A heuristic solution for a driver-vehicle scheduling problem

Benoît Laurent[1,2], Valérie Guihaire[1,2], and Jin-Kao Hao[2]

[1] Perez Informatique, 41 avenue Jean Jaures, 67000 Strasbourg, France
blaurent@perinfo.com, vguihaire@perinfo.com
[2] LERIA, Université d'Angers, 2 Bd Lavoisier, 49045 Angers cedex 01, France
jin-kao.hao@univ-angers.fr

Summary. A driver-vehicle scheduling problem in a limousines rental company is studied. Given a set of trip demands to be covered, the goal is to find a driver-vehicle schedule that covers as many as possible of the required demands while satisfying a set of imperative constraints and optimizing several cost objectives. A formulation of the problem is given and a solution approach using local search is developed.

1 Introduction

This paper deals with a driver and vehicle scheduling problem in a limousines rental company. This application context induces specific operational constraints making the problem fairly distinct from other known crew and vehicle scheduling problems. However, some neighboring problems can be found in the literature dealing mainly with transport by bus. The most recent approaches related to this issue are based on a complete integration of drivers and vehicles during the scheduling process (e.g., see [2], [4] and [1]).

In our case, we are given daily sets of trips, drivers and vehicles, with the goal of scheduling resources in order to cover the maximum possible workload. The quality of service being a crucial issue, a schedule must comply with a set of imperative constraints, while optimizing some economic objectives.

2 Problem description

Only a subset of the constraints and objectives are presented here. We state:

- A set $\mathcal{T}$ of trips, each being defined by a time and a place for the departure and the destination, a number of passengers, required driver skills, etc.
- A set $\mathcal{D}$ of drivers, each being characterized by a daily allowed time spread, a set of skills, etc.
- A set $\mathcal{V}$ of vehicles, each being characterized among others by its capacity.

The problem is to find a daily assignment of driver-vehicle couples to the trips that satisfies a set $\mathcal{C}$ of constraints while optimizing a set $\mathcal{O}$ of objectives.

2.1 Constraints

We consider any trip $t \in \mathcal{T}$, a vehicle $v \in \mathcal{V}$ and a driver $d \in \mathcal{D}$ assigned to t. For quality requirements, a solution must satisfy the following constraints:

- v must be compatible with t, i.e. there must be enough seats in v to accommodate all the passengers.
- d must have all the skills required by t.
- The duration between the pick-up time of the first trip and the end of the last trip cannot be greater than the maximum spread allowed for d.
- There must be enough time between successive trips for d to move from one trip to the next one.

2.2 Objectives

Main objective: It is primordial to meet customers' trip demands. However, the available resources may not be sufficient to satisfy all of them. Therefore, the first goal is to cover as many as possible of the trip demands. From the point of view of the rental company, it is preferable to maximize the sum of the durations of the assigned trips since long trips are more profitable than short ones. Notice that this objective is equivalent to minimizing the sum of the durations of the trips to which no couple of resources is assigned. In addition, the trips starting in an imminent way must be favored.

Secondary objectives: For evident economic reasons, it is desirable to reduce the running costs, i.e. the number of working drivers and used vehicles. Furthermore, it is useful to minimize the drivers' waiting times between trips.

3 Problem formulation

3.1 Notations

Let T, D and V be the number of trips, drivers and vehicles respectively. Given $t \in \mathcal{T}$, $d \in \mathcal{D}$, $v \in \mathcal{V}$, we state:

- $capa(v)$, the capacity of v,
- $pas(t)$, the number of passengers for t,
- $st(t), et(t), sp(t)$ and $ep(t)$ are respectively the start time, the end time, the start place and the end place of trip t,
- $sd(d)$ and $ed(d)$ are respectively the start time and the end time for d,
- $S_{max}(d)$ is the maximum spread time allowed for d.

We also define a set of binary relations:

- $sk(d, s) \Leftrightarrow$ driver d owns skill $s \in \mathcal{S}$, the set of skills,
- $Sk(t, s) \Leftrightarrow$ trip t requires skill s,
- $compat(t_i, t_j) \Leftrightarrow$ trips t_i and t_j can be done by the same resources,
- $dh(t_1, t_2)$ is the deadhead between trips t_1 and t_2,
- $wt(t_1, t_2)$ is the waiting time between trips t_1 and t_2.

