Citation for published version:
Ucar, E, Birbil, SI & Muter, I 2017, 'Managing disruptions in the multi-depot vehicle scheduling problem', Transportation Research Part B: Methodological, vol. 105, pp. 249–269.
https://doi.org/10.1016/j.trb.2017.09.002

DOI:
10.1016/j.trb.2017.09.002

Publication date:
2017

Document Version
Peer reviewed version

Link to publication

Publisher Rights
CC BY-NC-ND

Managing Disruptions in the Multi-Depot Vehicle Scheduling Problem

Ezgi Uçar, Ş. İlker Birbil

Sabancı University, Industrial Engineering Program, Orhanlı-Tuzla, 34956 Istanbul, Turkey

İbrahim Muter

School of Management, University of Bath, Claverton Down, Bath BA2 7AY, UK[*]

Abstract

We consider two types of disruptions arising in the multi-depot vehicle scheduling; the delays and the extra trips. These disruptions may or may not occur during operations, and hence they need to be indirectly incorporated into the planned schedule by anticipating their likely occurence times. We present a unique recovery method to handle these potential disruptions. Our method is based on partially swapping two planned routes in such a way that the effect on the planned schedule is minimal, if these disruptions are actually realized. The mathematical programming model for the multi-depot vehicle scheduling problem, which incorporates these robustness considerations, possesses a special structure. This special structure causes the conventional column generation method fall short as the resulting problem grows also row-wise when columns are generated. We design an *exact* simultaneous column-and-row generation algorithm to find a valid lower-bound. The novel aspect of this algorithm is the pricing subproblem, which generates pairs of routes that form recovery solutions. Compromising on exactness, we modify this algorithm in order to enable it to solve practical-sized instances efficiently. This heuristic algorithm is shown to provide very tight bounds on the randomly generated instances in a short computation time.

Keywords: multi-depot vehicle scheduling; robust planning; column-and-row generation

1. Introduction

Urban and intercity bus services are the pillars of the public transportation systems. The planning process for these systems involves constructing timetables according to the forecasted passenger demand and scheduling of the buses/crews. One of the components, namely the vehicle (bus) scheduling problem (VSP), concerns the assignment of a fleet of vehicles to a set of trips that have predetermined departure and arrival locations as well as fixed start and end times. The VSP is an important problem of public transportation and logistics, since the fuel cost accounts for one of the largest amount among the operational costs.

[*] Corresponding author.
E-mail Addresses: i.muter@bath.ac.uk (İbrahim Muter)

The VSP is studied extensively in the literature. Two general lines of research on this problem are the single depot vehicle scheduling problem (SDVSP) and the multi-depot vehicle scheduling problem (MDVSP). The SDVSP is polynomially solvable, and the models and algorithms for this problem are reviewed by Freling et al. (2001). However, the MDVSP, which is the problem we particularly consider in this paper, has been proven to be NP-Complete by Bertossi et al. (1987). The MDVSP involves assigning the timetabled trips feasibly to the available vehicles located at a set of depots. Here, a sequence of feasible trips corresponds to a *route* of a vehicle.

The routes that are found by the solution of MDVSP are subject to disruptions at the time of operation, such as; vehicle breakdowns, traffic congestions or adverse weather conditions. The consequences of such disruptions are cancellations and delays, which require changes in the planned schedule in order to satisfy the requests of the customers. The recovery procedures that are developed at the time of disruption may not lead to a feasible schedule, and even they do, they may inflict substantially large extra costs on the planned schedule. On the other hand, tackling disruptions at the planning stage is known as robust planning approach, which is based on anticipating disruption scenarios before they are actually realized. This approach readily provides a feasible recovery solution with the lowest possible increase in the planned cost, if the disruption scenario taken into account at the planning stage is realized at the time of operation. Robust planning has been considered in several studies for handling crew scheduling and vehicle scheduling (Huisman et al. 2004, Shebalov and Klabjan 2006).

In this paper, we consider two types of disruptions and present a robust planning approach that provides a unified way to recover them. The first type of disruptions consists of delays possibly caused by adverse conditions in weather and traffic. If long enough, a delay in the planned arrival time of a vehicle may, in turn, affect also the departure time of the successive trips assigned to the same vehicle. The second type of disruption is caused by the excess demand for a particular origin-destination pair that mandates the planners to insert an extra trip into the schedule at the operation phase (Tekiner et al. 2009, Muter et al. 2013b). Both types of disruptions are quite common in countries with a large bus transportation network like Turkey. A trivial recovery option for both types of disruptions (delay and extra trips) is to assign an idle vehicle, if exists, to cover the trips succeeding the delayed trip or the extra trip itself. This type of recovery solution is conceived when the disruption reveals itself and may result in additional costs associated with the relocation of the idle vehicle as well as other potential issues in crew scheduling. Such a recovery procedure is, in general, very hard and costly to handle. Hence, the only way to recover the delay or extra trip lies in the rescheduling of the vehicles (see Section 2 for studies handling disruptions by rescheduling).

Incorporating robustness into the planned schedule is possible, only if the sources of these two types of disruptions can be anticipated. Given a set of possibly delayed and extra trips, our robust planning approach provides a schedule that can be modified in such a way that exactly two routes deviate from the original plan in the recovery solution. The cost of the robust solution is generally higher than that corresponding to the planned schedule. However, if one of the disruptions is realized at the time of operation, then the cost savings achieved by the proposed robust schedule compensate for the loss due to execution of the robust plan.

The robust model that we tackle in this paper is the counterpart of that proposed in Muter et al. (2013b) for the robust airline crew pairing problem, in which the extra flights cause disturbances to the planned schedule. The novelty of this model lies in a set of constraints linking a pair of variables corresponding to two routes that constitute a recovery solution to an auxiliary variable. The number of linking constraints associated with recovery solutions is not only too large to consider explicitly in the model but also their enumeration requires the presence of the entire set of routes. The enumeration of the complete set of routes is generally not viable due to its large cardinality, and column generation is the prevalent methodology to attack such large-scale problems. During the course of column generation, as new routes are generated, those that pair up to form a recovery solution trigger the generation of new linking constraints. This raises two major issues in the application of the column generation approach: (i) The dual variable values of the new linking constraints, which appear as new recovery solutions are created during column generation, are not known. This causes the miscalculation of the reduced costs. (ii) Designing a pricing subproblem that generates pairs of routes forming recovery solutions for an extra trip is a challenging problem. To circumvent these difficulties in the solution of the robust airline crew pairing problem, Muter et al. (2013b) solve the linear programming (LP) relaxation of this problem only by a heuristic two-phase iterative column-and-row generation strategy. Thus, they are unable to provide a valid lower bound for the overall problem. Later, Muter et al. (2013a) focus on a general class of large-scale LP problems, referred to as problems with column-dependent-rows, and propose an exact solution method. This exact method, in theory, can be used to solve the robust airline crew pairing problem. In this paper, we not only employ the results proposed in Muter et al. (2013a) but also extend those results with a stronger termination condition.

The contributions of this paper are as follows:

- ◇ We present a unified methodology for the robust MDVSP when the sources of disruptions are extra trips and delays that can be anticipated at the planning phase.
- ◇ We give an *exact* simultaneous column-and-row generation algorithm to find a lower-bound for the robust MDVSP in which the pricing subproblem generates simultaneously pairs of routes forming recovery solutions.
- ◇ We design explicit search algorithms on graphs for the generation of route pairs used in pricing subproblems.
- ◇ We prove a stronger termination condition than the one given in Muter et al. (2013a) for the LP relaxation of the quadratic set covering problem. This is a new result in column-and-row generation.
- ◇ The termination condition of the proposed exact column-and-row generation algorithm may be computationally demanding to reach. We relax this condition and present a heuristic algorithm. We show that the proposed heuristic approach provides very tight bounds on the considered randomly generated instances.
- ◇ We conduct an extensive computational study to demonstrate the impact of the disruptions and to evaluate the performances of the proposed approaches.

2. Related Literature

In this section, we present the related studies in the literature on the MDVSP and disruption management. We refer the interested reader to Visentini et al. (2014) for an extensive review on various recovery methods. Although most of the solution methodologies are heuristics and handle the disruptions at the operational level, there exist few studies proposing exact approaches that incorporate robustness in their model in the planning stage.

One of the earliest studies on vehicle scheduling is by Bodin et al. (1978). Dell'Amico et al. (1993) solve the MDVSP with the objective of minimizing the number of vehicles and the overall operational cost. They use the structural properties of the problem to design a new polynomial time heuristic algorithm which always guarantees the use of the minimum number of vehicles. Haghani and Banihashemi (2002) develop new formulations for the MDVSP and its extension with route time constraints. To solve the latter problem, they propose two heuristic approaches and an exact method that is based on constraint generation. For the same problem, Haghani et al. (2003) propose two single depot vehicle scheduling with route time constraints model and present a comparative analysis of three vehicle scheduling models. Petersen et al. (2012) propose a planning approach to obtain a favorable trade-off between the two contrasting objectives, passenger service and operating cost, by modifying the timetable. The planning approach is referred to as the simultaneous vehicle scheduling and passenger service problem, and they solve this problem by large neighborhood search. The MDVSP with multiple vehicle types, where vehicle types are linked to each trip according to the trip's characteristics is studied by Ceder (2011). The author presents a heuristic algorithm based on the deficit function theory to solve this problem. Hassold and Ceder (2014) propose four exact methods and show the benefit of vehicle types substitutions on a real-life case study. Shen et al. (2016) consider randomness of the trip times and incorporate them into a probabilistic model of the vehicle scheduling problem with the objectives of minimizing the total cost and maximizing the on-time performance.

Besides the heuristic approaches, several exact methods are also applied to solve the MDVSP. Carpaneto et al. (1989) propose an exact branch-and-bound algorithm based on the computation of lower bounds by an additive scheme. Ribeiro and Soumis (1994) formulate the MDVSP as a set partitioning problem with side constraints and propose a column generation algorithm. They also show that the LP relaxation of the MDVSP generates a better bound than the additive bound proposed by Carpaneto et al. (1989). Fischetti et al. (2001) develop a branch-and-cut algorithm to solve the MDVRP in which limits are imposed on both the total time between the start and the end of any duty, and the total duty time. Löbel (1998) solves the LP relaxation of the multi-commodity flow formulation of the MDVSP by column generation. He devises two different Lagrangian relaxations of this formulation that constitute the pricing subproblem. To test the algorithm, a real-world data involving the cities of Germany is used. Hadjar et al. (2006) propose a branch-and-bound algorithm to solve the MDVSP. At each node in the tree, the LP relaxation is solved by column generation, and a variable fixing procedure is applied to reduce the number of variables. Moreover, to improve the LP bound, they introduce a class of valid inequalities.

In the literature, the solution methods to manage disruptions can be categorized under two groups. The first group is the online approaches that generate the schedules of vehicles on the fly during the execution and update the schedules at the operational level. The second group is the offline approaches that consider the possible disruptions at the planning stage. This class of solution procedures creates schedules that are likely to remain stable when disruptions occur. The studies in the vehicle scheduling literature presented below generally present methods to recover from disruptions in real-time through vehicle rescheduling, and hence, they fall in the category of online methods. The proposed approach in this study can be classified as an offline approach.

Huisman et al. (2004) develop a solution methodology for the dynamic vehicle scheduling problem, which consists of solving a sequence of optimization problems, where different scenarios for future travel times are taken into account. In the study of Kramkowski et al. (2009), the possible delays that can occur at the operational level are handled by inserting buffer times between trips. The critical point of this approach is to determine the right place to insert the buffer times. To solve the resulting robust vehicle scheduling problem, they apply a path based flow decomposition method on a network.

The disruptions in the SDVSP are handled by rescheduling the vehicles, and the resulting problem solved at the operational level is referred to as the vehicle rescheduling problem. Li et al. (2007) consider the disruptions due to a disabled vehicle and develop several fast algorithms to solve the mathematical model. For the same problem, Li et al. (2009) develop an arc-based formulation and present a Lagrangian heuristic. Their aim is to minimize the general arc costs, such as travel and idle times, as well as the trip cancellation costs and penalty costs incurred due to rescheduling. Sato et al. (2009) develop a recovery methodology to manage disruptions that are caused by delays and that can be applied both on vehicle and crew scheduling. They propose a network flow based formulation and a heuristic solution methodology for the rescheduling problem. The swap operation to recover disruptions developed in Sato et al. (2009) resembles the route swapping employed in this paper. However, the vehicle rescheduling problem is a dynamic problem which is solved when the disruption occurs at the time of operation.

The way we model robustness to handle disruptions in this paper is inspired by the works of Shebalov and Klabjan (2006) and Tekiner et al. (2009). The authors of the latter paper focus on a specific disruption caused by extra flights in airline crew pairing, where the pairings play a similar role as the routes in this work. To handle these disruptions at the planning stage, Tekiner et al. (2009) come up with two recovery solutions: type A and type B. Type A solution is similar to the recovery solutions that we have mentioned in the preceding paragraphs. Type B solution, on the other hand, is based on modification of a single pairing to cover an extra flight. This approach is based on enumeration of all possible pairings. Therefore, only small instances can be solved to optimality by standard commercial solvers. Muter et al. (2013b) propose a heuristic two-phase iterative column-and-row generation algorithm to attack large instances of the same problem. In the first phase, the number of constraints is fixed and then column generation is applied in a conventional manner. In the second phase, additional type A recovery solutions are identified based on the pairings generated during the last call to the column generation. In our present study, we utilize only type A solutions –referred to as recovery solutions in this paper– for recovery of disruptions. We do not make use of type B solutions as they introduce significantly long buffer times between trips in the planned schedule. Such long buffer times are unacceptable for real world planning.

3. Multi Depot Vehicle Scheduling Problem with Disruptions

In this section, we elaborate on the recovery solutions for the two types of disruptions defined earlier, namely the delayed and extra trips. We demonstrate that although these two disruption types are seemingly different, we may apply similar changes to the planned (robust) schedule to recover them. These changes are based on swapping two routes in such a way that if any one of these disruptions occurs, then the recovery is possible with the existing routes. Finally, the mathematical model for the robust MDVSP is presented.

