Joint Vehicle and Crew Routing and Scheduling

Edward Lam[1,2], Pascal Van Hentenryck[1,3], and Philip Kilby[1,3]

[1] NICTA, Eveleigh NSW 2015, Australia
[2] University of Melbourne, Parkville VIC 3010, Australia
[3] Australian National University, Acton ACT 2601, Australia

Abstract. Traditional vehicle routing problems implicitly assume only one crew operates a vehicle for the entirety of its journey. However, this assumption is violated in many applications arising in humanitarian and military logistics. This paper considers a Joint Vehicle and Crew Routing and Scheduling Problem, in which crews are able to interchange vehicles, resulting in space and time interdependencies between vehicle routes and crew routes. It proposes a constraint programming model that overlays crew routing constraints over a standard vehicle routing problem. The constraint programming model uses a novel optimization constraint that detects infeasibility and bounds crew objectives. Experimental results demonstrate significant benefits of using constraint programming over mixed integer programming and a vehicle-then-crew sequential approach.

1 Introduction

A vehicle routing problem (VRP) aims to design routes for a fleet of vehicles that minimize some cost measure, and perhaps, while adhering to side constraints, such as time windows, or pickup and delivery constraints. In practice, however, VRPs are not solved in isolation; they typically arise in a sequential optimization process that first optimizes vehicle routes and then crew schedules given the vehicle routes [33]. This sequential methodology has the advantage of reducing the computational complexity. However, by assigning vehicle routes first, this approach may lead to suboptimal, or even infeasible, crew schedules since decisions in the first stage ignore crew constraints and objectives. Hence it is imperative to simultaneously consider vehicle and crew constraints and objectives, particularly in industries in which crew costs outstrip vehicle costs.

This paper considers a Joint Vehicle and Crew Routing and Scheduling Problem (JVCRSP) motivated by applications in humanitarian and military logistics. In these settings, vehicles (e.g., airplanes) travel long routes, and serve a variety of pickup and delivery requests under various side constraints. Vehicles must be operated by crews who have limitations on their operating times. Crews are also able to interchange vehicles at different locations and to travel as passengers before and after their operating times. JVCRSPs are extremely challenging computationally since vehicle and crew routes are now interdependent. In addition, vehicle interchanges add an additional time complexity, since two vehicles must be synchronized to allow for the exchange to proceed. It is thus necessary to decide whether vehicles wait, and the waiting duration, since both vehicles must be present at the same location for an interchange to take place.

This paper proposes a constraint programming formulation of the JVCRSP that jointly optimizes vehicle and crew routing in the hope of remedying some limitations of a multi-stage model. The formulation overlays crew routing constraints over a traditional vehicle routing problem, and also adds a number of synchronization constraints to link vehicles and crews. In addition, the formulation includes a novel optimization constraint that checks feasibility and bounds crew costs early in the search while the focus is on vehicle routing. The constraint programming model is solved using a large neighborhood search (LNS) that explores both vehicle and crew neighborhoods.

The constraint program is compared to a mixed integer program and a two-stage method that separates the vehicle routing and crew scheduling problems. Experimental results on instances with up to 96 requests and different cost functions indicate that (1) the joint optimization of vehicle and crew routing produces considerable benefits over the sequential method, (2) the constraint program scales significantly better than the mixed integer program, and (3) vehicle interchanges are critical in obtaining high-quality solutions. These preliminary findings indicate that it is now in the realm of optimization technology to jointly optimize vehicle and crew routing and scheduling, and that constraint programming has a key role to play in this promising direction.

The rest of this paper is organized as follows. Section 2 describes the Joint Vehicle and Crew Routing and Scheduling Problem and some high-level modeling decisions. Section 3 outlines the two-stage methodology, while Section 4 describes the mixed integer formulation of the JVCRSP. Section 5 details the constraint programming formulation, two novel global constraints, and the search procedure. Section 6 reports the experimental results. Section 7 reviews related work and Section 8 concludes this paper.

2 Problem Description

The traditional vehicle routing problem with pickup and delivery and time windows (VRPPDTW) [30] consists of a fleet of vehicles, stationed at a common depot, that services pickup and delivery requests before returning to the depot. Every pickup request has one associated delivery request, with an equal but negated demand, that must be served after the pickup and by the same vehicle. In addition, every request has a time window, within which service must begin. Vehicles can arrive earlier than the time window, but must wait until the time window opens before starting service. Given a route, fixed travel times dictate the arrival and departure times of requests along the route.

The JVCRSP considered in this paper overlays, on top of the VRPPDTW, crew routing constraints that explicitly capture the movements of crews. The JVCRSP groups requests by location, and defines travel times between locations, thereby removing the one-to-one mapping between requests and locations found in many routing problems. Vehicles require a crew when moving between two

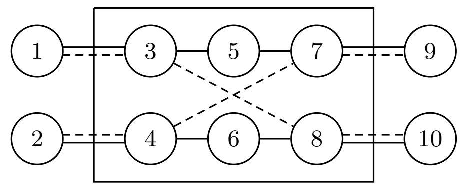

Fig. 1. A location at which vehicle interchange occurs. Nodes 3 to 8 belong to the same location. A vehicle (solid line) and a crew (dotted line) travel from node 1 to node 3, and another vehicle and crew travel from node 2 to node 4. The two vehicles service nodes 3 to 8, while the crews travel to nodes 7 and 8 to depart on a different vehicle. The movements into nodes 3 and 4, and out of nodes 7 and 8 must be synchronized between the vehicles and crews, but movements within the location are independent.

locations, but require no crew when servicing requests within a location. Similarly, crews require a vehicle to move between two locations, but can move independently of vehicles within a location. Crews are restricted to at most one driving shift, which has a maximum duration. No distance or time limitations are placed on crews before and after the driving shift.

The JVCRSP allows crews to switch vehicles at any given location. This idea is illustrated in Fig. 1. Vehicle interchange occurs when a crew moves from a node that is serviced by one vehicle to a node that is serviced by another vehicle, provided that the departure of the second vehicle is after the arrival of the first. Hence vehicle interchange requires explicit modelling of arrival and departure times. The JVCRSP retains the time windows on service start times and permits early arrival, but also allows vehicles to wait after service to depart at a later time. This functionality is necessary to facilitate vehicle interchange.

The objective function of the JVCRSP minimizes a weighted sum of the number of vehicles and crews used, and the total vehicle and crew travel distances.

3 A Two-Stage Method for the JVCRSP

One of the goals of this paper is to assess the benefits of jointly routing and scheduling vehicles and crews. A baseline to determine the potential of joint optimization is a two-stage approach that separates these two aspects by first solving a vehicle routing problem and then a crew pairing problem. The first stage can be solved with any appropriate technology, and this paper uses a large neighborhood search over a constraint programming model (e.g., [29,21,1]). Arrival and departure times are fixed at their earliest possible starting times to produce vehicle schedules. The crew pairing problem is modeled with a set covering problem typical in the airline industry [33]. It is solved using column generation with a resource-constrained shortest path pricing problem, and then a mixed integer programming solver to find integer solutions once the column generation process has converged.

