ a\ a\ a\ a\ a \ldots & a & a\ a\ c & f \\ a\ a\ a\ a\ a\ a \ldots & a & a\ a\ a\ c & f \\ a\ a\ a\ a\ a\ a \ldots & a & a\ a\ a\ a\ a & 0 \end{pmatrix}_{(Kn+n)\times[(K+1)m+m^2+1]} \tag{31}$$

Obviously, all the rows of the above partitioned matrix are linearly independent. Moreover, all the internal n rows in the kth row of matrix (31) are also linearly independent ($k = 1, \cdots, K$). So the rank of the first Kn rows of (A_2^E, b_2^E) is Kn. As we know, the rank of a is $n-1$. So the rank of the matrix in the last row of (A_2^E, b_2^E) is $n-1$. Therefore, the rank of (A_2^E, b_2^E) is $Kn + n - 1$. □

Based on the above Lemmas, the proof of Theorem 1 is given below.

Proof. Firstly, determine the upper bound of $Dim(P_{APAS}(D))$. Because the equations in (2) and (3) only associate with incidence vector $x \in \mathbb{B}^{K^2}$, and the equations in (6-8) associate with both incidence vector x and $y \in \mathbb{B}^{m(K+1)}$, it is evident that each row in (A_2^E, b_2^E) is not the linear combination of row vectors in (A_1^E, b_1^E). According to Lemma 1 and 2, the coefficient matrix (A^E, b^E) of equation set (2,3) and (6-8) satisfies $Rank(A^E, b^E) \geq Kn + 2K + n - 2$. As the $P_{APAS}(D)$ defines in space $\mathbb{B}^{m(K+1)+K^2}$, $Dim(P_{APAS}(D)) \leq K^2 + (m-n-2)(K+1) + 4$.

Secondly, determine the lower bound of $Dim(P_{APAS}(D))$.

According to Theorem 5, there are $(K-1)^2+1$ affinely independent RPPTDT-tours $\mathscr{F}^*$ which are induced from the linearly independent arc array set $\mathscr{X}^*$.

Moreover, another $m - n + 1$ affinely independent points, the set of which is denoted as $\mathscr{F}'$, can be constructed by the following way. According to the circuit structure of digraphs [10], there exist and only exist $m - n + 1$ linearly independent circuits in the network D, denoted as $C_1, \cdots, C_{m-n+1}$. For a given RPPTDT-tour $f \in \mathscr{F}^*$, we can obtain a new feasible solution by adding one of the linearly independent circuits to the incidence vector y_k of f. Then we can obtain at least $m - n + 1$ feasible solutions, the set of which is denoted as $\mathscr{F}_k$. Considering that the vector y_k might be linear combination of the $m - n + 1$ circuit vectors, the number of solutions in $\mathscr{F}_k$ which are affinely independent with $\mathscr{F}^*$ is at least $m - n$. Therefore, for all $k = 1, \cdots, K+1$, we can exhibit the other $(K+1)(m-n)$ affinely independent solutions in $\mathscr{F}$, denoted as $\mathscr{F}'$.

Finally, we will construct the last one affinely independent solution in $\mathscr{F}$ as follow. It is easy to see that there must exist two arcs (i, j) and (u, v) which are not adjacent, such that the number of paths between the above two arcs is at least 2, because $m - n \geq 1$. Let x' be an arc array with arcs (i, j) and (u, v) as its $k-1$th and kth arc respectively. Then we can exhibit two linearly independent solutions f_1'' and f_2'' by connecting arcs (i, j) and (u, v) in x' with the two paths respectively. Considering that one of the two paths might be associated with the incidence vector y_k in some RPPTDT-tour $f \in \mathscr{F}^*$, without loss of generality, let f_1'' be the solution with y^k appearing in the $\mathscr{F}^*$. Then there is at least one more affinely dependent solution f_2''. It is easy to show that the incidence matrix $(\mathscr{F}^*, \mathscr{F}', \{f_2''\})$ is non-singular. Thus, there are at least $K^2 + (m - n - 2)(K + 1) + 5$ affinely independent solutions in $\mathscr{F}$. Thus $Dim(P_{APAS}(D)) \geq K^2 + (m - n - 2)(K + 1) + 4$. □

Appendix B: Facet Defining Inequalities for the APAS Polytope

The proof of Theorem 2 is given below.