Figure 1 illustrates our notation. A set of trips is indexed by $t \in T$ with a given start time (s_t) and end time (e_t). There exists a set of depots indexed by $d \in D$, each of which hosts C^d vehicles. For a vehicle to operate trip $t_j \in T$ after trip $t_i \in T$, it needs to be relocated from the arrival location of t_i to the departure location of t_j, and the duration of the relocation, also known as *deadheading*, of the vehicle between t_i and t_j is denoted by $\delta_{t_i t_j}$. Trip t_j can be feasibly operated after trip t_i if the relocation time of the vehicle from the arrival location of t_i to the departure location of t_j satisfies $e_{t_i} + \delta_{t_i t_j} \leq s_{t_j}$. A pair of trips that satisfy this condition is called *compatible*. There exists an operational cost of operating these trips one after the other, denoted by $c_{t_i t_j}$. Such relocation times and costs are also applicable to the depot-trip and trip-depot pairs without any condition on compatibility of these pairs except that a vehicle must return to its original depot after completing its last trip.

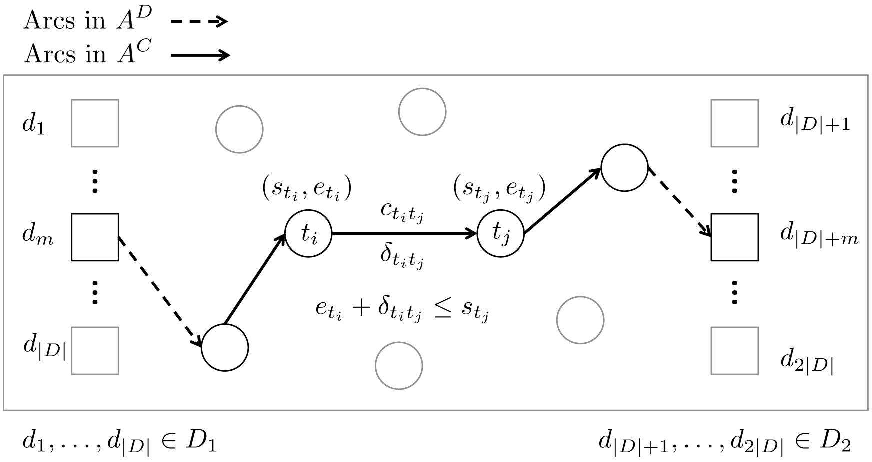


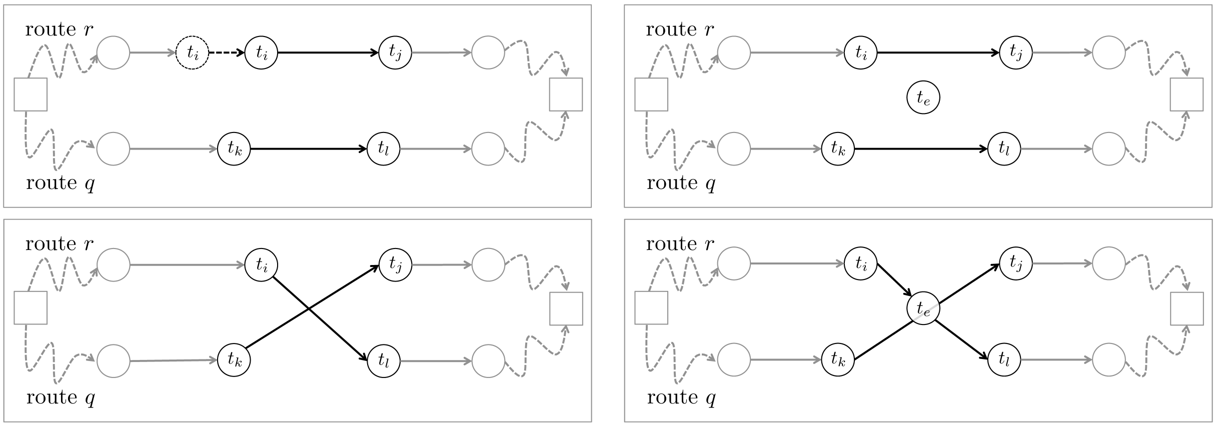
Figure 1: The notation used in the paper.

The MDVSP can be defined on a graph $G = (V, A)$, where $V = T \cup D_1 \cup D_2$ is the set of nodes which is itself composed of a set of trips indexed by $t \in T$, and two sets of nodes D_1 and D_2 both of which are the copies of D. While D_1 represents the source depot nodes denoted by $d_m, m = 1, ..., |D|$, D_2 contains the sink depot nodes denoted by $d_n, n = |D| + 1, ..., 2|D|$. Hence, there is a total of $|T| + 2|D|$ nodes. The arc set A is composed of two sets, $A = A^C \cup A^D$ where A^C connects compatible pairs of trips and A^D includes the arcs of the form (d_m, t_i) for $m = 1, ..., |D|$ and (t_i, d_n) for $t_i \in T$ and $n = |D| + 1, ..., 2|D|$. Hence, there is an arc from each source depot node to each trip node and from each trip node to each sink depot node. For A^C, $c_{t_i t_j}$ denotes the arc cost between trips t_i and t_j. For A^D, $c_{d_m t_i}$ and $c_{t_i d_n}$ denote the costs on arcs (d_m, t_i) and (t_i, d_n), respectively.

A feasible vehicle *route* starts from a depot, realizes a set of trips, and ends at the same depot. It can be represented as a vector $(d_m, t_1, t_2..., d_n)$, where $n = m + |D|$. The cost of a route is equal to the total costs of the arcs it traverses. The MDVSP is to determine a set of feasible vehicle routes that cover each trip once and has the smallest total cost. Letting R denote the set of all routes in G, the MDVSP can be stated as a set partitioning problem with side constraints (Hadjar et al. (2006)):

$$\begin{aligned}
\text{minimize} \quad & \sum_{r \in R} c_r y_r, && && (1)\\
\text{subject to} \quad & \sum_{r \in R} a_{tr} y_r = 1, && t \in T, && (2)\\
& \sum_{r \in R} b_{rd} y_r \leq C^d, && d \in D, && (3)\\
& y_r \in \{0, 1\}, && r \in R, && (4)
\end{aligned}$$

where c_r is the cost of route r, and $a_{tr} = 1$, if trip t is covered by route r and 0, otherwise. Decision variable y_r equals 1, if route r is selected and 0, otherwise. Binary parameter b_{rd} is 1, if route r starts and ends at depot d and 0, otherwise. The parameter C^d is the number of identical vehicles residing at depot d. The objective function minimizes the total cost of the selected routes. While constraints (2) impose that each trip should be covered exactly once, constraints (3) limit the number of vehicles leaving from each depot by the fleet size located at that depot.



(a) Delayed Trip

(b) Extra Trip

Figure 2: Swapping solutions for disruptions

At this point, we emphasize our assumption that the information pertinent to delayed trips and extra trips can be anticipated at the planning phase. Both the expected length of a delayed trip and the departure and arrival locations of an extra trip as well as its time-window – the earliest departure and the latest arrival time – are generally determined by using historical data and past experience. Extra trips are usually requested around the same time to popular destinations. A typical example is a summer Friday evening, when the walk-in customers are requesting tickets to a vacation resort. The swapping operation for delays and extra trips are illustrated in Figures 2(a) and 2(b), respectively. In the solution of (1)-(4), one of the vehicles covers trip t_i and continues with trip t_j, and another vehicle covers t_k and t_l in this order. In Figures 2(a), t_i is delayed, causing the connection between t_i and t_j to become incompatible. This disruption can be recovered by swapping the duties of the vehicles covering the trip pairs $t_i - t_j$ and $t_k - t_l$, assuming that the relocation time between these trip pairs allows a feasible connection. In Figure 2(b), emerging extra trip t_e is covered by the vehicle operating t_i, which then continues with t_l. The other vehicle covers t_k and continues with t_j. Again, such a swap is feasible, only if the relocation time between the newly connected trips are compatible. As illustrated in these figures, both delayed trips and extra trips can be recovered in a similar way by swapping the route assigned to the delayed or the extra trip with another route. In referring to the route pairs that form a recovery solution, we use the ordered-pair notation (r, q), where route $r \in R$ is the one that covers the delayed or extra trip, and latter route $q \in R$ recovers the rest of the trips of r. In this case, we refer to r and q as the primary and secondary routes of recovery solution (r, q), respectively.

By taking into consideration the recovery solutions for the disruptions, the robust MDVSP can be modeled as follows:

$$
\begin{aligned}
(\textbf{RMDVSP}) \qquad & \text{minimize} \quad && \sum_{r \in R} c_r y_r, && && (5)\\
& \text{subject to} \quad && \sum_{r \in R} a_{tr} y_r = 1, && t \in T, && (6)\\
& && \sum_{r \in R} b_{rd} y_r \leq C^d, && d \in D, && (7)\\
& && \sum_{(r,q) \in P(k)} x^k_{(r,q)} \geq 1, && k \in K, && (8)\\
& && y_r \geq x^k_{(r,q)}, && (r,q) \in P(k), k \in K, && (9)\\
& && y_q \geq x^k_{(r,q)}, && (r,q) \in P(k), k \in K, && (10)\\
& && y_r + y_q \leq 1 + x^k_{(r,q)}, && (r,q) \in P(k), k \in K, && (11)\\
& && y_r \in \{0,1\}, && r \in R, && (12)\\
& && x^k_{(r,q)} \in \{0,1\}, && (r,q) \in P(k), k \in K, && (13)
\end{aligned}
$$

where K denotes the set of all possible disruptions (delayed trips or extra trips). The index set $P(k)$ denotes the set of route pairs that form recovery solutions for disruption k. The auxiliary binary variable $x^k_{(r,q)}$ is set to 1, if the route pair $(r, q) \in P(k)$ and both routes r and q are selected, and to 0, otherwise. The set of constraints (9)-(11), which is referred to as the set of linking constraints, imposes that $x^k_{(r,q)}$ takes the value 1 if and only if $y_r = y_q = 1$. Through constraints (8), the selected routes of the above model provide at least one recovery solution for each $k \in K$.

4. Proposed Solution Methodology

Even for moderate size instances of (**RMDVSP**), the number of routes in R can be prohibitively large. To solve such large-scale models, column generation method, pioneered by Dantzig and Wolfe (1960) and Gilmore and Gomory (1961), is commonly employed. This method initializes a problem with a small subset of columns, referred to as the restricted master problem (RMP), and generates promising columns during the course of the algorithm. To that end, the dual information retrieved by solving the RMP is utilized in the solution of the pricing subproblem (PSP) which determines whether to add a new column or terminate the algorithm at the LP optimal solution (see Desaulniers et al. (2005) for a comprehensive survey on column generation). In the application of column generation to the conventional MDVSP given (1)-(4), the PSP is a shortest path problem that is solved for every source-sink pair each of which represent a depot. In the current study, this PSP constitutes only one element of our proposed methodology.

Notice that the number of linking constraints depends on the set of recovery solutions $P(k)$, $k \in K$. Hence, when the LP relaxation of (**RMDVSP**) is solved by column generation, a newly formed recovery solution that is comprised of the route pair $(r, q) \in P(k)$ introduces variable $x^k_{(r,q)}$ and three linking constraints (9)-(11). This structure of (**RMDVSP**) qualifies it as a problem with column-dependent-rows, for which a generic simultaneous column-and-row generation algorithm is proposed by Muter et al. (2013a). In this section, we first outline the application of the simultaneous column-and-row generation algorithm to (**RMDVSP**). Then, in the rest of the section, we elaborate on the details of the PSPs by addressing the associated difficulties and pointing out our approaches to alleviate them.

4.1. Simultaneous Column-and-Row Generation

In forming the RMP of the LP relaxation of (**RMDVSP**), we replace R by its subset $\bar{R}$. Moreover, we add those $(r, q) \in P(k)$, $k \in K$, where both r and q exist in $\bar{R}$. That is, the linking constraints given in (9)-(11) and the associated $x-$variables are only partially present in the RMP. The subset of the linking constraints and $x-$variables that are induced by $\bar{R}$ are denoted by $\bar{P}(k)$. The resulting RMP, which also lacks some of the linking constraints as well as columns, is referred to as the short RMP (SRMP):

$$
\begin{aligned}
\text{minimize} \quad & \sum_{r \in \bar{R}} c_r y_r, && && (14)\\
\text{subject to} \quad & \sum_{r \in \bar{R}} a_{tr} y_r = 1, && t \in T, && (15)\\
& \sum_{r \in \bar{R}} b_{rd} y_r \leq C^d, && d \in D, && (16)\\
& \sum_{(r,q) \in \bar{P}(k)} x^k_{(r,q)} \geq 1, && k \in K, && (17)\\
& y_r - x^k_{(r,q)} \geq 0, && (r,q) \in \bar{P}(k), k \in K, && (18)\\
& y_q - x^k_{(r,q)} \geq 0, && (r,q) \in \bar{P}(k), k \in K, && (19)\\
& y_r + y_q - x^k_{(r,q)} \leq 1, && (r,q) \in \bar{P}(k), k \in K, && (20)
\end{aligned}
$$

$$y_r \geq 0, \qquad r \in \bar{R}, \tag{21}$$

$$x^k_{(r,q)} \geq 0, \qquad (r,q) \in \bar{P}(k), k \in K. \tag{22}$$

According to the model above, when route pairs $\{(r,q),(q,r) \in P(k)\backslash\bar{P}(k)\}$ are completed in the SRMP, say with the generation of route r, a new set of linking constraints (18)-(20) and $x^k_{(r,q)}$ or $x^k_{(q,r)}$ associated with these pairs are also introduced into the SRMP. Route r can form a recovery solution both as the primary route in the form (r,q) or as the secondary route in the form (q,r) in a recovery solution. Therefore, when pricing y_r, we need to take into consideration the dual variable values associated with the newly introduced linking constraints currently missing from the SRMP. This shall be handled in a PSP that both generates recovery solutions and correctly calculates the values of these unknown dual variables.