4 A Mixed Integer Program for the JVCRSP

This section sketches a mixed integer programming (MIP) formulation for the JVCRSP. The model consists of a vehicle component and a crew component. The vehicle component is the traditional multi-commodity flow model of the VRPPDTW, which is omitted for familiarity and space reasons. The crew component contains similar routing constraints, but omits travel time and request cover constraints, since these are already included in the vehicle component. The vehicle and crew components are linked via synchronization constraints, which are key aspects of the formulation.

Table 1 lists the relevant inputs and decision variables. The usual vehicle flow variables $\mathrm{veh}_{v,i,j}$ are duplicated for crews; the variable $\mathrm{crew}_{c,i,j}$ denotes whether crew c traverses the edge (i, j). Additionally, the $\mathrm{driver}_{c,i,j}$ variable indicates whether crew c drives the vehicle that traverses the arc (i, j), and is used to synchronize the vehicles and crews in space.

Figure 2 describes the key synchronization constraints that coordinate the vehicles, crews, and drivers in the MIP model. Equations (1) and (2) state that when moving from one location to another, a crew requires a vehicle and a vehicle requires a driver. Equation (3) restricts the driver along an arc to be one of the crews that traverses that arc. Equations (4) and (5) link the arrival, service, and departure times. Equation (6) constrains the start and end nodes to a common arrival and departure time. Equations (7) to (9) synchronize time at request, start, and end nodes, respectively. For every crew c that drives from a request i to any request j, Eq. (10) bounds the start of driving to be no later than the departure time at i. Similarly, Eq. (11) bounds the end of driving. Equation (12) restricts the driving duration of each crew.

5 A Constraint Program for the JVCRSP

This section introduces the constraint programming (CP) formulation for the JVCRSP. It describes the instance data, decision variables, and constraints of the model, and the specifications of two global constraints, and the search procedure.

The data and decision variables are described in Table 2. Like the MIP model, the CP model is divided into a vehicle component and a crew component.

The vehicle component, described in Fig. 3, is based on the standard constraint programming modeling for vehicle routing (e.g., [21,28]). Successor variables $\mathrm{succ}(\cdot)$ capture the sequence of requests along vehicle routes. By artificially joining the end of one route to the start of another (Eqs. (13) and (14)), the successor variables then describe a Hamiltonian cycle, which enables the use of the Circuit global constraint for subtour elimination (Eq. (15)). Equations (16) and (17) track a vehicle along its route, and Eq. (18) determines if a vehicle is used. Equations (19) and (20) order the arrival, service, and departure times at each request. Equation (21) restricts the start and end nodes to one common arrival/departure time. Equation (22) enforces travel times. Symmetry-breaking, load, and pickup and delivery constraints are omitted due to space limitations.

Table 1. Instance data and decision variables of the MIP model.

Name	Description
$T > 0$	Time horizon.
$\mathcal{T} = [0, T]$	Time interval.
$\mathcal{V}$	Set of vehicles.
$\mathcal{C}$	Set of crews.
$\overline{T} \in \mathcal{T}$	Maximum driving duration of a crew.
s	Start node.
e	End node.
$\mathcal{R}$	Set of all requests.
$\mathcal{N} = \mathcal{R} \cup \{s, e\}$	Set of all nodes, including the start and end nodes.
$\mathcal{A}$	Arcs to be traversed by the vehicles and crews, excluding the arc (s, e) representing an unused vehicle/crew.
$\mathcal{L}$	Set of locations, including one depot location.
$l_i \in \mathcal{L}$	Location of $i \in \mathcal{N}$.
$a_i \in \mathcal{T}$	Earliest start of service at $i \in \mathcal{N}$.
$b_i \in \mathcal{T}$	Latest start of service at $i \in \mathcal{N}$.
$t_i \in \mathcal{T}$	Service duration of $i \in \mathcal{N}$.
$\mathrm{veh}_{v,i,j} \in \{0, 1\}$	1 if vehicle $v \in \mathcal{V}$ traverses $(i, j) \in \mathcal{A} \cup \{(s, e)\}$.
$\mathrm{crew}_{c,i,j} \in \{0, 1\}$	1 if crew $c \in \mathcal{C}$ traverses $(i, j) \in \mathcal{A} \cup \{(s, e)\}$.
$\mathrm{driver}_{c,i,j} \in \{0, 1\}$	1 if crew $c \in \mathcal{C}$ drives on $(i, j) \in \mathcal{A}$, $l_i \neq l_j$.
$\mathrm{arr}_{v,i} \in \mathcal{T}$	Arrival time of vehicle $v \in \mathcal{V}$ at $i \in \mathcal{N}$.
$\mathrm{serv}_{v,i} \in [a_i, b_i]$	Start of service by vehicle $v \in \mathcal{V}$ at $i \in \mathcal{N}$.
$\mathrm{dep}_{v,i} \in \mathcal{T}$	Departure time of vehicle $v \in \mathcal{V}$ at $i \in \mathcal{N}$.
$\mathrm{crewTime}_{c,i} \in \mathcal{T}$	Arrival/departure time of crew $c \in \mathcal{C}$ at $i \in \mathcal{N}$.
$\mathrm{driveStart}_c \in \mathcal{T}$	Start time of driving for crew $c \in \mathcal{C}$.
$\mathrm{driveEnd}_c \in \mathcal{T}$	End time of driving for crew $c \in \mathcal{C}$.
$\mathrm{driveDur}_c \in [0, \overline{T}]$	Driving duration of crew $c \in \mathcal{C}$.

