N° d'ordre :
EDSPIC :

Université Clermont Auvergne

École doctorale des
Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse

présentée par

Iván Guillermo Peña Arenas

pour obtenir le grade de

Docteur d'Université

Spécialité : Informatique

The Truck Driver Scheduling Problem under the European Community Social Legislation

Soutenue publiquement le 12 septembre 2022 devant le jury composé de :

Président :

Ammar OULAMARA — Professeur des Universités, LORIA Nancy

Examinateur :

Pierre CASTAGNA — Professeur des Universités, IUT Nantes

Rapporteurs :

Ammar OULAMARA — Professeur des Universités, LORIA Nancy
Caroline PRODHON — Maître de Conférences, HDR, Université de Technologie de Troyes

Directeurs de thèse :

Philippe LACOMME — Maître de Conférences, HDR, Université Clermont Auvergne
Thierry GARAIX — Professeur des Universités, École des Mines de Saint-Étienne
Nikolay TCHERNEV — Professeur des Universités, Université Clermont Auvergne

Ce travail a bénéficé d’un financemnt géré par l’Agence Nationale de la Recherche dans le cadre du projet FITS (Flexible and Intelligent Transportation Systems).

anr agence nationale de la recherche

AU SERVICE DE LA SCIENCE

ACKNOWLEDGEMENTS

RESUME

Cette thèse porte sur la modélisation et la résolution de différents problèmes d'ordonnancement en intégrant les contraintes légales qui portent sur les temps de conduite et les temps de travail des chauffeurs de camion. Ces problèmes demandent, entre autre, une coordination entre des activités de livraisons, qui se définissent par une date de début et une durée au niveau des clients (qui possèdent des contraintes horaires de passage), et des opérations de transport, qui se définissent par une date de début, une date de fin.

Pour résoudre ces problèmes, plusieurs méthodes d'optimisation ont été proposées afin d'obtenir des solutions de bonne qualité dans des temps raisonnables. La thèse commence par une introduction qui présente d'une façon générale le problème et des différentes approches qui ont été proposées dans la littérature pour le résoudre. Cette revue de la littérature suit deux axes, les problèmes liés au Truck Driver Scheduling Problem (TDSP) et le Vehicle Routing and Truck Driver Scheduling Problem (VRTDSP). Les méthodes de résolution sont développées par ordre de complexité, en commençant par des modèles utilisant un nombre restreint de points de décision et en terminant par des modèles qui prennent en compte des hypothèses de preemption.

Le premier problème est un problème d'ordonnancement/transport de type Truck and Driver dans lequel les pauses des chauffeurs ne sont planifiés qu'au niveau des clients ce qui impose que les temps de transport de dépassent pas 4h30. Il s'agit d'une modélisation parfaitement adaptée aux transports régionaux ou nationaux dans lesquels les distances entre deux clients ne dépassent pas 4h30. Il faut noter que le retour du camion au dépôt n'est pas modélisé dans l'évaluation de la tournée. Ce problème fait l'objet du chapitre 2 : une modélisation linéaire et un algorithme de programmation dynamique sont proposés et testés sur un nouveau jeu d'instances.

Le dernier problème traité concerne la résolution du Truck and Driver, dans lequel les pauses de chauffeurs peuvent être planifié à tout moment, que ce soit chez le client ou au milieu d'une activité (conduit ou service). Ce problème est plus complet car il modélise des cas plus généraux dont en particulier les transports grande distance entre les clients. Sur ce problème différentes contributions sont réalisées : la première concerne la proposition d'un modèle linéaire qui étend le modèle du chapitre 2 et trois versions d'un modèle de programmation dynamique.

Cette thèse présente différentes méthodes de solutions pour le problème TDSP tout en considérant les règles hebdomadaires de l'European Community Social Legislation, étendant dans ce sens toutes les contributions précédentes dans la littérature. En particulier, la règle du travail de nuit qui a été simplifiée ou rejetée dans le passé. En outre, un nouveau point de référence avec des solutions optimales détaillées est fourni. L'efficacité des méthodes proposées et les implications des différentes règles, en particulier la contrainte du travail de nuit, sont discutées.

ABSTRACT

This thesis is about models and solution methods for different routing and scheduling problems, which consider legal constraints related to the driving and working time of the truck drivers. These kind of problems require, among other things, coordination between delivery activities, which are defined by a starting date, the service duration at customers (who have time windows), and transport operations, which are defined by starting and finishing dates.

