

# Simultaneous Vehicle and Crew Scheduling for Extra Urban Transports

Benoît Laurent<sup>1,2</sup> and Jin-Kao Hao<sup>2</sup>

<sup>1</sup> Perinfo SA, 41 avenue Jean Jaurès, 67000 Strasbourg, France  
blaurent@perinfo.com

<sup>2</sup> LERIA, Université d'Angers, 2 boulevard Lavoisier, 49045 Angers Cedex 01, France  
jin-kao.hao@univ-angers.fr

**Abstract.** We present a *simultaneous* approach to solve the integrated vehicle and crew scheduling problem in an extra urban context. We consider the single depot case with a *heterogeneous* fleet of vehicles. We propose a constraint based model which is subsequently solved by a Greedy Randomized Adaptive Search Procedure. The construction phase of each initial solution relies on constraint programming techniques, while the local search phase exploits a powerful neighborhood exploration mechanism. The computational experiments conducted on real-world instances show the effectiveness and the flexibility of the approach compared with the classical sequential vehicle and crew scheduling.

## 1 Introduction

Crews and vehicles are the main resources to provide services in transport systems. The way these resources are employed directly impacts the quality of service and the cost of the whole transport system. It is thus primordial to optimize the utilization of these resources in any transportation scheduling systems.

The conventional crew and vehicle scheduling process is the *sequential* approach which determines first the vehicles schedule and then the crews schedule. This separation is mainly due to the complexity of each sub-problem. Indeed, the Multi-Depot Vehicle Scheduling Problem (MDVSP) is known to be NP-hard. The Bus Driver Scheduling Problem (BDSP) is usually modeled as a Set Covering or Set Partitioning Problem, both being NP-hard.

In the early 1980s, Ball et al. criticized this sequential approach [1], but the first real *integrated* solutions, in which vehicles and crews are simultaneously scheduled, were only developed in 1995 [5]. This integrated management of both resources demonstrated its efficiency in [6] where relief locations, i.e. places where a driver can be relieved by a colleague, are spatially distant. Integration is also profitable when a driver is not allowed to change from one vehicle to another.

The most popular approach to tackle the integrated scheduling problem is undoubtedly integer linear programming (ILP). In [5], Freling *et al.* suggest, for the single depot case, an ILP formulation comprising a quasi-assignment structure for the vehicle scheduling, a set partitioning model for the crew scheduling, and a set of binding constraints. The solution approach is an approximated method

combining Lagrangian heuristics and column generation. The first exact algorithm is described in [9]. Both aspects, crew and vehicle, are modeled by a set partitioning formulation. However, the branch-and-cut-and-price algorithm proposed can only solve small instances up to 20 trips. Another interesting exact solution for the single depot case is presented by Haase *et al.* in [8] where scenarios up to 150 trips are solved. For larger problems, a heuristic is employed to solve scenarios of 350 trips within 3 hours of CPU time.

Gaffi and Nonato in [6] were the first to address the integrated VCSP in the multiple-depot case. Like Freling and coauthors, they make use of Lagrangian relaxation and column generation. Once again, they confirmed the dominance of the simultaneous scheduling of crews and vehicles over the sequential approach on scenarios of Italian public transit operators. However, they observed that computational time remained a critical issue: it takes more than 24 hours to solve an instance of 257 trips. Recently, in [11], the models and algorithms introduced in [4] were successfully extended to tackle the integrated vehicle and crew scheduling problem with multiple depots.

The simultaneous approach begins to cover the gap between the academic world and the business world. It is now included in commercial packages like Hastus' suite (see [3]) or Microbus 2 (see [2]).

Although efficient and flexible solutions based on metaheuristics were successfully developed to deal with vehicle or driver scheduling separately, to the best of our knowledge, no similar study is reported in the literature for the integrated driver and vehicle scheduling problem.