$$\mathrm{crew}_{c,i,j} \leq \sum_{v \in \mathcal{V}} \mathrm{veh}_{v,i,j}, \qquad \forall c \in \mathcal{C}, (i,j) \in \mathcal{A}, l_i \neq l_j, \quad (1)$$

$$\sum_{v \in \mathcal{V}} \mathrm{veh}_{v,i,j} = \sum_{c \in \mathcal{C}} \mathrm{driver}_{c,i,j}, \qquad \forall (i,j) \in \mathcal{A}, l_i \neq l_j, \quad (2)$$

$$\mathrm{driver}_{c,i,j} \leq \mathrm{crew}_{c,i,j}, \qquad \forall c \in \mathcal{C}, (i,j) \in \mathcal{A}, l_i \neq l_j, \quad (3)$$

$$\mathrm{arr}_{v,i} \leq \mathrm{serv}_{v,i}, \qquad \forall v \in \mathcal{V}, i \in \mathcal{R}, \quad (4)$$

$$\mathrm{serv}_{v,i} + t_i \leq \mathrm{dep}_{v,i}, \qquad \forall v \in \mathcal{V}, i \in \mathcal{R}, \quad (5)$$

$$\mathrm{arr}_{v,i} = \mathrm{serv}_{v,i} = \mathrm{dep}_{v,i}, \qquad \forall v \in \mathcal{V}, i \in \{s, e\}, \quad (6)$$

$$\mathrm{arr}_{v,i} \leq \mathrm{crewTime}_{c,i} \leq \mathrm{dep}_{v,i}, \qquad \forall v \in \mathcal{V}, c \in \mathcal{C}, i \in \mathcal{R}, \quad (7)$$

$$\mathrm{crewTime}_{c,s} - \mathrm{dep}_{v,s} \leq M_1(2 - \mathrm{veh}_{v,s,j} - \mathrm{crew}_{c,s,j}), \quad \forall v \in \mathcal{V}, c \in \mathcal{C}, j : (s,j) \in \mathcal{A}, \quad (8)$$

$$\mathrm{arr}_{v,e} - \mathrm{crewTime}_{c,e} \leq M_2(2 - \mathrm{veh}_{v,i,e} - \mathrm{crew}_{c,i,e}), \quad \forall v \in \mathcal{V}, c \in \mathcal{C}, i : (i,e) \in \mathcal{A}, \quad (9)$$

$$\mathrm{driveStart}_c - \mathrm{crewTime}_{c,i} \leq M_3\left(1 - \sum_{j:(i,j) \in \mathcal{A}} \mathrm{driver}_{c,i,j}\right), \qquad \forall c \in \mathcal{C}, i \in \mathcal{R} \cup \{s\}, \quad (10)$$

$$\mathrm{crewTime}_{c,i} - \mathrm{driveEnd}_c \leq M_4\left(1 - \sum_{h:(h,i) \in \mathcal{A}} \mathrm{driver}_{c,h,i}\right), \qquad \forall c \in \mathcal{C}, i \in \mathcal{R} \cup \{e\}, \quad (11)$$

$$\mathrm{driveDur}_c = \mathrm{driveEnd}_c - \mathrm{driveStart}_c, \qquad \forall c \in \mathcal{C}. \quad (12)$$

Fig. 2. Core constraints of the MIP formulation.

Table 2. Instance data and the decision variables of the CP model.

Name	Description
$T \in \{1, \ldots, \infty\}$	Time horizon.
$\mathcal{T} = \{0, \ldots, T\}$	Set of time values.
$V \in \{1, \ldots, \infty\}$	Number of vehicles.
$\mathcal{V} = \{1, \ldots, V\}$	Set of vehicles.
$\mathcal{C} = \{1, \ldots, \lvert\mathcal{C}\rvert\}$	Set of crews.
$\mathcal{C}_0 = \mathcal{C} \cup \{0\}$	Set of crews, including a 0 value indicating no crew.
$\overline{T} \in \mathcal{T}$	Maximum driving duration of a crew.
$R \in \{1, \ldots, \infty\}$	Total number of requests.
$\mathcal{R} = \{1, \ldots, R\}$	Set of all requests.
$\mathcal{S} = \{R+1, \ldots, R+V\}$	Set of vehicle start nodes.
$\mathcal{E} = \{R+V+1, \ldots, R+2V\}$	Set of vehicle end nodes.
$\mathcal{N} = \mathcal{R} \cup \mathcal{S} \cup \mathcal{E}$	Set of all nodes.
$\mathcal{N}_0 = \mathcal{N} \cup \{0\}$	Set of all nodes, including a dummy node 0.
$s(v) = R + v$	Start node of vehicle $v \in \mathcal{V}$.
$e(v) = R + V + v$	End node of vehicle $v \in \mathcal{V}$.
$\mathcal{L}$	Set of locations, including one depot location.
$l(i) \in \mathcal{L}$	Location of $i \in \mathcal{N}$.
$d(i, j) \in \mathcal{T}$	Distance and travel time from $i \in \mathcal{N}$ to $j \in \mathcal{N}$.
$a(i) \in \mathcal{T}$	Earliest start of service at $i \in \mathcal{N}$.
$b(i) \in \mathcal{T}$	Latest start of service at $i \in \mathcal{N}$.
$t(i) \in \mathcal{T}$	Service duration of $i \in \mathcal{N}$.
w_1	Cost of using one vehicle.
w_2	Cost of using one crew.
w_3	Cost of one unit of vehicle distance.
w_4	Cost of one unit of crew distance.
$\mathrm{vehUsed}(v) \in \{0, 1\}$	Indicates if vehicle $v \in \mathcal{V}$ is used.
$\mathrm{succ}(i) \in \mathcal{N}$	Successor node of $i \in \mathcal{N}$.
$\mathrm{veh}(i) \in \mathcal{V}$	Vehicle that visits $i \in \mathcal{N}$.
$\mathrm{arr}(i) \in \mathcal{T}$	Arrival time at $i \in \mathcal{N}$.
$\mathrm{serv}(i) \in \{a(i), \ldots, b(i)\}$	Start of service at $i \in \mathcal{N}$.
$\mathrm{dep}(i) \in \mathcal{T}$	Departure time at $i \in \mathcal{N}$.
$\mathrm{crewUsed}(c) \in \{0, 1\}$	Indicates if crew $c \in \mathcal{C}$ is used.
$\mathrm{crewDist}(c) \in \{0, \ldots, \infty\}$	Distance traveled by crew $c \in \mathcal{C}$.
$\mathrm{crewSucc}(c, i) \in \mathcal{N}_0$	Successor of $i \in \mathcal{N}_0$ for crew $c \in \mathcal{C}_0$.
$\mathrm{crewTime}(i) \in \mathcal{T}$	Arrival/departure time of every crew at $i \in \mathcal{N}$.
$\mathrm{driveStart}(c) \in \mathcal{T}$	Start time of driving for crew $c \in \mathcal{C}_0$.
$\mathrm{driveEnd}(c) \in \mathcal{T}$	End time of driving for crew $c \in \mathcal{C}_0$.
$\mathrm{driveDur}(c) \in \{0, \ldots, \overline{T}\}$	Driving duration of crew $c \in \mathcal{C}$.
$\mathrm{driver}(i) \in \mathcal{C}_0$	Driver of vehicle $\mathrm{veh}(i)$ from $i \in \mathcal{R} \cup \mathcal{S}$ to $\mathrm{succ}(i)$, with 0 indicating no driver.

