Discrete Optimization

Vehicle routing and crew scheduling for metropolitan mail distribution at Australia Post

B.L. Hollis a,b,*, M.A. Forbes b, B.E. Douglas c

a Department of Mathematics, The University of Queensland, Brisbane, Qld 4072, Australia
b Opcom Pty Ltd, PO Box 149, Toowong, Qld 4066, Australia
c Network Planning and Design, Australia Post, 171521 Exhibition St, Melbourne 3000, Australia

Received 28 June 2004; accepted 4 January 2005
Available online 19 February 2005

Abstract

This paper presents a new multi-depot combined vehicle and crew scheduling algorithm, and uses it, in conjunction with a heuristic vehicle routing algorithm, to solve the intra-city mail distribution problem faced by Australia Post.

First we describe the Australia Post mail distribution problem and outline the heuristic vehicle routing algorithm used to find vehicle routes. We present a new multi-depot combined vehicle and crew scheduling algorithm based on set covering with column generation. The paper concludes with a computational investigation examining the affect of different types of vehicle routing solutions on the vehicle and crew scheduling solution, comparing the different levels of integration possible with the new vehicle and crew scheduling algorithm and comparing the results of sequential versus simultaneous vehicle and crew scheduling, using real life data for Australia Post distribution networks.

Keywords: Integer programming; Vehicle routing; Vehicle scheduling; Crew scheduling

1. Introduction

Vehicle routing and crew scheduling are part of many general distribution problems commonly faced by freight distribution and postal organisations. The traditional planning process used to find solutions to problems of this type typically involves constructing a set of vehicle routes then building a set of crew schedules to cover all vehicle routes and assigning vehicles to a set of vehicle routes to produce vehicle schedules.

The main contribution of this paper is a new mathematical formulation integrating the vehicle and crew scheduling stages of the planning process which is solved using an algorithm based on set covering with column generation.
Section 2 reviews the literature on integrated vehicle and crew scheduling. Section 3 gives a description of the intra-city vehicle routing and crew scheduling problem faced by Australia Post. Section 4 describes the heuristic vehicle routing algorithm used to construct vehicle routes. Section 5 describes the new vehicle and crew scheduling algorithm used to create vehicle and crew schedules. Section 6 undertakes a computational investigation examining the affect of different types of vehicle routing solutions on the vehicle and crew scheduling solution, and compares the different levels of integration possible with the new vehicle and crew scheduling algorithm, using data for the Australia Post Melbourne metropolitan mail distribution problem.

2. Literature review

Freling et al. (2003) and Haase et al. (2001) provide a good overview of methods for partially integrating vehicle and crew scheduling for the related problem of mass transit crew scheduling.

Full integration of vehicle and crew scheduling, so called simultaneous vehicle crew scheduling (referred to as either the single or multiple depot simultaneous vehicle crew scheduling problem—SDSV CSP, MDSVCSP, respectively) has only recently been considered in the literature. There are some papers that consider the SD SVCSP though relatively few that consider the MDSVCSP.

Haase and Friberg (1999) present an exact algorithm for the SD SVCSP using set partitioning with column generation, however, only very small problems were able to be solved (<30 trips). Haase et al. (2001) and Freling et al. (2003) consider the SD SVCSP in urban mass transit with a homogeneous vehicle fleet and two crew types where trip start and end times are fixed by a timetable. The approach presented by Haase et al. (2001) uses set partitioning with column generation and was used to solve problems with up to 400 tasks optimally and 700 tasks heuristically. Freling et al. (2003) use Lagrangian relaxation with column generation to find solutions for a set partitioning based model and solve problems with up to 238 tasks.

Huisman et al. (2003) extend the work of Freling et al. (2003) to the multiple depot case. They present two similar formulations for the MDSVCSP, incorporating variables for both crew schedules and vehicle arcs. Problems with up to 650 trips were solved using Lagrangian relaxation with column generation with the extra restrictions that: drivers are only allowed to operate vehicles stationed at their home depot; a maximum of only one vehicle change is permitted in a crew schedule, significantly simplifying the column generation sub algorithm; and not all trips can be driven by a vehicle operated out of any depot.

Klabjan et al. (2002) and Cordeau et al. (2001) solve a similar problem for aircraft routing and crew scheduling. Airline planning consists of several stages. The first stage is called Schedule Planning, deciding when and where to fly, the next stage is called Fleet Assignment where aircraft types are assigned to the schedule. The next stage involves finding a feasible set of aircraft routes, and the final stage is the crew scheduling stage which involves finding crew schedules and pairings, where a pairing is a sequence of crew schedules starting and ending at the same base and spanning less than a week.

For the problem considered by Klabjan et al. (2002) the Fleet Assignment stage of the planning process has been completed, dictating the number of aircraft on the ground at any one time. The crew scheduling formulation they present includes aircraft counting constraints to ensure that the crew schedules they develop adhere to the number of aircraft available, and in addition, allow small changes in the flight schedule in order to find better crew scheduling solutions.

Similarly, for the problem considered by Cordeau et al. (2001) the Fleet Assignment stage of the planning process has been completed, so that the type of aircraft assigned to each flight leg is known. They use Benders decomposition with column generation to solve problems where crew are only allowed to fly a single type of aircraft meaning that the problem decomposes into separate problems for each aircraft type. Problems with up to 500 flight legs were solved using this technique.

The simultaneous vehicle crew scheduling formulation we present is new in many respects: it is
for multiple depots and multiple vehicle types; includes both vehicle and driver fixed and variable costs allowing the numbers of each to be traded off directly; does not assume that vehicle types have been assigned to the set of routes to be covered; allows for an arbitrary number of vehicle changes within a crew schedule; does not enforce vehicles to be driven by drivers based at the vehicle’s depot; and incorporates a degree of re-timing of the vehicle routes to be covered. Furthermore, it only uses variables representing feasible crew schedules and vehicle counts and the types of problems we solve assume that vehicle routes can be covered by vehicles based at any depot and that drivers are qualified to drive any type of vehicle.

3. Problem description—the APMDP

The vehicle routing and crew scheduling problem for intra-city mail distribution at Australia Post (referred to as the Australia Post Mail Distribution Problem—APMDP) consists of finding a low cost set of vehicle routes, vehicle schedules and crew schedules based at multiple depots to deliver a given set of shipments within a 24 hour cyclical time period.

A shipment in the APMDP is defined as a quantity of mail with an origin, a destination and a specific time window for delivery, bounded by an earliest pick-up and a latest drop-off time. A vehicle route is a feasible sequence of pick-ups and drop-offs of individual shipments for a particular vehicle type starting and ending at a vehicle depot. A vehicle route is only complete when all shipments it delivers have been unloaded and it has returned to a depot. A vehicle schedule is a set of contiguous vehicle routes for the same vehicle type that starts and ends at the same depot and usually spans less than one 24 hour period. A crew schedule describes the tasks undertaken by one driver for one shift, it is a collection of activities whose composition is governed by the vehicle routes it covers, the vehicle schedules these routes are assigned to and a set of regulations that vary for different driver types.

The cyclical nature of the problem arises because the network is repeated (with minor operational variations) each weekday. In practice this means that a shipment with pick-up and drop-off times close to the start of the 24 hour time period could be part of a vehicle route that starts towards the end of the time period and straddles the start-end time of the cyclical time period.

A typical vehicle route for the APMDP may consist of pick-ups at the starting depot, pick-ups and drop-offs at locations within the distribution network and possible drop-offs at the ending depot. Vehicle routes may also include what are commonly called dead-head legs, which are empty positioning legs. Dead-head legs may be required to get from the starting depot to the location of the first pick-up, from the location of the last drop-off back to the ending depot or between the last drop-off of a set of shipments and the first pick-up of a new set of shipments within a route.

