Integer programs for a simultaneous vehicle routing and crew scheduling problem

Mauro Lucci
Daniel Severin
mlucci@fceia.unr.edu.ar
daniel@fceia.unr.edu.ar
CONICET - Universidad Nacional de Rosario
Rosario, Argentina

Paula Zabala
pzabala@dc.uba.ar
ICC, Universidad de Buenos Aires - CONICET
Buenos Aires, Argentina

ABSTRACT

In this work, we present two IP formulations for a simultaneous vehicle routing and crew scheduling problem, involving pickup-and-delivery requests with multiple time windows. Crews are composed of 1 or 2 drivers and any of them can descend in specific locations to rest or change trucks, which offers more planning options but creates a high interdependence. We perform tests on instances of up to 8 requests and a planning horizon of 1 week, obtaining that one of the formulations outperforms the other. These results are improved when a heuristic that provides an initial solution is used.

KEYWORDS

integer program, simultaneous, vehicle routing, crew scheduling

1 INTRODUCTION

A simultaneous *vehicle routing and crew scheduling problem* (VRCSP) consists of planning routes for a fleet of vehicles and scheduling their crews, where the vehicle-crew correspondence is dynamic through time, see e.g. [2]. Unlike the common assumption in which a crew operates the entire route in the same vehicle, the exchange of drivers allows a more efficient use of the fleet, since the rest constraints imposed on the drivers are not embedded in those associated with the vehicles. In contrast, a high synchronisation between vehicles and drivers is required.

In this work, we present two integer programming formulations that model a simultaneous VRCSP. Pickup-and-delivery requests with daily time windows have to be fulfilled over a weekly planning horizon by a fleet of homogeneous trucks and drivers. Here, crews can be composed of one or two drivers, and any of them can descend from the truck in specific locations. Thus, trucks play an additional role beyond carrying items: they can be used to transport drivers across the locations. For instance, a truck-driver couple might pause a delivery, travel somewhere else to pick up an additional driver and take her/him to another location, and then resume the delivery. Additionally, drivers are allowed to travel across the locations with non-company *shuttles*, in exchange for paying an extra rate.

2 PROBLEM DESCRIPTION

Let V be a set of homogeneous vehicles, D a set of drivers, and L a set of locations. Each $d \in D$ and $v \in V$ stay in an initial location $l_d \in L$ and $l_v \in L$, respectively. We denote $\text{dur}_V(l_1, l_2)$ and $\text{dur}_S(l_1, l_2)$ to the duration of a trip between the locations l_1 and l_2 in a company's vehicle and in a shuttle, respectively, and $\text{cost}_V(l_1, l_2)$ and $\text{cost}_S(l_1, l_2)$ to their costs.

Let R be a set of pickup-and-delivery requests. Each $r \in R$ demands shipping an item from a pickup location $l_r^p \in L$ to a delivery location $l_r^d \in L$, according to the following time constraints. The pickup of r must begin during the pickup time window $[a_r^p, b_r^p]$. The time window repeats everyday (at the same time) from an initial pickup day day_r^p, e.g. everyday from the third day of the planning horizon and during 8 am to 11 am. In the same way, there are a delivery time window $[a_r^d, b_r^d]$ and an initial delivery day day_r^d, for each request r. Each day of delay in the delivery of the request r carries a penalty, c_r. We use s_r^p and s_r^d to indicate the times taken to load or unload r respectively. Each vehicle can transport any of the requests, but can only do so one at a time. All the requests must be fulfilled within a planning horizon of H days.

In addition, certain work regulations for drivers must be complied, such as the maximum number of working hours and consecutive working days.

The objective is to provide a planning, that is, a set of routes for the vehicles and a schedule for the crew, at minimum operation cost. This cost is composed of the travel costs and the penalties in the delay of deliveries.

3 LITERATURE REVIEW

Vehicle routing problems in accordance with drivers' hours of service regulations have been studied since the 2000s, some heuristic approaches can be found in [3, 9] and exact approaches in [4, 10]. Unlike our problem, all these works assume that the same driver operates the entire route, i.e. no driver/vehicle changes are allowed.

Although the fixed vehicle-crew correspondence is barely abandoned throughout the literature on VRPs, there are various works that effectively deal with simultaneous VRCSPs.