The crew component, depicted in Fig. 4, overlays the vehicle component to upgrade the VRP into a JVCRSP. Crew successor variables $\mathrm{crewSucc}(c, \cdot)$ model a path for every crew c from any start node to any end node via a sequence of arcs that cover the vehicle movements. To allow crews to start and end on any vehicle, the model includes a dummy node 0; crews start and end at this dummy node, whose successor must be a start node and whose predecessor must be an end node. Equations (23) to (27) formalize this idea. The modeling uses the Subcircuit constraint [14] to enforce subtour elimination, since a crew does not visit all nodes. The Subcircuit constraint differs from the Circuit constraint seen in the vehicle component by allowing some nodes to be excluded from the Hamiltonian cycle. The successor of an excluded node is the node itself.

Every node i has an associated $\mathrm{driver}(i)$ variable that denotes the driver of vehicle $\mathrm{veh}(i)$ when it moves from i to its successor $\mathrm{succ}(i)$ at a different location,

$$
\begin{aligned}
&\mathrm{succ}(e(v)) = s(v+1), && \forall v \in \{1,\ldots,V-1\}, && (13)\\
&\mathrm{succ}(e(V)) = s(1), && && (14)\\
&\textsc{Circuit}(\mathrm{succ}(\cdot)), && && (15)\\
&\mathrm{veh}(s(v)) = \mathrm{veh}(e(v)) = v, && \forall v \in \mathcal{V}, && (16)\\
&\mathrm{veh}(\mathrm{succ}(i)) = \mathrm{veh}(i), && \forall i \in \mathcal{R}\cup\mathcal{S}, && (17)\\
&\mathrm{vehUsed}(v) \leftrightarrow \mathrm{succ}(s(v)) \neq e(v), && \forall v \in \mathcal{V}, && (18)\\
&\mathrm{arr}(i) \leq \mathrm{serv}(i), && \forall i \in \mathcal{R}, && (19)\\
&\mathrm{serv}(i) + t(i) \leq \mathrm{dep}(i), && \forall i \in \mathcal{R}, && (20)\\
&\mathrm{arr}(i) = \mathrm{serv}(i) = \mathrm{dep}(i), && \forall i \in \mathcal{S}\cup\mathcal{E}, && (21)\\
&\mathrm{dep}(i) + d(i,\mathrm{succ}(i)) = \mathrm{arr}(\mathrm{succ}(i)), && \forall i \in \mathcal{R}\cup\mathcal{S}. && (22)
\end{aligned}
$$

Fig. 3. Key constraints of the vehicle component of the CP model.

$$
\begin{aligned}
&\mathrm{crewSucc}(c,0) \in \mathcal{S}\cup\{0\}, && \forall c \in \mathcal{C}, && (23)\\
&\mathrm{crewSucc}(c,i) \in \mathcal{R}\cup\{i\}, && \forall c \in \mathcal{C}, i \in \mathcal{S}, && (24)\\
&\mathrm{crewSucc}(c,i) \in \mathcal{R}\cup\mathcal{E}, && \forall c \in \mathcal{C}, i \in \mathcal{R}, && (25)\\
&\mathrm{crewSucc}(c,i) \in \{0,i\}, && \forall c \in \mathcal{C}, i \in \mathcal{E}, && (26)\\
&\textsc{Subcircuit}(\mathrm{crewSucc}(c,\cdot)), && \forall c \in \mathcal{C}, && (27)\\
&l(\mathrm{succ}(i)) = l(i) \leftrightarrow \mathrm{driver}(i) = 0, && \forall i \in \mathcal{R}\cup\mathcal{S}, && (28)\\
&\mathrm{crewSucc}(\mathrm{driver}(i),i) = \mathrm{succ}(i), && \forall i \in \mathcal{R}\cup\mathcal{S}, && (29)\\
&\mathrm{crewUsed}(c) \leftrightarrow \bigvee_{i\in\mathcal{N}_0} \mathrm{crewSucc}(c,i) \neq i \leftrightarrow \bigvee_{i\in\mathcal{R}\cup\mathcal{S}} \mathrm{driver}(i) = c, && \forall c \in \mathcal{C}, && (30)\\
&\mathrm{crewUsed}(c) \geq \mathrm{crewUsed}(c+1), && \forall c \in \{1,\ldots,C-1\}, && (31)\\
&l(\mathrm{crewSucc}(c,i)) = l(i) \vee \mathrm{crewSucc}(c,i) = \mathrm{succ}(i), && \forall c \in \mathcal{C}, i \in \mathcal{R}\cup\mathcal{S}, && (32)\\
&\textsc{CrewShortcut}(\mathrm{succ}(\cdot),\mathrm{crewSucc}(\cdot,\cdot),\mathrm{arr}(\cdot),\mathrm{dep}(\cdot),\mathcal{C},\mathcal{R},\mathcal{S},\mathcal{E},l(\cdot)), && && (33)\\
&\mathrm{arr}(i) \leq \mathrm{crewTime}(i) \leq \mathrm{dep}(i), && \forall i \in \mathcal{N}, && (34)\\
&\mathrm{crewTime}(i) \leq \mathrm{crewTime}(\mathrm{crewSucc}(c,i)), && \forall c \in \mathcal{C}, i \in \mathcal{R}\cup\mathcal{S}, && (35)\\
&\textsc{CrewBound}(\mathrm{crewDist}(c),\mathrm{crewSucc}(c,\cdot),\mathrm{crewTime}(\cdot),\mathcal{R},\mathcal{S},\mathcal{E},d(\cdot,\cdot)), && \forall c \in \mathcal{C}, && (36)\\
&\mathrm{driveStart}(\mathrm{driver}(i)) \leq \mathrm{dep}(i), && \forall i \in \mathcal{R}\cup\mathcal{S}, && (37)\\
&\mathrm{driveEnd}(\mathrm{driver}(i)) \geq \mathrm{arr}(\mathrm{succ}(i)), && \forall i \in \mathcal{R}\cup\mathcal{S}, && (38)\\
&\mathrm{driveDur}(c) = \mathrm{driveEnd}(c) - \mathrm{driveStart}(c), && \forall c \in \mathcal{C}. && (39)
\end{aligned}
$$

Fig. 4. Crew component of the CP model.

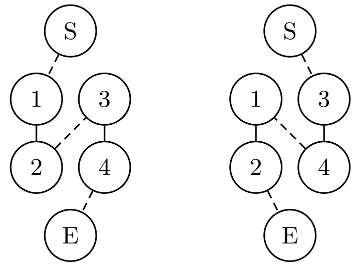

Fig. 5. A partial solution obtained at some stage during search. A crew is known to traverse the arcs $(1, 2)$ and $(3, 4)$, but it is not yet known whether the start node S connects (directly or via a path) to node 1 or node 3, and whether node 4 or node 2 connects to the end node. It is also not known whether such a path exists.

and takes the value 0 when the successor is at the same location, indicating that no driver is necessary (Eq. (28)). Equation (29) requires drivers to move with their vehicles. Equation (30) determines whether a crew is used. Equation (31) is a symmetry-breaking constraint. Equation (32) is a space synchronization constraint that requires crews to move to a node at their current location, or move with a vehicle (to a different location). The CrewShortcut global constraint in Eq. (33) enforces the crew shortcut property, and is detailed in Section 5.2. Equation (34) synchronizes time between vehicles and crews. The time window in this constraint allows a crew to switch vehicle by moving to another node at the same location, provided that this vehicle departs later than this crew. Equation (35) forces crews to only move forward in time. Equation (36) is an optimization constraint, described in Section 5.1, that bounds crew distances and checks whether crews can return to the depot. Equations (37) to (39) determine the driving time of each crew.