At this point, we discuss an important feature of the simultaneous column-and-row generation algorithm. When the solution of (**RMDVSP**) is obtained, the planners are given a set of recovery solutions for each disruption $k \in K$, namely pairs $(r,q) \in P(k)$ with $x^k_{(r,q)} = 1$. For a given disruption k, a selected route $r \in R$ may be part of more than one selected recovery solutions, say (r,q) and $(r,s) \in P(k)$, $s \in R$, and one of these pairs can be swapped to recover k, if that particular disruption is realized at the time of operation. On the other hand, suppose that r forms recovery solutions for more than one disruptions, say $(r,q) \in P(k_1)$ and $(s,r) \in P(k_2)$ for k_1, $k_2 \in K$, and these are the only selected recovery solutions for these disruptions. If both disruptions are realized at the time of operation, the planner can only recover either k_1 or k_2 but not both, since swapping a pair of routes to recover a disruption, say k_1, impairs the structure of r which can no longer recover k_2. This issue was referred to as *double counting* by Shebalov and Klabjan (2006), and was handled by adding a set of constraints to the robust model by Tekiner et al. (2009). In this paper, to alleviate this peculiarity, we generate the route pairs through simultaneous column-and-row generation for each $k \in K$ separately so that a route is not part of recovery solutions for more than one disruption. Even if the same route coincidentally takes part in recovery solutions for different disruptions, treating them as distinct columns for each disruption prevents double counting thanks to the partitioning constraints in (**RMDVSP**).

As a consequence of the above explanation on the structure of recovery solutions, for a given $k \in K$, the reduced costs of route r and $x-$variable associated with one of the recovery solutions that r induces, say $x^k_{(r,q)}$, are denoted by $\bar{c}_r$ and $\bar{d}_{(r,q),k}$, respectively. Formally

$$\bar{c}_r = c_r - \sum_{t \in T} a_{tr} u_t - \sum_{d \in D} b_{rd} v_d - \sum_{(r,q) \in P(k)} (\gamma^1_{(r,q),k} + \gamma^3_{(r,q),k}) + \sum_{(q,r) \in P(k)} (\gamma^2_{(q,r),k} + \gamma^3_{(q,r)}) \tag{23}$$

and

$$\bar{d}_{(r,q),k} = \gamma^1_{(r,q),k} + \gamma^2_{(r,q),k} + \gamma^3_{(r,q),k} - z_k, \tag{24}$$

where $u_t \in \mathbb{R}$, $v_d \in \mathbb{R}^-$, $z_k \in \mathbb{R}^+$, $\gamma^1_{(r,q),k}, \gamma^2_{(r,q),k} \in \mathbb{R}^+$ and $\gamma^3_{(r,q),k} \in \mathbb{R}^-$ are the dual variables corresponding to constraints (15)-(20), respectively. Note that newly generated variables y_r and $x^k_{(r,q)}$ do not reside in the existing linking constraints indexed by $\bar{P}(k)$, and hence, none of the dual variables associated with these constraints appear in (23) and (24). Each $(r,q) \in P(k)$ or $(q,r) \in P(k)$ triggers three constraints and one $x-$variable, and some $q \in R$ may not exist in the SRMP, i.e. $q \in R \backslash \bar{R}$. Therefore, there are two difficulties in designing a PSP for this problem: First, dual variables γ associated with the linking constraints currently missing from the SRMP are unknown. Second, the recovery solutions involving route pairs $\{(r,q),(q,r) \in P(k)\}$ must be generated simultaneously. The former issue has been tackled in Muter et al. (2013a) through the *thinking-ahead* approach for the closely related problem quadratic set covering, in which each pair of variables is associated with three linking constraints and an auxiliary $x-$variable. After giving the outline of the simultaneous column-and-row generation, we address these issues in a PSP which we call route-pair generating PSP that generates a set of recovery solutions. We also further the thinking-ahead approach in this paper to avoid degenerate iterations that stall the termination of the overall algorithm.

The simultaneous column-and-row generation algorithm to solve the LP relaxation of (**RMDVSP**) is illustrated in Figure 3, which starts with the construction of the SRMP with a small set of routes. After solving the SRMP and obtaining the optimal values of the dual variables, the first PSP, referred to as the individual route generating PSP, is called. This PSP is solved under the condition that no recovery solution is generated; that is, no unknown dual variable resides in (23). If at least one negative reduced cost route is found, then the route having the minimum reduced cost is added to the SRMP, and the algorithm continues with the same PSP. Otherwise, the column pool is searched for possible pairs since the set of routes generated via consecutive calls to the individual route generating PSP may pair up to create new recovery solutions. For each pair that can be swapped feasibly, the auxiliary $x-$variable and the associated linking constraints are added to SRMP. If at least one of the new linking constraints is violated, which is the case when the left-hand-side of (20) exceeds one, the SRMP is re-solved and the algorithm returns to individual route generating PSP. Otherwise, the route-pair generating PSP is called to generate pairs of routes forming recovery solutions. In this PSP, we check whether there is any column $r \in R$ having a negative reduced cost only after pairs of routes containing r are generated along with a set of linking constraints. If so, the columns and the rows are added to SRMP and the algorithm returns back to the first PSP. If the algorithm cannot find such pairs of routes for all disruptions, then the algorithm terminates with the LP optimal solution of (**RMDVSP**).

For the optimality of this column-and-row generation algorithm, we also need to ensure primal feasibility (in addition to the dual feasibility checked by the PSPs) as new rows are added iteratively to SRMP. At any iteration, constraints (18)-(20) for some $(r,q) \in P(k) \backslash \bar{P}(k)$, $k \in K$ are not violated, in fact they are redundant, unless both y_r and y_q are in the SRMP. As mentioned previously, some of the columns generated during the execution of the individual route generating PSP may form a recovery solution. The linking constraints associated with those recovery solutions are added after this PSP terminates. Hence, the primal feasibility may be impaired during the execution of the individual route generating PSP. However, since all the recovery solutions are extracted and the associated linking constraints are added before the route-pair generating PSP, the primal feasibility is guaranteed when the algorithm terminates.

4.2. Individual Route Generating Subproblem

Our first subproblem checks whether there is any route $r \in R$ with a negative reduced cost using only the optimal values of u_t, $t \in T$ and v_d, $d \in D$ obtained by solving the SRMP. Therefore, the recovery solutions coincidentally formed by this route and the dual variables of

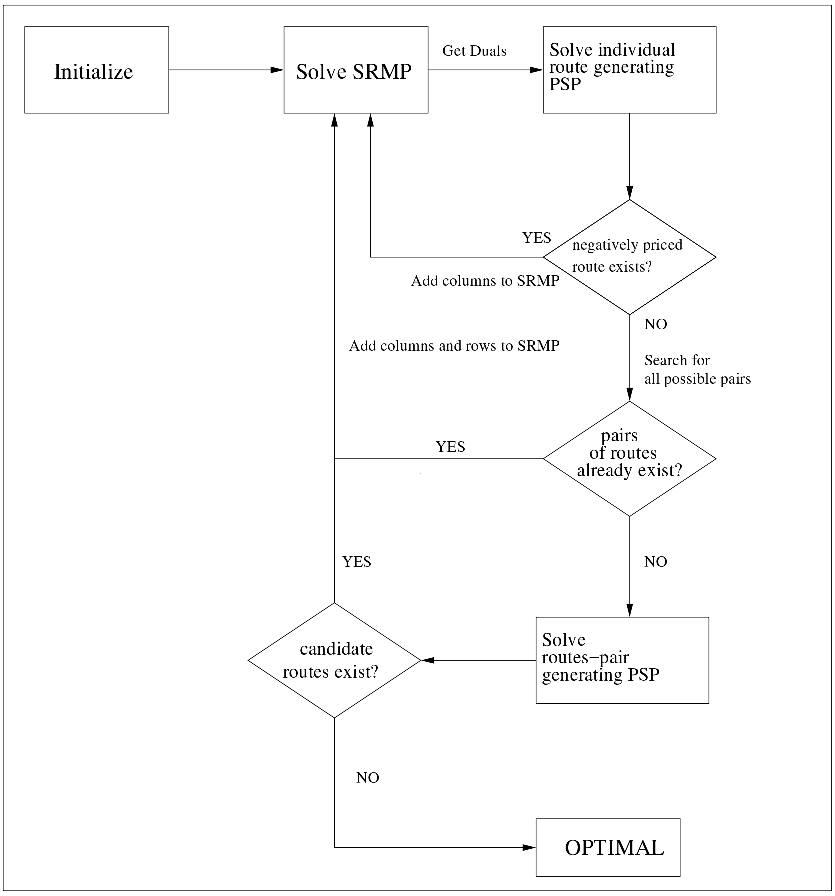

Initialize
Solve SRMP
Get Duals
Solve individual route generating PSP
YES
negatively priced route exists?
Add columns to SRMP
NO
Add columns and rows to SRMP
Search for all possible pairs
YES
pairs of routes already exist?
YES
NO
candidate routes exist?
Solve routes−pair generating PSP
NO
OPTIMAL

Figure 3: Flow of the proposed column-and-row generation algorithm

the linking constraints induced by these recovery solutions are first disregarded in this PSP. Formally, this PSP is given as

$$\zeta_y = \min_{r \in R} \left\{ c_r - \sum_{t \in T} a_{tr} u_t - \sum_{d \in D} b_{rd} v_d \right\}, \tag{25}$$

which is simply a shortest path problem for each $d \in D$. To that end, for each $d \in D$, we define $G^d = (V^d, A^d)$, where V^d is composed of T and two nodes associated with $d \in D$, say d_m and d_n, and A^d is composed of A^C and the set of arcs of the form (d_m, t_i) and (t_i, d_n) for $n = |D| + m$. Moreover, for all arcs in G^d, $\bar{c}_{t_i t_j} = c_{t_i t_j} - u_{t_i}$, $\bar{c}_{t_i d_n} = c_{t_i d_n} - u_{t_i}$ and $\bar{c}_{d_m t_i} = c_{d_m t_i}$ are the associated reduced cost parameters. If $\zeta_y < 0$ after solving the shortest path problems, the route with the smallest reduced cost is added to the SRMP. Otherwise, the algorithm continues as in Figure 3.

According to the flowchart given in Figure 3, the SRMP grows only column-wise until the individual route generating PSP cannot find any negatively priced columns. Some of the columns generated through solving the individual route generating PSP consecutively may incidentally form recovery solutions. Before moving to the route-pair generating PSP, we search for such recovery solutions that involve the routes generated by the individual route generating PSP. The identification procedure will be explained in Section 4.4 where the trip quadruples are used to forge the formation of candidate recovery solutions. In short, by checking whether the selected pair of routes from the SRMP possesses these quadruples, we will be able to detect the recovery solutions. Another characteristic of a pair of routes forming a recovery solution is that they must be disjoint. Otherwise, they can never be selected together in a feasible integer solution to recover a disruption due to the partitioning constraints of (**RMDVSP**). As mentioned previously, each route can be part of recovery solutions only for a single disruption. Hence, if route r generated during the individual route generating PSP constitutes a recovery solution for $k_1 \in K$ with $q \in \bar{R}$, which already forms a recovery solution for $k_2 \in K$, we add $y_{q'}$, a duplicate of y_q, to the model and consider this new variable only in the linking constraints associated with $(r, q') \in P(k_1)$. Next, we add the linking constraints and the $x-$variables associated with these recovery solutions to the SRMP. If at least one of the new linking constraints is violated by the current optimal solution of the SRMP, then we resolve it and return back to the first subproblem. Otherwise, the route-pair generating PSP is called.

4.3. Route-Pair Generating Subproblem

The objective of this subproblem is to identify new columns that price out favorably only after adding new linking constraints currently absent from the SRMP. The generation of new linking constraints is triggered by the generation of recovery solutions in $P(k)\backslash\bar{P}(k)$, which correspond to pairs of routes that can be swapped feasibly to recover disruption $k \in K$. Hence, to solve this subproblem, we need to design a methodology that simultaneously generates a set of routes (some of which can already exist in the SRMP) that forms recovery solutions and generates new linking constraints and $x-$variables. Recall, the reduced costs of $y-$ and $x-$variables in (23) and (24), respectively. As alluded to previously, the values of dual variables γ associated with the new linking constraints induced by the generated recovery solutions are unknown when this subproblem is called, and must be estimated to correctly calculate the reduced costs of the variables. In addressing the aforementioned difficulties of this subproblem, which are related to the generation of recovery solutions and the anticipation of the values of the unknown dual variables, we divide this section into two: First, we tackle the latter difficulty by demonstrating the application of the thinking-ahead approach and then present a stronger termination condition that improves the performance of the simultaneous column-and-row generation algorithm. Second, for the generation of recovery solutions consisting of route pairs, we present a model based on implicit enumeration of the recovery solutions. Admittedly, enumeration is not practical for large-scale instances. To circumvent this difficulty, we propose a heuristic algorithm that generates a single pair of routes with the minimum total reduced cost.

Since a route can form recovery solutions only for a single disruption, the route-pair generating PSP is solved for each disruption. For a given disruption $k \in K$, we define set $\mathcal{P}_r^k$ consisting of all recovery solutions for k containing route r. The outcome of this PSP is $\mathcal{F}_r^k$ which is a family of route pairs for disruption k of the form $\{(r, q), (q, r) \in P(k)\}$ involving route $r \in R\backslash\bar{R}$. While $\mathcal{P}_r^k$ contains all recovery solutions for k, $\mathcal{F}_r \subset \mathcal{P}_r$ may exclude some of the recovery solutions, which is justified by our termination condition given later in Lemma

4.1. The introduction of each $(r,q) \in \mathcal{F}_r$ or $(q,r) \in \mathcal{F}_r$ adds three constraints, an auxiliary $x-$variable and y_q, $q \in \Sigma_r \backslash \bar{R}$, where Σ_r is the index set of $y-$variables in the route pairs, to the SRMP.