The dwell time at a location within a vehicle route is governed by a specific loading and unloading regime for Australia Post distribution vehicles. A crew schedule in the APMDP consists of a set of contiguous activities dictated by the set of vehicle routes it covers. A crew schedule begins with a sign on activity followed by a set of drive activities describing the vehicle routes and dead-heads covered, with an optional intervening break activity which must occur at a depot, not necessarily the home depot. A crew schedule ends with a sign off activity. Dead-head legs may be needed in a crew schedule for many reasons: to connect different vehicle routes covered by the schedule; to travel from/to the depot to/from the first/last location in the first/last route covered by the schedule; or to return to a depot to incorporate a break activity.

A special set of activities are associated with any vehicle changes that occur within a crew schedule. Before a vehicle may be driven as part of a crew schedule, time is given to the driver to prep the vehicle and before a driver finishes using a vehicle, time is given to return it. Thus, from a vehicle view point, a crew schedule can be thought of as being composed of what are commonly called pieces of work. A piece of work is defined as a contiguous set of activities within the same crew schedule, bounded by prep and return activities, within which the same physical vehicle is being used.

A prescribed set of regulations exist for each driver type which govern the legality of a crew
schedule. This set of regulations includes: a maximum duration; the maximum duration before and after a break activity; and specific durations for sign on, prep, break, return and sign off activities. A set of vehicle movements for all vehicle types between all locations, that are allowed to accept vehicles of that size, is used to construct the vehicle routes and dead-heads that appear in crew schedules. This set of vehicle movements obeys triangle inequalities for cost, distance and duration.

4. Vehicle routing for the APMDP

The vehicle routing component of the APMDP can be described as a multiple depot, time constrained, vehicle routing problem with pick-ups and drop-offs, time windows and capacity constraints (referred to as the Australia Post Vehicle Routing Problem—APVRP). The routes in the APVRP need not return to base, they do however have to return to a depot. The vehicle schedules used to cover the routes must return to base.

The method needed to count vehicles in a non-return to base context within a cyclical time period makes the incorporation of vehicle fixed costs into a purely heuristic method of solving the APVRP difficult. In addition the APVRP is a low cardinality vehicle routing problem (see Table 2 to compare the average vehicle capacity with the average shipment volume), a problem domain where purely heuristic methods are less compelling (see for instance Xu et al. (2003) who solve a low cardinality vehicle routing problem using a set partitioning based heuristic column generation approach similar to that presented here). These two attributes are the reason why a purely heuristic method of solving the APVRP is not used.

4.1. Crew scheduling considerations

We need to impose a maximum route duration during the vehicle routing stage to ensure that legal crew schedules can be made to cover the set of vehicle routes generated. A crew schedule has a limit on the total work time that is allowed to elapse before and after a break activity. For simplicity we assume that the maximum work time after the break activity is always less than or equal to the maximum work time before a break activity. We ensure that the duration of each route, including time for a prep and return activity, is less than the maximum elapsed time before a break activity. In addition, as drivers are allowed to have a break at any depot, we need not force routes to return to base.

4.2. Mathematical formulation

The formulation we use to find solutions to the APVRP is based on set covering with column generation, an approach first described by Balinski and Quandt (1964) (see Toth and Vigo (2002) for a comprehensive summary of both exact and heuristic methods, including set covering with column generation, developed over the last few decades for the vehicle routing problem). As the variables within the model are legal routes, this approach makes it easy to model the time windows associated with each shipment and the loading and unloading regime required for Australia Post distribution vehicles. The APVRP formulation is very similar to the classical fleet assignment problem (see for instance Hane et al., 1995) in the airline industry.

We introduce the following notations and problem data and use them to construct a mixed integer programming formulation for the APVRP based on set covering with a circulation of vehicles.

Sets and associated subsets. Let \( T \), indexed by \( t \), be the set of time points representing a 1 minute discretisation of the cyclical 24 hour planning horizon. A cyclical planning horizon means that the minimum and maximum times within the planning horizon are represented by the same time index. Let \( t_c \) represent a particular time used to count vehicles in operation, the counting time. Let \( t^- \) represent the time point immediately before that represented by \( t \). Let \( L \), indexed by \( l \), be the set of locations. Let \( D \subset L \), indexed by \( d \), be the set of depot locations. Let \( V \), indexed by \( v \), be the set of vehicle types. Let \( M = D \times V \), indexed by \( m \), be the set of vehicle type–depot combinations. Let \( M_v \subset M \) be the set of vehicle type–depot combinations for vehicle type \( v \). Let \( S \), indexed by \( s \), be the set of shipments.

Let \( R \), indexed by \( r \), be the set of all feasible routes that could be generated given the specific column generation approach we use to construct
routes. Let \( R_s \subset R \) be the set of routes that cover, i.e. deliver, shipment \( s \). Let \( R_{\text{end}}^m \subset R \) be the set of routes that end (where end is denoted by the superscript “end”) at location \( l \) at time \( t \) using a vehicle of vehicle type–depot combination \( m \). Let \( R_{\text{beg}}^{mlt} \subset R \) be the set of routes that begin (where begin is denoted by the superscript “beg”) at location \( l \) at time \( t \) using a vehicle of vehicle type–depot combination \( m \). Let \( R_{\text{mt}}^m \subset R \) be the set of routes in operation (where in operation is denoted by the superscript “o”) at time \( t \) with vehicle type–depot combination \( m \).

**Variables.** The variables in the model are \( x_r \), \( N_{mlt} \) and \( q_m \). The binary variable \( x_r \) is equal to 1 if and only if route \( r \) is selected in the solution. The variable \( N_{mlt} \) is used to represent the number of vehicles of vehicle type–depot combination \( m \) at location \( l \) at time \( t \) that are not being used at time \( t \), i.e. not part of a route in operation at time \( t \). The variable \( q_m \) is used to represent the total number of vehicles of vehicle type–depot \( m \).

**Costs:** The costs in the model are \( g_m \) and \( c_r \). \( g_m \) is the fixed cost associated with each vehicle type–depot combination \( m \) and \( c_r \) is the variable cost of route \( r \).

**Problem data:** Let \( M_{\text{MAX}}^m \) be the capacity associated with vehicle type–depot combination \( m \). Let \( V_{\text{MAX}}^v \) be the maximum number of vehicles available of vehicle type \( v \).

The APVRP formulation is

\[
\text{Minimise } \sum_{r \in R} c_r x_r + \sum_{m \in M} g_m q_m \tag{1}
\]

subject to

\[
\sum_{r \in R_s} x_r \geq 1 \quad \forall s \in S, \tag{2}
\]

\[
N_{mlt} = \sum_{r \in R_{\text{beg}}^{mlt}} x_r - \sum_{r \in R_{\text{end}}^m} x_r \quad \forall m \in M, \quad \forall l \in L, \quad \forall t \in T, \tag{3}
\]

\[
\sum_{l \in L} N_{mlt} + \sum_{r \in R_{\text{mt}}^m} x_r = q_m \quad \forall m \in M, \tag{4}
\]

\[
q_m \leq M_{\text{MAX}}^m \quad \forall m \in M, \tag{5}
\]

\[
\sum_{m \in M_v} q_m \leq V_{\text{MAX}}^v \quad \forall v \in V, \tag{6}
\]

\[
N_{mlt} \geq 0 \quad \forall m \in M, \quad \forall l \in L, \quad \forall t \in T, \tag{7}
\]

\[
x_r \in \{0,1\} \quad \forall r \in R. \tag{8}
\]

Constraint (2) ensures that all shipments are covered by at least one route. The formulation is given as a set covering problem instead of a set partitioning problem because the heuristic solution technique employed favours building long routes, and, it is easier for commercially available linear programming packages to find good integer solutions to set covering problems. Note that with a set covering formulation, over-covering of shipments, when a shipment is delivered by more than one route in the solution, is possible (see Section 4.3 for the method used to resolve over-covering).