In this paper, we propose a new heuristic approach for the simultaneous vehicle and crew scheduling problem. In our situation, all vehicles are parked within the same depot. However, the problem is more general than the usual single depot case which imposes a homogeneous fleet of vehicles. Here, the vehicles may belong to different categories. The first contribution of this work is the introduction of a new model relying on the constraint satisfaction and optimization problem. This constraint based model has the main advantage of being intuitive and natural. It also offers much flexibility in terms of solution approaches since various metaheuristics can be developed. For the solution purpose, we implemented a Greedy Randomized Adaptive Search Procedure (GRASP) which constitutes, to our knowledge, the first application of metaheuristics to this problem. Within our GRASP algorithm, constraint programming techniques are used to build initial solutions. Improvements of these solutions are achieved with a local search algorithm which embeds a powerful "ejection chain" neighborhood exploration mechanism. The whole approach is assessed on a set of 7 real-world instances and compared with the conventional sequential methodology, clearly showing the superiority of the simultaneous scheduling approach.

## 2 Vehicle and Crew Scheduling: Problem Presentation

Given a set of trips within a one-day horizon, a set of drivers, a fleet of vehicles parked at a given depot, a set of workday types, the Vehicle and Crew Scheduling

Problem consists in determining a minimum cost schedule for crews and vehicles, such that generated duties are feasible and mutually compatible.

The trips are characterized by starting and ending locations with corresponding times. Each trip must be served by certain types (categories) of vehicles. Travel times between all pairs of locations are also known. The transfers without passengers between trips, or either coming from or returning to the depot are called deadheads. The other inputs concern the availability of vehicles in each category. Similar bounds exist for the crews. The workday types are defined by labor regulations. As mentioned in the introduction, crews and vehicles are supposed to share the same depot where all duties start and end.

The vehicle part of the problem consists in assigning a vehicle to each trip while satisfying the following constraints:

- **Category constraints.** The vehicle must belong to the required category (type). A hierarchy between types is defined so as to allow upgrades if no vehicle of the required type is available.
- **Feasible sequences constraints.** each vehicle must have enough time between consecutive trips to move from one to the other.

A sequence of compatible trips starting and ending at the depot, thus performed by the same vehicle, defines a vehicle block. Along these blocks, relief opportunities are time-place pairs defining when and where a driver can be relieved by a colleague. A portion of work between two relief points is thus necessarily accomplished by the same crew and defines a task, a piece of work being a sequence of tasks without interruption on a single vehicle. In our case, reliefs take place at the depot. In this context, pieces of work and vehicle blocks coincide.

The tasks from the vehicle scheduling phase are then assigned to crew members to compose crew duties. The following constraints must be satisfied (only a subset is presented here for the purpose of simplicity):

- **Maximum spread time.** For crew members, the duration between the pick-up time of the first trip and the drop-down time of the last one, must be less than or equal to the maximum spread time allowed.
- **Maximum working time.** For crew members, the total working time, corresponding to driving and possibly maintenance tasks, does not exceed the bound specified in each driver's contract.
- **Changeovers.** Depending on the companies, drivers are allowed or not to change from a vehicle to another during their duty.

Like many previously studied crew and vehicle scheduling problems, the cost structure for our problem is composed of fixed and operational costs. For evident economic reasons, the most important objective is to reduce the number of working drivers and running vehicles. In order to further reduce costs, it is also useful to minimize idle time and deadheads. For simplicity reasons, we do not mention these latter objectives in the rest of the paper.

### 3 Problem Formulation

In this part, we introduce an original formulation inspired by the study reported in [12] relying on a constraint satisfaction and optimization model. This constraint-based formulation offers a natural modeling of the initial problem and provides a flexible basis to implement various metaheuristics.

**Notations.** Let  $\mathcal{T}$  be the set of trips, numbered according to increasing starting time. The bounds for the vehicles and for each workday types being provided, we can define  $\mathcal{D}$  and  $\mathcal{V}$  the sets of drivers and vehicles respectively. Let  $T, D$  and  $V$  be the associated cardinalities. Given  $t \in \mathcal{T}, d \in \mathcal{D}, v \in \mathcal{V}$ , we define the notations figuring in Table 1 to formalize the constraints and objectives.

