Multi-resource routing with flexible tasks: an application in drayage operations

Karen Smilowitz

Department of Industrial Engineering and Management Sciences

Northwestern University

January 5, 2006

Abstract

This paper introduces an application of a Multi-Resource Routing Problem (MRRP) in drayage operations. Drayage involves the movement of loaded and empty equipment between rail yards, shippers, consignees and equipment yards. The problem of routing and scheduling drayage movements is modeled as an MRRP with flexible tasks, since the origins and destinations of some movements can be chosen from a set of possible nodes. The complexities added by routing choice are studied, along with the impact of these complexities on problem formulation. The solution approach developed to solve this problem includes column generation embedded in a branch and bound framework. Using this approach, efficient operating plans are designed to coordinate independent drayage operations in Chicago.

1 Introduction

This paper studies routing and scheduling for intermodal (rail/truck/maritime) drayage operations. Drayage refers to the regional movement of loaded and empty equipment (trailers and containers) by tractors between rail yards, shippers, consignees and equipment yards. Although drayage represents a small fraction of the total distance of an intermodal shipment, it accounts for a substantial share of shipping costs. Forty percent of a 900 mile movement cost is incurred in the drayage portions, typically less than 50 miles, see Morlok and Spasovic (1994). These costs are exacerbated by the lack of coordination among parties, including shippers, consignees, railroads, trucking companies, and intermodal marketing companies. This research considers coordination of drayage movements to increase efficiency by pooling resources.

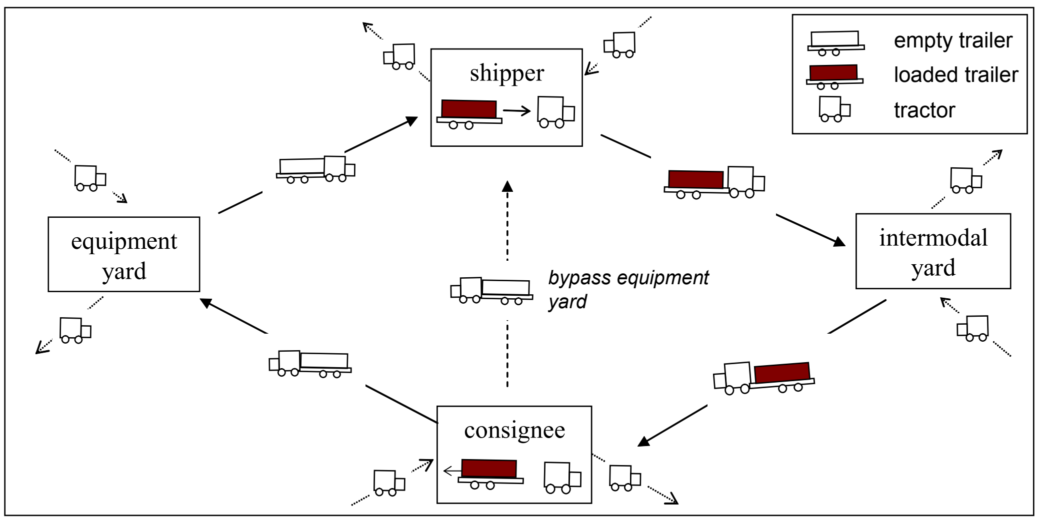


Figure 1: Illustration of drayage operations

Figure 1 illustrates the movement of trailers in intermodal drayage operations. For outbound operations, a tractor bobtails (i.e., a tractor movement without trailer) to an equipment yard to retrieve an empty trailer. The empty trailer is then transported by a tractor to the shipper for loading. Once the trailer is loaded, it is moved, again by a tractor, to an intermodal yard for a train departure. For inbound operations, a tractor bobtails to an intermodal yard to retrieve a loaded trailer that has arrived by rail. The loaded trailer is then delivered by a tractor to the consignee. After the trailer has been unloaded, the empty trailer is repositioned by a tractor to an equipment yard. In addition, cross-town movements reposition loaded trailers between intermodal yards.

If the movements involving the shipper and consignee are controlled by separate entities, there may be no possibility to pool equipment. However, if some degree of coordination is assumed in Figure 1, the empty trailer from the consignee can be redirected to the shipper, bypassing the equipment yard and reducing the empty movements. The choice appears obvious: select the option that reduces inefficiencies. Yet, as shown in this paper, due to the size of the problem, the existence of time windows and the need to model multiple resources, simply identifying these options for savings is challenging and finding a good solution is extremely difficult. This paper develops routing and scheduling models for intermodal drayage operations by solving a generalized version of the problem: the Multi-Resource Routing Problem (MRRP).

This paper presents a new approach to drayage operations. Section 2 generalizes the drayage problem as an MRRP with flexible tasks and highlights the novelty of our approach relative to existing work in this area. Section 3 presents modeling issues, including a new formulation for the MRRP. Section 4 describes the solution methodology for the MRRP. Section 5 applies this methodology to a series of test cases in Chicago. Section 6 summarizes the results and describes future work.

2 Multi-resource routing with flexible tasks

The drayage problem can be modeled as a multi-resource routing problem. Many distribution systems involve the routing of multiple *resources* (tractors, trailers, railcars, drivers, aircraft, crews) to perform a series of *tasks* (transporting goods, repositioning empty equipment). *Resources* are classified by the jobs they perform. For example, trailers are used for storage of items, but do not include a mechanism for locomotion. Tractors, on the other hand, provide locomotion but not storage and require another resource (a driver) to operate. *Tasks* are classified either as well-defined [meaning the origin, destination and time window for the movement are set] or flexible [either the origin or destination of the movement is unspecified]. Tasks are further classified by the resources which are required and the processing time for each resource.

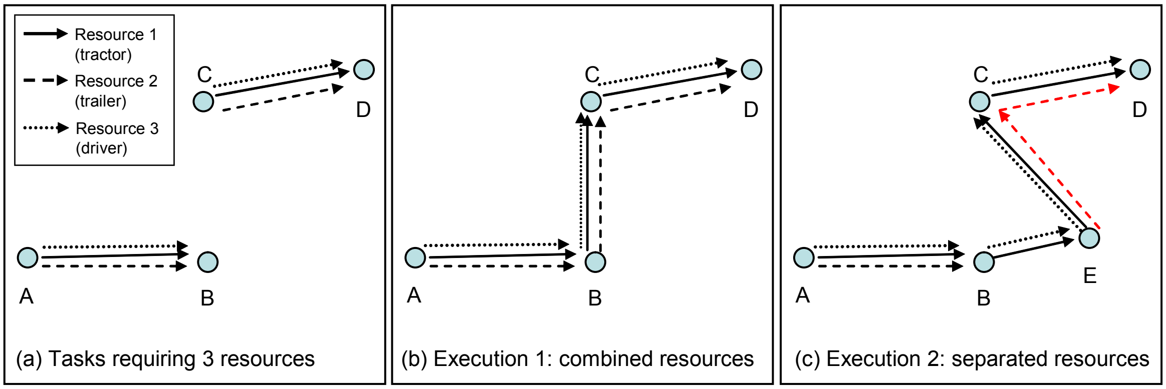


Figure 2: Interaction of tasks and resources

Figure 2(a) shows two well-defined tasks that each require three resources for completion. For example, resource 1 is a tractor; resource 2 is a trailer and resource 3 is a driver. If the processing time at node B is the same for all three resources, it is possible for the same tractor, trailer and driver combination to travel together to complete the second task, as shown in Figure 2(b). However, if the processing times are different, there is an incentive to separate the resources to avoid idle time. In Figure 2(c), the driver and tractor continue to node E to pickup a different trailer to complete the second task while the original trailer is still engaged at B. Due to this incentive, multi-resource routing problems consider various resources as separate, but related, entities. In our modeling of the MRRP, we consider the tractor and driver to be a single resource. The resulting model coordinates the activities of two resources: tractors (and drivers) and trailers.