The objective function (Eq. (40)) minimizes a weighted sum of the vehicle and crew counts, and the total vehicle and crew travel distances.

$$\min\ w_1 \sum_{v \in \mathcal{V}} \text{vehUsed}(v) + w_2 \sum_{c \in \mathcal{C}} \text{crewUsed}(c) + \\ w_3 \sum_{i \in \mathcal{R} \cup \mathcal{S}} d(i, \text{succ}(i)) + w_4 \sum_{c \in \mathcal{C}} \text{crewDist}(c). \quad (40)$$

5.1 Feasibility and Bounding of Crew Routes

Since crew routing is superimposed on a vehicle routing problem, it is important to determine during the search whether each crew has a feasible route and to compute a lower bound on the crew distance. Consider Fig. 5, which illustrates a partial solution. Here, the crew has been allocated to travel on some arcs but it is not known whether there exists a path from a start node to an end node through these arcs, and how long this path is, if it exists.

To detect infeasibility and to compute lower bounds to crew distances, an optimization constraint can be used [13,10,12,11]. It needs to find a shortest path

$$\min \sum_{i \in \mathcal{R} \cup \mathcal{S}} \sum_{j \in \mathrm{cSucc}(i)} d(i,j) \cdot x_{i,j}, \tag{41}$$

$$\sum_{i \in \mathcal{S}} \sum_{j:(i,j) \in \mathcal{B}} x_{i,j} = 1, \tag{42}$$

$$\sum_{i \in \mathcal{E}} \sum_{h:(h,i) \in \mathcal{B}} x_{h,i} = 1, \tag{43}$$

$$\sum_{h:(h,i) \in \mathcal{B}} x_{h,i} = \sum_{j:(i,j) \in \mathcal{B}} x_{i,j}, \qquad \forall i \in \mathcal{R}, \tag{44}$$

$$\sum_{j:(i,j) \in \mathcal{B}} x_{i,j} \leq 1, \qquad \forall i \in \mathcal{R} \cup \mathcal{S}, \tag{45}$$

$$t_i + d(i,j) - t_j \leq M(1 - x_{i,j}), \qquad (i,j) \in \mathcal{B}, \tag{46}$$

$$x_{i,j} \in [0,1], \qquad (i,j) \in \mathcal{B}, \tag{47}$$

$$t_i \in [\min(\mathrm{cTime}(i)), \max(\mathrm{cTime}(i))], \qquad i \in \mathcal{N}. \tag{48}$$

Fig. 6. The linear programming relaxation for optimization constraint CrewBound.

through specified nodes, which is NP-hard [24,20,9,32]. Instead, the CrewBound constraint solves a linear relaxation, which is presented in Fig. 6.

The set $\mathcal{B}$ represents the current set of arcs that can be traversed by the crew. It is defined as

$$\mathcal{B} = \{(i,j) : i \in \mathcal{R} \cup \mathcal{S}, j \in \mathrm{cSucc}(i), i \neq j\},$$

where the set $\mathrm{cSucc}(i)$ represents the domain of $\mathrm{crewSucc}(c,i)$ for every node $i \in \mathcal{R} \cup \mathcal{S}$ and the appropriate crew c. The set $\mathrm{cTime}(i)$ represents the domain of $\mathrm{crewTime}(i)$ for every node $i \in \mathcal{N}$.

The $x_{i,j}$ variables indicate whether the crew traverses the arc $(i,j) \in \mathcal{B}$, and the t_i variables represent the arrival time to node i.

The objective function minimizes the total distance (Eq. (41)). Equations (42) to (44) are the flow constraints, which ensure the existence of a path from a start node to an end node. Equation (45) is a redundant constraint that strengthens the linear program. Equation (46) is the time-based subtour elimination constraint, where M is a big-M constant. Equations (47) and (48) restrict the domains of the $x_{i,j}$ and t_i variables respectively.

5.2 Symmetry-Breaking within Locations

The constraint program also uses the global constraint CrewShortcut to remove symmetries in the crew routes. The intuition is best explained using Fig. 1. Recall that, in this example, one crew moves from node 3 to 8 and the other crew moves from node 4 to 7. Both crews thus shortcut the intermediate nodes, which are only visited by vehicles. The global constraint CrewShortcut ensures that crews never visit intermediate nodes, but move directly from incoming to outgoing nodes at a location. This pruning eliminates symmetric solutions in which crews visit various sequences of intermediate nodes within a location.

Let $\mathrm{pred}(i) \in \mathcal{N}$ be the predecessor of a node $i \in \mathcal{N}$. The constraint implements simple propagation rules, such as,

$$l(\mathrm{pred}(i)) = l(i) = l(\mathrm{succ}(i)) \leftrightarrow \bigwedge_{c \in \mathcal{C}} \mathrm{crewSucc}(c, i) = i, \qquad \forall i \in \mathcal{R}, \quad (49)$$

which ensures that crews never visit a node whose predecessor and successor are at the same location as the node itself.

5.3 The Search Procedure

This section briefly describes the labeling procedure and the LNS procedure.

The Labeling Procedure The labeling procedure first assigns vehicle routes, and then crew routes.

To determine vehicle routes, the search procedure considers each vehicle in turn and assigns the successor variables from the start node to the end node. The value selection strategy first chooses nodes at the same location ordered by earliest service time, and then considers nodes at other locations, also ordered by earliest service time.

To assign crew routes, the labeling procedure begins by ordering all nodes $i \in \mathcal{R} \cup \mathcal{S}$ that have an unlabeled $\mathrm{driver}(i)$ variable by earliest departure time. The first of these nodes is assigned a driver, and then a path from this node to any end node is constructed for this crew by labeling both the crew successor and the driver variables. This process is repeated until all $\mathrm{driver}(\cdot)$ variables are labeled. The value selection heuristic for crew successor variables favors the node that is visited next by the current vehicle of the crew, provided that the crew can drive from this node; otherwise, the successor is chosen randomly.

The LNS Procedure The LNS procedure randomly explores two neighborhoods:

- **Request Neighborhood**: This neighborhood relaxes a collection of pickup and delivery requests for insertion, in addition to all crew routes;
- **Crew Neighborhood**: This neighborhood fixes all vehicle routes, and relaxes a number of crew routes.