Constraints (3) and (7) ensure a legal circulation of vehicles in the network and are required due to the extra flexibility afforded the vehicle routing problem because routes do not have to return to base. Constraint (4) is used to count the number of vehicles of vehicle type–depot \( m \) by taking a slice through the circulation at a particular time, noting those vehicles in operation and those waiting at depots. Constraint (5) enforces the maximum number of vehicles of each type that are allowed to be depoted at a particular location. Constraint (6) enforces the maximum number of vehicles of a particular type that can be operated across all depots. Finally, binary requirements on vehicle route variables are given by constraint (8).

The size of the set of constraints (3), and thus the number of time points required in the set \( T \), can be reduced using node aggregation (see Hane et al., 1995). If we say that a route is picking up a vehicle at a depot when it starts and dropping off a vehicle at a depot when it finishes, then constraints are only required when a pick-up precedes a drop-off. Variables representing consecutive pick-ups, consecutive drop-offs or drop-offs occurring before pick-ups need not appear in separate constraints as legal connections within the circulation can be made in these cases. Conversely, illegal connections could be made, leading to an invalid circulation, if variables representing drop-offs occurring after pick-ups appeared in the same constraint.

A vehicle route in the APVRP formulation is characterised by the shipments it covers, the vehicle depots it starts and ends at, the times it starts
and ends as well as the vehicle depot of the vehicle schedule that the route is part of. Constraints (3) and (7), by ensuring that a legal circulation of vehicles using the set of vehicle routes chosen exists, ensures that a set of return to base vehicle schedules can be found.

Allowing the vehicle depot for the vehicle schedule that a vehicle route belongs to, to be different from both the start and end depot of the route itself, requires the introduction of an extra constraint (which simply fixes some variables to 0) and the imposition of an extra restriction on route creation to ensure that vehicle counting is performed correctly. Let \( \text{loc}(m) \) define the location associated with vehicle type–depot \( m \). The extra constraint required is

\[
N_{mlc} = 0 \quad \forall m \in M, \ l \in L \text{ such that } \text{loc}(m) \neq l.
\]

Constraint (9) ensures that no vehicles are waiting at a location that is different from their home depot at the counting time. We also have to restrict the depot of the possible vehicle schedules that vehicle routes in operation at the counting time can be part of, to be only either the start or end depot of the route. These two restrictions ensure that a vehicle schedule visits its home depot so that vehicle counting is performed correctly. The time chosen as the counting time in this regime becomes important as it can affect the solution generated. When allowing the depot for the vehicle schedule covering a vehicle route to differ from both the start and end depot of the route (we call this approach the allowing all depots approach denoted ADA, and the alternative approach, where the depot for the vehicle schedule covering the route is restricted to either the start or end depot of the route, the limited depots approach, denoted LDA), we take the time with the smallest total volume of shipments with their delivery windows active as the counting time. We call the approach where routes are forced to return to base, the return to base approach, denoted RTB. Note that for the RTB and LDA approaches the specific time chosen as the counting time is not important as its value has no affect on the solution.

4.3. Solution technique

We use a heuristic technique based on column generation to find solutions to the APVRP. First, all feasible routes covering one and two shipments that: return to base; connect to the closest start and end depot given the first pick-up and the last drop-off location of the shipments covered by the route; and are assigned to a vehicle schedule whose depot is either the start or end depot of the route (note the last two steps are omitted for the RTB approach), are enumerated. This set of routes forms an initial pool from which the linear relaxation of the model is solved to obtain a starting solution.

The column generation sub-problem is solved heuristically by using the dual values to price modifications to a subset of the routes currently in the pool. These modifications include (depending on the approach taken): inserting a new shipment into a route at various positions within the current pick-up drop-off sequence; changing the vehicle type–depot of a route; changing the start and end depot of a route; changing the depot of the vehicle schedule that a route is assigned to; changing the order of the pick-up drop-off sequence of a route; and changing the timings of a route. One column generation iteration is complete after a new set of routes has been created by applying this set of route modifications to a subset of the routes in the pool.

If the size of the pool becomes too large, reduced costs are used to cull those routes with the highest reduced cost until the size of the pool is reduced to a given preset limit, we call this the pool size limit. Routes appearing in intermediate linear solutions during the column generation process, along with all single shipment routes and a select subset of the lowest cost routes constructed for each legal two shipment combination, are immune from the culling process. Only a subset of the routes in the pool are allowed to be modified to produce new routes in each column generation iteration. We call this set the seed set. The seed set is chosen by taking a subset of the routes in the pool with the lowest reduced cost. Increasing the pool size limit and the size of the seed set affects both the quality of the solution and the time taken to solve the linear relaxation.
Column generation continues until either no new routes with a negative reduced cost can be found or the percentage improvement from the last three consecutive column generation iterations is <0.3% (this is used to reduce the tailing off effect of column generation, see Section 6.3).

Integer solutions are found by imposing the integrality constraint on the route variables, introducing branching variables with high branching priority based on the branching technique of Ryan and Foster (1981) and solving the model using all routes currently in the pool. Let $R_{ij}$ be the set of routes that cover shipments $i$ and $j$, the binary branching variables are introduced are defined by $\sum_{r \in R_{ij}} x_r = b_{ij}$ for all pairs of shipments $i, j$ that appear together in at least one route. As the APVRP is a low cardinality vehicle routing problem the number of branching variables introduced is not too large. The branching variables introduced still allow for over-covering to be present in the solution as they only explicitly prohibit over-covering of pairs of shipments that appear in at least one route together and so do not compromise the heuristic technique used that favours generating long routes.

To resolve over-covering, all subsets of routes which cover over-covered shipments are enumerated (by removing all combinations of the set of over-covered shipments within the route). These dominated routes, along with ones representing the application of the set of route modification steps to this new subset, are added to the set of routes in the solution to form a new larger set of routes. A final solution is obtained by solving the set partitioning version of the model, i.e. constraint (2) becomes $\sum_{r \in R_s} x_r = 1 \ \forall s \in S$, using this new set of routes.

In order to model the cyclical nature of the problem, all shipments with pick-up and drop-off times within a certain duration of the start-end time of the cyclical planning horizon, we call this duration the mirroring duration, are replicated at the same time on the next day, i.e. plus 24 hours. This means that such shipments can be covered by routes starting towards the beginning or near the end of the planning horizon. The cap on the maximum route duration, given by the maximum elapsed time before a driver can have a break, defines the length of the mirroring duration.

5. Vehicle and crew scheduling for the APMDP

The post vehicle routing vehicle and crew scheduling problem, is that of scheduling a set of different driver types based at a set of depots to cover a given set of vehicle routes, as well as assigning specific vehicles to a set of routes to produce vehicle schedules within a cyclical planning horizon.

5.1. Mathematical formulation

We introduce the following notations and problem data, in addition to those introduced in Section 4.2, and use them to construct a mixed integer programming formulation for the MDSVCSP based on set covering for crew schedules with an embedded circulation for vehicles.

Extra sets and associated subsets. Let $P$, indexed by $p$, be the set of crew types. Let $K$, indexed by $k$, be the set of tasks that have to be covered (i.e. vehicle routes in the APMDP context) by crew schedules. Let $J = D \times P$, indexed by $j$, be the set of crew type–depot combinations. Let $J_p \subset J$ be the set of crew type–depot combinations for crew type $p$.