The MRRP with flexible tasks is defined as:

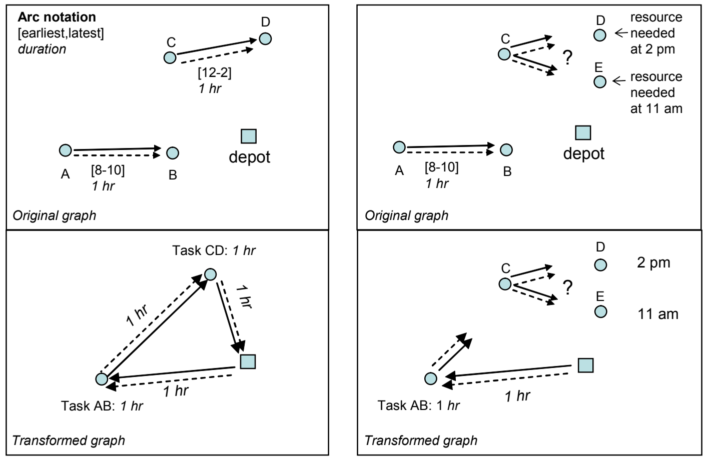
Given: a set of tasks (both well-defined and flexible) with required resources, processing times for resources and time windows; a fleet of each resource type; operating hours at all locations; and a network with travel times.

Find: a set of routes by resource type that satisfy all tasks while meeting a chosen objective function (minimizing fleet size, travel time) and observing operating rules for the tasks and resources.

Developing efficient operational plans is difficult even when all tasks are well-defined. In fact, it is often hard to find a feasible solution (i.e., satisfying all requests in the stated time windows with the available resources). In this case, penalties are introduced to allow service requests to be violated. The MRRP becomes further complicated when some tasks are flexible, as in drayage operations.

Multi-resource routing problems with well-defined tasks have typically been modeled in two ways: (1) as arc-based network flow problems and (2) as node-based vehicle routing problems. Ball *et al.* (1983) develop a network flow formulation for the distribution of trailers for a chemical company. They also transform the problem into a vehicle routing problem (VRP). The origin and destination of a movement are aggregated into a single node that represents the entire movement with all the characteristics of the movement (duration, origin, destination, time windows), see Figure 3(a). The transformed VRP creates tractor tours to serve requested trailer movements. Solution methods for the standard VRP can be applied. Subsequent work on related problems has focused on node-based VRP solution approaches, rather than the computationally intensive arc-based network flow formulations; see, for example, De Meulemeester *et al.* (1997) and Bodin *et al.* (2000). Ioannou *et al.* (2002) model the routing and scheduling of trucks to serve loaded container movements as a multi-Traveling Salesman Problem with Time Windows, a special case of the VRP with time windows.

Multi-resource routing problems with well-defined tasks can also be modeled as pickup and delivery problems with a vehicle capacity of 1 and restrictions on route lengths. A comprehensive review of the general pickup and delivery problem is presented in Savelsbergh and Sol (1995). Of particular interest is the use of set partitioning and column generation techniques for routing



(a) Well-defined tasks (b) Flexible tasks

Figure 3: Node-based representations for multi-resource problems

problems (for examples of applications, see Cullen *et al.* (1981), Dumas *et al.* (1991), Desrochers *et al.* (1992), Savelsbergh and Sol (1998), and Xu *et al.* (2003)). These formulations partition items (tasks to be performed) into disjoint sets (vehicle routes). One reason this approach is attractive for these types of problems is that it transfers the difficult routing constraints on various resources, such as time windows and driver shift constraints, to subproblems.

Applying these approaches to the MRRP with flexible tasks on a large-scale is problematic. Node-based vehicle routing methods are difficult for problems with flexible tasks since tasks which involve a choice of either origin or destination cannot be collapsed into a single node, see Figure 3(b). The task from node C has a flexible destination: a resource must be repositioned, but the destination of the resource can be either node D or node E (referred to as two potential *executions* of flexible task C). With an uncertain destination, this task cannot be collapsed into a single node with specific characteristics.

The MRRP with flexible tasks is further complicated by the presence of time windows. Consider again the flexible task from node C in Figure 3(b). Note that potential destinations D and E have different cut-off times for delivery of the resource. The timing of earlier tasks (for simplicity of illustration, assume only task AB) will determine which destinations are feasible. If task AB is performed at the earliest time (8 am), the resource will be available at node C at 10 am, assuming 1 hour to perform AB and 1 hour to move from B to C. Given a travel time of 1 hour from C to E, the resource can be delivered exactly at 11 am. However, if task AB is performed later, node E is no longer a feasible destination for the flexible task from C. This is a case that arises frequently in drayage operations. An empty trailer may be requested by a shipper by a certain time, without regard for its origin which may be chosen from a set of locations dependent on the completion of previous tasks. When tasks are well-defined, this issue is irrelevant since origins and destinations are specified.

Network flow formulations are well-suited for handling the dependency between tasks, see Powell (2002). However, network flow formulations can be challenging due to problem size. Such a formulation is studied in Morlok and Spasovic (1994) for drayage operations for a single rail carrier. Time is discretized over the planning horizon, resulting in a significant number of variables and constraints, which hinders the application of such a formulation to larger problem instances. The approach in this paper is novel in its use of node-based VRP formulations for the MRRP with flexible tasks.

3 Modeling approach

The MRRP is transformed into a modified Asymmetric Vehicle Routing Problem (AVRP) similar to the way in which Bodin *et al.* (2000) transform the Rollon-Rolloff Vehicle Routing Problem into an AVRP. We introduce a graph $G = (N, A)$ like the transformed graph in Figure 3(b). Nodes $i \in N$ represent all movements (tasks and all possible executions of flexible tasks) and the depot, $i = 0$. Let d_i be the duration of movement $i \in N$. An arc (i, j) between nodes $i, j \in N$ is included in the arc set A if movement j can be performed immediately after movement i. The length of an arc t_{ij} equals the travel time between the destination of movement i and the origin of movement j. Vehicle tours are constructed such that the length of a tour, including movement duration, travel time between movements, and waiting time due to time windows, does not exceed the driver shift. The objective of the AVRP is to minimize the number of tours and the duration of the tours required to visit all nodes in the graph. In the modified AVRP for the MRRP, not all nodes must be visited; only one execution of each flexible task is chosen. We show next how this is incorporated within a set partitioning formulation.

The time dependency between the set of possible executions of a flexible task and other tasks is addressed by placing conservative time windows on all tasks. Recall the simple example in Figure 3(b) with only two tasks. Node E is only a feasible destination for the resource at node C if task AB begins at 8 am. To remove this dependency and to guarantee that all potential executions of a flexible task are feasible, it is assumed that resources are unavailable throughout the entire time window, even if the duration of a task is shorter than its time window. In this case, the resources are considered occupied by task AB from 8 am to 10 am, and the trip from C to E is no longer a feasible execution. While these conservative time windows limit the available choices, they create a disjoint set of executions and a set partitioning formulation can be used to solve the MRRP.