These complex problems are commonly addressed with multi-stage algorithms, typically, routes for the vehicles are defined first and routes for the drivers are defined later, consistent with the decisions made in the previous stage. This approach can be found for example in [2, 8]. Clearly, multi-stage algorithms do not guarantee global optimality.

With regard to exact approaches for simultaneous VRCSPs, [6] proposed a constraint programming model and a mixed integer programming model, and [1] developed a mixed integer programming formulation and designed a branch-and-cut algorithm. Unlike the problem addressed in this work, both are restricted to single shifts, i.e. a multi-day planning horizon is not considered, and driver/vehicle changes are only possible in a limited set of locations.

A recent analysis concerning crews with more than one driver in the road transport sector is given by [5]. This work investigates under which conditions it is convenient to use single or team driving in European road freight transport, and concludes that the latter is beneficial for longer routes. However, a fixed vehicle-crew correspondence is assumed, and one driver of the team can rest while the other drives, which differs from our approach where both drivers are considered on service.

4 MATHEMATICAL MODEL

Our integer programming formulations are based on two weighted digraphs that model the vehicle and driver routes as certain directed paths. Also, a relationship is defined between the directed paths to keep both, trucks and drivers, synchronised in space and time.

We follow the usual definitions and notations from graph theory (see [11]). In particular, given a weighted digraph $G = (N, E, w)$, we refer to a directed path p from a node u to a node v as a (u, v)-directed path, the subset of arcs in p is denoted as $E(p)$ and the cost of p, $w(p)$, is given by $\sum_{e \in E(p)} w(e)$.

The constructions presented here use a time discretization of one hour: an instant of time is given by an integer t such that $0 \le t \le 24H$. Functions $day(t)$ and $hour(t)$ give the day and hour of the instant t respectively, i.e. $day(t) = \lfloor t/24 \rfloor$ and $hour(t) = t - 24day(t)$.

4.1 Vehicle routes

We define a weighted digraph $G_V = (N_V, E_V, w_V)$ as follows. The node set N_V has a node $n^r_{t,l}$ for each $l \in L$, $r \in R \cup \{0\}$ and $t \in \mathbb{Z}$ such that $0 \le t \le 24H$, and two special nodes: the source n_s and the sink $n_{\tilde{s}}$. The node $n^r_{t,l}$ represents the instant of time t in the location l loaded with the item demanded by request r if $r \in R$, and empty if $r = 0$. The arc set E_V is composed of the following sets:

- E_{VREST} has an arc $(n^r_{t,l}, n^r_{t+1,l})$ for each $l \in L$, $r \in R \cup \{0\}$ and $t \in \mathbb{Z}$ such that $0 \le t \le 24H - 1$. They represent a vehicle remaining parked for one hour.
- E_{VTRIP} has an arc $(n^r_{t,l_1}, n^r_{t+\Delta,l_2})$ for each pair of adjacent locations $l_1, l_2 \in L$, and for each $r \in R \cup \{0\}$ and $t \in \mathbb{Z}$ such that $0 \le t \le 24H - \Delta$, where $\Delta = \mathrm{dur}_V(l_1, l_2)$. They represent a vehicle travelling between locations l_1 and l_2.
- For all $r \in R$, E^r_{VPICK} has an arc $(n^0_{t,l^p_r}, n^r_{t+s^p_r,l^p_r})$ for each $t \in \mathbb{Z}$ such that $0 \le t \le 24H - s^p_r$ and the pickup time window of r is respected, i.e. $day(t) \ge day^p_r$ and $a^p_r \le hour(t) \le b^p_r$. They represent a vehicle loading an item.
- For all $r \in R$, E^r_{VDELI} has an arc $(n^r_{t,l^d_r}, n^0_{t+s^d_r,l^d_r})$ for each $t \in \mathbb{Z}$ such that $0 \le t \le 24H - s^d_r$ and the delivery time window of r is respected, i.e. $day(t) \ge day^d_r$ and $a^d_r \le hour(t) \le b^d_r$. They represent a vehicle unloading an item.
- E_{VSOUR} and E_{VSINK} have arcs $(n_s, n^0_{0,l})$ and $(n^0_{24H,l}, n_{\tilde{s}})$, respectively, for all $l \in L$. They are useful to model truck routes as $(n_s, n_{\tilde{s}})$-directed paths.