In order to solve this models, different optimization methods were proposed to achieve good quality solutions in a reasonable running times. The thesis starts with an introduction, which presents an overview of the problem and the different approaches that have been proposed in the literature to solve it. This literature review follows two lines, problems related with the Truck Driver Scheduling Problem (TDSP) and the combined Vehicle Routing and Truck Driver Scheduling Problem (VRTDSP). The solution methods are developed in order of complexity, starting with models using a restricted number of decision points and ending with models that can handle pre-emption assumptions.

The first problem is a Truck Driver Scheduling problem in which breaks and daily rests are only scheduled at the customer locations, which means that transport times between customers must not exceed 4.5 hours. This model is perfectly suited to regional or national transport where distance between two customers does not exceed 4.5h. This problem is the subject of chapter 2: a linear model and a label setting algorithm are proposed and tested on a new set of instances.

The second problem addressed concerns to the solution of the Truck Driver Scheduling problem, in which breaks can be scheduled at any point of time, whether at customer locations or in the middle of an activity (driving or service). This problem is more complete because it models more general cases including in particular long distance transports between customers. On this problem different contributions are made: the first is a linear model which extends the model from chapter 2, and three versions of a label setting algorithm.

This thesis presents different solutions methods for the TDSP problem while considering all the weekly rules from the European Community Social Legislation, extending in this sense all previous contributions in the literature. In particular, the night working rule that has been whether simplified or discarded in the past. In addition, a new benchmark with detail optimal solutions is provided. The efficiency of the proposed methods and the implications of the different rules in particular the night working constraint are discussed.

CONTENTS

GENERAL INTRODUCTION

To improve working conditions, to define a clear set of rules for the competition between overland transport companies and to increase road safety, the European Union has established the European Community Social Legislation on driver's and working hours Rules. This legislation defines a framework of rules that impose breaks or daily rests according to the working time. Since, freight cargo companies are responsible for any infringement to these set of rules committed by their employees, the application of this regulation has modified in a strong way their operations.

The problem of scheduling breaks and rests for a given sequence of customers, considering a given regulation is the Truck Drivers Scheduling Problem (TDPS). There are previous works considering the problem of including breaks and rests for drivers while solving the vehicle routing problem, albeit it is just after the entrance in vigor of this regulation that the problem starts to receive more attention. When the TDSP problem is considered into a vehicle routing problem, it is referred as the Vehicle Routing and Truck Driver Scheduling Problem (VRTDSP). The regulations can vary from one country to another and in this thesis we considered the EU social legislation.

There are diverse contributions with regard of the different regulations. In the case of the EC Social legislation, at the beginning they only consider a basic set of rules to bring a feasible or legal solution. Later, they become more comprehensive adding extensions from the driving hours rules and the directive 2002/15/EC related to working hours. However, with respect to the working hours directive, the night working rule has been discarded or more lately, implemented as forbidding the night working. In particular, the night working rule states: if night work is performed, the daily working time does not exceed ten hours in each 24h period. From the algorithmic point of view this entails two challenges, especially for label setting algorithms, one of the most used methods to solve the TDSP. First, updating the resources in a sliding 24h time window. Second, designing efficient dominance rules to achieve optimality in a competitive computational time.

There are two types of options while modeling the places to schedule a break or a daily rest. First, breaks can only take place at customer or at specific locations, i.e. parking lots. The approach reduces the complexity of the problem, since the number of variables diminishes and it makes a lot of sense from the real life application, that drivers does not take breaks or rests anywhere. Second, breaks take place as soon as they are required, including in the middle of an activity. In the literature, this is the most common approach, since it is more complex from the algorithmic point of view. This preemption assumption has been considered for the driving activity but not for the service activity. In the thesis, we consider the two approaches, starting with the less complex and finishing with the case that works under pre-emption assumptions.

Several types of solutions methods exact and heuristic have been proposed to solve the TDSP and the VRTDSP. The vast majority of contributions working on a planning horizon of one week. Among them, linear formulations and label setting algorithms are the most widespread. With respect to linear formulations, even the most recent do not consider all the rules from the EC Social Legislation, i.e. split daily rests and the directive on working hours. Moreover, in terms of the complexity of the problem, there is a conjecture that the TDSP is NP-Complete under the EC Social Legislation. Thus, the rule of thumb has been to develop methods to find a feasible or legal schedule for a given sequence of customers, instead of solving the problem to optimality.

This Thesis works on the Truck Drivers Scheduling Problem. In order to gain a better understanding of the regulation, we develop linear formulations, first scheduling breaks at fixed points in the sequence, and later using a preemption assumption. The main contribution of the thesis is to develop efficient and optimal solution methods to solve the TDSP for a given sequence of customers. The thesis is divided into 3 chapters:

Chapter 1 presents a description of the European Community Social Legislation alongside with the terminology used throughout the thesis, and a literature review on the Truck Driver Scheduling Problem. The focus of the thesis is the TDSP Problem for scheduling driving and working hours of truck drivers with respect to the European Community social legislation. However, since the TDSP is generally used as a subroutine in the Vehicle Routing and Truck Driver Scheduling Problem, this kind of problems are also subject of the literature review. This chapter brings an overview of what has been done on the subject and positions the objectives and the scope of the thesis with respect to this research field.