**Table 1.** Notations

$Cat(t)$	set of categories (types) of vehicle that can serve trip $t$
$cat(v)$	category (type) of vehicle $v$
$st(t), et(t)$	respectively start and end time of trip $t$
$S_{\max}(d)$	maximum spread time allowed for driver $d$
$W_{\max}(d)$	maximum working time allowed for driver $d$
$VC_{\max}(d)$	maximum number of vehicle changes for driver $d$

We also use the following notations to handle "driver-vehicle to trip" assignments:

- $wd(d) = 1 \Leftrightarrow d$  is assigned to at least one trip,
- $vu(v) = 1 \Leftrightarrow v$  is assigned to at least one trip,
- $Seq(d)$  is the set of pairs  $(t_k, t_l)$  that  $d \in \mathcal{D}$  handles consecutively.

Finally,  $dh'(t_k, t_l), wt'(t_k, t_l)$  and  $Seq'(d)$  are similar to  $dh(t_k, t_l), wt(t_k, t_l)$  and  $Seq(d)$  respectively, except that they take into account a stop at the depot between trips.

#### 3.1 Decision Variables and Domains

In our problem, we aim to simultaneously assign a couple (driver and vehicle) to each trip. Therefore, we define the set of decision variables as the set of trips  $\mathcal{T}$ . Naturally, the associated value domain  $\mathcal{I}_k$  for each variable corresponds to driver-vehicle pairs. Initially, all domains  $\mathcal{I}_k$  are equal to  $\mathcal{I} = \mathcal{D} \times \mathcal{V}$ . The set of constraints is exposed in section 3.2, the pursued objectives in section 3.3.

#### 3.2 Constraints

This section provides a mathematical definition of the types of constraints.

##### Category constraints

$$\forall t \in \mathcal{T}, \forall v \in \mathcal{V}, t = (., v), \quad CATEGORY(t, v) \Leftrightarrow cat(v) \in Cat(t)$$

**Feasible sequences constraints**

$$\forall d \in \mathcal{D}, \forall (t_k, t_l) \in \mathcal{T}^2, t_k = (d, \cdot), t_l = (d, \cdot), t_k \prec t_l,$$

$$FEASIBLE\_D(t_k, t_l, d) \Leftrightarrow \begin{cases} compat(t_k, t_l) \wedge (t_k, t_l) \in Seq(d) \\ \vee \\ compat'(t_k, t_l) \end{cases}$$

Similar constraints exist for the vehicles.

**Maximum spread time constraints**

$$\forall d \in \mathcal{D}, \forall (t_k, t_l) \in \mathcal{T}^2, t_k = (d, \cdot), t_l = (d, \cdot), t_k \prec t_l,$$

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

**Maximum working time constraints**

$$\forall d \in \mathcal{D}, MAX\_WORK(t_k, t_l, d) \Leftrightarrow$$

$$\sum_{t_k \in \mathcal{T}, t_k = (d, \cdot)} (et(t_k) - st(t_k)) + \sum_{(t_k, t_l) \in Seq(d)} dh(t_k, t_l) + \sum_{(t_k, t_l) \in Seq'(d)} dh'(t_k, t_l) \leq W_{\max}(d)$$

**Changeovers constraints**

$$\forall d \in \mathcal{D}, CHANGEOVERS \Leftrightarrow |\{v \in \mathcal{V} \mid \exists t_k \in \mathcal{T}, t_k = (d, v)\}| \leq VC_{\max}(d)$$

Constraints will be used to pre-process the value domains of the variables and to build the initial schedules.

**3.3 Objectives**

The following two objectives are considered: minimization of the number of working drivers and vehicles.

$$f_1 = Min \sum_{d \in \mathcal{D}} wd(d) \text{ and } f_2 = Min \sum_{v \in \mathcal{V}} vu(v)$$

These objectives will be combined into a weighted evaluation function which is used by the GRASP algorithm (see section 4.4).

**4 Solution Approach****4.1 Constraint Based Pre-processing**

The number of potential assignments is exponential subject to the cardinality of the set of decision variables and to the cardinality of the domains. To reduce this number, we borrow filtering techniques from constraint programming [14].

The node consistency property allows to remove from domains all values that are inconsistent with at least one constraint. Let us consider, for example, the CATEGORY constraints:

$$\forall t_k \in \mathcal{T}, \forall (d, v) \in \mathcal{I}, \text{cat}(v) \notin \text{Cat}(t_k), \Rightarrow (d, v) \notin \mathcal{I}_k$$

For a given trip, all pairs including vehicles forbidden for category reasons are removed from the trip's domain.