The weight vector w_V is given by:

$$w_V(e) = \begin{cases} \mathrm{cost}_V(l_1, l_2) & \text{if } e = (n^r_{t_1,l_1}, n^r_{t_2,l_2}) \in E_{VTRIP} \\ (day(t_1) - day^d_r)c_r & \text{if } e = (n^r_{t_1,l}, n^0_{t_2,l}) \in E^r_{VDELI} \\ 0 & \text{otherwise} \end{cases}$$

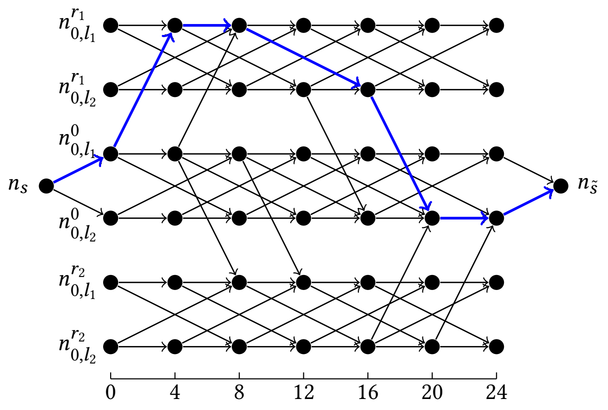

Figure 1: Example of G_V and a $(n_s, n_{\tilde{s}})$-directed path.

The first row in the equation given above considers the cost of travelling among locations and the second row penalizes late deliveries proportional to the number of surplus days.

Observe that G_V is an acyclic digraph, i.e. without directed cycles. Also, every $(n_s, n_{\tilde{s}})$-directed path p that passes through a node $n^0_{0,l}$ represents a vehicle route for some vehicle v with $l_v = l$, and vice versa. In fact, p alternates between a pickup arc and a delivery arc of the same request, which means that the vehicle transports just one item at a time and every loaded item is eventually unloaded. Also, $w_V(p)$ represents the travel cost and the delay cost of the vehicle route.

We present an example of this construction in Figure 1. Let us consider 2 locations l_1 and l_2 with $\mathrm{dur}_V(l_1, l_2) = 8$, a planning horizon of 1 day discretized every 4 hours, and the following 2 requests:

- r_1 with $l^p_{r_1} = l_1$, $day^p_{r_1} = 0$, $[a^p_{r_1}, b^p_{r_1}] = [0, 4]$, $l^d_{r_1} = l_2$, $day^d_{r_1} = 0$, $[a^d_{r_1}, b^d_{r_1}] = [12, 16]$ and $s^p_{r_1} = s^d_{r_1} = 4$.
- r_2 with $l^p_{r_2} = l_1$, $day^p_{r_2} = 0$, $[a^p_{r_2}, b^p_{r_2}] = [4, 8]$, $l^d_{r_2} = l_2$, $day^d_{r_2} = 0$, $[a^d_{r_2}, b^d_{r_2}] = [16, 20]$ and $s^p_{r_2} = s^d_{r_2} = 4$.

The top nodes have the superscript r_1 and the bottom ones, r_2. Nodes at the same height correspond to the same location, but at a later instant of time. The blue $(n_s, n_{\tilde{s}})$-directed path represents a vehicle that starts in l_1, loads the item demanded by r_1 from 0 to 4, parks from 4 to 8, trips to l_2 from 8 to 16, unloads the item demanded by r_1 from 16 to 20 and parks from 20 to 24. The cost of this path is just $\mathrm{cost}_V(l_1, l_2)$, since the delivery of r_1 is done on schedule.

4.1.1 *Alternative digraph.*

The previous digraph presents the disadvantage that it quickly grows with the number of requests. We now propose an alternative digraph $\tilde{G}_V$ whose size is independent from that number.

The construction is very similar, but it uses only two superscripts: 0 and 1. Now, a node $n^r_{t,l}$ represents the instant of time t in the location l loaded with some item if $r = 1$, and empty if $r = 0$, i.e. nodes no longer distinguish the request. We need to redefine E_V, which now is a multiset. Both E_{VREST} and E_{VTRIP} are only defined for $r \in \{0, 1\}$. Each arc $(n^0_{t,l}, n^r_{t',l}) \in E^r_{VPICK}$ is now $(n^0_{t,l}, n^1_{t',l})$, and each arc $(n^r_{t,l}, n^0_{t',l}) \in E^r_{VDELI}$ is now $(n^1_{t,l}, n^0_{t',l})$.