Let $I$, indexed by $i$, be the set of all legal crew schedules. Let $I^{\text{ad}} \subset I$ be the set of crew schedules for crew type–depot $j$ (where for crew type–depot is denoted by the superscript “ctd”). Let $I^{\text{ask}}_k$ be the set of crew schedules that cover task $k$ (where cover task is denoted by the superscript “task”). Let $I^{\text{fr}}_m$ be the set of crew schedules that finish a return activity (where finish return is denoted by the superscript “fr”) for a vehicle of vehicle type–depot combination $m$ at location $l$ at time $t$. Let $I^{\text{bp}}_m$, be the set of crew schedules that begin a prep activity (where begin prep is denoted by the superscript “bp”) for a vehicle of vehicle type–depot combination $m$ at location $l$ at time $t$. Let $I^{\text{ml}}_m \subset I$ be the set of crew schedules that are using a vehicle of vehicle type–depot combination $m$ at time $t$, i.e. a piece of work within the crew schedule being covered by a vehicle of vehicle type–depot $m$ spans the time point (or is in operation at the
Extra variables. The extra variables in the model are \( y_i \) and \( n_j \). The binary variable \( y_i \) is equal to one if and only if crew schedule \( i \) is in the solution. The variable \( n_j \) is only used to represent the total number of crew schedules of crew type–depot \( j \).

Extra costs. The extra costs in the model are \( h_i \) and \( f_j \). \( h_i \) is the combined variable vehicle and crew costs for crew schedule \( i \). \( f_j \) is the fixed cost for a crew schedule of crew type–depot \( j \).

Extra problem data. Let \( PMAX_p \) be the maximum number of drivers available for crew type \( p \) and let \( JMAX_j \) be the maximum number of drivers available for crew type–depot \( j \).

The MDSVCSP formulation is

\[
\text{Minimise} \quad \sum_{i \in I} h_i y_i + \sum_{j \in J} f_j n_j + \sum_{m \in M} g_m q_m \quad (10)
\]

subject to

\[
\sum_{i \in P_k^{ad}} y_i \geq 1 \quad \forall k \in K, \quad (11)
\]

\[
\sum_{i \in P_j^{ad}} y_i = n_j \quad \forall j \in J, \quad (12)
\]

\[
n_j \leq JMAX_j \quad \forall j \in J, \quad (13)
\]

\[
\sum_{j \in J_p} n_j \leq PMAX_p \quad \forall p \in P, \quad (14)
\]

\[
N_{mlt} + \sum_{i \in P_{l\cdot t}^{ad}} y_i - \sum_{i \in P_{l\cdot t}^{bd}} y_i = N_{mlt} \quad \forall m \in M, \quad l \in L, \quad t \in T, \quad (15)
\]

\[
\sum_{l \in L} N_{mlt} + \sum_{i \in P_{l\cdot t}^{bd}} y_i = q_m \quad \forall m \in M, \quad (16)
\]

\[
q_m \leq MMAX_m \quad \forall m \in M, \quad (17)
\]

\[
\sum_{m \in M_v} q_m \leq VMAX_v \quad \forall v \in V, \quad (18)
\]

\[
N_{mlt} \geq 0 \quad \forall m \in M, \quad l \in L, \quad t \in T, \quad (19)
\]

\[
y_i \in \{0,1\} \quad \forall i \in I. \quad (20)
\]

Constraint (11) ensures that all tasks are covered by at least one crew schedule. Constraints (12)–(14) model the crew type restrictions. Constraint (12) is used to define variables representing the number of crew schedules for a particular crew type–depot. Constraint (13) enforces the maximum number of crew schedules of a particular type that can be operated out of a particular depot. Constraint (14) enforces the maximum number of crew schedules of a particular type that can be operated across all depots.

Constraints (15)–(18) model the vehicle type restrictions and are essentially the same as those used in the APVRP formulation given in Section 4.2. Constraints (15) and (19) ensure a legal circulation of vehicles in the network. Constraint (16) is used to count the number of vehicles of vehicle type–depot \( m \) by taking a slice through the circulation at a particular time, noting those vehicles in operation and those waiting at depots. Constraint (17) enforces the maximum number of vehicles of each type that are allowed to be depoted at a particular location. Constraint (18) enforces the maximum number of vehicles of a particular type that can be operated across all depots. Finally, binary requirements on crew schedule variables are given by (20).

As with the APVRP formulation we use set covering instead of set partitioning for the MDSVCSP formulation. In addition to making intermediate LPs and the final IP easier to solve, the set covering approach allows the enumeration heuristic we use to find initial solutions, to incorporate a domination criteria vastly decreasing the number of crew schedules created while still ensuring good initial solutions. Over-covering is resolved in the same manner as for the APVRP as described in Section 4.3. To resolve over-covering, all subsets of crew schedules which cover over-covered tasks are enumerated by removing all combinations of the set of task-instances representing the over-covered tasks within the crew schedule (see Section 5.2 for an explanation of the term \textit{task-instance}). These \textit{dominated} crew schedules are added to the set of crew schedules in the solution to form a new larger set of crew schedules. A final solution is obtained by solving the set partitioning version of the model using this new set of crew schedules.

As with the APVRP formulation, the size of the set of constraints (15), and thus the number of time points required in the set \( T \), can be reduced using \textit{node aggregation}. If we say that a crew schedule is picking up a vehicle at a depot when it begins a prep activity and dropping off a vehicle at a depot when it finishes a return activity, then con-
straints are only required when a pick-up precedes a drop-off.

5.2. Exploiting the flexibility of the integrated approach

There are many properties of the APMDP that can be exploited when the vehicle and crew scheduling problem is solved simultaneously. These include

- The incorporation of both fixed and variable driver and vehicle costs within the one optimisation leading to a solution where these costs can be traded off directly.
- Given that a simultaneous approach can guarantee that all vehicles return to base, there may be opportunities for drivers to swap vehicles at intermediate depots.
- Prep and Return activities are created in crew schedules as required and not dictated by a predetermined set of vehicle schedules.
- A given vehicle route may be able to be operated at any time within a given time window and still ensure that all shipments are delivered on time.
- The vehicle type for a given vehicle route may, where legal, be changed to another vehicle type.
- The dead-head legs in the input set of vehicle routes may be removed leading to a larger set of smaller routes with increased flexibility, resulting in a lower cost solution overall.

The set covering approach taken in the MDSVCSP formulation makes it easy to model the variable timing and the vehicle substitutability of the vehicle routes as well as the cyclical nature of the problem. For ease of exposition we introduce the concept of a task and a task-instance. A task represents a vehicle route and has an associated set of task-instances. Different task-instances for a given task vary according to start and end time and/or vehicle type. The allowable timings of the set of task-instances for a task is governed by the time window within which the task may be operated and the chosen discretisation of this time window as well as its proximity to the start of the planning horizon. All task-instances within a certain duration (the mirroring duration, which is set to the longest crew schedule duration for the particular problem) of the start of the planning horizon are replicated at the same time on the next day, i.e. plus 24 hours. The set of task-instances are used as the building blocks for generating crew schedules. The set of tasks covered by a crew schedule is governed by the set of task-instances that have been used to construct the crew schedule as each task-instance is associated with a specific parent task.

5.3. Pieces of work within a crew schedule and vehicle schedules

Each piece of work within a crew schedule (recall that a piece of work is a group of contiguous activities within a crew schedule bounded by prep and return activities within which the same physical vehicle is being used) in the solution must be covered by a vehicle schedule of the appropriate vehicle type based at a particular depot. To model the different depots that vehicles covering pieces of work within a crew schedule can be assigned to, we introduce specific variables into the MDSVCSP formulation to represent each possible combination. There are three approaches we use to select the potential depots that vehicles covering pieces of work can be assigned to.