For given $k \in K$ and $\mathcal{F}_r^k$, we restate the reduced cost of y_r as follows

$$\bar{c}_r = c_r - \sum_{t \in T} a_{tr} u_t - \sum_{d \in D} b_r^d v_d - \sum_{(r,q) \in \mathcal{F}_r^k} (\gamma_{(r,q),k}^1 + \gamma_{(r,q),k}^3) - \sum_{(q,r) \in \mathcal{F}_r^k} (\gamma_{(q,r),k}^2 + \gamma_{(q,r),k}^3) \tag{26}$$

where $\gamma_{(r,q),k}$ and $\gamma_{(q,r),k}$ are the unknown dual variables associated with the linking constraints induced by $(r,q) \in \mathcal{F}_r$ and $(q,r) \in \mathcal{F}_r$, respectively. Since new $x-$variables and y_q, $q \in \Sigma_r \backslash \{r\}$ reside in these linking constraints, the reduced costs of these variables also include the unknown dual variables. We point out that $\mathcal{F}_r^k$ and the unknown dual variables induced by $\mathcal{F}_r^k$ are determined to minimize the reduced cost of y_r. The rationale of the thinking-ahead approach can be summarized as follows: For recovery solutions $(r,q) \in \mathcal{F}_r^k$ and $(q,r) \in \mathcal{F}_r^k$, we conceive a new SRMP, referred to as the augmented SRMP, which is an extension of current SRMP with the new linking constraints and their slack and surplus variables; $x_{(r,q)}^k$ and $x_{(q,r)}^k$, respectively. Then, the optimal basis of the SRMP, denoted by $\mathbf{B}$, is expanded with new columns to construct the optimal basis for the augmented SRMP, denoted by $\mathbf{B}_r$. This is called *basis augmentation*, of which the most crucial point is that the values of the existing dual variables remain the same so that the reduced costs of the variables can be calculated correctly. As the term *thinking-ahead* implies, we can correctly anticipate the optimal values of the dual variables of the new linking constraints without actually forming and solving to optimality the augmented SRMP. In our analysis, we formally show that the basis augmentation prescribed by the thinking-ahead approach not only preserves $\mathbf{B}$ and the values of the existing dual variables but also ensures that the reduced costs of the variables in the SRMP do not change.

At first sight, it may seem that the construction of the augmented basis after the addition of the set of linking constraints can be easily handled using standard sensitivity analysis. This could be achieved by adding the slack/surplus variables provided that primal feasibility is not impaired. However, in our case, for a recovery solution, say $(r,q) \in \mathcal{F}_r^k$, the associated variable $x_{(r,q)}^k$, whose reduced cost is affected by the dual variables associated with these new constraints, is also considered in the augmented SRMP. The dual constraint associated with $x_{(r,q)}^k$, which is given by

$$\gamma_{(r,q),k}^1 + \gamma_{(r,q),k}^2 + \gamma_{(r,q),k}^3 - z_k \geq 0, \tag{27}$$

is violated, if $z_k > 0$ and the slack/surplus variables associated with the linking constraints are selected as basic, which renders $\gamma_{(r,q),k}^1 = \gamma_{(r,q),k}^2 = \gamma_{(r,q),k}^3 = 0$. When $x_{(r,q)}^k$ with a negative reduced cost enters the basis, the iteration is degenerate due to the associated linking constraints which force $x_{(r,q)}^k$ to be zero unless y_r and y_q are in the basis.

Taking into account the above considerations, we define the conditions that the dual variables associated with each $(r,q) \in \mathcal{F}_r^k$ –which are also valid for $(q,r) \in \mathcal{F}_r^k$– should satisfy:

1. The inequalities $\gamma_{(r,q),k}^1 \geq 0$, $\gamma_{(r,q),k}^2 \geq 0$ and $\gamma_{(r,q),k}^3 \leq 0$ must hold.

2. The inequality (27), which ensures the feasibility of the dual constraint associated with $x^k_{(r,q)}$, is satisfied.
3. To warm-start the augmented SRMP in the next iteration, three new variables among $x^k_{(r,q)}$ and slack/surplus variables associated with the linking constraints must be selected in the augmentation of $\mathbf{B}$. By complementary slackness, this condition amounts to selecting values of γ in such a way that three dual constraints among those given in 1 and 2 become tight.
4. The values of the new dual variables must not affect the reduced costs of y_q, $q \in \bar{R}$, especially the ones in the basis. Otherwise, the values of existing dual variables change, and the reduced cost calculation of r cannot be done correctly.

Note that no condition is imposed on y_q, $q \in \Sigma_r \backslash \bar{R}$ whose reduced costs are considered separately in part of the route-pair generating PSP. Hence, in pricing y_r, we do not let the new dual variables affect the reduced costs of these variables. The optimality of the augmented basis can be argued by the primal feasibility, which is satisfied since the set of linking constraints induced by $\mathcal{F}^k_r$ is redundant when pricing $r \in R \backslash \bar{R}$, the dual feasibility (by conditions 1, 2 and 4 which enforce that the reduced costs of the variables in the augmented SRMP be nonnegative), and the complementary slackness (by 3 which adds three basic variables to $\mathbf{B}$ for each $(r,q) \in \mathcal{F}^k_r$ and 4 which preserves the current basis $\mathbf{B}$). Condition 4 can be satisfied only if the dual values of the new linking constraints in which the existing $y-$variables reside are set to zero; i.e., $\gamma^2_{(r,q),k} = 0$ and $\gamma^3_{(r,q),k} = 0$, and for (q,r), pairs $\gamma^1_{(q,r),k} = 0$ and $\gamma^3_{(q,r),k} = 0$. By complementary slackness, these two equalities prescribe the surplus and slack variables as basic for (19) and (20), respectively. The third basic variable required for condition 3 must be determined for (18), in which the only $y-$variable is y_r, and either the surplus variable associated with this constraint or $x^k_{(r,q)}$ can be selected. The above conditions translate formally into the following definition of the route-pair generating PSP:

$$\zeta_{yx} = \min_{r \in (R \backslash \bar{R})} \left\{ c_r - \sum_{t \in T} a_{tr} u_t - \sum_{d \in D} b_{rd} v_d - \max_{k \in K, \mathcal{F}_k \subset \mathcal{P}_k} \left(\sum_{(r,q) \in \mathcal{F}^k_r} \alpha_{(r,q)} + \sum_{(q,r) \in \mathcal{F}_k} \alpha_{(q,r)} \right) \right\}, \text{ where} \tag{28}$$

$$\begin{aligned}
\alpha_{(r,q)} = \text{maximize} \quad & \gamma^1_{(r,q),k} + \gamma^3_{(r,q),k}, && \text{(29a)}\\
\text{subject to} \quad & -\gamma^1_{(r,q),k} - \gamma^2_{(r,q),k} - \gamma^3_{(r,q),k} \leq -z_k, && \text{(29b)}\\
& \gamma^2_{(r,q),k} = \gamma^3_{(r,q),k} = 0, && \text{(29c)}\\
& \gamma^1_{(r,q),k} \geq 0, && \text{(29d)}\\
& \text{At least one of (29b) or (29d) is tight,} && \text{(29e)}\\
\alpha_{(q,r)} = \text{maximize} \quad & \gamma^2_{(q,r),k} + \gamma^3_{(q,r),k}, && \text{(29f)}\\
\text{subject to} \quad & -\gamma^1_{(q,r),k} - \gamma^2_{(q,r),k} - \gamma^3_{(q,r),k} \leq -z_k, && \text{(29g)}\\
& \gamma^1_{(q,r),k} = \gamma^3_{(q,r),k} = 0, && \text{(29h)}
\end{aligned}$$

$$\gamma^2_{(q,r),k} \geq 0, \tag{29i}$$

$$\text{At least one of (29g) or (29i) is tight.} \tag{29j}$$

In the above two-level problem, the lower-level problems (29a)-(29e) and (29f)-(29j) determine the values of the dual variables associated with the new linking constraints induced by $(r, q) \in \mathcal{F}^k_r$ and $(q, r) \in \mathcal{F}^k_r$, respectively. The constraints imposed in these problems stem from the previously defined conditions entailing the optimality of the augmented basis $\mathbf{B}_r$. The upper-level problem (28) finds the route with the minimum reduced cost together with recovery solutions set $\mathcal{F}^k_r$. The solution of (29a)-(29e) and (29f)-(29j) can be easily obtained by setting $\gamma^1_{(r,q),k} = \gamma^2_{(q,r),k} = z_k$. When $z_k > 0$, $x^k_{(r,q)}$ and $x^k_{(q,r)}$ are selected as basic for the constraints (18) and (19), respectively. The surplus variables are added to form $\mathbf{B}_r$ when $z_k = 0$. After incorporating the solution of the lower-problems, the route-pair generating PSP becomes

$$\begin{aligned}\zeta_{yx} &= \min_{r \in R \setminus \bar{R}} \left\{ c_r - \sum_{t \in T} a_{tr} u_t - \sum_{d \in D} b_{rd} v_d - \max_{k \in K, \mathcal{F}^k_r \subset \mathcal{P}_k} \left\{ \sum_{(r,q) \in \mathcal{F}^k_r} z_k + \sum_{(q,r) \in \mathcal{F}^k_r} z_k \right\} \right\} \\ &= \min_{r \in R \setminus \bar{R}} \left\{ c_r - \sum_{t \in T} a_{tr} u_t - \sum_{d \in D} b_{rd} v_d - \max_{k \in K, \mathcal{F}^k_r \subset \mathcal{P}_k} \left\{ |\mathcal{F}^k_r| z_k \right\} \right\}. \end{aligned} \tag{30}$$

When $z_k = 0$, there is no need to solve this PSP since we have confirmed by individual route generating PSP that no route can have negative reduced cost without the dual values of the new linking constraints. The formal proof showing the optimality of $\mathbf{B}_r$ for the augmented SRMP together with the optimality of the simultaneous column-and-row generation algorithm can be found in Muter et al. (2013a) for general problems with column-dependent-rows. We show through Lemma 4.1 that augmenting $\mathbf{B}$ to $\mathbf{B}_r$ does not change the values of the existing dual variables, and the dual values of the newly generated constraints induced by $\mathcal{F}^k_r$ are as anticipated by the thinking-ahead approach applied through the constraints in (29). Using these dual values induced by $\mathbf{B}_r$ as calculated in the proof of Lemma 4.1, we calculate the reduced cost of y_r correctly as defined in (30). We give the proof of the next lemma in the appendix for completeness.

Lemma 4.1 *For the SRMP augmented with a set of linking constraints and $x-$variables associated with $\mathcal{F}^k_r$, $\mathbf{B}_r$ found by the thinking-ahead approach is optimal.*

Proof. See Appendix A.

□

It can be inferred from the definition of the route-pair generating PSP given in (30) that when $z_k > 0$ for some $k \in K$, the family $\mathcal{F}^k_r$ contains all possible route pairs $\{(r, q), (q, r) \in P(k)\}$ since each of them decreases $\bar{c}_r$ by z_k. Solving (30) boils down to enumerating all possible recovery solutions involving r, even the ones that may never enter the basis. In the following theorem, we demonstrate that after the variable y_r with negative reduced cost enters the basis, it only causes a degenerate iteration without changing the values of the existing dual variables, if $y-$variables which y_r pairs up with in the new recovery solutions do not enter the basis in the consecutive iterations. Hence, solving (30) using solely the thinking-ahead approach may retard the termination of the algorithm by adding many variables together without changing the solution. The theorem given below proposes a stronger termination condition than that found by the thinking-ahead approach. We redefine $\mathcal{F}_r^k$ so that it contains only those variables for which the reduced costs are negative after the dual value of the linking constraint is subtracted. That is, $\mathcal{F}_r^k = \{(r,q),(q,r) \in P(k)\backslash\bar{P}(k) \mid \bar{c}_q - z_k < 0\}$.

Theorem 4.1 *Given an optimal basis $\mathbf{B}_r$, a set of optimal dual values (u, v, z, γ), in which missing values in γ are obtained as prescribed by* (29b)-(29e) ((29g)-(29j))*, and reduced cost of variables $\bar{c}_r$, $r \in R$, the proposed algorithm terminates with an optimal solution for the LP relaxation of* (**RMDVSP**) *when*

$$\min_{r\in(R\backslash\bar{R})}\{\bar{c}_r + \textstyle\sum_{q\in S_r^k} \bar{c}_q - \mid \mathcal{F}_r^k \mid z_k\} \geq 0.$$

where $S_r^k = \{q \in R|(r,q),(q,r) \in \mathcal{F}_r^k\}$.

Proof. See Appendix B.

□

4.4. Generation of the Recovery Solutions

In this section, we present our graph search algorithm that is designed to generate recovery solutions. At this point, we formally describe the characteristics of the route pairs as follows:

Definition 4.1 *A pair of routes (r,q) is a recovery solution, that is $(r,q) \in P(k)$, $k \in K$, if it satisfies the following conditions:*

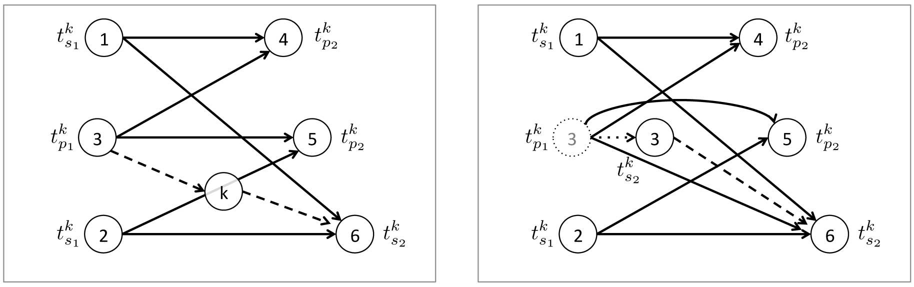
(i) r and q belong to the same depot to ensure that after swapping r and q, these routes still end at their base depot.

(ii) r and q are disjoint, that is, no trip can be visited by both of these routes.

(iii) r and q satisfy the swappability conditions given in Section 3.

In generating the recovery solutions, we first label nodes in G that forge swappability for the routes visiting them. Then, the routes that visit these labeled nodes are constructed, and an optimization problem is solved to identify $\mathcal{F}_r^k$ that results in a minimum reduced cost variable y_r. This approach boils down to implicit enumeration of a subset of routes that exists in recovery solution set and solving an integer program. Clearly, such an approach could be computationally very demanding especially for large problems. Therefore, we instead employ a heuristic method to simultaneously generate a single route pair with the smallest total reduced cost.