In this alternative digraph, a $(n_s, n_{\tilde{s}})$-directed path no longer guarantees alternation between pickup arcs and delivery arcs of the same request. For example, a directed path might refer to a vehicle that pick up r_1, deliver r_2, pick up r_2, and then deliver r_1. Thus, additional restrictions, called *pairing constraints*, have to be handled externally.

4.2 Driver routes

Now, we define a weighted digraph $G_D = (N_D, E_D, w_D)$. The node set N_D has a node $n_{t,l}$ for each $l \in L$ and $t \in \mathbb{Z}$ such that $0 \leq t \leq 24H$, and two special nodes: the source n_s and the sink $n_{\tilde{s}}$. The node $n_{t,l}$ represents the instant of time t in the location l. Observe that in this case, it is not necessary to mark the nodes with the requests. The arc multiset E_D is composed of the following sets:

- E_{DREST} has an arc $(n_{t,l}, n_{t+1,l})$ for each $l \in L$ and $t \in \mathbb{Z}$ such that $0 \leq t \leq 24H - 1$. They represent a driver resting for one hour.
- E_{DTRIP} has an arc $(n_{t,l_1}, n_{t+\Delta,l_2})$ for each pair of adjacent locations $l_1, l_2 \in L$, and for each $t \in \mathbb{Z}$ such that $0 \leq t \leq 24H - \Delta$, where $\Delta = \text{dur}_V(l_1, l_2)$. They represent a driver travelling between locations l_1 and l_2 on a truck.
- E_{DSHUT} has an arc $(n_{t,l_1}, n_{t+\Delta,l_2})$ for each pair of adjacent locations $l_1, l_2 \in L$, and for each $t \in \mathbb{Z}$ such that $0 \leq t \leq 24H - \Delta$, where $\Delta = \text{dur}_S(l_1, l_2)$. They represent a driver travelling between locations l_1 and l_2 on an external shuttle.
- For all $r \in R$, E^r_{DPICK} has an arc $(n_{t,l^p_r}, n_{t+s^p_r,l^p_r})$ for each $t \in \mathbb{Z}$ such that $0 \leq t \leq 24H - s^p_r$ and the pickup time window of r is respected. They represent a driver loading an item.
- For all $r \in R$, E^r_{DDELI} has an arc $(n_{t,l^d_r}, n_{t+s^d_r,l^d_r})$ for each $t \in \mathbb{Z}$ such that $0 \leq t \leq 24H - s^d_r$ and the delivery time window of r is respected. They represent a driver unloading an item.
- E_{DSOUR} and E_{DSINK} have arcs $(n_s, n_{0,l})$ and $(n_{24H,l}, n_{\tilde{s}})$, for all $l \in L$. They are useful to model driver routes as $(n_s, n_{\tilde{s}})$-directed paths.

The weight vector w_D is given by:

$$w_D(e) = \begin{cases} \text{cost}_S(l_1, l_2) & \text{if } e = (n_{t_1,l_1}, n_{t_2,l_2}) \in E_{DSHUT} \\ 0 & \text{otherwise} \end{cases}$$

Observe that in G_D every driver route for some driver d determines a $(n_s, n_{\tilde{s}})$-directed path p that passes through the node n_{0,l_d}. However, the converse is not true due to the *rest* constraints imposed on drivers. These constraints have to be enforced externally. Also, $w_D(p)$ represents the shuttle cost of the driver route.

In Figure 2 we show the result of applying the above construction to the previous example. We suppose $\text{dur}_S(l_1, l_2) = 8$. In this case, the arcs in E_{DSHUT} are drawn with dashed lines (to distinguish them from the arcs in E_{DTRIP}). Besides, we displayed the arcs in $E^{r_i}_{DPICK} \cup E^{r_i}_{DDELI}$ with $i \in \{1, 2\}$ with curved arrows and labeled with p_i or d_i depending on the action (picking or delivering) of the request r_i. The red $(n_s, n_{\tilde{s}})$-directed path represents a driver that starts in l_1, takes a shuttle to l_2 from 0 to 8, rests from 8 to 12, unloads the item demanded by r_1 from 12 to 16, and rests from 16 to 24. The cost of this path is $\text{cost}_S(l_1, l_2)$.