The second chapter presents a novel Mixed Integer Linear Formulation and a Label Setting algorithm for the Truck Driver Scheduling problem considering the European driver regulations. Both solution methods use the assumption that breaks are only possible at customers or at specific locations, as well as previous contributions. The linear formulation presented in this chapter includes some week rules that have not been included in the past, for instance: breaks due to working time and split daily rests. Thus providing in this sense a more complete mathematical description of the EC Social Legislation. Finally, even though the problem has received a lot of attention it is difficult to find a set of instances with a thorough description of their solutions, thus a new set of instances are proposed in order to test the performance of the models and detail optimal solutions are provided.

The chapter 3 proposes a MILP formulation and a Label Setting algorithm to solve the TDSP problem under preemption assumptions, that is to say, breaks could take place in the middle of a transport or a service activity. Previous approaches only consider preemption on driving activities. Moreover, these models include all weekly rules from the EC Social Legislation using a fixed sequence of customers. In this sense, both models extend previous contributions since they include the night working constraint. In addition, they do not only strive to find a feasible solution, but to provide an optimal solution. Therefore, the major contribution of this chapter is to bring two optimal procedures to schedule breaks and rests considering the EC Social Legislation. Computational experiments measure the efficiency of the proposed methods and the effect of the night working constraint on the feasibility of the schedules.

List of publications

International Conferences

Garaix, T., Lacomme, P., Tchernev, N. and Peña-Arenas, I., "An exact label setting algorithm for the truck driver scheduling problem considering the European Community Social Legislation", Odysseus 2022, May 2022.

Peña-Arenas, I., Garaix, T., Lacomme, P. and Tchernev, N., "A mixed integer programming formulation for the truc drivers scheduling problem considering the European Union driver rules", 2021 IEEE 17th International Conference on Automation Science and Engineering, CASE 2021: 101-106.

Bourreau, E., Farias, K., Garaix, T., Lacomme, P., Martino, D., Peña-Arenas, I. "On the Truck Driver Scheduling Problem: A Constraint Programming based approach". Euro Athens 2021.

National Conferences

Garaix, T., Lacomme, P., Peña-Arenas, I., Tchernev, N. "Linear formulation of the driver-scheduling problem under the european driving rules regulation". 22éme Congrès annuel de la Société français de Recherche Opérationnelle et d'Aide à la Décision (ROADEF), Avril 2021. France.

Bourreau, E., Farias, K., Garaix, T., Lacomme, P., Martino, D., Peña-Arenas, I. "First Constraint programming based approach for the Truck Driver Scheduling Problem". 22éme Congrès annuel de la Société français de Recherche Opérationnelle et d'Aide à la Décision (ROADEF), Avril 2021, France.

Garaix, T., Lacomme, P., Peña-Arenas, I., Tchernev, N. "A label setting algorithm for the truck driver scheduling problem under the European community social legislation". 23éme Congrès annuel de la Société français de Recherche Opérationnelle et d'Aide à la Décision (ROADEF), Février 2022. France.

CHAPTER 1

Introduction

This chapter presents a description of the European social legislation, the problems that are going to be treated, the objectives and the structure of the thesis. The objective of the European Community social legislation on driver's and working hour rules is to harmonise the competition environment among transport companies, improve the working conditions and to increase the road safety. A description of this set of rules is presented along side with the terminology used throughout the thesis. Since the freight companies have to follow this legislation, dispatchers or planners have to include this new set of constraints increasing the complexity of the routing problem, which itself was already a difficult problem. This is the reason why the problem has received a lot of attention by researchers. In particular, our main focus are two problems the Truck Driver Scheduling Problem (TDSP), for scheduling driving and working hours of truck drivers with respect to the European Community social legislation and the combined Vehicle Routing and Truck Driver Scheduling Problem (VRTDSP). Finally, the objectives and the structure of the thesis are outlined.

1.1 European Community social legislation on driving and working hours in road transportation

An important actor in the number of fatalities in road crashes are the Heavy Goods Vehicles (HGV). Since fatal road collisions involving HGVs is more frequent than for other types of vehicles due to the vehicle's size and mass. The number of people who dies in collisions involving HGV vehicles per-Km basis, is up to three times as many people who die in collisions involving non-goods vehicles [1]. The EU-average fatality rate in accidents involving HGVs is 8.1 per million population, and ranges from 1.5 in Estonia to 20.6 in Poland [2]. The percentage of all road fatalities involving HGV between 2010 and 2018 in the EU is on average 14.85% [3], see Figure 1-1.