Consider a crew schedule that is composed of two pieces of work, the first starting at the home depot of the crew schedule, A, and ending at another depot, B, the second starting at B and ending at A. If we restrict the possible depot locations for vehicles covering these pieces of work to either the start or end locations of the piece of work, then there are four possible ways of assigning the vehicles used to cover these pieces of work to vehicle depots; both pieces of work are covered by a vehicle depoted at A; the first piece of work is covered by a vehicle depoted at A and the second by a vehicle depoted at B; the first piece of work is covered by a vehicle depoted at B and the second by a vehicle depoted at A; and both pieces of work are covered by vehicles depoted at B. As with the corresponding approach outlined for the APVRP, we call this approach the limited depots approach, denoted LDA. In this approach, for any crew
schedule with \( n \) pieces of work, there are \( 2^n \) possible ways (hence \( 2^n \) variables are introduced for each crew schedule made) of assigning the vehicles used to cover the pieces of work to vehicle depots. Clearly this approach will be impractical for problems where crew schedules incorporate many pieces of work. Crew schedules generated in the APMDP typically have no more than three pieces of work, the majority have either one or two due to the duration of the vehicle routes to be covered and the time lost in a schedule when a vehicle change is performed.

Relaxing this restriction so that any depot could be the base for a vehicle covering a particular piece of work would mean that if there are \( m \) depots there would be \( m^n \) possible combinations for a crew schedule with \( n \) pieces of work. As with the corresponding approach outlined for the APVRP, we call this approach the allowing all depots approach, denoted ADA. Introducing \( m^n \) variables for each crew schedule is clearly impractical for even a modest number of depots in the problem. Restricting the set of locations that can be depots for vehicles used by crew schedules to be the set of locations where prep and return activities have taken place within the crew schedule, plus, optionally, a set of nearby depots, could be used to temper the exponential increase in variables for problems with large numbers of depots when using this approach. As with the APVRP formulation given in Section 4.2, we have to introduce constraint (9) into the MDSVCSP formulation to ensure that vehicle counting is performed correctly in this case. Similarly, we have to restrict the depots that vehicles in operation at the counting time can be assigned to, to be only those at either end of the current piece of work. The time chosen as the counting time in this approach becomes important as it can affect the solution generated (see Section 6.4 to see how we determine the counting time).

Restricting the candidate depots for vehicles covering pieces of work within crew schedules to be the home depot of the crew schedule means that we need only introduce a single variable into the MDSVCSP formulation for each crew schedule. We refer to this approach as the restricted depots approach denoted RDA. In this approach drivers only drive vehicles stationed at their home depot.

As each crew schedule in the model encodes the specific depot that vehicles covering each of its pieces of work are assigned to, retrieving vehicle schedules from the solution is a simple matter of applying a first in first out heuristic to the network defined by the pieces of work assigned to each separate vehicle type–depot.

### 5.4. Solution technique

We use restricted enumeration followed by column generation to solve the linear relaxation of the MDSVCSP formulation, a technique commonly used to solve large vehicle routing and crew scheduling problems (see for instance Desrochers and Soumis, 1989). The enumeration heuristic is used to find a good initial solution quickly, column generation is then used to further improve the solution. Once column generation terminates, either when no further negative reduced cost columns can be found or a given set of termination criteria are met, an integral solution is found heuristically using branch and bound.

As with the APVRP we introduce branching variables with high branching priority, this time based on the follow-on branching rule, Desrosiers et al. (1991) which is motivated by the branching technique of Ryan and Foster (1981). Let \( I_{jk} \) be the set of crew schedules that cover tasks \( j \) and \( k \) consecutively, we introduce binary variables defined by \( \sum_{j \in I_{jk}} y_j = e_{jk} \) for all pairs of tasks \( j,k \) that appear consecutively in at least one crew schedule. Again, these constraints do not prohibit over-covering from being present in the solution and so do not compromise the heuristic enumeration technique used to find initial solutions which is designed to generate long crew schedules. As the Australia Post MDSVCSP is a low cardinality crew scheduling problem the number of branching variables introduced is quite small.

The column generation sub problem corresponds to the standard resource constrained shortest path problem, where the resource is the elapsed time after a break activity, and is solved using dynamic programming (see Ziegelmann (2001) for a comprehensive overview of the literature for solving resource constrained shortest path problems).
5.4.1. Enumeration

The enumeration heuristic is based on a simple depth first recursive search algorithm (see Sedgewick (1992) for a description of depth first recursive search algorithms). First, the set of task-instances is sorted by start time. Then, for each task-instance in this ordered set, starting with the task-instance with the earliest start time, a depth first recursive search through the ordered array of task instances is performed. Crew schedules are created for all legal driver types based at all legal driver depots when it is not possible to add another task-instance to the current search tree and create a legal crew schedule using any driver type based at any depot. This approach creates only long crew schedules, i.e. the crew schedules created contain as many task-instances as legally possible. Furthermore, crew schedules are not created if the task-instances covered are a subset of the task-instances covered by an existing crew schedule with the same driver type operating out of the same depot, i.e. only undominated crew schedules are created.

For most problems the set of undominated crew schedules is too large to enumerate. There are two techniques we use to combat this problem. The first involves adding extra restrictions to the legality criteria for crew schedules created within the enumeration heuristic to limit the creation of undesirable crew schedules (e.g. placing a cap on the total amount of unproductive time in a crew schedule). The second involves solving MDSVCSP LPs during the enumeration heuristic when the number of crew schedules created reaches a certain threshold and then culling the crew schedules not in this partial solution.

The first step of the enumeration process involves creating the set of all single task-instance crew schedules for all legal driver types for all legal driver depots. This set of single task-instance crew schedules guarantees feasibility when solving intermediate LPs during the enumeration heuristic. Using a set covering formulation for the MDSVCSP in conjunction with the domination criteria enforced in the enumeration process, allows lower cost solutions to be found from fewer crew schedules than would be possible using a set partitioning formulation as over-covering would not be allowed in this case.

6. Computational experimentation

In this section we present a computational study using the algorithms and techniques given in the previous sections to find solutions to the APMDP using three data instances labelled M1, M2 and M3.

The aims of this study are as follows: first, to investigate the impact of different vehicle routing solutions on vehicle crew scheduling solutions; second, to investigate the different ways that the simultaneous vehicle crew scheduling algorithm presented can be used to integrate vehicles and drivers and in doing so analyse the affect of allowing drivers to drive vehicles based at depots different from their own; third, to measure the improvement from solving the vehicle and crew scheduling problem simultaneously as opposed to sequentially; and finally, to demonstrate significant cost savings for Australia Post. All problems were solved using a Pentium 3.6GHz CPU with 2GB RAM, with the optimiser CPLEX (v9.0) used to solve both linear and integer programs.

First we give a brief description of the data used. We present the results of solving the APVRP formulation using all approaches for a range of parameter settings. We describe a set of acceleration strategies used to reduce the time taken to solve the linear relaxation of the MDSVCSP formulation and outline a method for building solutions as the level of integration between vehicles and drivers increases. We compare the affect of different types of vehicle routing solutions on the vehicle crew scheduling solution. Finally, we measure the affect of increasing the level of integration between vehicles and drivers and the difference between simultaneous and sequential solution approaches.

6.1. APMDP data

Table 1 describes the crew schedule regulations for both types of drivers used by Australia Post for the Melbourne metropolitan mail distribution network. The duration of all prep activities is 15 minutes, the duration of all return activities is 10 minutes and the sign on and sign off activity durations are 0 minute for both driver types.
Three models of the Australia Post Melbourne metropolitan mail distribution network were used to test the various solution techniques. The models, called M1, M2 and M3 encapsulate the distribution network from the years 2001, 2002 and 2003, respectively. Table 2 displays: the number of locations; the number of depots; the number of vehicle types; the average vehicle capacity; the number of shipments; and the average shipment volume for each model. Note the location of the depots in model M3 is different to those in M1 and M2.

### 6.2. Vehicle routing solutions

As mentioned in Section 4.3 there are two parameters that control the heuristic method used to solve the APVRP. These parameters are the pool size limit and the size of the seed set. Increasing the pool size limit and the size of the seed set affects both the quality of the solution and the time taken to solve the linear relaxation.