For some extra trip k, if it is possible to add an arc from trip $t \in T$ to k, $e_t + \delta_{tk} \leq s_k$, this trip and its successors are labeled as $t_{p_1}^k$ and $t_{p_2}^k$, respectively. This indicates that these trips can be a part of primary routes in possible recovery solutions. If the type of disruption is delay, the delayed trip itself and its successor are labeled as $t_{p_1}^k$ and $t_{p_2}^k$, respectively. If it is possible to add an arc from k, which can be either extra or delayed trip, to trip $t \in T$ $(e_k + \delta_{kt} \leq s_t)$, then this trip and its predecessors are labeled as $t_{s_2}^k$ and $t_{s_1}^k$, respectively. This shows that these trips can be a part of secondary routes in possible recovery solutions. Since



(a) Extra trip

(b) Delayed trip

Figure 4: Search algorithm for swapping solution of disruptions

there is a time-window on the operation of the extra trips, we consider the late start time and the early finish time of extra trips in the above calculations. If there is an arc from $t^k_{s_1}$ to $t^k_{p_2}$, then these four nodes, constituting a quadruple $(t^k_{p_1}, t^k_{p_2}, t^k_{s_1}, t^k_{s_2})$, can be connected to the source and sink depot nodes through partial paths to form recovery solutions $(r, q) \in P(k)$ for extra trip k. Arcs $(t^k_{p_1}, t^k_{p_2})$ and $(t^k_{s_1}, t^k_{s_2})$ are referred to as the primary and secondary parts of quadruple $(t^k_{p_1}, t^k_{p_2}, t^k_{s_1}, t^k_{s_2})$. If the disruption is realized at the time of operation, then $t^k_{p_1}$ covers the delayed trip or the extra trip k and connects to $t^k_{s_2}$ while $t^k_{s_1}$ continues with $t^k_{p_2}$. Moreover, Q^k denotes the set of all quadruples for $k \in K$. Therefore, the last condition in Definition 4.1 can be updated by stating that r and q, which form a recovery solution for $k \in K$, contain the primary and the secondary parts of a quadruple in Q^k.

In Figures 4(a) and 4(b), we demonstrate the labeling technique to identify the quadruples that are potentially part of the recovery solutions for disruptions caused by extra trip and delayed trip, respectively. In Figure 4(a), trips 3 and 6 are labeled as $t^k_{p_1}$ and $t^k_{s_2}$, respectively. Likewise, trips 4 and 5 are labeled as $t^k_{p_2}$, and trips 1 and 2 correspond to $t^k_{s_1}$. Then, we should check whether there is an arc from 1 to 4, 1 to 5, 2 to 4 and 2 to 5. If so, then this enables swapping possible route pairs for the recovery of k. The existing arcs $(2, 5)$ and $(1, 4)$ render that $\{(3, 5, 2, 6), (3, 4, 1, 6)\} \in Q^k$. That is, a route pair of the form (r, q), in which primary route r involves arcs $(3, 5)$ and $(3, 4)$, and secondary route q traverses $(2, 6)$ and $(1, 6)$, constitutes a candidate recovery solution for the extra trip k. In Figure 4(b), the labeling for a delayed trip is illustrated. Trip 3 is labeled as $t^k_{p_1}$ and connected to trips 4, 5 and 6. Delayed trip 3 can be connected to trip 6 with label $t^k_{s_2}$ whose predecessors 1 and 2 have the label $t^k_{s_1}$. As for the extra trip case, $(3, 5, 2, 6)$ and $(3, 4, 1, 6)$ are the quadruples.

The enumeration of Q^k, $k \in K$ is executed only once at the beginning of the overall algorithm. When the individual route generating PSP terminates, the routes generated during the consecutive calls to this PSP are examined for possible formation of recovery solutions before we move to the route-pair generating PSP. This is accomplished for each $k \in K$ by checking whether a given pair of routes covers the primary and secondary parts of any $o \in Q^k$.

In order to terminate the route-pair generating PSP with the condition stated in Theorem 4.1, an enumeration-based algorithm is needed to construct the set of route pairs conforming to the conditions on the recovery solutions given in Definition 4.1. Let R^o be the set of all routes that contain either the primary part $(t^k_{p_1}, t^k_{p_2})$ or the secondary part $(t^k_{s_1}, t^k_{s_2})$ of

$o = (t^k_{p_1}, t^k_{p_2}, t^k_{s_1}, t^k_{s_2}) \in Q^k$ and $R^{Q_k} = \bigcup_{o \in Q^k} R^o$. While some route in R^{Q_k} covers the primary part of one quadruple, it can contain the secondary part of another quadruple, which is possible for only extra trips since a delayed trip always corresponds to $t^k_{p_1}$. According to Theorem 4.1, we aim to find route $r \in R^{Q_k}$ that has negative reduced cost when added to the SRMP with recovery solutions $\{(r,q),(q,r) \in \mathcal{F}^k_r : \bar{c}_q - z_k < 0\}$. Hence, any route $q \in R$ that satisfies $\bar{c}_q - z_k \geq 0$ can be discarded from R^{Q_k}. The routes in R^{Q_k} are enumerated on G using a depth-first search algorithm. Whenever a partial path covers one of $(t^k_{p_1}, t^k_{p_2})$ or $(t^k_{s_1}, t^k_{s_2})$ for $(t^k_{p_1}, t^k_{p_2}, t^k_{s_1}, t^k_{s_2}) \in Q^k$ during the course of the algorithm, it is flagged as a prospective member of R^{Q_k}. On the other hand, an unflagged partial path reaching a node that is topologically later than the latest start times of $t^k_{p_1}$ or $t^k_{s_1}$ for all $(t^k_{p_1}, t^k_{p_2}, t^k_{s_1}, t^k_{s_2}) \in Q^k$ can be discarded since it cannot be a member of R^{Q_k}. The search algorithm can be enhanced through a bounding method that eliminates partial routes which can never satisfy $\bar{c}_q - z_k < 0$ at the final depot node when completed. To that end, we find the shortest path from the sink nodes to the source nodes associated with all depots $d \in D$ using only the reverse arcs in A (the reverse graph is explained later in this section). Let the reduced cost of the partial path in the enumeration tree from source node d_m to t and the value of the shortest path from corresponding sink node d_n to t on the reversed graph be denoted by $\bar{c}_{d_m,t}$ and $\hat{c}_{d_n,t}$, respectively. Thus, if $\bar{c}_{d_m,t} + \hat{c}_{d_n,t} - v_{d_m} - z_k \geq 0$, the partial path is discarded from the search algorithm.

Given the set of routes R^{Q_k}, the route-pair generating PSP is a subset selection problem that determines the minimum reduced cost route r together with $\mathcal{F}^k_r$. This problem can be formulated as an integer programming problem in which binary variable θ_q indicates that route $q \in R^{Q_k}$ takes part in one of the recovery solutions in $\mathcal{F}^k_r$ and $\lambda_r = 1$ if and only if r is the route whose reduced cost is to be minimized with incorporation of $\mathcal{F}^k_r$. The constraints of the model given below are based on the conditions that are defined in Definition 4.1. The mathematical model is as follows

$$\text{minimize} \quad \sum_{r \in R^{Q^k}} \bar{c}_r \theta_r - z_k \left(\sum_{r \in R^{Q^k}} \theta_r - 1 \right), \tag{31}$$

$$\text{subject to} \quad \sum_{r \in R^{Q^k}} \lambda_r = 1, \tag{32}$$

$$\lambda_r \leq \theta_r, \qquad r \in R^{Q^k} \tag{33}$$

$$\sum_{q \in R^{Q^k}} a_{tq}(\theta_q - \lambda_q) \leq M(1 - \sum_{r \in R^{Q^k}} a_{tr}\lambda_r), \qquad t \in T, \tag{34}$$

$$\sum_{o \in Q^k} \sum_{r \in R^{Q^k} : \beta^1_{or} = \beta^2_{oq} = 1 \vee \beta^2_{or} = \beta^1_{oq} = 1} \lambda_r \geq (\theta_q - \lambda_q), \qquad q \in R^{Q^k}, \tag{35}$$

$$M \sum_{r \in R^{Q^k}} b_{rd}\lambda_r \geq \sum_{q \in R^{Q^k}} b_{qd}(\theta_q - \lambda_q), \qquad d \in D, \tag{36}$$

$$\theta_r, \lambda_r \in \{0,1\}, \qquad r \in R^{Q^k}, \tag{37}$$

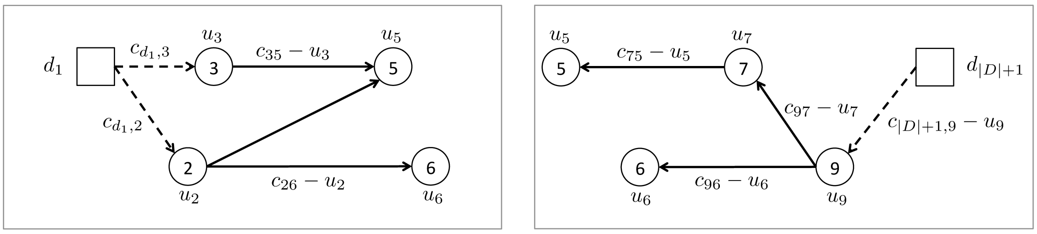
where binary parameters β^1_{qo} and β^2_{qo} are 1 if and only if route $q \in R^{Q^k}$ covers the primary and the secondary parts of quadruple $o \in Q^k$, respectively. Through constraints (32) and (33), only one route $r \in R^{Q_k}$ is selected for reduced cost minimization, and it must exist in recovery solutions by fixing $\theta_r = 1$. Constraints (34) ensure that route r, for which $\lambda_r = 1$, is disjoint from the other selected routes. Constraints (35) enforce that if route q with $\lambda_q = 0$ does not form a recovery solution with route r with $\lambda_r = 1$ for any one of the quadruples $o \in Q^k$, then route q cannot be selected. The depot compatibility of r and the other selected routes are satisfied through (36). The objective is to minimize the total reduced cost of the selected routes subtracted by the dual value of all recovery solutions.

There are two major difficulties in (31)-(37). First, the enumeration of R^{Q_k} requires a depth-first search algorithm for all $k \in K$, which can be cumbersome. Moreover, the cardinality of R^{Q_k} may be very large, which then results in a large-scale integer program that should be solved for each $k \in K$. Due to these complications that may lead to impractical solution times, we employ a heuristic methodology that compromises optimality for efficiency. Instead of generating all possible routes R^{Q_k} and solving (31)-(37), we only generate two routes with minimum reduced cost; one covering $(t^k_{p_1}, t^k_{p_2})$ and the other covering $(t^k_{s_1}, t^k_{s_2})$, for each $(t^k_{p_1}, t^k_{p_2}, t^k_{s_1}, t^k_{s_2}) \in Q^k$, $k \in K$. Therefore, two routes (r, q) forming a recovery solution for $k \in K$ are selected from R^o for each $o \in Q^k$, and the route-pair generating PSP becomes

$$\zeta_{yx} = \min_{(r,q)\in R^o, o\in Q^k, k\in K} \left\{\bar{c}_r + \bar{c}_q - z_k\right\}. \tag{38}$$

When $\zeta_{yx} < 0$, variables y_r, y_q and $x^k_{(r,q)}$ and the associated linking constraints are added to the SRMP. Then, the algorithm continues. If $\zeta_{yx} \geq 0$, then the algorithm terminates.

Generating the minimum reduced cost routes, each covering one of the two arcs associated with a quadruple, can be achieved by solving a shortest path problem. The proposed methodology is illustrated in Figures 5(a) and 5(b). Let $(3, 5, 2, 6) \in Q^k$ be one of the quadruples for extra trip k. To generate a recovery solution associated with this quadruple, two routes are generated, one including trips 3 and 5, and the other containing trips 2 and 6. As mentioned previously, these routes must belong to the same depot to ensure compatibility of them. The minimum reduced cost primary route starting from a source depot, say d_1, covering a set of trips and arcs, including $(3, 5)$, and ending at the sink node $d_{|D|+1}$ can be achieved by finding two shortest paths, one from d_1 to node 3 and the other from 5 to $d_{|D|+1}$. For the secondary



(a) Original Graph

(b) Reversed Graph

Figure 5: Reduced cost calculation for route-pair pricing subproblem

route, the minimum reduced cost path is obtained by finding two shortest paths, one from d_1 to node 2 and the other from 6 to $d_{|D|+1}$. The shortest paths from d_1 to nodes 3 and 2 are already available from the last iteration of the individual route generating PSP, in which the algorithm for the shortest path from d_1 to $d_{|D|+1}$ also provides the shortest paths to each node $t \in T$. To calculate the shortest paths from nodes 5 and 6 to sink node $d_{|D|+1}$ we should solve two shortest path problems from each of these nodes to $d_{|D|+1}$. However, when the number of quadruples in Q^k is large, solving many shortest path problems inflicts a computational burden on the solution of the PSP. Instead, we find the shortest path from $d_{|D|+1}$ to each $t \in T$ in the reversed graph by solving only one shortest path problem from $d_{|D|+1}$ to d_1. Figure 5(a) shows the shortest paths to nodes 2 and 3 with the reduced costs on the original graph. In Figure 5(b), the reversed graph is illustrated with the reversed arcs. This graph is used to find the shortest paths from the sink node to nodes 5 and 6. The reduced cost of the route r that contains trips 3 and 5 and the reduced cost of route q that contains trips 2 and 6 are

$$\begin{aligned}\bar{c}_r &= \bar{c}_{d_1,3} + \hat{c}_{d_{|D|+1},5} + c_{3,5} - u_3 - v_{d_1},\\ \bar{c}_q &= \bar{c}_{d_1,2} + \hat{c}_{d_{|D|+1},6} + c_{2,6} - u_2 - v_{d_1}.\end{aligned}$$

We show in the next section that this heuristic method reaches the optimal solution in most of the small instances and solves instances with as many as 500 trips efficiently.