6 Experimental Results

This section compares the solution quality and features of the MIP and CP models, and the two-stage procedure.

The Instances The instances are generated to capture the essence of applications in humanitarian and military logistics. These problems typically feature fewer locations than traditional vehicle routing applications, but comprise multiple requests at each location. Both the locations and the time windows are generated randomly. Table 3 describes the size of the instances. The maximum number of vehicles and crews are not limited in the instances.

Table 3. The number of locations and requests in each of the ten JVCRSP instances.

	A	B	C	D	E	F	G	H	I	J
$\vert\mathcal{L}\vert$	7	7	7	11	11	12	11	11	5	6
$\vert\mathcal{R}\vert$	12	16	20	40	46	46	40	80	54	96

The Algorithms As mentioned previously, the two-stage method uses a constraint programming model for its first step. The model is solved using a large neighborhood search with a time limit of one minute, and the first step takes the best VRP solution out of ten runs. The crew pairing problem is solved optimally using the output of the first stage. All instances are solved to optimality in under 0.02 seconds. Hence the two-stage method requires a total of ten minutes. Obviously, the VRP runs can be performed in parallel.

The MIP model is solved using Gurobi 6.0.0 on an eight-core CPU for twelve hours with default parameters. The linear programming relaxation is rather weak and the dual bounds are not informative, which is why they are not reported here. An alternative MIP model that folds Eqs. (8) and (9) into Eq. (7) by modifying the network is also tested, but is inferior to the one presented.

The CP model is implemented in Objective-CP [31], and initialized with the same VRP solution as in the two-stage approach. This allows us to assess the benefits of the joint optimization reliably, since the second stage and the CP model are seeded with the same vehicle routes. The LNS procedure over the constraint model is then performed ten times for five minutes each. The model essentially requires a total of sixty minutes, which includes ten minutes to produce the initial VRP solution. Of course, all these computations can be executed in parallel in six minutes of wall clock time.

Table 4. Solutions found by the three models with equal vehicle and crew costs ($w_1 = w_2 = 1000$, $w_3 = w_4 = 1$). The objective value columns separate vehicle and crew costs in parenthesis. Time (in seconds) is provided for the MIP model. The last column lists the percentage improvement of the CP model compared against the two-stage method.

	Two-Stage	MIP		CP	
	Objective Value	Objective Value	Time	Objective Value	%
A	8328 (2086 + 6242)	8318 (2086 + 6232)	5	8318 (2086 + 6232)	-0.1%
B	7198 (2065 + 5133)	6154 (2063 + 4091)	27132	6167 (2068 + 4099)	-16.7%
C	8216 (2079 + 6137)	6192 (2088 + 4104)	37254	6200 (2079 + 4121)	-32.5%
D	14879 (3298 + 11581)	-	-	13855 (3302 + 10553)	-7.4%
E	18105 (2275 + 15830)	-	-	17181 (2278 + 14903)	-5.4%
F	26334 (4406 + 21928)	-	-	24570 (4412 + 20158)	-7.2%
G	14110 (3351 + 10759)	-	-	13074 (3355 + 9719)	-7.9%
H	25617 (6585 + 19032)	-	-	24975 (6587 + 18388)	-2.6%
I	8894 (2316 + 6578)	-	-	7795 (2309 + 5486)	-14.1%
J	16932 (3710 + 13222)	-	-	13877 (3710 + 10167)	-22.0%

Table 5. Best solutions from the three models with higher crew costs ($w_1 = 1000$, $w_2 = 5000$, $w_3 = w_4 = 1$).

	Two-Stage	MIP		CP	
	Objective Value	Objective Value	Time	Objective Value	%
A	32328 (2086 + 30242)	32318 (2086 + 30232)	5	32318 (2086 + 30232)	0.0%
B	27198 (2065 + 25133)	22154 (2056 + 20098)	8352	22167 (2068 + 20099)	-22.7%
C	32216 (2079 + 30137)	28198 (3086 + 25112)	4869	22200 (2079 + 20121)	-45.1%
D	58879 (3298 + 55581)	-	-	53905 (3317 + 50588)	-9.2%
E	78105 (2275 + 75830)	-	-	73175 (2280 + 70895)	-6.7%
F	110334 (4406 + 105928)	-	-	100502 (4413 + 96089)	-9.8%
G	54110 (3351 + 50759)	-	-	43996 (3353 + 40643)	-23.0%
H	97617 (6585 + 91032)	-	-	92928 (6585 + 86343)	-5.0%
I	32894 (2316 + 30578)	-	-	27817 (2326 + 25491)	-18.3%
J	64932 (3710 + 61222)	-	-	49930 (3710 + 46220)	-30.0%

Table 6. Best solutions from the three models with higher vehicle costs ($w_1 = 5000$, $w_2 = 1000$, $w_3 = w_4 = 1$).

	Two-Stage	MIP		CP	
	Objective Value	Objective Value	Time	Objective Value	%
A	16328 (10086 + 6242)	16318 (10086 + 6232)	7	16318 (10086 + 6232)	-0.1%
B	15198 (10065 + 5133)	14154 (10056 + 4098)	2327	14167 (10068 + 4099)	-7.3%
C	16216 (10079 + 6137)	14200 (10091 + 4109)	32135	14200 (10079 + 4121)	-14.2%
D	26879 (15298 + 11581)	-	-	24946 (15303 + 9643)	-7.7%
E	26105 (10275 + 15830)	-	-	25172 (10278 + 14894)	-3.7%
F	42334 (20406 + 21928)	-	-	40597 (20412 + 20185)	-4.3%
G	26110 (15351 + 10759)	-	-	24008 (15353 + 8655)	-8.8%
H	49617 (30585 + 19032)	-	-	48911 (30585 + 18326)	-1.4%
I	16894 (10316 + 6578)	-	-	15800 (10326 + 5474)	-6.9%
J	28932 (15710 + 13222)	-	-	25930 (15710 + 10220)	-11.6%

Solution Quality Tables 4 to 6 depict the experimental results for various vehicle and crew costs. The tables only report the CPU time for the MIP model, since the two-stage and CP models have been given short execution times, as explained above. The experimental results highlight a few important findings:

1. Jointly optimizing vehicle and crew routing and scheduling achieves significant benefits. On the above instances, the average benefit of the CP model over the two-stage approach is 12%. When crew costs are higher, the benefits can exceed 45%.
2. The benefits obviously increase when the crew costs increase. However, even when vehicle costs are five times as large as crew costs, the benefits are still substantial and exceed 5% in more than half of the instances.
3. On the smaller instances, the MIP marginally outperforms the CP model when run over a significantly longer period of time.
4. The MIP model contains numerous big-M synchronization constraints that weaken its linear relaxation, making it difficult to solve. It can only prove

optimality on the trivial instance A, and it also fails to scale to large instances, contrary to the CP model, which quickly finds high-quality solutions.