## 4.2 General Algorithm

GRASP is a multi-start metaheuristic for combinatorial problems, in which each iteration consists basically of two phases: construction and local search. The skeleton of the procedure is described in Algorithm 1. Our construction phase, detailed in subsection 4.3, relies on constraint programming techniques. The resulting solution is then improved by a local search algorithm (see subsection 4.4) embedding a powerful neighborhood exploration mechanism.

---

### Algorithm 1: Greedy randomized adaptive search procedure

---

**Data** :  $f$  evaluation function,  $it_{max}$  maximum number of iterations, restricted candidate list  
**Result** : Best solution found  $f^*$

```

begin
    /*  $x^*$  is the best solution found so far and  $f^* = f(x^*)$  */ ;
     $f^* \leftarrow \infty$  ;
    while  $it \leq it_{max}$  do
        /* Generate a greedy randomized solution  $x$  */ ;
         $x \leftarrow \text{greedy\_rand}()$  /* See Section 4.3 */ ;
        /* Apply a local search algorithm,  $x_l$  is a local minimum */ ;
         $x_l \leftarrow \text{descent}()$  /* See Section 4.4 */ ;
        /* Record best solution */ ;
        if  $f(x_l) \leq f^*$  then
             $f^* \leftarrow f(x_l)$ ;
             $x^* \leftarrow x_l$ ;
        end
        /* Increment number of iterations */ ;
         $it \leftarrow it + 1$  ;
    end
end
    
```

---

## 4.3 Initial Schedule

The initial solution aims to find a feasible crew and vehicle schedule by assigning a driver-vehicle pair to each decision variable (trip) while satisfying all the constraints. For this purpose, we developed a constructive greedy heuristic sketched

hereafter. The strategy employed here is similar to the Best Fit Decreasing Strategy developed for the Bin Packing Problem [13].

At each step of the greedy construction step, a trip is picked from a restricted candidate list of fixed size. This list contains the most-constrained trips. The selected trip is then assigned a value (a driver-vehicle pair) that can handle it.

After each assignment, a forward checking procedure is applied to prevent future conflicts. It performs arc consistency between the currently instantiated variable and the not-yet instantiated ones. Considering the MAX\_SPREAD constraints for example, when a driver-vehicle pair  $(d_k, v_k)$  is assigned to a variable  $t_k$ , all pairs including  $d_k$  are removed from the domain of the variables bound to  $t_k$  within constraints of type MAX\_SPREAD.

$$\forall t_j \in \mathcal{T}, (et(t_k) - st(t_j) \geq S_{max}(d_k)) \Rightarrow \forall v \in \mathcal{V}, (d_k, v) \notin \mathcal{I}_j$$

This forward-checking process is similarly applied to the other constraints.

#### 4.4 Improvement by Local Search

**Search Space.** A configuration  $\sigma$  is a *consistent assignment* of "driver-vehicle" pairs in  $\mathcal{I}$  to trips in  $\mathcal{T}$ . The search space  $\Omega$  is then defined as the set of all such assignments. Notice that our representation remedies in essence, i.e. without any additional constraint, some drawbacks raised by set partitioning or set covering models largely used for the Crew Scheduling Problem. The former is rather hard to solve and is often relaxed in the set covering formulation while the latter can lead to over-covered trips (see [10] for instance).

**Evaluation Function.** In order to guide the algorithm to visit the search space, one needs a function to evaluate the configurations. The quality of a configuration  $\sigma$  is estimated through a weighted aggregation of the different objectives.

$$\forall \sigma \in \Omega, f(\sigma) = w_1 \times f_1(\sigma) + w_2 \times f_2(\sigma)$$

$w_i > 0$  ( $i=1,2$ ) are the weights associated to the two vehicle and driver objectives.

**Neighborhood Operator.** The neighborhood is one of the most important component of any local search algorithm. In our case, we have experimented several neighborhoods using different principles such as one-change (changing the value of a variable), swap (exchanging the values of two variables), change-and-repair and ejection chain. Among these alternatives, the neighborhood based on ejection chain proves clearly to be the most powerful.