In the MRRP, requested tasks (here, trailer movements) are partitioned into resource (tractor) routes. The tractor routes must comply with the operating rules and tasks must be performed within time windows with the required resources. These difficult constraints on tractor routing and task completion (work hours, time windows on requests) are moved to subproblems. Route generation is discussed in Section 4.4. It is assumed that all requests are known at the time of routing. We assume two resources: one for locomotion (tractor) and one for storage (trailer). Since the tractor and the driver have similar processing times, the driver is linked to the tractor. The following notation is used.

Sets

$\mathcal{T}$: Set of tasks ($\mathcal{T} = \mathcal{T}_w \cup \mathcal{T}_f$) where $\mathcal{T}_w$ = well-defined tasks; $\mathcal{T}_f$ = flexible tasks

$\mathcal{E}_i$: Set of possible executions of flexible task $i \in \mathcal{T}_f$

$\mathcal{M}$: Set of movements [all tasks in $\mathcal{T}$ and all possible executions of flexible tasks]

$\mathcal{R}$: Set of feasible routes

Parameters

c_r : Cost of route $r \in \mathcal{R}$

a_{ri} : Covering parameter: $= \begin{cases} 1 & \text{if movement } i \in \mathcal{M} \text{ is on route } r \in \mathcal{R} \\ 0 & \text{otherwise} \end{cases}$

Decision variables

$$x_r = \begin{cases} 1 & \text{if route } r \in \mathcal{R} \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$$

The set partitioning formulation of the MRRP with well-defined tasks is formulated as follows.

$$\min \sum_{r \in \mathcal{R}} c_r x_r \tag{1a}$$

subject to

$$\sum_{r \in \mathcal{R}} a_{ri} x_r = 1 \qquad \forall i \in \mathcal{T}_w \tag{1b}$$

$$\sum_{r \in \mathcal{R}} \sum_{e \in \mathcal{E}_i} a_{re} x_r = 1 \qquad \forall i \in \mathcal{T}_f \tag{1c}$$

$$x_r \in \{0, 1\} \qquad \forall r \in \mathcal{R} \tag{1d}$$

The objective function (1a) minimizes the fleet size and the travel time of all routes. Therefore, c_r is defined as $c_r = C + \tau_r$ where C is a constant large enough to first minimize vehicles and then travel time, τ_r. Equations (1b) are the partitioning constraints that ensure that all well-defined tasks are executed exactly once. Equations (1c) are the partitioning constraints for all flexible tasks. These constraints are similar to those found in the Multi-dimension, Multiple-Choice Knapsack Problem, see Sinha and Zoltners (1979), and similar solution methods can be applied. Finally, equations (1d) define the binary decision variables for each route.

To generate the set $\mathcal{E}_i$, we define a region of geographic feasibility for executions of a flexible task. A set of potential empty trailer movements that are feasible by time and geographic location for a flexible task from a consignee is shown in Figure 4. The set includes a default equipment yard to ensure problem feasibility. The shaded circle depicts the region of geographic feasibility for the empty trailer available at the consignee. The executions must also be time-feasible. In Figure 4, although shipper S_3 is within the radius of the consignee, it is not a time-feasible option since the resource is not available until after the cut-off time at S_3. The feasible execution from the consignee to shipper S_1 will appear both in the set $\mathcal{E}_i$ where i represents the flexible task requested by the consignee and in the set $\mathcal{E}_j$ where j represents the flexible task requested by the shipper.

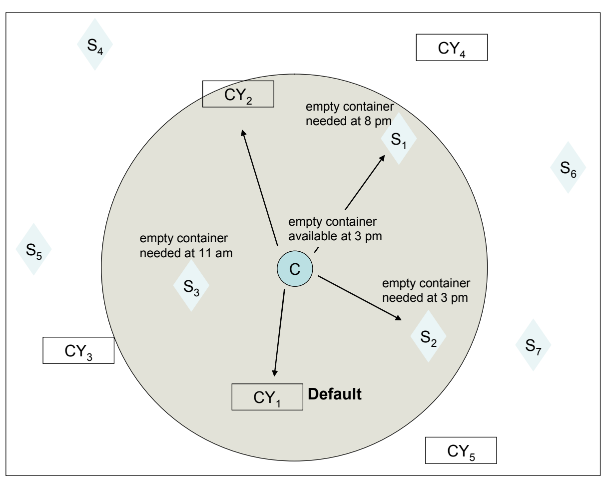


Figure 4: Possible empty movements for a drop and pick consignee

Francis *et al.* (2004) propose an alternative approach to generating the set $\mathcal{E}_i$ which defines a node-specific area of geographic feasibility rather than setting a single region size for all nodes. The application of their approach to the test cases in this paper suggests that due to the node density in Chicago, a fixed radius does not impact solution quality significantly. Allowing for node-specific regions of feasibility can manage problem size efficiently.

4 Solution approach

In this section, we present a branch and bound framework to solve the MRRP which uses column generation to solve the linear relaxation of formulation (1) at the nodes. Section 4.1 describes the general branch and bound approach. Sections 4.2 and 4.3 introduce the upper and lower bounds used within the solution method, respectively. Section 4.4 presents the column generation method used to solve the linear relaxation at each node of the branch and bound tree.

4.1 Branch and bound framework

We use branch and bound to obtain an integer solution to the set partitioning formulation of the MRRP. The solution to the MRRP is obtained by solving the linear relaxation of (1) at each node of the branch and bound tree and then branching on the route variables. At each node, if the solution to (1) is integer, we check for a new upper bound. Otherwise, if the current solution is greater than or equal to the upper bound, then the node is truncated; if not, we branch on the route variable closest to 1. The initial upper bound is obtained with the method described in Section 4.2.

The number of route possibilities for problem instances with many flexible tasks may lead to an exorbitant number of nodes to explore. Therefore, we iteratively relax the truncation criteria within the branch and bound procedure as the number of nodes explored increases. Typically, a node i is truncated if z_i, the solution to linear relaxation of (1), is greater than or equal to z_{UB}, the upper bound. However, we introduce ϵ_i such that a node is truncated if $z_i \geq (1-\epsilon_i)z_{UB}$. At the root node, we set $\epsilon_0 = 0$. At each subsequent node i, ϵ_i slowly increases with the number of nodes and the solution time. As a result, the branch and bound algorithm is run to optimality for small problem instances, but not for larger problem instances.