The Impact of Vehicle Interchanges One of the main advantages of jointly optimizing vehicle and crew routing and scheduling is the additional flexibility for vehicle interchanges. For example, instance C in Tables 4 to 6 highlight some of the benefits of this flexibility. The CP model is able to delay vehicles to favor interchanges, thereby decreasing crew costs significantly.

Table 7 quantifies vehicle interchanges for the instances with equal vehicle and crew costs. It reports the total number of crews required in the solution, the number of crews participating in vehicle interchanges, the total number of vehicle interchanges, and the average number of interchanges per crew that switches vehicle. These results indicate that the constraint programming model allows more vehicle interchanges on average. For the larger instances, each crew travels on almost three vehicles on average. The total number of vehicle interchanges is roughly the same for the CP model and the two-stage approach, but the CP model produces solutions with fewer crews.

Table 7. Statistics on vehicle interchange in the solutions to the runs with equal vehicle and crew costs. For both the two-stage and the CP models, this table provides the number of crews used in the solution, the number of crews that participates in vehicle interchange, the total number of interchanges, and the average number of interchanges for each crew that undertakes interchange.

	Two-Stage				CP			
	Crews Used	Intchng Crews	Vehicle Intchngs	Average Intchngs	Crews Used	Intchng Crews	Vehicle Intchngs	Average Intchngs
A	6	2	2	1.0	6	3	3	1.0
B	5	2	2	1.0	4	3	3	1.0
C	6	3	3	1.0	4	4	4	1.0
D	11	8	12	1.5	10	7	7	1.0
E	15	9	14	1.6	14	12	21	1.8
F	21	20	35	1.8	19	19	36	1.9
G	10	8	9	1.1	9	8	11	1.4
H	18	17	28	1.6	17	16	29	1.8
I	6	5	5	1.0	5	3	5	1.7
J	12	11	15	1.4	9	8	15	1.9

The Impact of the Joint Vehicle and Crew Routing and Scheduling It is also interesting to notice in Tables 4 to 6 that the CP model often prefers vehicle routes whose costs are higher than those of the two-stage approach. This is the case despite the fact that the CP search is initialized with the first-stage VRP solution. This indicates that the CP model trades off crews costs for vehicle costs. Running a two-stage version of the CP model, in which the vehicle routes (but not the vehicle schedules) are fixed, also confirms these benefits. The CP model significantly outperforms this two-stage CP approach on a number of instances. These results indicate that both the flexibility in interchanges and the joint optimization are key benefits of the CP model.

7 Related Work

JVCRSPs have not attracted very much interest in the literature at this point, probably due to their inherent complexity [5]. This contradicts with the numerous studies of VRPs and related problems (e.g., [30,23,2]). This section reviews the most relevant work.

A JVCRSP is considered in [22], in which vehicles transport teams to service customers, and are able to move without any team on board. The problem features three types of tasks that must be serviced in order, and all customers have one task of each type. Each team can only service one compatible type of task. The MIP formulation has variables indexed by five dimensions, and is intractable. The paper also develops a simple iterative local search algorithm embedded within a particle swarm metaheuristic. This approach is very specific and cannot accommodate side constraints easily.

A JVCRSP modeling a mail distribution application is solved using a two-stage heuristic column generation approach in [19]. This model features multiple depots at which vehicle routes begin and end. In the first stage, trips that begin and end at these depots are computed. The second stage takes a scheduling approach and assigns vehicles and crews to the trips. Vehicle interchange can only occur at the depots at the start or end of a trip. In addition, the model features a 24-hour cyclic time period, and variables indexed by discretized time blocks. Since the solution method is based on two stages, it cannot jointly optimize vehicles and crews. Observe also that the CP formulation in our paper does not require time-indexed variables, and allows for vehicle interchange at all locations.

Another JVCRSP considers European legislation and includes relay stations where drivers are allowed to rest and interchange vehicles [8]. Upon reaching a relay station, the vehicle must wait a fixed amount of time. Vehicle interchange can only occur if other drivers arrive at this relay station during this time interval. The problem also provides a shuttle service, separate to the fleet of vehicles, that can be used to move drivers between relay stations. The problem is solved using a two-stage LNS method. In the first stage, a VRP is solved and the resulting routes form the customers of a VRP in the second stage, in which the crews perform the role of vehicles. Observe that this approach also cannot jointly optimize vehicle and crew routing. The model features several fixed parameters, such as the duration during which a vehicle waits at a relay station, and a search procedure that only explores a limited number of nearby relay stations. Both these restrictions reduce the search space, but negatively impact vehicle interchange, leading the authors to conclude that allowing for vehicle interchange does not significantly improve the objective value. Our model proves the contrary since it does not fix the duration that a crew waits, and can explore all locations for vehicle interchange.

JVCRSPs belong in the class of problems known as the VRPs with multiple synchronization constraints [5]. Synchronization is a feature present in certain VRPs, in which decisions about one object (e.g., vehicle, route, request) imply actions that may or must be taken on other objects. A complex VRP with synchronization is the VRP with Trailers and Transshipments (VRPTT) [7,4]. It features two vehicle classes: lorries which can move independently, and trailers which must be towed by an accompanying lorry. All lorries begin at a single depot with or without a trailer. A lorry can detach its trailer at transshipment locations in order to visit customers who are unable to accommodate a trailer (e.g., due to size). Lorries can transfer load into and/or attach with any trailer at any transshipment location. A lorry that has detached its trailer can also return to the depot without reattaching a trailer, leaving its trailer behind at a transshipment location to be collected by another lorry at a future time. Several sophisticated MIP formulations are presented, which are solved using branch-and-cut on instances with up to eight customers, eight transshipment locations and eight vehicles. The VRPTT can be reformulated into a JVCRSP by casting crews as lorries with zero capacity, and vehicles as trailers [6].

The VRPTT is most closely related to our JVCRSP model, since lorry routes and trailer routes are jointly computed, and the search space is not artificially limited. However, one major difference is that the VRPTT incorporates load synchronization, which the JVCRSP does not consider.

Finally, observe that vehicle and crew *scheduling* problems (VCSPs) are thoroughly studied (e.g., [18,27,17,15,16,3,25,26]). VCSPs aim to assign vehicles and crews to a predetermined set of trips, with each trip consisting of a fixed route, and usually fixed arrival and departure times. Trips in VCSPs correspond to parts of a route in JVCRSPs, which are not available a priori, but instead, must be computed during search, thereby increasing the computational challenges.