Recall from Section 4.3 that there are two different approaches that can be used to solve the APVRP when routes do not have to return to base, the allowing all depots approach (ADA) and the limited depots approach (LDA). The two approaches differ in the set of allowable depots for a vehicle schedule covering a particular vehicle route. Table 3 shows the results of using all three approaches (RTB, LDA, ADA) when solving the APVRP formulation for a range of values of the pool size limit and the seed set size for model M3. The table shows: the approach taken; the pool size limit; the size of the seed set; the number of column generation iterations before termination; the total number of columns generated; the total time spent solving the linear relaxation (which includes column generation time); the number of columns in the MIP; the number of branch and bound nodes; the branch and bound CPU time; the optimality gap; the number of over-covered shipments; the improvement from resolving over-covering; the number of vehicles; the number of routes; the total route duration and the total number of kilometers travelled in the solution. The final row shows a percentage indicating how close each solution was to the best solution found from all approaches. The optimality gap is defined as the difference between the value of the best solution found and the smallest bound associated with an unexplored node in the branch and bound search tree expressed as a percentage of the best solution.

Branch and bound was terminated if there had been no improvement in the integer solution over the previous 1500 iterations. Note the optimality gaps across all approaches for this model are large. The average optimality gap across all models and approaches was 2.47%. However, the solutions typically exhibit over-covering and the improvement gained from removing this over-covering while being modest is still significant. The average percentage improvement from removing over-covering across all models and approaches was 0.59%. The table also shows that increasing the controlling parameters (the size of the seed set and the pool size limit) for each approach produces a better solution at the cost of an increase in the time taken for the solution to both the LP and IP to reach our termination criteria.

### Table 1

Crew schedule regulations (times are in hh:mm format)

<table>
<thead>
<tr>
<th>Regulation</th>
<th>Driver type 1</th>
<th>Driver type 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work time</td>
<td>7:21</td>
<td>7:21</td>
</tr>
<tr>
<td>Break length</td>
<td>0:30</td>
<td>0:30</td>
</tr>
<tr>
<td>Work time before break</td>
<td>0:00</td>
<td>5:00</td>
</tr>
<tr>
<td>Work time after break</td>
<td>0:00</td>
<td>5:00</td>
</tr>
</tbody>
</table>

### Table 2

APMDP models

<table>
<thead>
<tr>
<th>Model</th>
<th>Locations</th>
<th>Depots</th>
<th>Veh. types</th>
<th>Av. veh. cap.</th>
<th>Shipments</th>
<th>Av. volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>339</td>
<td>5</td>
<td>5</td>
<td>2200</td>
<td>1127</td>
<td>598</td>
</tr>
<tr>
<td>M2</td>
<td>305</td>
<td>5</td>
<td>5</td>
<td>2200</td>
<td>1090</td>
<td>580</td>
</tr>
<tr>
<td>M3</td>
<td>256</td>
<td>5</td>
<td>4</td>
<td>2550</td>
<td>1181</td>
<td>800</td>
</tr>
</tbody>
</table>
Table 4 summarises the results of solving the APVRP formulation for all approaches for all models. The solution labels from Table 3 are reused in Table 4 (and Tables 5 and 6) indicating the APVRP approach taken and the controlling parameters used for each set of results. For each model Table 4 displays: the number of vehicles; the number of routes; the total route duration; the number of kilometers; and a percentage indicating how close each solution was to the best solution found from all approaches.

For each model the LDA approach with the largest controlling parameters produced the best solution. Notably, the number of routes in the solutions produced using the RTB approach is significantly less than the number of routes produced using the other solution approaches. This will mean fewer tasks and task-instances when the

<table>
<thead>
<tr>
<th>Model</th>
<th>APVRP approach</th>
<th>Solution label</th>
<th>VR1</th>
<th>VR2</th>
<th>VR3</th>
<th>VR4</th>
<th>VR5</th>
<th>VR6</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>ADA</td>
<td>ADA</td>
<td>LDA</td>
<td>LDA</td>
<td>RTB</td>
<td>RTB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>No. vehicles</td>
<td>45</td>
<td>44</td>
<td>43</td>
<td>43</td>
<td>45</td>
<td>43</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>No. routes</td>
<td>282</td>
<td>282</td>
<td>253</td>
<td>249</td>
<td>257</td>
<td>235</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tot. route dur.</td>
<td>498:35</td>
<td>496:33</td>
<td>489:05</td>
<td>487:35</td>
<td>511:12</td>
<td>496:30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No. km</td>
<td>13,319</td>
<td>13,133</td>
<td>12,969</td>
<td>12,924</td>
<td>14,001</td>
<td>13,506</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ranking (+X%)</td>
<td>2.47</td>
<td>1.68</td>
<td>0.44</td>
<td>0.00</td>
<td>4.77</td>
<td>1.76</td>
<td></td>
</tr>
<tr>
<td>M2</td>
<td>ADA</td>
<td>ADA</td>
<td>LDA</td>
<td>LDA</td>
<td>RTB</td>
<td>RTB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>No. vehicles</td>
<td>43</td>
<td>42</td>
<td>41</td>
<td>41</td>
<td>43</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No. routes</td>
<td>275</td>
<td>275</td>
<td>261</td>
<td>248</td>
<td>235</td>
<td>230</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No. km</td>
<td>14,296</td>
<td>14,244</td>
<td>14,502</td>
<td>14,281</td>
<td>14,878</td>
<td>14,287</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ranking (+X%)</td>
<td>2.49</td>
<td>1.70</td>
<td>1.76</td>
<td>0.00</td>
<td>3.70</td>
<td>1.55</td>
<td></td>
</tr>
<tr>
<td>M3</td>
<td>ADA</td>
<td>ADA</td>
<td>LDA</td>
<td>LDA</td>
<td>RTB</td>
<td>RTB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>No. vehicles</td>
<td>44</td>
<td>45</td>
<td>47</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No. routes</td>
<td>336</td>
<td>327</td>
<td>334</td>
<td>317</td>
<td>291</td>
<td>295</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No. km</td>
<td>18,909</td>
<td>18,330</td>
<td>18,666</td>
<td>18,201</td>
<td>19,533</td>
<td>19,256</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ranking (+X%)</td>
<td>2.28</td>
<td>0.70</td>
<td>2.24</td>
<td>0.00</td>
<td>5.34</td>
<td>4.36</td>
<td></td>
</tr>
</tbody>
</table>
associated vehicle crew scheduling problem is solved.

For each model, solutions found using the ADA approach were generally of a lower quality than those found using the LDA approach. There are a range of factors that could be contributing to this behaviour. First, the lowest cost solutions may have significant numbers of vehicles waiting at locations different from their home depot over the time chosen as the counting time, making them infeasible for the ADA approach but feasible for the LDA approach. In addition, the number of possible variables for the ADA approach is much larger making the column generation heuristic less effective at finding low cost linear solutions. Finally, the structure of the problem, in particular the low number of depots and their relative proximity, may not be particularly conducive to producing solutions that exploit the extra flexibility of the ADA approach.

