An efficient algorithm for the truck driver scheduling problem

Niels De Walsche[0009−0003−0643−4352], Greet Vanden Berghe[0000−0002−0275−5568], and Pieter Smet[0000−0002−3955−7725]

KU Leuven, Ghent Campus, Gebroeders De Smetstraat 1, 9000 Ghent, Belgium
{niels.dewalsche,greet.vanden.berghe,pieter.smet}@kuleuven.be

Abstract. In a world where the wellbeing of drivers is, quite rightly, receiving more attention, algorithms that can efficiently create schedules that comply with the regulations are becoming increasingly important. This extended abstract investigates how to schedule long-distance truck drivers in accordance with the European Union's Hours-of-Service regulations. We introduce a vehicle routing algorithm that takes into account both the working hours of drivers and time windows of customers. The heuristic algorithm outperforms the current state-of-the-art approach for a public data set.

Keywords: Truck Driver Scheduling Problem, Hours-of-Service regulations, Vehicle scheduling

1 Introduction

In recent years, there has been an increased focus on employee wellbeing. This is particularly relevant for truck drivers, who are often on the road for long periods of time. The European Union has imposed regulations to ensure the safety and wellbeing of such drivers, namely (EC) No. 561/2006. These regulations are also important for drivers' employers, who risk significant penalties if their drivers are not compliant. Goel and Vidal [3] highlighted the fact that adopting hours-of-service regulations can more than double the total travel duration. It is therefore crucial for companies to carefully optimize the routes of their drivers.

The truck driver scheduling problem (TDSP), introduced by Goel [2], aims at creating schedules that comply with the aforementioned European Union regulations. Ensuring compliant schedules represents a crucial component of the Vehicle Routing and Truck Driver Scheduling Problem (VRTDSP), where the goal is to find routes for the trucks and schedules for the drivers that minimize both the total distance traveled and the number of drivers.

2 Problem description

A solution of the TDSP, consists of a sequence of distinct activities, that describe what a driver is doing at certain points in time: service, driving, break and rest. Feasible schedules must comply with a number of constraints:

- The driver must carry out a service activity at each customer location, the execution of which is restricted by a time window.
- The time it takes to travel between two customers is equal to the total driving time, which may be interrupted by breaks or rests.
- Compliance with the European Union regulations, namely:
  - A driver must take a break of at least 45 minutes if they have driven for 4.5 hours since the end of their last break or rest period.
  - A driver must take a rest of at least 11 hours if they have driven for 9 hours since the end of their last rest period.
  - A driver must rest for at least 11 hours within 24 hours of the end of their last rest period.

3 Proposed algorithm

Our truck driver scheduling algorithm is based on the multilabel method introduced by Goel [2]. Compared to the original method, we have made several modifications, such as (1) an improved labeling strategy that finds a higher number of feasible schedules and (2) an effective pruning strategy that reduces the number of schedules that have to be checked for feasibility. The vehicle routing component of the problem is solved by the SISRs algorithm introduced by Christiaens and Vanden Berghe [1]. SISRs suggests routes for the drivers and our algorithm determines whether an EU-compliant schedule can be generated for these routes.

4 Preliminary results

The performance of the proposed algorithm is evaluated using the VRTDSP instances introduced by Goel [2]. These instances are based on the Vehicle Routing Problem with Time Windows instances of Solomon [4] and modified for the VRTDSP. The instances are divided into three geographical layouts: random (R), clustered (C) and random-clustered (RC). Each geographical layout consists of two types of instances: type 1 and type 2 (e.g. R1 & R2). Type 2 instances have a higher vehicle capacity and longer time windows compared to type 1 instances.

We compare our results against the Hybrid Genetic Search with Adaptive Diversity Control (HGSADC) algorithm proposed by Goel and Vidal [3], which is currently the best performing heuristic algorithm for the VRTDSP. Given that our algorithm does not consider split breaks, split rests or extended driving times, the results are compared against the "No split" version of HGSADC. Similar to Goel and Vidal [3], we solve the problem in accordance with a hierarchical objective, where minimizing the number of drivers is prioritized over minimizing the total distance traveled. Note that the number of drivers determines the fleet size. Also note that the results for HGSADC are directly taken from the original paper and thus there is a difference in the hardware used for the experiments.

The results in Table 1 are presented in terms of the average and fewest number of drivers in addition to the average and shortest total distance traveled. Column $T_{Avg.}$ details the average execution time of our algorithm in seconds, averaged over five runs per instance. The algorithm ran for 50000 iterations, with 10% of the iterations in the fleet minimization phase. A time limit of 1 hour was set, again with a maximum of 10% of the time spent in the fleet minimization phase.

The algorithm was implemented in Rust (1.78.0). All our experiments were conducted on a computer with an AMD Ryzen 7 5800X processor at 3.8 GHz with 8 cores, 32GB of RAM and Windows 10 operating system.

Table 1: Results for the Goel [2] instances

	Our algorithm					HGSADC [3]				
	Avg. Fleet	Avg. Dist.	Best Fleet	Best Dist.	$T_{Avg.}$	Avg. Fleet	Avg. Dist.	Best Fleet	Best Dist.	$T_{Avg.}$
R1	98.60	11 860.77	**98.00**	**11 810.67**	7.09	98.80	11 769.13	98.00	11 835.89	–
R2	55.00	10 807.90	**54.00**	**10 694.42**	27.90	62.60	10 294.36	62.00	10 279.25	–
C1	90.00	7618.23	**90.00**	**7609.33**	3.23	90.40	7630.25	90.00	7628.73	–
C2	35.00	5486.92	**35.00**	**5440.08**	15.56	40.00	5754.04	40.00	5753.30	–
RC1	72.00	9101.35	72.00	8918.92	5.84	72.00	8915.07	**72.00**	**8892.74**	–
RC2	47.00	9204.53	**45.00**	**9220.92**	12.78	50.00	8960.99	50.00	8917.25	–
All	397.60	54079.70	**394.00**	**53694.33**	12.40	413.80	53323.84	412.00	53307.16	3240

The first observation we can make from from Table 1 is that our algorithm performs well when considering the primary objective: minimizing the number of drivers required, especially for the type 2 instances. Moreover, despite the reduction concerning the number of drivers, the total distance traveled remains close to the values produced by the HGSADC algorithm. Finally, it is also worth observing the significant variation concerning average execution time between type 1 and 2 instances.

By the time of our presentation at PATAT, we will present results regarding the completeness of our algorithm. Put another way: what percentage of feasible schedules does the algorithm determine to be feasible. We also aim to gain further insight into the results by investigating the difference in difficulty between the type 1 and 2 instances.

This research was supported by KU Leuven IOF (C2E/23/012: Human-centred decision support based on a new theory for personnel rostering).

References

1. Christiaens, J., Vanden Berghe, G.: Slack induction by string removals for vehicle routing problems. Transportation Science **54**(2), 417–433 (2020)
2. Goel, A.: Vehicle scheduling and routing with drivers' working hours. Transportation Science **43**(1), 17–26 (2009)
3. Goel, A., Vidal, T.: Hours of service regulations in road freight transport: An optimization-based international assessment. Transportation science **48**(3), 391–412 (2014)
4. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research **35**(2), 254–265 (1987)