Proof. we only need to find $K^2 + (m - n - 2)(K + 1) + 4$ affinely independent points in $\mathscr{F}$ satisfying the constraint (22) as an equality. For the given j and k, one can find an arc starting at vertex v_j in the connected network D, denoted by (i, j), and let $\mathscr{F}_{ij}^k$ be the set of RPPTDT-tours whose kth serviced arc is (i, j). These affinely independent points can be found in $\mathscr{F}_{ij}^k$ as follows.

Firstly, according to Theorem 5, there are $(K - 2)^2 + 1$ affinely independent RPPTDT-tours in $\mathscr{F}_{ij}^k$, the set of which is denoted as $\mathscr{F}_1$. Without loss of generality, the path connecting each $k-1$th and kth $(k = 1, \cdots, K)$ serviced arcs of the RPPTDT-tours in $\mathscr{F}_1$ is a simple path (if the two arc are adjacent, then the path is void). It means that there is no isolated circuit in each $f \in \mathscr{F}_1$. That is, $\mathscr{F}_1$ is the first set of affinelly independent RPPTDT-tours satisfying (22) as an equality in $\mathscr{F}$.

Secondly, according to the circuit structure of digraphs [10], there are $m-n+1$ linearly independent circuits in D. For each $k' = 1, \cdots, K + 1$, we can construct $m - n + 1$ affinely independent solutions by adding these linearly independent circuits to $y_{k'}$ of a RPPTDT-tour $f \in \mathscr{F}_1$. For the case $k' \neq k$, the circuit added to the k'th path does not change the incidence vector y_k, which satisfies (22) . Thus the number of affinely independent solutions constructed by adding linearly independent circuit to the k'th path is at least $m - n + 1$ $(k' \neq k)$. On the other hand, for the case $k' = k$, the (22) will be satisfied as an equality if the added circuit is not isolated or does not contain v_l, otherwise it will be violated or be satisfied as an inequality. As the out-degree or the in-degree of v_l is at most 3, there are at most 3 isolated circuits in D passing though v_l. That is, we can construct at least $m-n-2$ affinely independent solutions by adding linearly independent circuit to the kth path. Considering that the vector $y_{k'}$ might be linear combination of the $m - n + 1$ circuit vectors $(k' = 1, \cdots, K + 1)$, at least $(m - n)K + m - n - 3$ affinely independent solutions can be constructed by adding linearly independent circuit for all $k = 1, \cdots, K + 1$, the set of which is denoted as $\mathscr{F}_2$.

Thirdly, for each two arcs (s,t) and (u,v) which are not adjacent, there must exist at least three paths connecting v_t and v_u, denoted as p^1_{tu}, p^2_{tu} and p^3_{tu}, since the assumption that the number of outgoing or incoming arcs of v_t exceeds 2. Let x' be the arc array containing arcs (s,t) and (u,v) as its $k'-1$th and k'th arc respectively. Then we can exhibit three linearly independent solutions, denoted as f'_1, f'_2 and f'_3, by connecting arcs (s,t) and (u,v) in x' through the paths p^1_{tu}, p^2_{tu} and p^3_{tu} respectively. Considering that one of the three paths might be associated with the incidence vector $y_{k'}$ of some RPPTDT-tour in the above $\mathscr{F}_1$, without loss of generality, let p^1_{ut} be the path which has been associated with the incidence vector $y_{k'}$ of some solution in $\mathscr{F}_1$. Then there is at least two more affinely independent solutions f'_2 and f'_3. For all $k' = 1, \cdots, K+1$, we can exhibit $2K$ affinely independent solutions satisfying (22) as an equality, which is denoted as $\mathscr{F}_3$.

It is easy to show that the incidence matrix of $(\mathscr{F}_1, \mathscr{F}_2, \mathscr{F}_3)$ is non singular, hence, the number of affinely independent solutions in $\mathscr{F}$ is at least $K^2 + (m - n - 2)(K+1) + 4$. $\square$