4.3 Synchronisation of routes

Truck and driver routes must be synchronised in space and time. For example, a truck and a driver need to be idle in the same location at the same time for a trip to take place, and the same

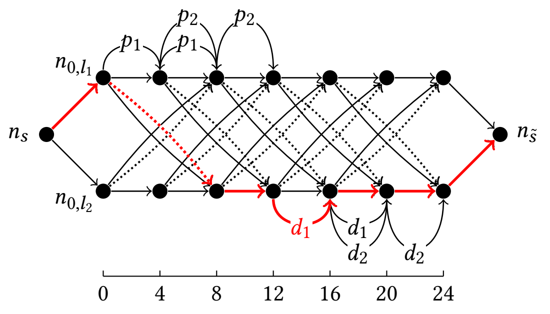

Figure 2: Example of G_D and a $(n_s, n_{\tilde{s}})$-directed path.

happens with the loading/unloading of an item. Furthermore, no more than two drivers can be in the same vehicle at the same time. Thus, despite having two separated digraphs, G_V and G_D, the directed paths of each are related. We need to introduce additional notations before formalising the synchronisation constraints.

We use a placeholder in an arc, e.g $\{(\_, w) \in E_V\}$, to denote the subset of arcs whose endpoints exactly match the given pattern, in this case, $\{(u, w') \in E_V : w' = w\}$. In the same way, sometimes we use more than one placeholder or leave the super/subscript of a node blank, e.g. $\{(n^-_{t_1,l_1}, n^-_{t_2,l_2}) \in E_V\}$ to denote $\{(n^{r_1}_{t'_1,l'_1}, n^{r_2}_{t'_2,l'_2}) \in E_V : t'_1 = t_1, l'_1 = l_1, t'_2 = t_2, l'_2 = l_2\}$.

Let E_{DSYNC} be the multiset of arcs in E_D that need synchronisation, i.e.

$$E_{DSYNC} = E_{DTRIP} \cup (\bigcup_{r \in R} E^r_{DPICK}) \cup (\bigcup_{r \in R} E^r_{DDELI}),$$

and let $\mathcal{M}$ be the function that maps arcs in E_{DSYNC} to their *corresponding* arcs in E_V, defined as:

$$\mathcal{M}(e) = \begin{cases} \{(n^-_{t,l}, n^-_{t',l'}) \in E_{VTRIP}\} & \text{if } e = (n_{t,l}, n_{t',l'}) \in E_{DTRIP} \\ \{(n^0_{t,l}, n^-_{t',l}) \in E^r_{VPICK}\} & \text{if } e = (n_{t,l}, n_{t',l}) \in E^r_{DPICK} \\ \{(n^-_{t,l}, n^0_{t',l}) \in E^r_{VDELI}\} & \text{if } e = (n_{t,l}, n_{t',l}) \in E^r_{DDELI} \end{cases}$$

Observe that $\{e\}$ and $\mathcal{M}(e)$ contain the arcs in G_D and G_V, respectively, that have the same type and whose endpoints have equal locations and instants of time. Besides, the last two cases in the definition of $\mathcal{M}$ are equal to $\{(n^0_{t,l}, n^r_{t',l})\}$ and $\{(n^r_{t,l}, n^0_{t',l})\}$, respectively, but we keep the placeholder notation so that it also applies to the alternative digraph $\tilde{G}_V$ (where they correspond to $\{(n^0_{t,l}, n^1_{t',l})\}$ and $\{(n^1_{t,l}, n^0_{t',l})\}$, respectively).

Now the synchronisation constraints can be stated as follows. Let P_V be a set of vehicle routes and P_D be a set of driver routes. Then for all $e \in E_{DSYNC}$, the inequality $m \leq n \leq 2m$ must hold, where n is the number of driver routes in P_D that contain e and m is the number of vehicle routes in P_V that contain an arc of $\mathcal{M}(e)$, i.e. $n = |\{p \in P_D : e \in E(p)\}|$ and $m = |\{p \in P_V : \mathcal{M}(e) \cap E(p) \neq \emptyset\}|$.

4.4 Integer program

We define the following variables:

- X_e integer for all $e \in E_V$ whose value is the number of vehicle routes that traverse e,
- Y_{de} binary for all $d \in D$ and $e \in E^d_D$, where $E^d_D = E_D \setminus \{(n_s, n_{0,l}) : l \neq l_d\}$, whose value is 1 if the driver route of d traverses e, and
- Z_{dj} binary for all $d \in D$ and $0 \leq j < H$ whose value is 1 if d has day j off.