8 Conclusion

Motivated by applications arising in humanitarian and military logistics, this paper studied a Joint Vehicle and Crew Routing and Scheduling Problem, in which crews are able to interchange vehicles, resulting in interdependent vehicle routes and crew routes. The paper proposed a constraint programming model that overlays crew routing constraints over a standard vehicle routing problem. The model uses a novel optimization constraint to detect infeasibility and to bound crew objectives, and a symmetry-breaking global constraint. The model is compared to a sequential method and a mixed integer programming model. Experimental results demonstrate significant benefits of the constraint programming model, which reduces costs by 12% on average compared to the two-stage method, and scales significantly better than the mixed integer programming model. The benefits of the constraint programming model are influenced by both the additional flexibility in vehicle interchanges and in trading crew costs for vehicle costs. Future work will investigate more sophisticated search techniques and global constraints to scale to larger instances.

References

1. Bent, R., Van Hentenryck, P.: A two-stage hybrid local search for the vehicle routing problem with time windows. Transportation Science 38(4), 515–530 (2004)
2. Berbeglia, G., Cordeau, J.F., Gribkovskaia, I., Laporte, G.: Static pickup and delivery problems: a classification scheme and survey. TOP 15(1), 1–31 (2007)
3. Cordeau, J.F., Stojković, G., Soumis, F., Desrosiers, J.: Benders decomposition for simultaneous aircraft routing and crew scheduling. Transportation Science 35(4), 375–388 (2001)
4. Drexl, M.: On some generalized routing problems. Ph.D. thesis, RWTH Aachen University, Aachen (2007)
5. Drexl, M.: Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints. Transportation Science 46(3), 297–316 (2012)
6. Drexl, M.: Applications of the vehicle routing problem with trailers and transshipments. European Journal of Operational Research 227(2), 275 – 283 (2013)
7. Drexl, M.: Branch-and-cut algorithms for the vehicle routing problem with trailers and transshipments. Networks 63(1), 119–133 (2014)
8. Drexl, M., Rieck, J., Sigl, T., Press, B.: Simultaneous vehicle and crew routing and scheduling for partial- and full-load long-distance road transport. BuR - Business Research 6(2), 242–264 (2013)
9. Dreyfus, S.E.: An appraisal of some shortest-path algorithms. Operations Research 17(3), 395–412 (1969)
10. Focacci, F., Lodi, A., Milano, M.: Embedding relaxations in global constraints for solving TSP and TSPTW. Annals of Mathematics and Artificial Intelligence 34(4), 291–311 (2002)
11. Focacci, F., Lodi, A., Milano, M.: Optimization-oriented global constraints. Constraints 7(3-4), 351–365 (2002)
12. Focacci, F., Lodi, A., Milano, M.: Exploiting relaxations in CP. In: Milano, M. (ed.) Constraint and Integer Programming, Operations Research/Computer Science Interfaces Series, vol. 27, pp. 137–167. Springer US (2004)
13. Focacci, F., Lodi, A., Milano, M., Vigo, D.: Solving TSP through the integration of OR and CP techniques. Electronic Notes in Discrete Mathematics 1(0), 13 – 25 (1999)
14. Francis, K.G., Stuckey, P.J.: Explaining circuit propagation. Constraints 19(1), 1–29 (2014)
15. Freling, R., Huisman, D., Wagelmans, A.: Applying an integrated approach to vehicle and crew scheduling in practice. In: Voß, S., Daduna, J. (eds.) Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems, vol. 505, pp. 73–90. Springer Berlin Heidelberg (2001)
16. Freling, R., Huisman, D., Wagelmans, A.: Models and algorithms for integration of vehicle and crew scheduling. Journal of Scheduling 6(1), 63–85 (2003)
17. Freling, R., Wagelmans, A., Paixão, J.: An overview of models and techniques for integrating vehicle and crew scheduling. In: Wilson, N. (ed.) Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems, vol. 471, pp. 441–460. Springer Berlin Heidelberg (1999)
18. Haase, K., Desaulniers, G., Desrosiers, J.: Simultaneous vehicle and crew scheduling in urban mass transit systems. Transportation Science 35(3), 286–303 (2001)
19. Hollis, B., Forbes, M., Douglas, B.: Vehicle routing and crew scheduling for metropolitan mail distribution at Australia Post. European Journal of Operational Research 173(1), 133 – 150 (2006)

20. Ibaraki, T.: Algorithms for obtaining shortest paths visiting specified nodes. SIAM Review 15(2), 309–317 (1973)
21. Kilby, P., Prosser, P., Shaw, P.: A comparison of traditional and constraint-based heuristic methods on vehicle routing problems with side constraints. Constraints 5(4), 389–414 (2000)
22. Kim, B.I., Koo, J., Park, J.: The combined manpower-vehicle routing problem for multi-staged services. Expert Systems with Applications 37(12), 8424 – 8431 (2010)
23. Laporte, G.: What you should know about the vehicle routing problem. Naval Research Logistics (NRL) 54(8), 811–819 (2007)
24. Laporte, G., Mercure, H., Norbert, Y.: Optimal tour planning with specified nodes. RAIRO - Operations Research - Recherche Opérationnelle 18(3), 203–210 (1984)
25. Mercier, A., Cordeau, J.F., Soumis, F.: A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. Computers & Operations Research 32(6), 1451 – 1476 (2005)
26. Mercier, A., Soumis, F.: An integrated aircraft routing, crew scheduling and flight retiming model. Computers & Operations Research 34(8), 2251–2265 (2007)
27. Mesquita, M., Paias, A.: Set partitioning/covering-based approaches for the integrated vehicle and crew scheduling problem. Computers & Operations Research 35(5), 1562 – 1575 (2008), part Special Issue: Algorithms and Computational Methods in Feasibility and Infeasibility
28. Rousseau, L.M., Gendreau, M., Pesant, G.: Using constraint-based operators to solve the vehicle routing problem with time windows. Journal of Heuristics 8(1), 43–58 (2002)
29. Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: Maher, M., Puget, J.F. (eds.) Principles and Practice of Constraint Programming — CP98, Lecture Notes in Computer Science, vol. 1520, pp. 417–431. Springer Berlin Heidelberg (1998)
30. Toth, P., Vigo, D.: The Vehicle Routing Problem. Society for Industrial and Applied Mathematics (2002)
31. Van Hentenryck, P., Michel, L.: The Objective-CP optimization system. In: Schulte, C. (ed.) Principles and Practice of Constraint Programming, Lecture Notes in Computer Science, vol. 8124, pp. 8–29. Springer Berlin Heidelberg (2013)
32. Volgenant, T., Jonker, R.: On some generalizations of the travelling-salesman problem. The Journal of the Operational Research Society 38(11), pp. 1073–1079 (1987)
33. Yu, G.: Operations Research in the Airline Industry. International Series in Operations Research & Management Science: 9, Springer US (1998)