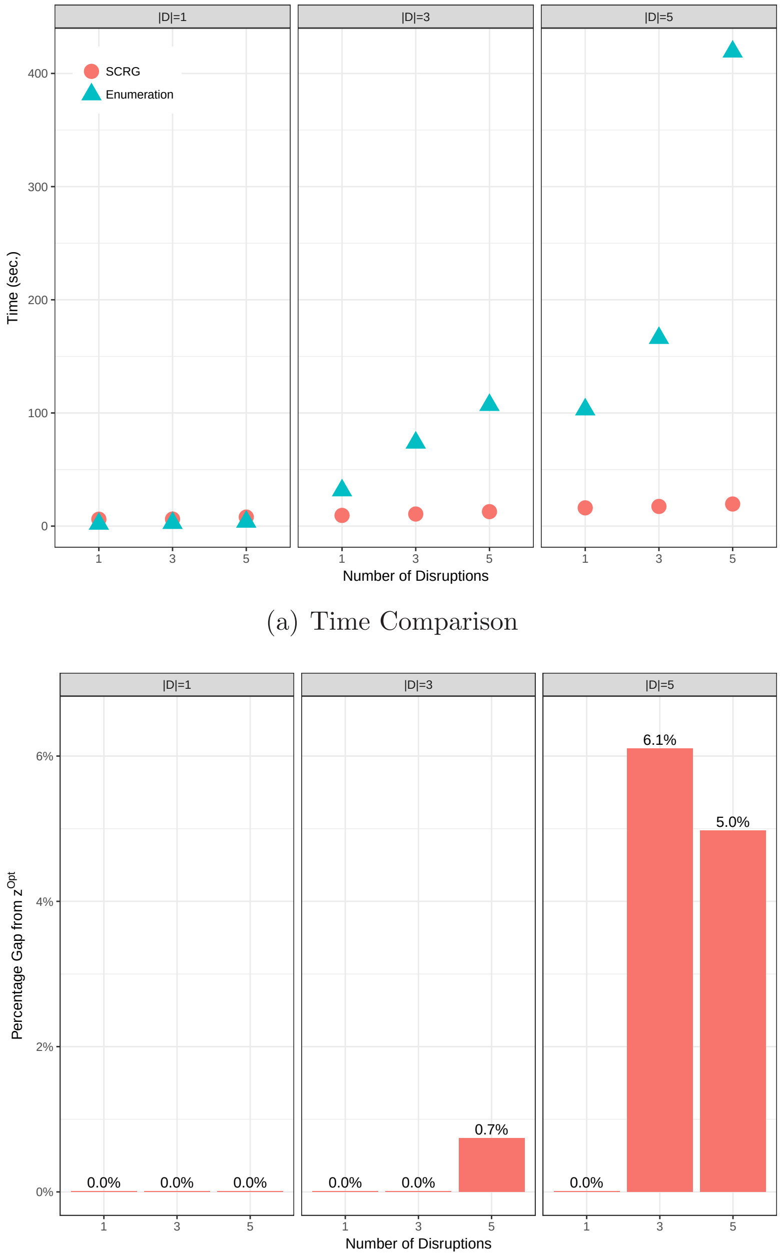
5. Computational Experiments

In this section, we present the setting of the computational experiments and report the results obtained by solving a set of randomly generated instances using our proposed simultaneous column-and-row generation algorithm. We conduct our computational experiments on a machine with a 2.66 GHz Intel(R) Core(TM)2 Quad CPU and 8 GB of RAM. The algorithm is implemented in Visual C++, and the SRMPs formed at each iteration of our algorithm are solved by ILOG CPLEX 12.1 using ILOG Concert Technology 2.9. Since our proposed algorithm solves the LP relaxation of (**RMDVSP**), in order to obtain an integer solution, we solve the final SRMP with the mixed integer programming (MIP) solver of ILOG CPLEX 12.1 which is also employed to find the optimal solution of (**RMDVSP**) when enumeration is possible. We impose a one-hour time limit in the solution of each instance.

To test the computational efficiency of the proposed simultaneous column-and-row generation algorithm, we randomly generate a set of timetabled trips and depots along with disruptions. We select 33 cities in Turkey as the set of locations and utilize up to five of them as depot locations. Similar to the random data generator utilized by Carpaneto et al. (1989), we determine the arrival and departure locations of each trip by sampling from uniform distribution. To generate the departure times of the trips, we discretize the time horizon into 30 minutes intervals and randomly select the intervals.

For the generation of the extra trips, we first select the possible departure and arrival locations of extra trips, and then we form these trips similar to the timetabled trips. For the generation of the delayed trips, we choose the possible delayed trips randomly from the set of timetabled trips. The length of the delay is also uniformly distributed between 60 and 180 minutes. We determine the fleet size at each depot randomly between $\left[\frac{|T|}{3|D|}, \frac{|T|}{2|D|}\right]$. In the experiments, we test the algorithms on instances with different values for $|T| \in \{100, 300, 500\}$. Moreover, the number of depots $|D|$ is selected from the set $\{1, 3, 5\}$.

The performance of our proposed heuristic is compared against an optimal approach for (**RMDVSP**). To that end, we enumerate all feasible routes and identify all recovery solutions for each disruption. Since the size of the route set and the recovery solution set are prohibitively large, the enumeration is possible only for problems with a limited size. In Figure 6, the results of the tests conducted to assess the performance of the proposed methodology are given. A set of instances with 100 trips and varying number of depots and disruptions is solved by our proposed heuristic simultaneous column-and-row generation algorithm (SCRG) and an enumeration-based approach (Enumeration) that finds the optimal solution of (**RMDVSP**). In Figure 6(a), the total computation time of SCRG combined with the solution time of the MIP solver – when the solution is fractional – and the computation time for Enumeration are compared. As the number of disruptions and depots increases, the total number of routes and the total number of recovery solutions increase dramatically. For instance, in the case of 5 disruptions and 5 depots, 118,995 recovery solutions are available. According to the mathematical model, each recovery solution induces three linking constraints, so that the total number of rows is larger than 350,000, which causes Enumeration to bear a huge burden in solving (**RMDVSP**). It is evident from this figure that Enumeration is faster than SCRG in solving the small instances, however, for large instances, using SCRG is inevitable. For 5 disruptions and 5 depots, SCRG is much faster than Enumeration. We also evaluate the performance of our proposed method in terms of the solution quality by calculating the gap between the value obtained by our methodology and the value of the optimal integral solution, known as the optimality gap, in Figure 6(b). In many instances, we observe that SCRG itself or the subsequent MIP solution result in an objective value coinciding with the value of the optimal integer solution of (**RMDVSP**) obtained by Enumeration. This finding is one way of justifying the usefulness of the heuristic approach explained in Section 4.4. We note that the average optimality gap using the proposed SRCG is only 1.3% for these 9 instances.



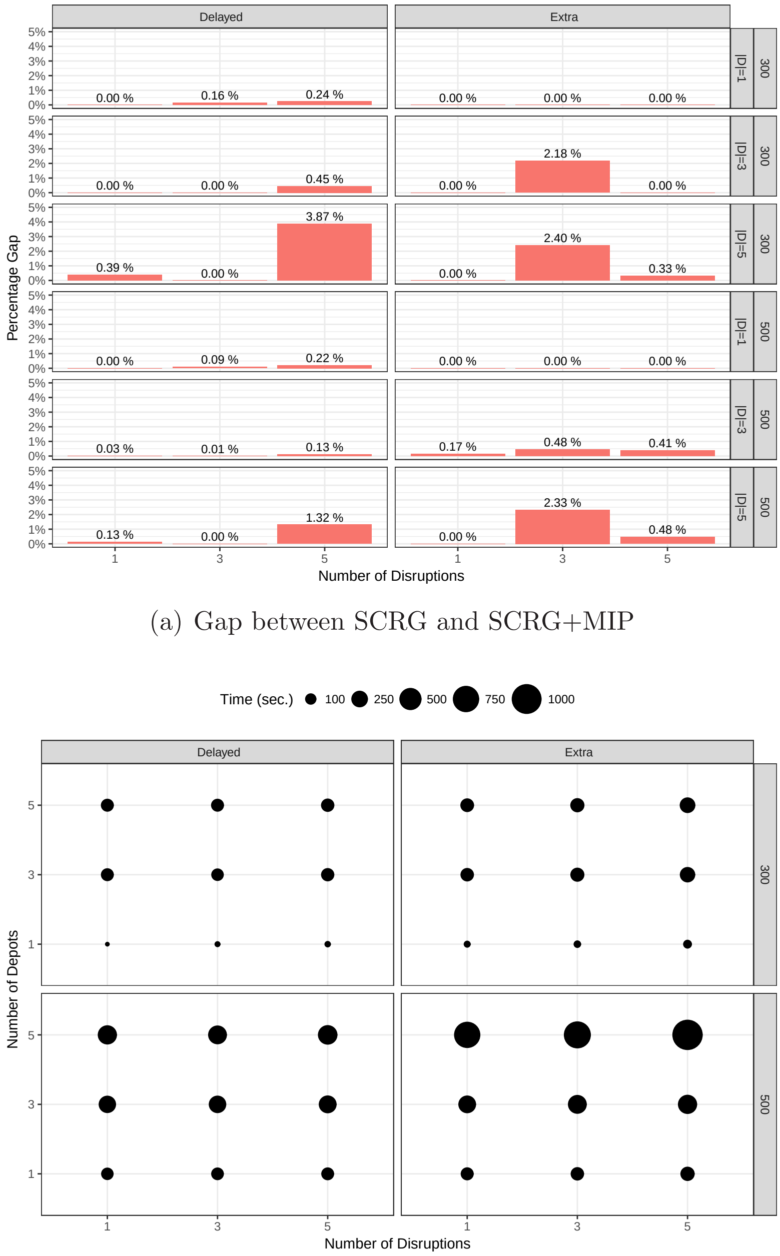
(a) Time Comparison

(b) Bound Comparison

Figure 6: Comparison of SCRG and Enumeration

In Figure 7, the results of large instances that are obtained by the application of SCRG are given for extra trips and delayed trips. Figure 7(a) shows the percentage gap between the objective function value reached by SCRG and the value obtained by solving the final SRMP with the MIP solver. Solving the final SRMP by the MIP solver leads, on average, to a gap of 0.4 % for delayed trips and to a gap of 0.5 % for extra trips. While this tight gap does not provide any evidence on the quality of these integer solutions, it justifies the use of the MIP solver instead of integrating SCRG within a branch-and-bound framework which would increase the solution time drastically. The computation times to find the integer solution, which includes the solution time of both SCRG and MIP solver, are reported in Figure 7(b). The results of tests with both delayed and extra trips convey that the solution time of the MIP solver is negligible, namely smaller than a second on the average. There is a discernible increase in the solution time as the number of disruptions increases. In the tests with larger number of disruptions, the algorithm terminates due to time limit for the instances with 500 trips.

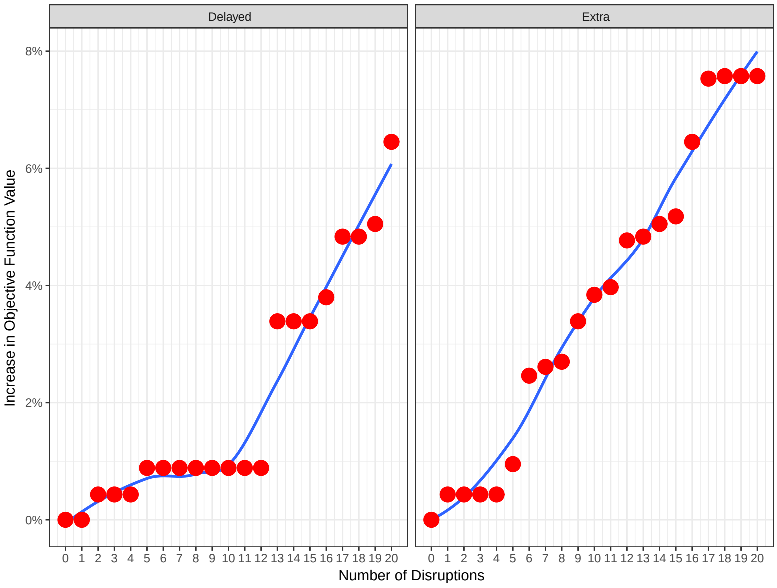
To investigate the effect of the number of disruptions in the total cost of the planned schedule, we add up to 20 disruptions to a medium size problem (300 trips, 3 depots) and obtain the solution with SCRG combined with the MIP solver. To that end, we find the objective function value of each instance without disruptions as the basis for comparison and calculate the percentage increase in the objective function value when extra trips and delayed trips are incorporated. These results are illustrated in Figure 8(a). Even though these values are not obtained by an exact approach, they still provide an insight about the behavior of the robust model proposed in this paper when the number of disruptions increases. Analyzing the results in terms of the types of disruptions, we observe that the average increase in the total cost is slightly larger when extra trips are handled. The largest increase in the total cost is observed in the case of 20 extra trips and delayed trips with approximately 7.6% and 6.5%, respectively. However, we re-emphasize that if the proposed solution method is not used, it would be not only difficult but also more costly for the planner to handle these disruptions with the solution of the conventional MDVSP. Moreover, in Figure 8(b), the total computation times including the MIP solver times are reported for extra trips and delayed trips. According to these results, there is an upward trend in the computation time as the number of disruptions increases.



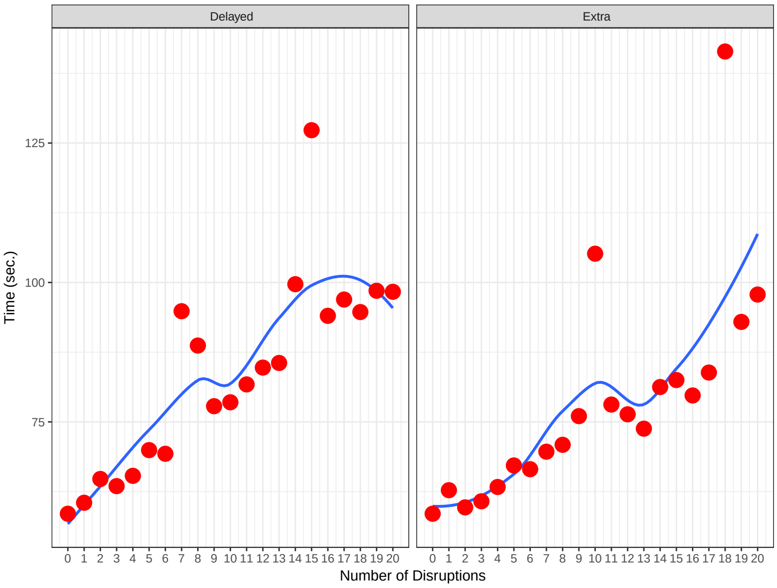
(a) Gap between SCRG and SCRG+MIP

(b) Computation Times (SCRG+MIP)

Figure 7: Results for a collection of instances with delayed and extra trips



(a) The effect of disruptions on total cost



(b) The effect of disruptions on computation time

Figure 8: Results for varying numbers of disruptions for $|T| = 300$

6. Conclusions

In this paper, we have studied disruptions to the planned vehicle schedules, and presented a unique recovery method based on partially swapping two planned routes. The linear programming relaxation of the proposed mathematical model has been solved by a simultaneous column-and-row generation algorithm. The unique feature of this algorithm is that a set of variables forming recovery solutions is generated simultaneously in a novel pricing subproblem. We have accelerated this algorithm by limiting this pricing subproblem to generate a single pair of variables with the smallest total reduced cost. Our computational experiments convey that the resulting heuristic method is capable of yielding small optimality gaps and more importantly, present recovery solutions that can be operated when the disruptions are realized.