The objective is to minimise:

$$\sum_{e \in E_{VDELI}} w_V(e) X_e \tag{1}$$

$$\sum_{e \in E_{VTRIP}} w_V(e) X_e \tag{2}$$

$$\sum_{e \in E_{DSHUT}} w_D(e) \sum_{d \in D} Y_{de} \tag{3}$$

subject to:

$$\sum_{e \in E^r_{VPICK}} X_e = 1 \qquad \forall r \in R \tag{4}$$

$$\sum_{e \in \{(\_,u) \in E_V\}} X_e = \sum_{e \in \{(u,\_) \in E_V\}} X_e \qquad \forall u \in N_V \setminus \{n_s, n_{\tilde{s}}\} \tag{5}$$

$$\sum_{e \in \{(\_,u) \in E^d_D\}} Y_{de} = \sum_{e \in \{(u,\_) \in E^d_D\}} Y_{de} \qquad \forall d \in D, u \in N_D \setminus \{n_s, n_{\tilde{s}}\} \tag{6}$$

$$\sum_{e \in \mathcal{M}(e)} X_e \le \sum_{d \in D} Y_{de} \le 2 \sum_{e \in \mathcal{M}(e)} X_e \qquad \forall e \in E_{DSYNC} \tag{7}$$

$$\sum_{h=t}^{t+23} \sum_{\substack{e \in \{(n_{h,\_,\_},\_) \in \\ E_{DREST}\}}} Y_{de} \ge 12 \qquad \forall d \in D, 0 \le t \le 24(H-1) \tag{8}$$

$$\sum_{i=j}^{j+6} Z_{di} \ge 1 \qquad \forall d \in D, 0 \le j \le H-7 \tag{9}$$

$$24 Z_{dj} \le \sum_{h=24j}^{24j+23} \sum_{\substack{e \in \{(n_{h,\_,\_},\_) \in \\ E_{DREST}\}}} Y_{de} \qquad \forall d \in D, 0 \le j < H \tag{10}$$

$$X_e \in \{0, \dots, \text{cap}_V(e)\} \qquad \forall e \in E_V \tag{11}$$

$$Y_{de} \in \{0,1\} \qquad \forall d \in D, e \in E^d_D \tag{12}$$

$$Z_{dj} \in \{0,1\} \qquad \forall d \in D, 0 \le j < H \tag{13}$$

The objective functions (1), (2), and (3) are the costs for late deliveries, company's vehicle trips, and shuttle trips, respectively. Constraints (4) say that every pickup request is serviced by exactly one vehicle. (5) and (6) are the flow conservation constraints for the vehicle routes and driver routes, respectively. (7) are the synchronization constraints. (8) say that drivers must rest at least 12 hours in every 24-hour interval. (9) say that drivers must have a day off in every 7-day interval. (10) say that drivers must not work on days off. (11), (12), and (13) define the possible values for the variables, where the capacity of the vehicle arcs is given by

$$\text{cap}_V(e) = \begin{cases} |V| & \text{if } e \in E_{VSINK} \cup \{(n^0_{\_,\_}, n^0_{\_,\_}) \in E_V\} \\ |\{v \in V : l_v = l\}| & \text{if } e = (n_s, n^0_{0,l}) \in E_{VSOUR} \\ 1 & \text{otherwise} \end{cases}$$

It is clear that the total number of vehicles bounds the capacity of all the arcs in G_V. But, in particular, the number of vehicles sharing its initial location bounds the capacity of the arcs adjacent to the source node. Also, the capacity of the arcs adjacent to a node with a superscript in R is bounded by 1, since each request can be served by a single truck.

We refer to this integer programming formulation as $\mathcal{F}$.

4.4.1 Alternative integer program. We enumerate the changes needed in the integer program for the alternative digraph presented in 4.1.1.