4.2 Upper bound

Approaches for the VRP with time windows (VRPTW) are used to obtain upper bounds on the MRRP with special consideration for flexible tasks. The method described below is a modified trip insertion heuristic based on a method for the VRPTW with worker shift constraints, see Campbell and Savelsbergh (2004). Let $\mathcal{U}$ be the set of movements not yet assigned to a route, and let $\mathcal{R}$ be the set of routes constructed. An algorithmic representation of the method is shown below:

Step 0:

$\mathcal{U} = \mathcal{M}$ all movements unassigned

$\mathcal{R} = \emptyset$ empty set of routes

Step 1: $\forall j \in \mathcal{U}$:

(1) $\forall r \in \mathcal{R}$: find least-cost, feasible insertion of j into r

(2) $\forall k \in \mathcal{U}$: find least-cost, feasible merger of j and k

Step 2: select best (least-cost) option from Step 1

If selection comes from (1) in Step 1

(a) update r by inserting j: $\mathcal{U} = \mathcal{U} \setminus j$

(b) if $j \in \mathcal{E}_i$ for some $i \in \mathcal{T}_f$ then $\mathcal{U} = \mathcal{U} \setminus m \quad \forall m \in \mathcal{E}_i$

If selection comes from (2) in Step 1

(a) create $\hat{r}$: merger of j & k: $\mathcal{R} = \mathcal{R} \cup \hat{r}$ and $\mathcal{U} = \mathcal{U} \setminus j, k$

(b) if j or $k \in \mathcal{E}_i$ for some $i \in \mathcal{T}_f$ then $\mathcal{U} = \mathcal{U} \setminus m \quad \forall m \in \mathcal{E}_i$

or if $j \in \mathcal{E}_i$ and $k \in \mathcal{E}_l$ for $i, l \in \mathcal{T}_f$ then $\mathcal{U} = \mathcal{U} \setminus m \qquad \forall m \in \mathcal{E}_i \cup \mathcal{E}_l$

Step 3: Repeat steps 1 and 2 while $\mathcal{U} \neq \emptyset$

In step 0, the set of unassigned movements $\mathcal{U}$ is equal to the set of all movements, $\mathcal{M}$. The route set $\mathcal{R}$ is empty. Steps 1 and 2 are repeated while there are still unassigned movements. In Step 1, all options are considered for both inserting unassigned movements into existing routes and merging two unassigned movements to form a new route. Insertions must meet time window requirements and the duration of the resulting route must not exceed the driver work shift L. In Step 2, the best option from Step 1 is executed. In this step, an additional search is needed for the MRRP with flexible tasks to eliminate multiple executions of the same flexible task. As a result, the complexity increases from $O(n^3)$ in Campbell and Savelsbergh (2004) to $O(n^4)$ where n is the number of tasks. The method terminates with a set of feasible routes to complete all tasks.

4.3 Lower bound

A lower bound on solutions to the MRRP is obtained by modifying the minimum trip scheduling approach in Bodin *et al.* (2000) to allow for the existence of time windows and the choice of task executions. This approach first determines $Z(d)$, a lower bound on travel distances for all movements performed, and V, a lower bound on the number of vehicles required. Recall that d_i is the duration of task i. Let $\hat{d}_i$ equal the duration of the shortest (least-time) execution for flexible task $i \in \mathcal{T}_f$. A lower bound on total travel distance is:

$$Z(d) = \sum_{i \in \mathcal{T}_w} d_i + \sum_{i \in \mathcal{T}_f} \hat{d}_i$$

Since each well-defined task, and one execution of each flexible task, must be executed, then summing the minimum durations of these movements is clearly a lower bound on total travel distance. The distance lower bound is divided by the length of the work day, L, to obtain the following lower bound on the number of vehicles:

$$V = \left\lceil \frac{Z(d)}{L} \right\rceil$$

The above lower bounds do not account for repositioning between movements; they are simply bounds on the execution of tasks. Recall the network $G = (N, A)$ described in Section 3. Further recall that the node set N is comprised of the task sets $\mathcal{T}_w$ and $\mathcal{T}_f$, as well as the set of executions for each flexible task, $\mathcal{E}_i, \forall i \in \mathcal{T}_f$. Each node $i \in N$ has an associated subset of nodes, $N^+(i)$, that can be visited after i and a subset $N^-(i)$, that can be visited before i. These subsets are determined by time windows of the movements and the restriction on route length. Note that all subsets include the depot. Let $\hat{t}_{ij}$ be a modified travel time on arc $(i, j) \in A$ that includes the waiting time due to time windows (i.e., the earliest start time at j if the beginning of the time window for movement $j \in \mathcal{M}$ is later than the latest arrival to j from i).

From Bodin *et al.* (2000), we introduce a partial graph $\bar{G} = (\bar{N}, \bar{A})$ of G which corresponds to any feasible solution to the MRRP. The set $\bar{A}$ must have at least $(|\mathcal{T}| + V)$ arcs, and V arcs must start at the depot and V arcs must end at the depot. One arc must start and another must end at each well-defined task and at one execution for each flexible task. The set $\bar{N}$ will consist of $|\mathcal{T}|$ nodes plus the depot.

The problem of determining a set of minimum cost paths on $\bar{G}$ yields a lower bound on the cost of repositioning between movements. Let $y_{ij} = 1$ if arc $(i, j) \in A$ is included in $\bar{A}$ (and consequently nodes i, j are included in $\bar{N}$), and 0 otherwise. We introduce p_i to penalize an execution of a flexible task that is not of minimum duration: $p_i = d_i - \hat{d}_k$ if $i \in \mathcal{E}_k$ for some $k \in \mathcal{T}_f$; $p_i = 0$ otherwise. An execution that is not of minimum duration will be chosen if the reduction in repositioning cost offsets the penalty. The problem formulation is as follows.

$$Z(\bar{G}) = \min \sum_{(i,j) \in A} (\hat{t}_{ij} + p_i) y_{ij} \tag{2a}$$

subject to

$$\sum_{j \in N^+(i)} y_{ij} = 1 \qquad \forall i \in \mathcal{T}_w \tag{2b}$$

$$\sum_{j \in N^-(i)} y_{ji} = 1 \qquad \forall i \in \mathcal{T}_w \tag{2c}$$

$$\sum_{e \in \mathcal{E}_i} \sum_{j \in N^+(e)} y_{ej} = 1 \qquad \forall i \in \mathcal{T}_f \tag{2d}$$

$$\sum_{e \in \mathcal{E}_i} \sum_{j \in N^-(e)} y_{je} = 1 \qquad \forall i \in \mathcal{T}_f \tag{2e}$$

$$\sum_{j \in \mathcal{M}} y_{0j} \geq V \tag{2f}$$

$$\sum_{j \in \mathcal{M}} y_{j0} \geq V \tag{2g}$$

$$y_{ij} \in \{0, 1\} \qquad \forall i, j \in N \tag{2h}$$

The objective function minimizes the cost of connecting movements and additional penalties. Constraints (2b) and (2c) ensure that each well-defined task is assigned to a route. Constraints (2d) and (2e) ensure that one execution in $\mathcal{E}_i$ for each flexible task i is assigned to a route. Constraints (2f) and (2g) ensure that at least V total routes begin and end at the depot.

Let Z_{LB} be a lower bound on the solution to the MRRP: $Z_{LB} = Z(d) + Z(\bar{G})$. The lower bound on vehicles is updated as follows:

$$V = \left\lceil \frac{Z_{LB}}{L} \right\rceil$$

Formulation (2) is solved again with the updated value of V in constraints (2f) and (2g). The lower bounds Z_{LB} and V are updated, and formulation (2) is resolved until V does not change from one iteration to the next.

Formulation (2) is a lower bound on the MRRP as formulated in (1) because, like the MRRP, all tasks must be visited during time windows and exactly one execution of each flexible task is chosen. However, the MRRP ensures that the length of *each* route does not exceed the shift limit; formulation (2) ensures that *on average* each route does not exceed the shift limit.