The principle of ejection chains was introduced in [7] and defined in a very general way. An ejection chain is initiated by selecting a set of elements to undergo a change of state. The result of this change leads to identifying a collection of other sets, with the property that the elements of at least one must be "ejected from" their current state. In our case, we devised an ejection chain as follows:

1. pick a trip (variable) at random,
2. pick a new driver-vehicle pair,
3. if assigning the pair to the trip does not create any conflict, the assignment becomes effective, otherwise, a new driver-vehicle pair is assigned to all the trips in conflict if a consistent assignment is possible. As a last resort, these conflicting variables are unassigned.

## 5 Computational Experiments

In this section, we assess the usefulness of our *integrated approach* and compare it with the *conventional sequential approach*. For this purpose, we carry out a series of experimentations using a set of data instances coming from real situations.

### 5.1 Data and Experimental Settings

Computational experiments were based on 7 real-world instances representing different workloads. The main characteristics of these instances are shown in Table 2 (Left Part), namely, the number of scheduled trips, available drivers and vehicles. Notice that the size of these instances are comparable to those mentioned in previous studies (see Section 1).

In our experimentations, all drivers are subjected to the same rules, namely a maximum working time of 9:00 and a maximum spread time of 12:00. Changeovers are not allowed. Two types of vehicles are considered with the following hierarchy: vehicles of type 2 can handle trips of type 1 or 2, whereas vehicles of type 1 are dedicated to type 1 only.

Our search procedures were coded in C++, compiled with VC++ 8.0 (flag -O3), on a PC running Windows XP (1Go RAM, 2.8Ghz). For the sequential approach, we developed a Branch & Bound algorithm on a multi-commodity flow model for the vehicle part and on a set covering model for the crew part (see [10]). For our GRASP procedure, 20 independent runs were carried out on each instance with different random seeds, each run being limited to 10 minutes of CPU time.

Concerning the tuning of GRASP, after some preliminary experimentations, we obtained the best results with the following set of parameters: 10 iterations ( $it_{max}=10$  in Algorithm 1), a size of 5 for the restricted candidate list, a stop criterion for each local search set to  $100 \times T$  iterations without improvement.

### 5.2 Comparison between Sequential and Integrated Scheduling

The right part of Table 2 displays the results of the sequential and the integrated approach on the 7 data sets. For each instance, we report the number of required



drivers and vehicles<sup>1</sup>. Figures for the simultaneous approach correspond to the best solution over the 20 runs (mean and standard deviation between brackets).

From Table 2, we can make two main comments. First, one observes that the integrated approach always outperforms the sequential one except for 2 instances where both approaches furnish equivalent results. In particular, the savings in terms of number of drivers are significant with up to 5 in two cases (cor\_67 and dij\_159). The sequential approach provides a lower bound for the number of vehicles that is always reached in the integrated solutions. Across the 7 instances, the results are also quite stable with very small standard deviations.

Second, the integrated approach is more powerful than the sequential one in the sense that the sequential approach failed to solve the instance "opt\_215" while solutions are possible when crews and vehicles are considered simultaneously. Indeed, when the sequential scheduling is applied to this instance, the vehicle phase results in a schedule with no relief opportunity along one bus duty and consequently leads to an insolvable problem for the driver scheduling part.

**Table 2.** Comparison between sequential and simultaneous scheduling

	Trips Drivers		Vehicles		Sequential		Simultaneous	
			type 1	type 2	drivers	vehicles	drivers	vehicles
bea_59	59	20	11	5	18	16	<b>16</b> (16.0, 0.0)	16 (16.0, 0.0)
cor_67	67	20	18	0	20	15	<b>15</b> (15.8, 0.4)	15 (15.0, 0.0)
cha_105	105	25	14	10	22	22	22 (22.0, 0.0)	22 (22.0, 0.0)
sem_151	151	30	18	12	27	27	27 (27.0, 0.0)	27 (27.0, 0.0)
dij_159	159	36	20	20	34	29	<b>29</b> (29.0, 0.0)	29 (29.0, 0.0)
otp_215	215	50	50	0	-	48	<b>49</b> (49.0, 0.0)	49 (49.0, 0.0)
aux_249	249	50	35	12	48	44	<b>46</b> (46.8, 0.4)	44 (44.0, 0.0)

These results show the dominance of the integrated approach over the sequential one. A more complete assessment would compare the results with tight lower bounds, which are unfortunately unavailable yet.