We define a total order $\prec$ on the set of trips $\mathcal{T}$ by:

$$\forall (t_i, t_j) \in \mathcal{T}^2, t_i \prec t_j \Leftrightarrow \begin{cases} st(t_i) < st(t_j) \\ \vee \\ st(t_i) = st(t_j) \wedge i < j \end{cases}$$

The following notations are used to handle "driver-vehicle to trip" assignments:

- $wd(d)$, (resp. $vu(v)$) $\Leftrightarrow d$ (resp. v) is assigned to at least one trip,
- $Seq(d)$ is the set of couples (t_1, t_2) that $d \in \mathcal{D}$ handles consecutively.

Eventually, we define $compat'(t_i, t_j), dh'(t_i, t_j), wt'(t_i, t_j)$ and $Seq'(d)$ that are similar to $compat(t_i, t_j), dh(t_i, t_j), wt(t_i, t_j)$ and $Seq(d)$ respectively except that they take into account a stop at the depot between trips.

3.2 Constraints

Capacity constraints The first type of constraints imposes that the vehicle is big enough to carry all the passengers. For each $t \in \mathcal{T}$ and each $v \in \mathcal{V}$:

$$CAPA(t, v) \Leftrightarrow pas(t) \leq capa(v)$$

Skills constraints Some trips require special skills from the driver, for instance spoken languages. For each $t \in \mathcal{T}$, each $d \in \mathcal{D}$ and each $s \in \mathcal{S}$:

$$SKILLS(t, d, s) \Leftrightarrow \neg Sk(t, s) \vee sk(d, s)$$

A similar type of constraints exists with vehicles features.

Maximum spread time constraints The third type of constraints imposes the maximum spread time for each driver. For each $d \in \mathcal{D}$ and each $(t_i, t_j) \in \mathcal{T}^2, t_i = (d, .), t_j = (d, .)$:

$$MAX\_SPREAD(t_i, t_j, d) \Leftrightarrow (et(t_j) - st(t_i)) \leq S_{max}(d)$$

Feasible sequences constraints This type of constraints imposes that the sequence of trips assigned to a driver is feasible, i.e. the driver has enough time to move from the end of a trip to the start of the following one. A possible change of vehicle, if needed, must take place at the depot. For each

$d \in \mathcal{D}$ and each $(t_i, t_j) \in \mathcal{T}^2, t_i = (d, v_k), t_j = (d, v_l), t_i \prec t_j$:

$$FEASIBLE\_D(t_i, t_j, d) \Leftrightarrow \begin{cases} ((v_k = v_l) \wedge compat(t_i, t_j)) \\ \vee \\ ((v_k \neq v_l) \wedge compat'(t_i, t_j)) \end{cases}$$

3.3 Objectives

Main objective Minimizing the total duration of the unassigned trips:

$$Min \sum_{t \in \mathcal{T}, t=(\epsilon,.) \vee t=(.,\epsilon)} (et(t) - st(t))$$

where $t = (\epsilon, .) \vee t = (., \epsilon)$ means the trip t is not covered.

Secondary objectives Minimizing the number of working drivers and vehicles in use:

$$Min \sum_{d \in \mathcal{D}} wd(d) + \sum_{v \in \mathcal{V}} vu(v)$$

Minimizing deadheads:

$$Min \sum_{d \in \mathcal{D}} \left(\sum_{(t_1,t_2) \in Seq(d)} dh(t_1, t_2) + \sum_{(t_1,t_2) \in Seq'(d)} dh'(t_1, t_2) \right)$$

Minimizing the total waiting time:

$$Min \sum_{d \in \mathcal{D}} \left(\sum_{(t_1,t_2) \in Seq(d)} wt(t_1, t_2) + \sum_{(t_1,t_2) \in Seq'(d)} wt'(t_1, t_2) \right)$$

3.4 Configuration and evaluation function

A configuration σ is a consistent assignment of "driver-vehicle" couples in $\mathcal{I} = (\mathcal{D} \cup \{\epsilon\}) \times (\mathcal{V} \cup \{\epsilon\})$ to trips in $\mathcal{T}$. The search space Ω is the set of all such assignments. A configuration is evaluated by a weighted aggregation of the objectives, augmented by a penalty function for broken constraints [3].

$$\forall \sigma \in \Omega, \quad eval(\sigma) = wbc \times \sum_{c \in \mathcal{C}} f_c(\sigma) + \sum_{i \in \{1,\ldots,O\}} w_i \times f_i(\sigma)$$

with:

- $wbc > 0$ the weight associated to broken constraints,
- f_c the penalty for c. $f_c = 1$ if c is broken by σ, $f_c = 0$ otherwise,
- O the number of objectives,
- w_i the associated weight for i^{th} objective function,
- f_i the value of i^{th} objective function.