Table 5
Multiple depot simultaneous vehicle crew scheduling solutions for model M3 using the LDA approach (all times are in hh:mm format)

<table>
<thead>
<tr>
<th>VR solution label</th>
<th>VR1</th>
<th>VR2</th>
<th>VR3</th>
<th>VR4</th>
<th>VR5</th>
<th>VR6</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. tasks</td>
<td>336</td>
<td>327</td>
<td>334</td>
<td>317</td>
<td>291</td>
<td>295</td>
</tr>
<tr>
<td>No. task instances</td>
<td>1335</td>
<td>1213</td>
<td>1263</td>
<td>1145</td>
<td>1063</td>
<td>1099</td>
</tr>
<tr>
<td>No. cols made by heur.</td>
<td>560,373</td>
<td>451,269</td>
<td>611,083</td>
<td>350,100</td>
<td>212,024</td>
<td>256,040</td>
</tr>
<tr>
<td>Heuristic CPU time</td>
<td>0:05</td>
<td>0:03</td>
<td>0:05</td>
<td>0:02</td>
<td>0:01</td>
<td>0:01</td>
</tr>
<tr>
<td>No. CG iterations</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>No. cols generated</td>
<td>1,205,105</td>
<td>1,201,902</td>
<td>1,202,483</td>
<td>1,156,308</td>
<td>1,204,160</td>
<td>1,062,597</td>
</tr>
<tr>
<td>Impr. from heur. LP Obj.</td>
<td>0.0</td>
<td>0.13</td>
<td>1.22</td>
<td>2.09</td>
<td>1.20</td>
<td>1.29</td>
</tr>
<tr>
<td>Total LP CPU time</td>
<td>1:38</td>
<td>1:11</td>
<td>1:55</td>
<td>2:20</td>
<td>1:09</td>
<td>1:15</td>
</tr>
<tr>
<td>No. B&amp;B nodes</td>
<td>5000</td>
<td>3595</td>
<td>18,000</td>
<td>5000</td>
<td>1673</td>
<td>5000</td>
</tr>
<tr>
<td>B&amp;B CPU time</td>
<td>0:37</td>
<td>0:06</td>
<td>0:43</td>
<td>0:05</td>
<td>0:01</td>
<td>0:04</td>
</tr>
<tr>
<td>Optimality gap (%)</td>
<td>1.01</td>
<td>0.00</td>
<td>0.96</td>
<td>0.09</td>
<td>0.00</td>
<td>0.11</td>
</tr>
<tr>
<td>No. over-cov. tasks</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Over-cov. impr. (%)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>No. CS type 1</td>
<td>30</td>
<td>31</td>
<td>35</td>
<td>45</td>
<td>43</td>
<td>39</td>
</tr>
<tr>
<td>No. CS type 2</td>
<td>101</td>
<td>97</td>
<td>93</td>
<td>72</td>
<td>79</td>
<td>83</td>
</tr>
<tr>
<td>Total no. CS</td>
<td>131</td>
<td>128</td>
<td>128</td>
<td>117</td>
<td>122</td>
<td>122</td>
</tr>
<tr>
<td>Total CS duration</td>
<td>717:32</td>
<td>703:29</td>
<td>721:26</td>
<td>700:33</td>
<td>712:58</td>
<td>704:03</td>
</tr>
<tr>
<td>Total drive time</td>
<td>539:59</td>
<td>533:02</td>
<td>538:42</td>
<td>528:12</td>
<td>548:50</td>
<td>544:31</td>
</tr>
<tr>
<td>No. vehicles</td>
<td>53</td>
<td>52</td>
<td>58</td>
<td>48</td>
<td>52</td>
<td>52</td>
</tr>
<tr>
<td>No. km</td>
<td>19,670</td>
<td>19,222</td>
<td>19,399</td>
<td>19,010</td>
<td>19,580</td>
<td>19,620</td>
</tr>
<tr>
<td>Ranking (+X%)</td>
<td>3.58</td>
<td>2.11</td>
<td>4.32</td>
<td>0.00</td>
<td>2.61</td>
<td>1.57</td>
</tr>
</tbody>
</table>

Table 6
Comparison of different MDSVCSP approaches for all models for each APVRP approach (all times are in hh:mm format)

<table>
<thead>
<tr>
<th>Model</th>
<th>Solution label</th>
<th>VR1 ADA (%)</th>
<th>VR2 ADA (%)</th>
<th>VR3 LDA (%)</th>
<th>VR4 ADA (%)</th>
<th>VR5 ADA (%)</th>
<th>VR6 ADA (%)</th>
<th>Avg. CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>RDA</td>
<td>4.68</td>
<td>3.50</td>
<td>2.89</td>
<td>0.91</td>
<td>1.78</td>
<td>0.19</td>
<td>1:13</td>
</tr>
<tr>
<td></td>
<td>LDA</td>
<td>4.68</td>
<td>3.13</td>
<td>2.69</td>
<td>0.91</td>
<td>1.53</td>
<td>0.19</td>
<td>1:48</td>
</tr>
<tr>
<td></td>
<td>ADA</td>
<td>5.53</td>
<td>3.91</td>
<td>2.59</td>
<td>1.30</td>
<td>1.53</td>
<td>0.00</td>
<td>3:12</td>
</tr>
<tr>
<td>M2</td>
<td>RDA</td>
<td>4.16</td>
<td>4.69</td>
<td>4.02</td>
<td>4.82</td>
<td>2.56</td>
<td>0.01</td>
<td>0:34</td>
</tr>
<tr>
<td></td>
<td>LDA</td>
<td>4.16</td>
<td>4.69</td>
<td>4.02</td>
<td>4.82</td>
<td>2.23</td>
<td>0.00</td>
<td>1:45</td>
</tr>
<tr>
<td></td>
<td>ADA</td>
<td>4.16</td>
<td>4.69</td>
<td>4.93</td>
<td>4.59</td>
<td>2.23</td>
<td>0.23</td>
<td>3:04</td>
</tr>
<tr>
<td>M3</td>
<td>RDA</td>
<td>3.58</td>
<td>2.17</td>
<td>4.56</td>
<td>0.00</td>
<td>2.61</td>
<td>1.57</td>
<td>1:02</td>
</tr>
<tr>
<td></td>
<td>LDA</td>
<td>3.58</td>
<td>2.11</td>
<td>4.33</td>
<td>0.00</td>
<td>2.61</td>
<td>1.57</td>
<td>1:51</td>
</tr>
<tr>
<td></td>
<td>ADA</td>
<td>3.58</td>
<td>1.89</td>
<td>3.69</td>
<td>0.00</td>
<td>2.21</td>
<td>1.57</td>
<td>4:06</td>
</tr>
</tbody>
</table>
6.3. Strategies for solving the MDSVCSP formulation

Intelligent use of the enumeration heuristic, early termination of the solution to the linear relaxation, capping the number of columns generated in an iteration and aggressively culling the pool of columns maintained by column generation, were all used in an attempt to reduce the time taken to solve the linear relaxation of the MDSVCSP formulation.

First note that any solution found using the RDA approach is a legal solution for the LDA approach, and with the exception of constraint (9), any legal solution found using the LDA approach is a legal solution for the ADA approach. As the level of integration increases (going from the RDA approach to the ADA approach) the number of variables in the MDSVCSP formulation increases, making the problems much harder to solve. These two observations suggest that it would be faster to find solutions to the LDA-(ADA) approach by building on an existing solution found using the RDA(LDA) approach.

Finding a solution by building upon a previous solution to a restricted version of the problem has three main advantages for our particular application. First, the heuristic enumeration technique can now be employed when solving using the LDA and the ADA approaches much more effectively as it need not introduce more than a single variable into the formulation for each crew schedule to model the interaction between drivers and vehicles for the now, initial, RDA approach.

Second, once the linear relaxation using the RDA approach has been solved, we can find an integer solution to the problem, ensure the set of variables in this integer solution are not culled by subsequent column generation iterations and use the objective value to set an upper bound on the integer objective to be found using the LDA approach.

Third, using an existing solution to the LDA approach allows us to intelligently choose the counting time required for the ADA approach. Using an integer solution calculated from the LDA approach the counting time is taken as the time with the least total cost of vehicles waiting at locations different from their home depot.