## 6 Conclusion

In this paper, we proposed a new heuristic for a simultaneous drivers and vehicles scheduling problem in an extra-urban area. The assumptions retained are suitable to tackle practical problems in rural areas. The ability of managing a heterogeneous fleet for a given depot is thus especially relevant.

The problem formulation as a constraint satisfaction and optimization model proves to be a very flexible framework for designing heuristic algorithms. From this model, we developed a solution procedure relying on a Greedy Randomized Adaptive Search Procedure which integrates constraint programming techniques and a powerful neighborhood based local search. This proposed approach constitutes the first application of metaheuristics to this difficult application.

<sup>1</sup> Information related to the total duration of deadheads and idle periods is also available, but not reported here.

The computational study carried out on a set of real-world instances clearly shows the dominance of the integrated approach over the conventional sequential one. In particular, a simultaneous schedule of drivers and vehicles achieves better utilization of both resources. Moreover, in some cases, the integrated approach is indispensable, especially when relief opportunities are rare.

Finally, let us mention that the general solution approach shown in the paper is also suitable for dynamic adjustment of schedules by local re-optimization.

**Acknowledgments.** This work was partially supported by the French Ministry for Research and Education through a CIFRE contract (number 176/2004). The reviewers of the paper are greatly acknowledged for their helpful comments.

## References

1. Ball, M., Bodin, L., Dial, R.: A matching based heuristic for scheduling mass transit crews and vehicles. *Transportation Science* 17, 4–31 (1983)
2. Borndorfer, R., Loebel, A., Weider, S.: A bundle method for integrated multi-depot vehicle and duty scheduling in public transit. TR 04-14 (2004)
3. Dallaire, A., Fleurent, C., Rousseau, J.M.: Dynamic Constraint Generation in CrewOpt, a Column Generation Approach for Transit Crew Scheduling. In: Voß, S., Daduna, J.R. (eds.) *Computer-Aided Scheduling of Public Transport (CASPT)*, pp. 73–90. Springer, Berlin (2004)
4. Freling, R.: *Models and Techniques for Integrating Vehicle and Crew Scheduling*. PhD thesis, Tinbergen Institute, Erasmus University Rotterdam (1997)
5. Freling, R., Boender, G., Paixão, J.M.P.: An integrated approach to vehicle and crew scheduling. Technical Report 9503/A, Economie Institute, Erasmus University Rotterdam, Rotterdam (1995)
6. Gaffi, A., Nonato, M.: An integrated approach to the extra-urban crew and vehicle scheduling problem. In: Wilson, N.H.M. (ed.) *Computer-Aided Transit Scheduling*, pp. 103–128. Springer, Berlin (1999)
7. Glover, F.: Ejection chains, reference structures and alternating path methods for traveling salesman problems. *Discrete Applied Math.* 65(1-3), 223–253 (1996)
8. Haase, K., Desaulniers, G., Desrosiers, J.: Simultaneous vehicle and crew scheduling in urban mass transit systems. *Transportation Science* 35(3), 286–303 (2001)
9. Haase, K., Friberg, C.: An exact branch and cut algorithm for the vehicle and crew scheduling problem. In: Wilson, N.H.M. (ed.) *Computer-Aided Transit Scheduling*, pp. 63–80. Springer, Berlin (1999)
10. Huisman, D.: *Integrated and Dynamic Vehicle and Crew Scheduling*. PhD thesis, Tinbergen Institute, Erasmus University Rotterdam (2004)
11. Huisman, D., Freling, R., Wagelmans, A.P.M.: Multiple-depot integrated vehicle and crew scheduling. *Trans. Sci.* 39, 491–502 (2005)
12. Laurent, B., Hao, J.K.: Simultaneous vehicle and driver scheduling: a case study in a limousine rental company. *Computers & Industrial Engineering* 53(3), 542–558 (2007)
13. Martello, S., Toth, P.: *Knapsack Problems: Algorithms and Computer Implementations*. Wiley, New York (1990)
14. Tsang, E.: *Foundations of Constraint Satisfaction*. Academic Press, London (1993)