Acknowledgement

This study is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under grants 113M014 and 113M480. The authors also acknowledge the contribution of the COST Action TD1207.

References

Bertossi, A. A., Carraresi, P., and Gallo, G. (1987). On some matching problems arising in vehicle scheduling models. *Networks*, 17(3):271–281.

Bodin, L., Rosenfield, D., and Kydes, A. (1978). Ucost: a micro approach to a transportation planning problem. *Journal of Urban Analysis*, 5(1):47–69.

Carpaneto, G., Dell'Amico, M., Fischetti, M., and Toth, P. (1989). A branch and bound algorithm for the multiple depot vehicle scheduling problem. *Networks*, 19(5):531–548.

Ceder, A. A. (2011). Public-transport vehicle scheduling with multi vehicle type. *Transportation Research Part C: Emerging Technologies*, 19(3):485–497.

Dantzig, G. and Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research*, 8:101–111.

Dell'Amico, M., Fischetti, M., and Toth, P. (1993). Heuristic algorithms for the multiple depot vehicle scheduling problem. *Management Science*, 39(1):115–125.

Desaulniers, G., Desrosiers, J., and Solomon, M., editors (2005). *Column Generation*, New York. Springer.

Fischetti, M., Lodi, A., Martello, S., and Toth, P. (2001). A polyhedral approach to simplified crew scheduling and vehicle scheduling problems. *Management Science*, 47(6):833–850.

Freling, R., Wagelmans, A. P., and Paixão, J. M. P. (2001). Models and algorithms for single-depot vehicle scheduling. *Transportation Science*, 35(2):165–180.

Gilmore, P. C. and Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, 9:849–859.

Hadjar, A., Marcotte, O., and Soumis, F. (2006). A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. *Operations Research*, 54(1):130–149.

Haghani, A. and Banihashemi, M. (2002). Heuristic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints. *Transportation Research Part A: Policy and Practice*, 36(4):309–333.

Haghani, A., Banihashemi, M., and Chiang, K.-H. (2003). A comparative analysis of bus transit vehicle scheduling models. *Transportation Research Part B: Methodological*, 37(4):301 – 322.

Hassold, S. and Ceder, A. A. (2014). Public transport vehicle scheduling featuring multiple vehicle types. *Transportation Research Part B: Methodological*, 67:129 – 143.

Huisman, D., Freling, R., and Wagelmans, A. P. (2004). A robust solution approach to the dynamic vehicle scheduling problem. *Transportation Science*, 38(4):447–458.

Kramkowski, S., Kliewer, N., and Meier, C. (2009). Increasing delay-tolerance of vehicle schedules in public bus transport. In *Proceedings of the Metaheuristic International Conference VIII*, Hamburg, Germany.

Li, J.-Q., Mirchandani, P. B., and Borenstein, D. (2007). The vehicle rescheduling problem: Model and algorithms. *Networks*, 50(3):211–229.

Li, J, Q., Borensteinb, D., and Mirchandania, P, B. (2009). A Lagrangian heuristic for the real-time vehicle rescheduling problem. *Transportation Research Part E*, 45(3):419–433.

Löbel, A. (1998). Vehicle scheduling in public transit and Lagrangean pricing. *Management Science*, 44(12-part-1):1637–1649.

Muter, İ., Birbil, Ş. İ., and Bülbül, K. (2013a). Simultaneous column-and-row generation for large-scale linear programs with column-dependent-rows. *Mathematical Programming*, 142(1-2):47–82.

Muter, İ., Birbil, Ş. İ., Bülbül, K., Şahin, G., Yenigün, H., Taş, D., and Tüzün, D. (2013b). Solving a robust airline crew pairing problem with column generation. *Computers & Operations Research*, 40(3):815–830.

Petersen, H. L., Larsen, A., Madsen, O. B., Petersen, B., and Ropke, S. (2012). The simultaneous vehicle scheduling and passenger service problem. *Transportation Science*, 47(4):603–616.

Ribeiro, C. C. and Soumis, F. (1994). A column generation approach to the multiple-depot vehicle scheduling problem. *Operations Research*, 42(1):41–52.

Sato, T., Sakikawa, S., Morita, T., Ueki, N., and Murata, T. (2009). Crew and vehicle rescheduling based on a network flow model and its application to a railway train operation. *IAENG International Journal of Applied Mathematics*, 39(3):142–150.

Shebalov, S. and Klabjan, D. (2006). Robust airline crew pairing: Move-up crews. *Transportation Science*, 40:300–312.

Shen, Y., Xu, J., and Li, J. (2016). A probabilistic model for vehicle scheduling based on stochastic trip times. *Transportation Research Part B: Methodological*, 85:19–31.

Tekiner, H., Birbil, Ş. İ., and Bülbül, K. (2009). Robust crew pairing for managing extra flights. *Computers & Operations Research*, 36(6):2031–2048.

Visentini, M. S., Borenstein, D., Li, J.-Q., and Mirchandani, P. B. (2014). Review of real-time vehicle schedule recovery methods in transportation services. *Journal of Scheduling*, 17(6):541–567.

Appendix A. Proof of Lemma 4.1

We first explain the structure of $\mathbf{B}$ and the augmented basis $\mathbf{B}_r$, and then show in the proof that with the proposed basis augmentation, the values of the existing dual variables do not change and the values of the new dual variables are anticipated correctly. While we denote the values of the dual variables associated with $\mathbf{B}$ (existing dual variables) and those associated with $\mathbf{B_r}$ by $\mathbf{w}$ and $\mathbf{w}_{aug}$, respectively, the reduced cost of variable y_r calculated by using $\mathbf{w}_{aug}$ is denoted by $\bar{c}_{aug_r}$. Finally, $\mathbf{A}_r$ and $\mathbf{A}_{aug_r}$ denote the columns of variable y_r in the rows of the SRMP and the augmented SRMP, respectively.

We define the optimal basis of the SRMP as $\mathbf{B} = \begin{pmatrix} \mathbf{A_1} & \mathbf{0} & \mathbf{E_1} \\ \mathbf{0} & \mathbf{B_1} & \mathbf{E_2} \\ \mathbf{C_1} & \mathbf{D_1} & \mathbf{E_3} \end{pmatrix}$. Here, the matrix $\begin{pmatrix} \mathbf{A_1} \\ \mathbf{0} \\ \mathbf{C_1} \end{pmatrix}$ shows the columns of $y-$variables in the basis and the matrix $\begin{pmatrix} \mathbf{A_1} & \mathbf{0} & \mathbf{E_1} \end{pmatrix}$ represents the set of constraints (15) and (16). The matrix $\begin{pmatrix} \mathbf{0} \\ \mathbf{B_1} \\ \mathbf{D_1} \end{pmatrix}$ includes the columns of x variables. The matrix $\begin{pmatrix} \mathbf{0} & \mathbf{B_1} & \mathbf{E_2} \end{pmatrix}$ is associated with constraints (17), the matrix $\begin{pmatrix} \mathbf{E_1} \\ \mathbf{E_2} \\ \mathbf{E_3} \end{pmatrix}$ corresponds to the columns of the basic surplus and slack variables, and the matrix $\begin{pmatrix} \mathbf{C_1} & \mathbf{D_1} & \mathbf{E_3} \end{pmatrix}$ represents the existing linking constraints. The augmented basis for recovery solution $(r,q) \in \mathcal{F}_r^k$ can be constructed as

$$\mathbf{B}_r = \left(\begin{array}{ccc|ccc} \mathbf{A_1} & \mathbf{0} & \mathbf{E_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B_1} & \mathbf{E_2} & \mathbf{B_2} & \mathbf{0} & \mathbf{0} \\ \mathbf{C_1} & \mathbf{D_1} & \mathbf{E_3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & -1 & 0 & 0 \\ \mathbf{C_2} & \mathbf{0} & \mathbf{0} & -1 & -1 & 0 \\ \mathbf{C_3} & \mathbf{0} & \mathbf{0} & -1 & 0 & 1 \end{array}\right) = \left(\begin{array}{c|ccc} \mathbf{B} & \mathbf{F} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & -1 & 0 & 0 \\ \mathbf{G} & -1 & -1 & 0 \\ \mathbf{H} & -1 & 0 & 1 \end{array}\right) \tag{A.1}$$

where $\mathbf{F} = \begin{pmatrix} \mathbf{0} \\ \mathbf{B_2} \\ \mathbf{0} \end{pmatrix}$, $\mathbf{G} = \begin{pmatrix} \mathbf{C_2} & \mathbf{0} & \mathbf{0} \end{pmatrix}$ and $\mathbf{H} = \begin{pmatrix} \mathbf{C_3} & \mathbf{0} & \mathbf{0} \end{pmatrix}$. The fourth column contains the coefficient of x_{rq}, and the fifth and the sixth columns represent the surplus and the slack variables associated with constraints (19) and (20), respectively. If y_q is already part of the basis, $\mathbf{C_2}$ and $\mathbf{C_3}$ are zero vectors containing a single 1, and they are $\mathbf{0}$, otherwise. According to the basis augmentation designed in (29a)-(29e), the dual variables associated with the rows in which an existing variable reside are set to 0 through selecting the corresponding surplus and slack variables as basic. Assuming $z_k > 0$, $x^k_{(r,q)}$ is chosen as basic for constraint (18) which is the fourth row in $\mathbf{B}_r$, where only y_r resides. Instead of only one route pair (r,q), we can construct $\mathbf{B}_r$ for all $\{(r,q),(q,r) \in \mathcal{F}_r^k\}$ by combining new linking constraints in three groups as those for which $x-$variables, surplus and slack variables are basic.

$$\mathbf{B}_r = \left(\begin{array}{c|ccc} \mathbf{B} & \mathbf{F} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{G} & -\mathbf{I} & -\mathbf{I} & \mathbf{0} \\ \mathbf{H} & -\mathbf{I} & \mathbf{0} & \mathbf{I} \end{array}\right) \tag{A.2}$$

The fourth set of rows involves constraints for which $x-$variables are selected as basic, and the fifth and sixth sets of rows correspond to constraints for which surplus and slack variables are basic, respectively. Lemma 4.1 shows that the augmented basis is optimal for the augmented SRMP. Here is the proof of this lemma.

Proof. To prove the optimality of the augmented basis $\mathbf{B}_r$, we need to show that the primal and dual feasibility and complementary slackness are satisfied. The primal feasibility is satisfied because all linking constraints and $x-$variables associated with the existing recovery solutions have already been added to the SRMP before the route-pair generating PSP is called, and the new linking constraints of type (18)-(20) induced by $\mathcal{F}_r^k$ are redundant before y_r is added to the SRMP.

To prove the dual feasibility and complementary slackness, we first present the inverse of $\mathbf{B}_r$. Let $\mathbf{J} = \begin{pmatrix} \mathbf{B} & \mathbf{F} \\ \mathbf{0} & -\mathbf{I} \end{pmatrix}$. The inverse is $\mathbf{J}^{-1} = \begin{pmatrix} \mathbf{B}^{-1} & \mathbf{B}^{-1}\mathbf{F} \\ \mathbf{0} & -\mathbf{I} \end{pmatrix}$. Letting $\mathbf{M} = \begin{pmatrix} \mathbf{G} & -\mathbf{I} \\ \mathbf{H} & \mathbf{0} \end{pmatrix}$ and $\mathbf{K} = \begin{pmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$, $\mathbf{B}_r$ and $\mathbf{B}_r^{-1}$ can be written as $\mathbf{B}_r = \begin{pmatrix} \mathbf{J} & \mathbf{0} \\ \mathbf{M} & \mathbf{K} \end{pmatrix}$ and $\mathbf{B_r}^{-1} = \begin{pmatrix} \mathbf{J}^{-1} & \mathbf{0} \\ -\mathbf{K}^{-1}\mathbf{M}\mathbf{J}^{-1} & \mathbf{K}^{-1} \end{pmatrix}$, respectively. Thus,

$$\mathbf{B}_r^{-1} = \left(\begin{array}{c|ccc} \mathbf{B}^{-1} & \mathbf{B}^{-1}\mathbf{F} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{G}\mathbf{B}^{-1} & \mathbf{G}\mathbf{B}^{-1}\mathbf{F} + \mathbf{I} & -\mathbf{I} & \mathbf{0} \\ -\mathbf{H}\mathbf{B}^{-1} & -\mathbf{H}\mathbf{B}^{-1}\mathbf{F} - \mathbf{I} & \mathbf{0} & \mathbf{I} \end{array} \right). \tag{A.3}$$

Let the objective function values of the variables in the augmented basis be $\mathbf{c}_{\mathbf{B}aug} = \begin{pmatrix} \mathbf{c_B} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$ where $\mathbf{c_B}$ is comprised of the objective function coefficients of the basic variables whose columns form $\mathbf{B}$, and the next three entries are the objective function coefficients of x, surplus and slack variables, respectively. The values of the dual variables induced by $\mathbf{B}_r$ become

$$\mathbf{w}_{aug}^r = \left(\begin{array}{c|ccc} \mathbf{c_B} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right) \left(\begin{array}{c|ccc} \mathbf{B}^{-1} & \mathbf{B}^{-1}\mathbf{F} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{G}\mathbf{B}^{-1} & \mathbf{G}\mathbf{B}^{-1}\mathbf{F} + \mathbf{I} & -\mathbf{I} & \mathbf{0} \\ -\mathbf{H}\mathbf{B}^{-1} & -\mathbf{H}\mathbf{B}^{-1}\mathbf{F} - \mathbf{I} & \mathbf{0} & \mathbf{I} \end{array} \right) = \left(\begin{array}{c|ccc} \mathbf{c_B}\mathbf{B}^{-1} & \mathbf{c_B}\mathbf{B}^{-1}\mathbf{F} & \mathbf{0} & \mathbf{0} \end{array} \right), \tag{A.4}$$

where $\mathbf{F}$ now stands for the coefficients of the new $x-$variables induced by $\mathcal{F}_r^k$ in the current SRMP. Note that each column of $\mathbf{F}$ has a single nonzero entry with value one which is located at the kth row of (17) so that $\mathbf{c_B}\mathbf{B}^{-1}\mathbf{F} = \mathbf{w}\mathbf{F} = z_k$. Thus, $\mathbf{w}_{aug}^r = \left(\begin{array}{c|ccc} \mathbf{w} & \mathbf{z_k} & \mathbf{0} & \mathbf{0} \end{array} \right)$, in which the values of the dual variables induced by $\mathbf{B}$ do not change, and the values of the dual variables associated with the new linking constraints, denoted by vector $\mathbf{z_k}$ with size 1x$|\mathcal{F}_r^k|$, are equal to those designated by the route-pair generating PSP.