4.4 Column generation

At each node of the branch and bound tree, column generation is used to solve the linear relaxation of (1). At the root node, we begin with a feasible subset of routes $\mathcal{R}'$. This subset can be found as follows. Each task is assigned to its own route. For flexible tasks, a default execution is chosen. For example, in the drayage operation, the default equipment yard is used. Penalties are added if resulting routes are longer than driver work shifts. The penalties are designated such that the model always finds feasible solutions if they exist. However, some test cases in Section 5 include fixed tasks with durations greater than L or flexible tasks for which all executions (including the default yard) exceed the time limit. In these cases, the final solution includes a route that exceeds the driver work shift.[1] Alternatively, one could begin with a set of routes generated with the upper bound method introduced in Section 4.2, after adapting the method to allow for well-defined tasks with duration greater than L. Given a route subset $\mathcal{R}'$, the following is solved.

$$\min \sum_{r \in \mathcal{R}'} c_r x_r \tag{3a}$$

subject to

$$\sum_{r \in \mathcal{R}'} a_{ri} x_r = 1 \qquad \forall i \in \mathcal{T}_w \tag{3b}$$

$$\sum_{r \in \mathcal{R}'} \sum_{e \in \mathcal{E}_i} a_{re} x_r = 1 \qquad \forall i \in \mathcal{T}_f \tag{3c}$$

$$0 \le x_r \le 1 \qquad \forall r \in \mathcal{R}' \tag{3d}$$

[1]Results in Section 5 present the true cost without penalties.

New routes are added iteratively to $\mathcal{R}'$ if the addition of such routes could improve the objective function. A modified route cost is defined by the dual variable π_i of the covering constraint for task i; (3b) for well-defined tasks and (3c) for flexible tasks. This cost corresponds to the reduced cost of the candidate routes:

$$\bar{c}_r = c_r - \sum_{i \in \mathcal{M}} a_{ri} \pi_i.$$

At each iteration, routes are generated using methods described below. A candidate route r is added to R' only if $\bar{c}_r$ is negative and if its duration does not exceed L. We control the number of routes added to the master problem at each iteration using the following column dominance rule. A route with a negative reduced cost is added only if no other candidate route at that iteration satisfies the same fixed and flexible tasks (non-zero elements in the column) and has a lower actual cost. If new columns exist, problem (3) is resolved and the dual variables are updated, otherwise we conclude that the linear relaxation has been solved at that node.

Formulation (3) can be strengthened with the lower bound on vehicles determined in Section 4.3.

$$\sum_{r \in \mathcal{R}'} x_r \geq V \tag{3e}$$

Constraint (3e) requires at least V vehicles in the relaxed solution.

The pricing problem to generate new columns can be solved as an elementary shortest path problems with resource constraints (ESPPRC) which is NP-Hard, see Dror (1994). Desrochers *et al.* (1992) introduce a label-correcting dynamic programming algorithm to solve the constrained shortest path. The label-correcting dynamic programming algorithm for the MRRP creates paths beginning and ending at the depot that visit each fixed task at most once and visit at most one execution of each flexible task. Further, the duration of each path can not exceed L and time windows must be observed.

Irnich and Villeneuve (2003) and Feillet *et al.* (2004) highlight the difficulties encountered when attempting to find only elementary path in cases such as the MRRP when nodes may only be visited once on a route. These difficulties are compounded with flexible tasks since a vehicle may not visit more than one execution of the same flexible task on a route. As noted in Feillet *et al.* (2004), while many applications of the ESPPRC simply relax the elementary path constraints and employ standard label correcting methods, there are some cases in which this approach yields poor solutions. The MRRP is such a case since a node cannot be visited more than once on a route, and nodes that serve the same flexible task cannot be on the same route. Feillet *et al.* (2004) show how data requirements increase significantly when using a label-correcting approach to the ESPPRC. The sets of non-dominated paths that must be maintained throughout the algorithm can grow prohibitively large in the case of the MRRP with flexible tasks. As discussed in the next section, these data requirements limit the use of the label-correcting method to small instances of the MRRP with flexible tasks.

Alternatively we propose the use of the greedy trip insertion heuristic from Section 4.2 to generate new routes. Further, since each solution obtained with the greedy trip insertion is feasible, we can also search for improved upper bounds throughout column generation. As with the label-correcting method, only routes with negative reduced costs are added to the master problem.

In the computational tests presented in Section 5 we compare the following two methods. As shown in Savelsbergh and Sol (1998), approximation methods may be combined with exact methods to significantly reduce the time required to obtain optimal solutions to routing problems. This leads to the first method for the MRRP: the heuristic method is employed first to check for new columns. If no new columns are found, the label-correcting method is run. Alternatively, in the second method, only the heuristic method is used to generate new columns. In this second case, the parameter ϵ_i is employed to reduce the number of branch and bound nodes explored.

5 Computational study

The solution approaches for the MRRP with flexible tasks are applied to drayage operations in Chicago. Section 5.1 presents the test cases. Section 5.2 discusses the performance of the solution methods in terms of solution quality and computational effort. Section 5.3 discusses the operational savings from coordination of drayage operations.

5.1 Test cases

Data are provided by local Chicago drayage companies and third party logistics companies, (Dahnke (2003); Corinescu (2003); Grosz (2003)). The region under consideration, shown in Figure 5, covers greater Chicagoland and parts of central Illinois, southern Wisconsin and western Indiana. The figure identifies the intermodal yards, but the individual consignees and shippers remain anonymous. It is assumed that equipment yards are located at intermodal yards. It is further assumed that all tractor routes begin and end at the same depot. The model captures one day of operation, assuming the loads for the day are known when decisions must be made. It is assumed that drivers work a continuous ten hour work shift. Table 1 indicates the operating parameters used with all test cases.

Table 2 presents two groups of test cases from industry. For groups A and B, the instances labeled by numbers represent independent drayage companies and the coordinated instances (ACrd and BCrd) represent the same system with the independent drayage companies operating under a single dispatcher. The columns list the number of well-defined tasks; the number of flexible tasks; and the number of possible executions generated for flexible tasks, given the full 95 nodes in the network shown in Figure 5, and a reduced set of 35 nodes. The reduced set is used in Section 5.2 to demonstrate the ability of the label-correcting method to solve small instances of the MRRP with flexible tasks. The test cases are small compared with the total number of movements in Chicago

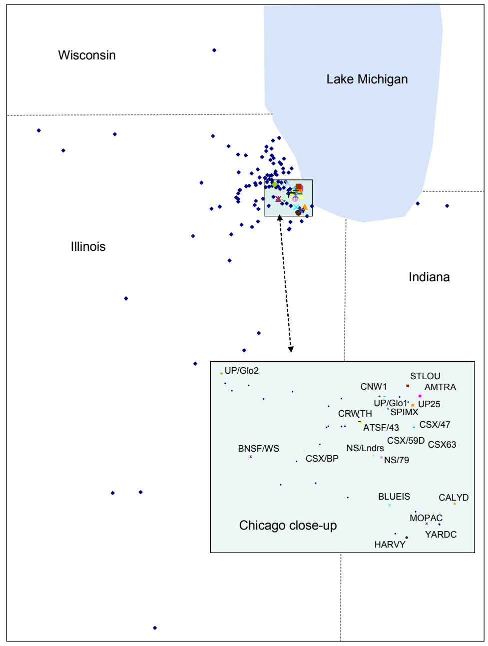


Figure 5: Pick up and delivery locations in the Chicago region. *Intermodal yards are labeled by rail carrier; shippers and consignees are shown as diamonds*

performed each day; however, they represent the largest in our current test bed from industry. Therefore, a second set of test cases, randomly generated based on the nodes in Figure 5 and the operating parameters in Table 1, is used to study the application of the solution methodology to larger problems instances. These test cases are shown in Table 3.