3.5 Greedy algorithm for initial configuration

The first step is a pre-processing of the data. By examining the incompatibilities between drivers and vehicles, drivers and trips, and trips and drivers, we reduce the domains of the variables. A constructive greedy heuristic combined with constraint propagation techniques is then used to create an initial configuration σ. The stop criterion is that no additional assignment can be made without violating constraints.

3.6 Hill climbing and simulated annealing

In order to improve the initial configuration σ, both hill climbing and simulated annealing are experimented, based on a 1_change neighborhood. From the current configuration, we obtain a neighboring configuration by changing the driver and/or the vehicle assigned to one trip.

$$\forall \sigma \in \Omega, 1\_change(\sigma) = \{(t,(d,v)) \in \mathcal{T} \times \mathcal{I} | \exists \text{ a unique } t \text{ such that } \sigma(t) \neq (d,v)\}$$

4 Experimentations and results

Computational experiments were carried out on five real instances, representing different workloads. Table 1 shows the main characteristics of the instances and the results manually obtained in the limousines rental company for comparison purpose.

Our algorithms were programmed in C++, compiled with gcc 3.4.2, on a PC running Windows XP (256Mo RAM, 2.4Ghz). The program was run 10 times on each instance with different random seeds. The stop condition used is a maximum duration fixed to 10 minutes.

Table 1. Characteristics of the five instances and results of manual scheduling

Date	trips	total duration (hhh:mm)	Manual scheduling				
			broken constraints	drivers	vehicles	deadheads (hh:mm)	waiting time (hhh:mm)
08_05	91	133:46	27	44	52	27:01	119:04
10_05	126	238:58	48	65	70	39:32	200:32
18_05	153	453:06	47	74	74	56:30	196:20
19_05	163	504:47	48	79	77	55:26	165:35
23_05	202	457:59	79	84	81	63:53	248:07

Table 2 shows the results obtained with hill climbing and simulated annealing algorithms. The rules are slightly different between manual and computed scheduling: in the second case, broken constraints are strictly forbidden but unassigned trips are allowed. These are manually handled afterwards. Therefore, a column was added in Table 2 giving the percentage of unassigned work.

Table 2. Results using hill climbing and simulated annealing

Date	hill climbing					simulated annealing				
	Unassigned trips (%)	drivers	vehicles	deadheads (hh:mm)	waiting time (hh:mm)	Unassigned work (%)	drivers	vehicles	deadheads (hh:mm)	waiting time (hh:mm)
08_05	1.9	35	37	29:28	57:51	0.93	36	37	28:39	70:34
10_05	9.2	44	46	39:12	74:57	9.2	44	46	35:15	66:44
18_05	5.7	67	69	49:49	87:34	5.3	65	68	56:46	88:22
19_05	15	63	69	50:04	77:27	15	63	67	49:30	88:09
23_05	12	68	67	63:16	81:01	9.9	74	70	62:49	94:31

These results show a significant improvement regarding the actual practice. Without breaking any constraint, both algorithms assign most of the work. Furthermore, the number of required resources is substantially reduced. The total waiting time is divided by a factor of 2.36 in average. Actually, the workload for human schedulers is so high that costs reduction is only a secondary concern. This leads to the other major contribution of this work: the time needed to elaborate a planning is drastically decreased. Whereas people spend nearly 4 hours on this task, our program only takes a few minutes to get a much better quality schedule.

5 Conclusion

We tackled a practical driver and vehicle scheduling problem in an original context. The solution approach combines a pre-processing phase using constraint programming techniques and an optimization phase using local search. Results obtained on real data showed significant improvements compared with the actual practice in terms of solution quality and computing time.

Acknowledgments
This work was carried out within the framework of a CIFRE grant from "Agence Nationale de la Recherche Technique" (ANRT), which is acknowledged. Special thanks go to Joël Thibault for his role in fully supporting this work.

References

1. R. Borndörfer, A. Löbel, and S. Weider. A bundle method for integrated multi-depot vehicle and duty scheduling in public transit,CASPT 2004, Aug. 2004.
2. R. Freling, D. Huisman, and A.P.M. Wagelmans. Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling*, 6:63–85, 2003.
3. P. Galinier and J.K. Hao. A general approach for constraint solving by local search. *Journal of Mathematical Modelling and Algorithms*, 3(1):63–85, 2004.
4. D. Huisman. *Integrated and Dynamic Vehicle and Crew Scheduling*. PhD thesis, Erasmus University Rotterdam, 2004.