Since existing $x-$variables do not appear in the new linking constraints, their reduced cost remains non-negative. Non-negativity of the reduced costs of the new $x-$variables and slack/surplus variables associated with the new linking constraints induced by $\mathcal{F}_r^k$ are imposed through the constraints in (29). The reduced cost of $q \in \bar{R}$ incorporating the dual values of the new constraints associated with $(r, q) \in P(k) \backslash \bar{P}(k)$ is calculated as

$$\bar{c}_{aug_q} = c_q - \mathbf{w}_{aug}\mathbf{A}_{aug_q} = c_q - \left(\begin{array}{c|ccc} \mathbf{w} & z_k & 0 & 0 \end{array} \right) \left(\begin{array}{c|ccc} \mathbf{A}_q & 0 & 1 & 1 \end{array} \right)^T = \bar{c}_q. \tag{A.5}$$

which also holds for $(q, r) \in P(k) \backslash \bar{P}(k)$. Hence, the values of the dual variables induced by $\mathbf{B}_r$ satisfy dual feasibility. This concludes the proof. □

Appendix B. Proof of Theorem 4.1

We only consider the case $z_k > 0$ for a given $k \in K$ since otherwise $\mathcal{F}_r^k$ would be an empty set. For ease of illustration, we take two route pairs, (r, q) and (r, l), in $\mathcal{F}_r^k$, i.e. $S_r^k = \{q, l\}$. However, the result can simply be extended to larger sets in which r is the secondary route in some of the route pairs. In that case, our proof follows with the same steps but with larger matrices.

Proof. To cover possible cases, let us assume that y_q is not part of the basis and y_l is part of the basis. The termination condition then becomes

$$\bar{c}_r + \bar{c}_q + \bar{c}_l - 2z_k \geq 0, \tag{B.1}$$

where $\bar{c}_r, \bar{c}_q \geq 0$ and $\bar{c}_l = 0$. We show that if the above condition is satisfied for a given $\mathcal{F}_r^k$, only degenerate iterations can be performed without changing the values of the existing dual variables.

According to the basis augmentation defined previously, for both route pairs, $(r, m) \in \mathcal{F}_r^k$, we can augment the basis as $\{x_{(r,m)}^k, s_{rm}^{[2]}, s_{rm}^{[3]}\}$ where $s_{rm}^{\cdot}$ is the slack/surplus variable associated with the second or the third row of the linking constraint set. The augmented basis can be written as

$$\mathbf{B}_r = \begin{pmatrix} \mathbf{B} & \mathbf{F} & \mathbf{0} & \mathbf{0} & \mathbf{F} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -1 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{0} & -1 & -1 & 0 & 0 & 0 & 0 \\ \mathbf{0} & -1 & 0 & 1 & 0 & 0 & 0 \\ \mathbf{0} & 0 & 0 & 0 & -1 & 0 & 0 \\ \mathbf{G} & 0 & 0 & 0 & -1 & -1 & 0 \\ \mathbf{H} & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}, \tag{B.2}$$

where the second, third and fourth columns are associated with $x_{(r,q)}^k$, $s_{rq}^{[2]}$ and $s_{rq}^{[3]}$, respectively. The rows with the same order correspond to the linking constraints associated with (r, q). The rest of the columns and rows are associated with (r, l). In order to facilitate the inverse operations, we multiply the last row by -1 and define $\mathbf{D_1} = \left(\begin{smallmatrix} -1 \\ 1 \end{smallmatrix}\right)$, $\mathbf{D_2} = \left(\begin{smallmatrix} -1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & 0 & 1 \end{smallmatrix}\right)$, $\mathbf{M} = \left(\begin{smallmatrix} \mathbf{G} \\ -\mathbf{H} \end{smallmatrix}\right)$ and $\mathbf{X_1} = \left(\begin{smallmatrix} \mathbf{F} & \mathbf{0} & \mathbf{0} \end{smallmatrix}\right)$. Hence, the augmented basis and its inverse become

$$\mathbf{B}_r = \begin{pmatrix} \mathbf{B} & \mathbf{X_1} & \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -1 & \mathbf{0} \\ \mathbf{M} & \mathbf{0} & \mathbf{D_1} & -\mathbf{I} \end{pmatrix} \tag{B.3}$$

and

$$\mathbf{B}_r^{-1} = \begin{pmatrix} \mathbf{B}^{-1} & -\mathbf{B}^{-1}\mathbf{X_1}\mathbf{D_2}^{-1} & \mathbf{B}^{-1}\mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_2}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -1 & \mathbf{0} \\ \mathbf{M}\mathbf{B}^{-1} & -\mathbf{M}\mathbf{B}^{-1}\mathbf{X_1}\mathbf{D_2}^{-1} & \mathbf{M}\mathbf{B}^{-1}\mathbf{F} - \mathbf{D_1} & -\mathbf{I} \end{pmatrix}, \tag{B.4}$$

respectively. The values of the dual variables induced by the augmented basis given above are calculated as

$$\begin{aligned}\mathbf{w}_{aug} &= \begin{pmatrix}\mathbf{c_B} & \mathbf{0} & 0 & \mathbf{0}\end{pmatrix}\mathbf{B}_r{}^{-1}\\ &= \begin{pmatrix}\mathbf{c_B B}^{-1} & -\mathbf{c_B B}^{-1}\mathbf{X_1 D_2}{}^{-1} & \mathbf{c_B B}^{-1}\mathbf{F} & \mathbf{0}\end{pmatrix}\\ &= \begin{pmatrix}\mathbf{c_B B}^{-1} & \begin{pmatrix} z_k & 0 & 0\end{pmatrix} & z_k & \mathbf{0}\end{pmatrix}.\end{aligned}$$

According to the thinking-ahead approach, only the reduced cost of variable y_r is affected by the newly added linking constraints, which is shown in the proof of Lemma 4.1 in Appendix A. The reduced cost of y_r

$$\bar{c}_{aug_r} = c_r - \mathbf{w}_{aug}\begin{pmatrix}\mathbf{A}_r & 1 & 0 & 1 & 1 & 0 & 1\end{pmatrix}^T = \bar{c}_r - 2z_k$$

is negative given that (B.1) is not satisfied and $\bar{c}_q, \bar{c}_l \geq 0$. Hence, y_r is the only candidate to enter the basis. Next, we show that $s_{rq}^{[2]}$, which corresponds to the third column in $\mathbf{B}_r$, can be selected as the leaving variable after the minimum ratio test. To that end, we calculate the third entries of $\bar{\mathbf{A}}_{r_{aug}} = \mathbf{B}_r{}^{-1}\mathbf{A}_{r_{aug}}$ and $\bar{\mathbf{b}}_{aug} = \mathbf{B}_r^{-1}b_{aug}$, which are denoted by $\bar{A}_{r_{aug}}^{[3]}$ and $\bar{b}_{aug}^{[3]}$, respectively. That is

$$\begin{aligned}\bar{\mathbf{A}}_{r_{aug}} &= \mathbf{B}_r{}^{-1}\mathbf{A}_{raug} = \mathbf{B}_r{}^{-1}\begin{pmatrix}\mathbf{A}_r & 1 & 0 & 1 & 1 & 0 & 1\end{pmatrix}^T\\ \bar{A}_{r_{aug}}^{[3]} &= \begin{pmatrix}\mathbf{0} & 1 & -1 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\mathbf{A}_r & 1 & 0 & 1 & 1 & 0 & 1\end{pmatrix}^T = 1\end{aligned}$$

and

$$\begin{aligned}\bar{\mathbf{b}}_{aug} &= \mathbf{B}_r{}^{-1}\mathbf{b}_{aug} = \mathbf{B}_r{}^{-1}\begin{pmatrix}\mathbf{b} & 0 & 0 & 1 & 0 & 0 & -1\end{pmatrix}^T\\ \bar{b}_{aug}^{[3]} &= \begin{pmatrix}\mathbf{0} & 1 & -1 & 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\mathbf{b} & 0 & 0 & 1 & 0 & 0 & -1\end{pmatrix} = 0.\end{aligned}$$

Consequently, y_r enters the basis and $s_{rq}^{[2]}$ leaves the basis in a degenerate iteration. The augmented basis $\mathbf{B}_r$ is updated as

$$\mathbf{B}_r = \begin{pmatrix}\mathbf{B} & \mathbf{F} & \mathbf{A}_r & \mathbf{0} & \mathbf{F} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & -1 & 1 & 0 & 0 & 0 & 0\\ \mathbf{0} & -1 & 0 & 0 & 0 & 0 & 0\\ \mathbf{0} & -1 & 1 & 1 & 0 & 0 & 0\\ \mathbf{0} & 0 & 1 & 0 & -1 & 0 & 0\\ \mathbf{G} & 0 & 0 & 0 & -1 & -1 & 0\\ -\mathbf{H} & 0 & -1 & 0 & +1 & 0 & -1\end{pmatrix} = \begin{pmatrix}\mathbf{B} & \mathbf{X_2} & \mathbf{F} & \mathbf{0}\\ \mathbf{0} & \mathbf{D_3} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{D_4} & -1 & \mathbf{0}\\ \mathbf{M} & \mathbf{D_5} & \mathbf{D_1} & -\mathbf{I}\end{pmatrix}, \tag{B.5}$$

where $\mathbf{D_3} = \left(\begin{smallmatrix}-1 & 1 & 0\\ -1 & 0 & 0\\ -1 & 1 & 1\end{smallmatrix}\right)$, $\mathbf{D_4} = \left(\begin{smallmatrix}0 & 1 & 0\end{smallmatrix}\right)$, $\mathbf{D_5} = \left(\begin{smallmatrix}0 & 0 & 0\\ 0 & -1 & 0\end{smallmatrix}\right)$ and $\mathbf{X_2} = \left(\begin{smallmatrix}\mathbf{F} & \mathbf{A}_r & \mathbf{0}\end{smallmatrix}\right)$. The inverse of the basis is given by

$$\mathbf{B}_r^{-1} = \begin{pmatrix}\mathbf{B}^{-1} & -\mathbf{B}^{-1}\mathbf{X_2 D_3}{}^{-1} - \mathbf{B}^{-1}\mathbf{F D_4 D_3}{}^{-1} & \mathbf{B}^{-1}\mathbf{F} & \mathbf{0}\\ \mathbf{0} & \mathbf{D_3}{}^{-1} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{D_4 D_3}{}^{-1} & -1 & 0\\ \mathbf{MB}^{-1} & \substack{-\mathbf{MB}^{-1}\mathbf{X_2 D_3}{}^{-1} - \mathbf{MB}^{-1}\mathbf{F D_4 D_3}{}^{-1}\\ +\mathbf{D_5 D_3}{}^{-1} + \mathbf{D_1 D_4 D_3}{}^{-1}} & \mathbf{MB}^{-1}\mathbf{F} - \mathbf{D_1} & -\mathbf{I}\end{pmatrix}. \tag{B.6}$$

Thus, the values of the dual variables become

$$\begin{gathered}
\mathbf{w}_{aug} = \begin{pmatrix}\mathbf{c_B} & \mathbf{c_{X_2}} & 0 & 0\end{pmatrix} \mathbf{B}_r{}^{-1} \\
= \begin{pmatrix}\mathbf{c_B B}^{-1} & -\mathbf{c_B B}^{-1}\mathbf{X_2 D_3}{}^{-1} - \mathbf{c_B B}^{-1}\mathbf{F D_4 D_3}{}^{-1} + \mathbf{c_{X_2} D_3}{}^{-1} & \mathbf{c_B B}^{-1}\mathbf{F} & \mathbf{0}\end{pmatrix} \\
= \begin{pmatrix}\mathbf{c_B B}^{-1} & \left(-\mathbf{c_B B}^{-1}\mathbf{A}_r,\, \mathbf{c_B B}^{-1}\mathbf{F}+\mathbf{c_B B}^{-1}\mathbf{A}_r,\, 0 \right) - \left(z_k,\, -z_k,\, 0 \right) + \left(c_r,\, -c_r,\, 0 \right) & z_k & \mathbf{0}\end{pmatrix} \\
= \begin{pmatrix}\mathbf{c_B B}^{-1} & \left(\bar{c}_r - z_k,\, 2z_k - \bar{c}_r,\, 0 \right) & z_k & \mathbf{0}\end{pmatrix},
\end{gathered}$$

where $\mathbf{c_{X_2}} = \left(0 \; c_r \; 0 \right)$. Neither the values of $\mathbf{w}$ nor the reduced costs of the variables other than y_q change as

$$\bar{c}_{aug_q} = c_q - \mathbf{w}_{aug} \begin{pmatrix}\mathbf{A}_q & 0 & 1 & 1 & 0 & 0 & 0\end{pmatrix}^T = \bar{c}_q + \bar{c}_r - 2z_k.$$

If (B.1) is not satisfied; i.e., $\bar{c}_l + \bar{c}_q + \bar{c}_r - 2z_k = \bar{c}_{aug_q} < 0$, then y_q enters the basis and changes the values of the existing dual variables as well as, possibly, the solution. Otherwise, y_q and no other variable can enter the basis so that the algorithm can be terminated.

Observe that each recovery solution in $\mathcal{F}_r^k$ decreases the reduced cost of y_r by $\bar{c}_q - z_k$, $q \in S_r^k$. The row-generating PSP boils down to finding $r \in R$ and $\mathcal{F}_r^k$ with the minimum reduced cost. If variable q that pairs with r does not satisfy $\bar{c}_q - z_k < 0$, then it cannot exist in $\mathcal{F}_r^k$ since the reduced cost of y_r would be smaller without y_q belonging to S_r^k. With this last observation, we conclude the proof. □