5.2 Solution method performance

All solution methods are implemented using C with the CPLEX Callable Library Interface and a CPLEX 8.1 solver, running on a Sun Fire v250 1.28-GHz UltraSPARC IIIi computer with two processors.

Use of the label-correcting method to solve the MRRP is limited to the small test cases due to the computational requirements described in Section 4.4. The label-correcting method found optimal solutions for 15 test cases of 50 total test cases in under ten hours. For the other test cases, the solution approach reached the 10-hour time limit while solving the pricing problem for the first time at the root node.

Table 4 presents the statistics on computational effort for the test cases solved to optimality with the label-correcting method. The first column lists the rows in the master problem and the second column lists the final column count in the master problem. The third and fourth columns list the solution time for the entire algorithm and the number of branch and bound nodes explored, respectively. Optimal solutions for many of the smaller instances are found at the root node. The numbers of nodes for most remaining instances are relatively low.

Table 5 presents the optimal solutions for these test cases. Table 5 presents (i) the number of tractors ("fleet") and the total travel time ("Travel time") in the optimal solution, (ii) the lower bounds on fleet and travel time with optimality gaps in parenthesis, and (iii) the fleet and travel time obtained with the heuristic approach, again with optimality gaps in parenthesis.

The optimal solutions are used to assess the strength of the lower bounds and the quality of heuristic solutions. The results in Table 5 suggest that the heuristic gaps are tighter than those for the lower bound. The heuristic performs well for all test cases, obtaining optimal fleet sizes in 9 of the test cases. Optimal travel times are found with the heuristic method for 6 of the test cases, and the gap is within 22% for the remaining test cases.

Tables 6 and 7 present the lower bounds and heuristic solutions, and the gap between the two, for the remaining industry test cases and the randomly generated test cases, respectively. These problem instances cannot be solved with the label-correcting method; in fact, in all cases the time limit is reached in the first iteration. The gaps between the lower bounds and heuristic solutions range from 0% to 28% for the randomly generated problems with average gaps of 20% for fleet size and 13% for travel time. However, the results in Table 5 suggest that the heuristic solutions may be closer to the optimal solutions than the lower bounds. Further, these results improve on earlier results obtained with a simple heuristic in Neuman and Smilowitz (2002) based on the well-known savings algorithm (Clarke and Wright (1964)).

Tables 8 and 9 report statistics on the computational effort required for the heuristic and lower bounds. Columns 1-3 list the problem sizes (numbers of rows and columns) and solution times for the lower bounds. The number of rows is equal to $2|\mathcal{T}| + 2$. The number of columns is equal to the number of feasible arcs between tasks (the set A), which can grow significantly large as the number of tasks increase. Despite the large number of columns, the lower bounds can be obtained quickly (under a minute for all but one test case). The remaining columns in Tables 8 and 9 report statistics for the heuristic method, including the master problem rows and columns, solution times, branch and bound nodes, and the final value of ϵ. In the master problem, the number of rows is equal to the total number of tasks in $\mathcal{T}$ plus 1 when constraint (3e) is added. Initially, the number of columns is equal to number of tasks since the original feasible solution assigns each task to its own route. The number of columns added at each iteration is at most equal to the total number of tasks; however, this number is significantly lower in practice. Solutions are found at the root node in 5 of the industry test cases and 1 of the randomly generated test cases. The addition of the parameter ϵ is useful in controlling solution times. In most test cases, the final $\epsilon = 0$; however, increases in ϵ are observed for a few problems with high solution times, which can be reduced by increasing ϵ more rapidly.

5.3 Case study: Chicago drayage operations

Individual operations of multiple entities are compared with a hypothetical system in which all entities are controlled by a single dispatcher to test the benefits of coordination. Such coordinated systems have the potential to address many issues raised by the recently created Intermodal Advisory Task Force (IATF) in Chicago.[2] This task force is considering means to measure and alleviate problems caused by the large number of drayage movements in Chicago. A model that can quantify the benefits of coordination is a critical tool in the process of gaining support for a coordinated system.

As shown in the previous section, the only test group in which all instances could be solved to optimality is Group A with 35 nodes. For Group A with 95 nodes and Group B with 35 nodes, all instances are solved with the heuristic. The results in Figure 6 represent solutions obtained with the best solution method possible for that test group. Although we cannot guarantee optimality for the solutions found with the heuristic method, we can obtain good feasible solutions. These solutions can then be used to measure the efficiencies possible with coordination. Efficiency is measured by reduced deadhead travel which can lead to higher productivity rates for drivers and decreased congestion on city streets, which are key goals of the IATF.

Recall that test cases ACrd and BCrd represent coordination of dray companies A1-A6 and B1-B8, respectively. The results in Section 5.2 show that coordination reduces the fleet size: from 33 total vehicles to 30 vehicles in ACrd_{35}; from 13 to 9 in ACrd_{95}; and from 374 to 345 in BCrd_{35}. The total travel time decreases as well: from 284 total hours to 269 total hours in ACrd_{35}; from 94 to 76 total hours in ACrd_{95}; and 3,367 to 3,343 in BCrd_{35}. Figure 6 depicts other measures of improved efficiency: movements per vehicle run ("moves/run") and travel time per vehicle run ("time/run"). In the figure, the statistics for individual dray companies (diamonds and squares) are compared with statistics for coordinated operations (lines) for all groups. By reducing the number of tractors needed in the coordinated systems, the number of movements per run is higher in coordinated systems.

[2]See United States Department of Transportation - Federal Highway Administration (1996) for details.

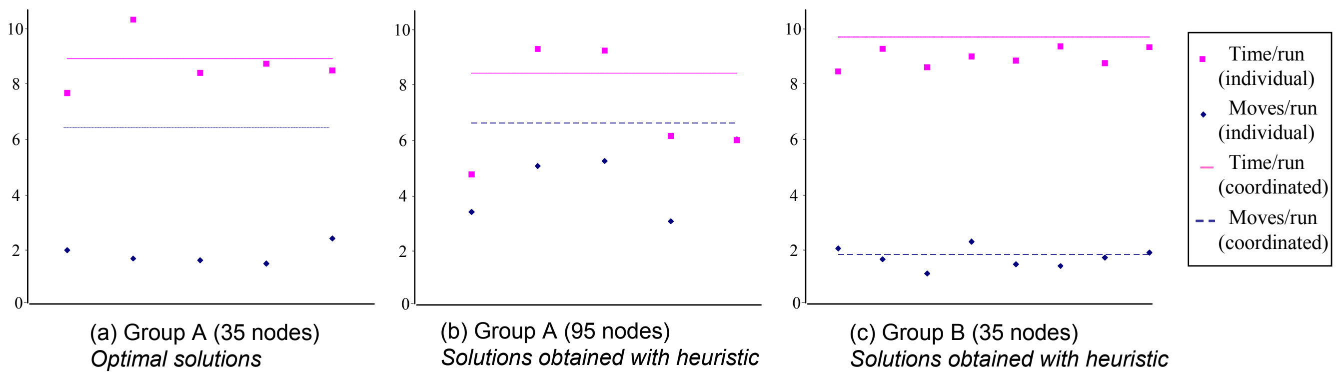


Figure 6: The impact of coordination on efficiency measures.