- We have a binary variable $X_{ve} \in \{0,1\}$ for each $v \in V$ and $e \in E^v_V$, where $E^v_V = E_V \setminus \{(n_s, n^0_{0,l}) : l \neq l_v\}$, and constraints (11) are replaced by:

$$X_{ve} \in \{0,1\} \qquad \forall v \in V, e \in E^v_V$$

- The vehicle flow conservation constraints become:

$$\sum_{e \in \{(\_,u) \in E^v_V\}} X_{ve} = \sum_{e \in \{(u,\_) \in E^v_V\}} X_{ve} \qquad \forall v \in V, u \in N_V \setminus \{n_s, n_{\tilde{s}}\}$$

- In the rest of the constraints and objectives, the expression X_e becomes an alias for $\sum_{v \in V} X_{ve}$.
- We add pairing constraints:

$$X_{ve} \le \sum_{h=0}^{t} \Big(\sum_{\substack{e' \in \{(\_, n^{\bar{r}}_{h,\_}) \\ \in E^r_{VPICK}\}}} X_{ve'} - \sum_{\substack{e' \in \{(\_, n^{\bar{r}}_{h,\_}) \\ \in E^r_{VDELI}\}}} X_{ve'} \Big) \qquad \forall v \in V, r \in R, e \in \{(n^{\bar{r}}_{t,\_}, \_) \in E^r_{VDELI}\}$$

  that forbid the delivery of a request that has already been delivered or that has not yet been picked up by the vehicle.

We refer to the resulting integer programming formulation as $\mathcal{F}_{ALT}$.

5 RESULTS

We use a set of generated instances to evaluate the performance of $\mathcal{F}$ and $\mathcal{F}_{Alt}$. We consider a real map with 6 Argentinian cities (the average distance among them is 470km), a planning horizon of 7 days, and a variable number of requests, vehicles, and drivers. A further description of this generation is found in [7]. The multiple objectives were converted into one by a weighted sum, considering a weighting factor of $\frac{1}{3}$ for each objective.

The experiments were performed in a desktop computer with an Intel Core i5 2.6GHz processor, 5.7GB of memory, Ubuntu 20.04, and CPLEX 12.7 with default parameters (but using a single thread). The code is written in C++11 with the CPLEX's API. We fixed a time limit of 7200s for each execution.

The results are reported in Table 1. Each row gathers three different instances generated with same parameters, but different seeds. Columns 1-3 display the number of requests, vehicles, and drivers, respectively, of the instances. The next columns show the results obtained with each formulation. Those columns labeled with "n" report the number of instances where the solver proves optimality before the time limit, with "t" the average time in seconds consumed for those instances, with "m" the number of instances where the solver only proves feasibility (but not optimality), and with "gap" the averages of relative gap reported by the solver for those instances. We do not display the number of instances where the solver does not find any integer solution before the time limit, but this number can be calculated as $3 - (m + n)$ as all the instances were feasible. We write a mark "–" in the columns t and gap, respectively, when n and m are equal to 0.

As we can see from the table, for $|R| = 4$, $\mathcal{F}_{Alt}$ achieves better gaps and finds an integer solution in one more instance than $\mathcal{F}$, while the latter exhibits better times when optimality is proven. For $|R| \in \{6, 8\}$, $\mathcal{F}$ clearly outperforms $\mathcal{F}_{Alt}$, finding integer solutions in 9 instances and proving optimality in 7 of them. In particular, for the 3 instances where $\mathcal{F}_{Alt}$ finds a solution, $\mathcal{F}$ gets better ones.

Table 1: Comparison of formulations

$\|R\|$	$\|V\|$	$\|D\|$	$\mathcal{F}_{Alt}$ n	t (s)	m	gap	$\mathcal{F}$ n	t (s)	m	gap	$\mathcal{F}$ + initial heur. n	t (s)	m	gap
4	1	2	2	4074	1	8 %	2	2305	0	–	2	784	1	4 %
4	2	4	0	–	2	20 %	0	–	2	39 %	0	–	3	23 %
6	2	4	0	–	1	56 %	1	1553	2	25 %	1	693	2	17 %
6	3	6	0	–	1	30 %	2	974	0	–	2	905	1	8 %
8	2	4	0	–	1	65 %	1	1569	0	–	2	2018	1	11 %
8	3	6	0	–	0	–	1	2183	0	–	1	953	2	14 %
8	4	4	0	–	0	–	2	4495	0	–	3	963	0	–

Regarding $\mathcal{F}$, we detected that the value of the linear relaxation coincided with the optimal value in, at least, 8 instances. Also, the solver was not able to find any integer solution in 8 instances within the time limit. Thus, this suggests that the solver already starts with good dual bounds, but has a hard time finding primal solutions. For this reason, we repeated the experiment providing an initial heuristic solution with the CPLEX's *MIP start* functionality. The initial solution was generated with the heuristic described in [7] and required 450 seconds per instance. The results are reported in the last columns of Table 1, and the time consumed by the initial heuristic is considered in column "t".