Instead of generating the entire set of columns with negative reduced cost for each column generation iteration, we only generate the first $N$ columns where $N$ is varied for different approaches. This is a commonly used technique (see for instance Freling et al., 2003) to reduce the overall time required to solve the linear relaxation. For each approach the value of $N$ was dictated by the amount of memory available. For the RDA approach, we set $N = 100,000$, for the LDA approach, $N = 400,000$ and for the ADA approach, $N = 2,000,000$. Note that these limits are seldom reached except for the ADA approach.

We terminate the solution to the linear relaxation if the objective function has not decreased by more than 0.3% over the last 3 column generation iterations. This is a commonly used strategy (see for instance Haase et al., 2001; Freling et al., 2003) to reduce the tailing-off effect of column generation, i.e. requiring a large number of iterations to prove LP optimality. We terminate branch and bound if there has been no improvement in the integer solution over the previous 5000 iterations.

Culling the pool of columns maintained by column generation back to those that appeared in the basis from the previous LP solve plus a select subset (based on reduced cost) of those generated in the current iteration, was used to reduce the size of each LP and thus the time spent solving LPs during column generation. For the RDA approach the pool was culled back to 25,000, for the LDA approach, back to 100,000, and for the ADA approach, back to 400,000.

For each model the set of task-instances was generated from the given set of tasks, i.e. routes, by: removing all dead-head legs from either end of each route; allowing all legal vehicle type substitutions to be made; then, if legal, re-timing this set of task-instances by ±60 minutes; and finally mirroring a subset of this set of task-instances to produce the final set of task-instances.

6.4. Vehicle crew scheduling solutions

In this section we display the results of solving the MDSVCSP formulation using the LDA approach for model M3. We examine the affect of different vehicle routing solutions on the vehicle
crew scheduling solution and investigate the benefit of increasing levels of integration between vehicles and drivers. Finally we compare the results of simultaneous versus sequential vehicle crew scheduling for all three models.

Table 5 shows the results of solving the MDSVCSP formulation using the LDA approach for model M3, for each solution produced using the APVRP formulation displayed in Table 3. Table 5 displays: the number of tasks; the number of task-instances; the number of columns produced using the heuristic enumeration algorithm; the CPU time taken to enumerate this set of columns; the total number of column generation iterations; the total number of columns generated; the improvement that column generation obtained over the objective found using the enumeration heuristic; the total time spent solving the linear relaxation (which includes column generation time); the branch and bound CPU time; the optimality gap; the number of over-covered tasks; the improvement found from removing over-covering; the number of crew schedules of type 1; the number of crew schedules of type 2; the total number of crew schedules; the total crew schedule duration; the total drive time in all crew schedules; the number of vehicles; and the number of kilometers travelled in the solution. The final row shows a percentage indicating how close each solution was to the best solution found from all approaches.

Note that the optimality gaps across all APVRP approaches for this particular MDSVCSP approach, i.e. the LDA approach, is quite low. The average optimality gap across all models, all APVRP approaches and all MDSVCSP approaches was 0.16%. Note that the improvement obtained by column generation over the linear objective found using the enumeration heuristic is also quite low. The average improvement from the linear objective found by the enumeration heuristic across all models, all APVRP approaches and all MDSVCSP approaches was 1.57% and the average running time for the heuristic was 2 minutes. This shows the enumeration heuristic to be very effective at quickly finding high quality initial solutions.

Table 6 summarises the comparative performance of the different MDSVCSP approaches, it shows percentages indicating how close each MDSVCSP approach was to the best MDSVCSP approach for each model for all APVRP approaches. It also includes the average CPU time for each MDSVCSP approach for all models for all APVRP approaches. Table 6 shows that for model M1 and M2 the best return to base vehicle routing solution, even though in each case it was only the second best vehicle routing solution overall, produced the best vehicle crew scheduling solution, by a considerable percentage. However for model M3, for which the location of depots and number of vehicle types used is slightly different, the best vehicle crew scheduling solution was found starting with the solution to the LDA vehicle routing approach. This indicates that some experimentation in initial vehicle routing solutions is required to find the best vehicle and crew scheduling solution as the best approach may change depending on the number of vehicle types and the depot configuration used.

Table 6 also shows that the LDA approach, on average, improves upon the solution found using the RDA approach by only 0.08%. It also shows that the ADA approach seldom improves upon the solution found using the LDA approach and frequently results in a worse solution. The factors contributing to this are similar to those outlined in Section 6.2 when analyzing a similar discrepancy between the corresponding approaches for the APVRP formulation. These two observations indicate that the improvement to be gained by allowing drivers to drive vehicles stationed at locations different from their home depot does not outweigh the loss of operational simplicity.

Table 7 displays the results of solving each model using both a sequential and the best simultaneous vehicle crew scheduling method. Sequential solutions differ from simultaneous solutions in that: only return to base vehicle routing solutions can form the set of tasks to be covered, hence the best RTB vehicle routing solutions were used (ones with solution label VR6) for each model for the sequential approach; neither vehicle substitution, sliding or the removal of start and end empty dead-head legs are allowed when creating the set of task-instances from the set of routes in the initial vehicle routing solution; drivers are only
allowed to drive vehicles based at their home depot; and finally dead-head legs are not allowed to be created in crew schedules ensuring that each return to base vehicle route is covered by a vehicle based at the same depot and crew schedules only cover return to base routes based at their home depot. The last restriction imposed on the sequential approach means that the problem breaks down into separate problems for each depot.

For each model Table 7 shows the label for the vehicle routing solution used, the total number of crew schedules, the total number of vehicles and the total number of kilometers in solutions generated using both the sequential and the best simultaneous approaches. The last row in Table 7 shows a percentage indicating the difference between the objectives found using the sequential and simultaneous approaches for each model indicating that, on average, simultaneous solutions improve upon those found using the sequential method by 8.31%. Note that vehicle counts increase in the sequential vehicle crew scheduling solutions from those of the initial vehicle routing solution because prep and return activity times now need to be included when counting vehicles.

Table 7
Comparison of sequential versus simultaneous vehicle crew scheduling methods for all models

<table>
<thead>
<tr>
<th>VCS approach</th>
<th>Model: M1</th>
<th></th>
<th>Model: M2</th>
<th></th>
<th>Model: M3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution label</td>
<td>Seq.</td>
<td>Sim.</td>
<td>Seq.</td>
<td>Sim.</td>
<td>Seq.</td>
<td>Sim.</td>
</tr>
<tr>
<td>No. CS</td>
<td>121</td>
<td>105</td>
<td>116</td>
<td>98</td>
<td>119</td>
<td>117</td>
</tr>
<tr>
<td>No. vehicles</td>
<td>58</td>
<td>49</td>
<td>57</td>
<td>45</td>
<td>60</td>
<td>48</td>
</tr>
<tr>
<td>No. km</td>
<td>13,506</td>
<td>13,650</td>
<td>14,287</td>
<td>14,720</td>
<td>19,288</td>
<td>19,010</td>
</tr>
<tr>
<td>Ranking (+X%)</td>
<td>7.38</td>
<td>0.00</td>
<td>9.95</td>
<td>0.00</td>
<td>7.62</td>
<td>0.00</td>
</tr>
</tbody>
</table>

The formulation introduced naturally models the properties of the mail distribution problem that can be varied to find better combined vehicle and crew scheduling solutions. The solution technique employed exploits the ability of the formulation to build upon existing solutions as the level of integration between vehicles and drivers is increased, and has been shown to find high quality solutions. The algorithms and solution techniques presented in this paper have been used by network planners at Australia Post to demonstrate a potential transport network operational cost saving of 10% for the 2003 Melbourne metropolitan mail distribution network. Australia Post is currently using software developed based upon the ideas presented in this paper to assist in the management of ongoing changes to the mail distribution networks in major cities throughout Australia.

Acknowledgement

The authors would like to thank Australia Post for its support and anonymous referees for their comments.

References