6 Conclusions

It is shown in this paper that coordinating drayage activities of multiple parties can lead to increased overall system productivity. These results are valuable for congested urban areas with significant drayage activities such as Chicago. At the same time, this paper demonstrates the challenges of identifying opportunities from coordination and solving large problem instances. Advances are made in both areas. Intermodal drayage operations with empty repositioning choice are modeled as a multi-resource routing problem (MRRP) with flexible tasks. A new formulation and solution method are developed for the MRRP with flexible tasks. The heuristic solution method performs well relative to the label-correcting method that yields optimal solutions.

Future work in modeling should include the consideration of additional resources (most importantly, container and chassis). Further, relaxing the assumption of static requests is an important extension to this work. According to discussions with industry, sixty percent of drayage requests are known before the day of operation and the remaining forty percent become known over the day.

References

Ball, M., Golden, B., Assad, A., and Bodin, L. (1983). Planning for truck fleet size in the presence of common-carrier options. *Decision Sciences*, **14**(1), 103–120.

Bodin, L., Mingozzi, A., Baldacci, R., and Ball, M. (2000). The roll-on, roll-off vehicle routing problem. *Transportation Science*, **34**(3), 271–288.

Campbell, A. and Savelsbergh, M. (2004). Efficiently handling practical complexities in insertion heuristics. *Transportation Science*, **38**(3), 369–378.

Clarke, G. and Wright, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, **12**, 568–581.

Corinescu, E. (2003). Drayage data files from 2002. Hub City Terminals, Inc.

Cullen, F., Jarvis, J., and Ratliff, H. (1981). Set partitioning based heuristics for interactive routing. *Networks*, **11**, 125–143.

Dahnke, B. (2003). Drayage data files from 1998. Laser Trucking, personal communication.

De Meulemeester, L., Laporte, G., Louveaux, F., and Semet, F. (1997). Optimal sequencing of skip collections. *Journal of Operations Research Society*, **48**, 57–64.

Desrochers, M., Desrosiers, J., and Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, **40**(2), 342–354.

Dror, M. (1994). Note on the complexity of the shortest path problem with resource constraints for column generation in VRPTW. *Operations Research*, **42**, 977–978.

Dumas, Y., Desrosiers, J., and Soumis, F. (1991). The pickup and delivery problem with time windows. *European Journal of Operations Research*, **54**, 7–22.

Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004). An exact algorithm for elementary shortest path problems with resource constraints: application to some vehicle routing problems. *Networks*, **44**(3), 216–229.

Francis, P., Smilowitz, K., and Zhang, G. (2004). Improved modeling and solution methods for the multi-resource routing problem. Working paper 04-003, Northwestern University.

Grosz, J. (2003). Personal communications. Cushing Transportation , Inc.

Ioannou, P., Chassiakos, A., Jula, H., and Unglaub, R. (2002). Dynamic optimization of cargo movement by trucks in metropolitan areas with adjacent ports. Research report, University of Southern California, METRANS Transportation Center.

Irnich, S. and Villeneuve, D. (2003). The shortest path problem with resource constraints and k-cycle elimination for $k \geq 3$. Technical Report Les Cahiers du GERAD G-2003-55, École des Hautes Études Commerciales, Montréal.

Morlok, E. K. and Spasovic, L. N. (1994). Approaches for improving drayage in rail-truck intermodal service. Research report, University of Pennsylvania.

Neuman, C. and Smilowitz, K. (2002). Strategies for coordinated drayage movements. In P. Mirchandani and R. Hall, editors, *NSF Real Time Logistics Workshop*, Long Beach, CA.

Powell, W. (2002). Dynamic models of transportation operations. In S. Graves and T. A. G. de Tok, editors, *Handbooks in Operations Research, Volume on Supply Chain Management*. North Holland.

Savelsbergh, M. and Sol, M. (1995). The general pickup and delivery problem. *Transportation Science*, **29**, 17–29.

Savelsbergh, M. and Sol, M. (1998). DRIVE: Dynamic routing of independent vehicles. *Operations Research*, **46**, 474–490.

Sinha, P. and Zoltners, A. (1979). The multiple choice knapsack problem. *Operations Research*, **27**, 503–515.

United States Department of Transportation - Federal Highway Administration (1996). Case study: Improving mobility in the chicago region. *http://www.fhwa.dot.gov/freightplanning/chicago.html*.

Xu, H., Chen, Z.-L., Rajagopal, S., and Arunapuram, S. (2003). Solving a practical pickup and delivery problem. *Transportation Science*, **37**(3), 347– 364.

Acknowledgment

The author would like to thank the Associate Editor and the anonymous reviewers for their valuable comments and Peter Francis and Guangming Zhang for help with the computational work. This research has been supported by grant DMI-0348622 from the National Science Foundation.

Parameter	Value
Maximum radius for empty trailer movements	2 miles
Time to pick up loaded trailer	30 minutes
Time to drop off loaded trailer	30 minutes
Time to pick up empty trailer	15 minutes
Time to drop off empty trailer	15 minutes
Time to load trailer	1 hour
Time to unload trailer	1 hour
Driver work shift	10 hours (continuous)

Table 1: Operating parameters.

Test case	Well-defined tasks	Flexible tasks	Potential executions (95)	Potential executions (35)
Group A				
A1	5	5	85	16
A2	3	2	34	7
A3	16	10	164	26
A4	4	2	32	4
A5	7	5	83	17
ACrd	35	24	406	74
Group B				
B1	35	28	491	66
B2	88	67	1,089	139
B3	23	19	293	27
B4	42	29	495	83
B5	37	26	435	59
B6	31	26	391	45
B7	52	41	717	104
B8	44	33	536	81
BCrd	351	270	5,083	705

Table 2: Test cases from industry

Test case	Well-defined tasks	Flexible tasks	Potential executions
1	100	65	923
2	100	57	819
3	100	48	660
4	100	51	768
5	125	75	1147
6	125	64	908
7	125	66	897
8	125	70	1014
9	150	81	1159
10	150	76	1090
11	150	78	1124
12	150	79	1258
13	175	94	1378
14	175	92	1362
15	175	88	1199
16	175	80	1188
17	200	96	1316
18	200	107	1678
19	200	90	1310
20	200	107	1331

Table 3: Test cases: randomly generated cases

Case	Rows	Columns	Solution time	B&B nodes
$A1_{35}$	11	45	0.0	1
$A2_{35}$	6	5	0.0	1
$A3_{35}$	30	73	0.0	3
$A4_{35}$	7	8	0.0	1
$A5_{35}$	13	207	0.5	3603
$ACrd_{35}$	63	677	0.0	1
$B1_{35}$	66	5675	0.5	1
$B3_{35}$	43	59	0.0	1
$B4_{35}$	74	27739	13.2	19
$B5_{35}$	63	672	0.0	3
$B6_{35}$	58	123	0.0	1
$B8_{35}$	78	2288	0.1	1
$A1_{95}$	11	205	2.1	1
$A2_{95}$	6	16	0.0	1
$A4_{95}$	7	18	0.0	1

Table 4: Label-correcting method computational statistics; solution times in minutes