Starting with an integer solution was beneficial in all the cases. In particular, the execution was 3 times faster on average for all the instances previously reported as optimal, and the gap improved for all the instances previously reported as feasible. Additionally, two new instances are solved to optimality, for which no integer solutions had been discovered during the previous experiment.

6 CONCLUSIONS

We addressed the exact resolution of a simultaneous vehicle routing and crew scheduling problem. We consider a dynamic driver-vehicle correspondence, where crews can have one or two drivers, and they can be exchanged in a set of locations. The result is a hard combinatorial optimization problem, with many interrelated resources to program, subject to a variety of operational and labour requirements.

We proposed two integer programming formulations. They are based on two digraphs that model the vehicle and driver routes, respectively. The structure of the digraphs captures many of the requirements of the problem, while the rest are handled as constraints in the formulations.

We used a set of generated instances to evaluate the performance of both. The instances involved 4-8 requests, 1-4 vehicles, 2-8 drivers, 1 week, and 6 locations. We detected that the formulation based on the digraph that captured the greatest number of requirements outperformed the other, and, in general, it provided good dual bounds. As expected, larger instances cannot be solved through these IP models and heuristic approaches should be considered. The option of starting the optimization with an initial heuristic solution provided by the metaheuristic presented in [7], greatly improved the results.

As future work, we plan to design and incorporate ad-hoc callbacks during the optimization that help us to improve the overall performance, e.g. the results suggest that implementing a primal heuristic would be beneficial. It would also be interesting to investigate other alternatives to deal with the multi-objective optimization, e.g. normalise the objective functions or consider a hierarchical optimization.

ACKNOWLEDGMENTS

This work was partially supported by grants PICT-2016-0410 and PICT-2017-1826 (ANPCyT), PID ING538 (UNR) and UBACyT 200201-70100484BA.

REFERENCES

[1] B. Domínguez-Martín, I. Rodriguez-Martin, and JJ. Salazar-Gonzalez. 2018. The driver and vehicle routing problem. *Computers & Operations Research* 92 (2018), 56–64.

[2] M. Drexl, J. Rieck, T. Sigl, and B. Press. 2013. Simultaneous Vehicle and Crew Routing and Scheduling for Partial- and Full-Load Long-Distance Road Transport. *Business Research* 6, 2 (2013), 242–264.

[3] A. Goel. 2009. Vehicle Scheduling and Routing with Drivers' Working Hours. *Transportation Science* 43, 1 (2009), 17–26. https://doi.org/10.1287/trsc.1070.0226

[4] A. Goel and S. Irnich. 2017. An Exact Method for Vehicle Routing and Truck Driver Scheduling Problems. *Transportation Science* 51, 2 (2017), 737–754. https://doi.org/10.1287/trsc.2016.0678

[5] A. Goel, T. Vidal, and A. L. Kok. 2019. *To team up or not–Single versus team driving in European road freight transport.* Technical Report. Tech. rep., PUC–Rio, Rio de Janeiro, Brasil.

[6] E. Lam, P. Van Hentenryck, and P. Kilby. 2015. Joint Vehicle and Crew Routing and Scheduling. In *Principles and Practice of Constraint Programming*, Gilles Pesant (Ed.). Springer International Publishing, Cham, 654–670.

[7] Mauro Lucci, Daniel Severín, and Paula Zabala. 2021. A metaheuristic for crew scheduling in a pickup-and-delivery problem with time windows. *International Transactions in Operational Research* (2021). https://doi.org/10.1111/itor.13096

[8] N. Mendes and M. Iori. 2020. A Decision Support System for a Multi-trip Vehicle Routing Problem with Trucks and Drivers Scheduling.. In *ICEIS (1).* 339–349.

[9] ME. Rancourt, JF. Cordeau, and G. Laporte. 2013. Long-Haul Vehicle Routing and Scheduling with Working Hour Rules. *Transportation Science* 47, 1 (2013), 81–107. https://doi.org/10.1287/trsc.1120.0417

[10] C. Tilk and A. Goel. 2020. Bidirectional labeling for solving vehicle routing and truck driver scheduling problems. *European Journal of Operational Research* 283, 1 (2020), 108–124. https://doi.org/10.1016/j.ejor.2019.10.038

[11] D. West. 2001. *Introduction to graph theory.* Prentice hall Upper Saddle River.