	Optimal solution		Lower bound		Heuristic solution	
test case	fleet	travel time	fleet	travel time	fleet	travel time
$A1_{35}$	5	38.7	4 (20%)	36.2 (6%)	5 (0%)	45.6 (18%)
$A2_{35}$	3	31.6	3 (0%)	30.2 (4%)	3 (0%)	31.6 (0%)
$A3_{35}$	16	135.6	11 (31%)	108.3 (20%)	18 (13%)	147.5 (9%)
$A4_{35}$	4	35.2	4 (0%)	30.6 (13%)	4 (0%)	36.4 (3%)
$A5_{35}$	5	42.8	4 (20%)	31.5 (26%)	5 (0%)	46.6 (9%)
$ACrd_{35}$	30	269.2	21 (30%)	208.8 (22%)	34 (13%)	295.7 (10%)
		Average gap	(17%)	(15%)	(4%)	(8%)
$B1_{35}$	23	214.3	17 (26%)	160.6 (25%)	31 (35%)	262.1 (22%)
$B3_{35}$	37	317.9	27 (27%)	263.9 (17%)	37 (0%)	317.9 (0%)
$B4_{35}$	25	237.8	18 (28%)	174.6 (27%)	31 (24%)	278.8 (17%)
$B5_{35}$	39	356.3	27 (31%)	269.3 (24%)	43 (10%)	380.3 (7%)
$B6_{35}$	41	383.1	30 (27%)	294.5 (23%)	41 (0%)	383.2 (0%)
$B8_{35}$	41	381.2	30 (27%)	292.9 (23%)	41 (0%)	383.1 (0%)
		Average gap	(28%)	(23%)	(12%)	(8%)
$A1_{95}$	2	13.8	1 (50%)	7.1 (49%)	3 (50%)	14.1 (2%)
$A2_{95}$	1	9.3	1 (0%)	8.3 (11%)	1 (0%)	9.3 (0%)
$A4_{95}$	2	12.2	2 (0%)	11 (10%)	2 (0%)	12.2 (0%)
		Average gap	(17%)	(23%)	(17%)	(1%)

Table 5: Comparison of solution method performance: Gap from optimal listed in parentheses

	Lower bound		Heuristic solution	
test case	fleet	travel time	fleet	travel time
$A1_{95}$	1	7.1	3 (67%)	14.1 (50%)
$A2_{95}$	1	8.3	1 (0%)	9.3 (11%)
$A3_{95}$	4	37.8	5 (20%)	46.2 (18%)
$A4_{95}$	2	11	2 (0%)	12.2 (10%)
$A5_{95}$	2	11.6	2 (0%)	11.9 (3%)
$ACrd_{95}$	8	71	9 (11%)	75.6 (6%)
		Average gap	(16%)	(16%)

$B1_{95}$	6	57.2	8 (25%)	65.5 (13%)
$B2_{95}$	22	217.1	25 (12%)	236.5 (8%)
$B3_{95}$	7	66	9 (22%)	79.7 (17%)
$B4_{95}$	10	95.2	12 (17%)	104.2 (9%)
$B5_{95}$	8	70.5	9 (11%)	83.1 (15%)
$B6_{95}$	9	87.6	12 (25%)	118.1 (26%)
$B7_{95}$	10	93.4	12 (17%)	110.9 (16%)
$B8_{95}$	11	108.3	13 (15%)	130 (17%)
$BCrd_{95}$	76	759.7		
		Average gap	(18%)	(15%)

Table 6: Solution method performance: Gap between bounds listed in parentheses

	Lower bound		Heuristic solution	
Case	fleet	travel time	fleet	travel time
1	21	204	25 (19%)	205 (0%)
2	19	182.8	24 (26%)	192.5 (5%)
3	22	212.2	22 (0%)	216.2 (2%)
4	18	176.2	23 (28%)	199.5 (13%)
5	23	226.5	28 (22%)	257.7 (14%)
6	25	247.9	32 (28%)	293 (18%)
7	29	280.6	32 (10%)	326.3 (16%)
8	30	294.7	33 (10%)	338.7 (15%)
9	33	326.5	38 (15%)	364.5 (12%)
10	30	298.3	37 (23%)	341.4 (14%)
11	30	296.7	37 (23%)	350.2 (18%)
12	25	249.2	30 (20%)	282.8 (13%)
13	38	375.6	43 (13%)	401.4 (7%)
14	31	307.6	37 (19%)	334.6 (9%)
15	39	385.8	47 (21%)	440.8 (14%)
16	33	327.7	40 (21%)	367.9 (12%)
17	45	444	57 (27%)	547.5 (23%)
18	37	368.6	46 (24%)	417.5 (13%)
19	38	378	47 (24%)	454.4 (20%)
20	39	389.4	48 (23%)	455.7 (17%)
		Average gap	(20%)	(13%)

Table 7: Solution method performance: Gap between bounds listed in parentheses

Test case	Lower bound Rows	Columns	Solution time	Heuristic solution Rows	Columns	Solution time	B&B nodes	Final ϵ
$A1_{95}$	22	7,484	0	11	13	0.0	1	0
$A2_{95}$	12	1,299	0	6	8	0.0	1	0
$A3_{95}$	54	25,195	0	27	31	0.0	1	0
$A4_{95}$	14	1,192	0	7	6	0.0	1	0
$A5_{95}$	26	6,052	0	13	266	0.3	1413	0
$ACrd_{95}$	120	141,460	0	60	5,931	226.6	31,143	0.07
$B1_{95}$	128	199,868	0	64	9,002	273.9	28,900	0.08
$B2_{95}$	312	1,014,837	0	156	1,083	34.2	131	0
$B3_{95}$	86	79,753	0	43	1,168	136.9	23,700	0.04
$B4_{95}$	144	208,087	0	72	603	2.9	161	0
$B5_{95}$	128	163,974	0	64	2,412	197.3	21,300	0.06
$B6_{95}$	116	134,356	0	58	58	0.0	1	0
$B7_{95}$	188	451,753	0	94	9,141	274.2	10,841	0.08
$B8_{95}$	156	237,223	0	78	413	2.4	117	0
$BCrd_{95}$	1240	19,126,798	38	620				

Table 8: Solution method performance: computational effort; solution times in minutes

Test case	Lower bound Rows	Lower bound Columns	Lower bound Solution time	Heuristic solution Rows	Heuristic solution Columns	Heuristic solution Solution time	Heuristic solution B&B nodes	Heuristic solution Final ϵ
1	332	695,867	0	166	676	12.1	119	0
2	316	564,859	0	158	169	0.2	1	0
3	298	411,912	0	149	918	8.6	113	0
4	304	516,273	0	152	1,923	888.8	14,165	0.27
5	402	1,125,417	0	201	1,078	28.5	121	0
6	380	718,763	0	190	1,045	21.2	123	0
7	384	712,820	0	192	921	21.1	123	0
8	392	874,183	0	196	1,059	23.2	123	0
9	464	1,175,534	0	232	1,230	41.7	129	0
10	454	1,034,771	0	227	1,236	42.2	129	0
11	458	1,106,768	0	229	1,324	37.1	127	0
12	460	1,353,662	0	230	1,267	54.4	127	0
13	540	1,656,916	0	270	1,460	80.9	133	0.01
14	536	1,630,583	0	268	1,535	128.7	133	0.01
15	528	1,286,917	0	264	1,331	55.4	131	0
16	512	1,320,903	0	256	1,676	72.2	133	0.01
17	594	1,553,397	0	297	1,810	105.6	141	0.01
18	616	2,361,260	0	308	1,690	91.1	131	0.01
19	582	1,554,966	0	291	1,800	143.1	137	0.01
20	586	1,649,184	0	293	1,972	146.7	131	0.01

Table 9: Solution method performance: computational effort; solution times in minutes