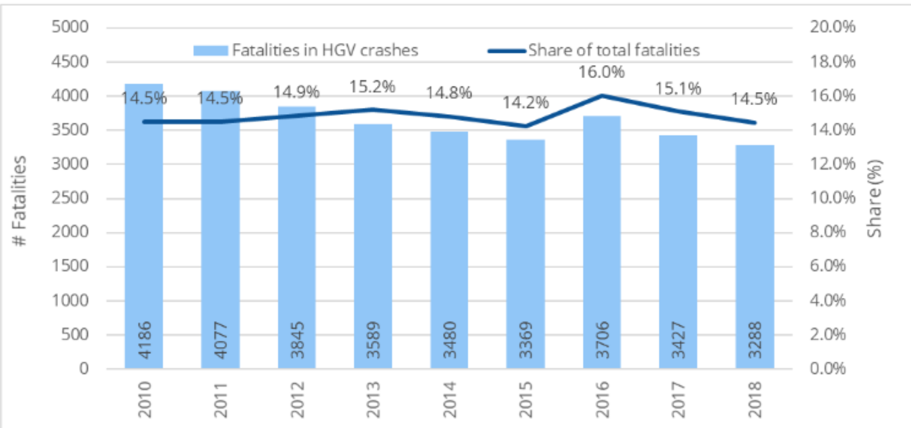


Figure 1-1. Annual number of fatalities in HGV crashes EU27 (2010-2018). Source: CARE.

Driver fatigue is one of the main reasons that reduces road safety and approximately 20% of the commercial road transport collisions are due to fatigue and over 50% of long haul drivers have at some time fallen asleep at the wheel [1][7]. Some factors that increase fatigue are the time of the day, sleep deprivation and time spent driving. During the day there are two periods of time were the maximum threat of having an accident are the highest, these are between 2:00am - 5:00am and 2:00pm – 4:00pm, this fact is related with the circadian rhythm [4]. Reduction in the amounts of sleep has a deep correlation with the probability of crashes, even mild degrees of sleep loss, i.e. from 9h to 7h of sleep, affects the driving performance; moreover, extended expositions to sleep deprivation, carries a large and cumulative alertness deterioration, which is not recovery by consecutive large periods of sleep [5]. Finally, the length of the time spent driving diminishes the driving performance, higher rates of accidents are related with both, working long hours and long periods of driving [6].

Therefore, in order increase road safety by preventing driving fatigue, to improve working conditions and to define a clear set of rules for the competition among modes of overland transport, especially the road sector, the EU has established the European Community social legislation on driver's and working hours Rules, from now on in the document EC social legislation. It basically relies in two legislative acts EC No 561/2006 related with driving hours and Directive 2002/15/EC referring to working hours for people engaged in road transportation.

The EC social legislation concerns to drivers of good vehicles where the maximum valid weight, including any trailer, or semi-trailer, exceeds 3.5 tonnes, or of passengers with more than 9 seats including the driver. This regulation applies regardless of the country of registration of the vehicle and to road transport within the European Community or between the European Community and Switzerland and the countries party to the agreement on the European Economic Area. If the transport operation is partially done outside of the areas previously mentioned, the agreement of the Work of Crews of Vehicles Engaged in International Road Transport (AETR) applies if the vehicle is registered in the EC or in a member country of the AETR [8]. For more details about the legislation, exemptions and national derogations please refer to https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32006R0561

1.1.1 Definitions

In this section some terms that are going to be used throughout the thesis are introduced. Additionally, they are going to be useful to describe the set of rules included in the EC social legislation that are going to be considered. Most of these set of terms rely on the legislation, although, some others are included for the sake of explanation.

Driving. Transportation activity between clients and depot(s) and the other way round.

Service/other work. They are all the activities such as, loading/unloading, cleaning and technical maintenance, etc. that are not driving.

Break. It is the period of time where the driver does not have to develop a service or driving activity, and use it for recuperation.

Full break. A break with a duration of at least 45 minutes.

Week. Encompass the period of time between 0:00 hours on Monday and 24:00 hours on Sunday.

Driving time. Duration of the driving activity.

Working time. It is related to the time assigned to process service and driving activities.

Waiting or Period of Availability (POA). Period of time where a driver is not processing working time nor taking a break, although remains available to resume his activities.

Shift spread. It is the working day between rest periods composed by working time, breaks, and POA, as stressed on Figure 1-2.

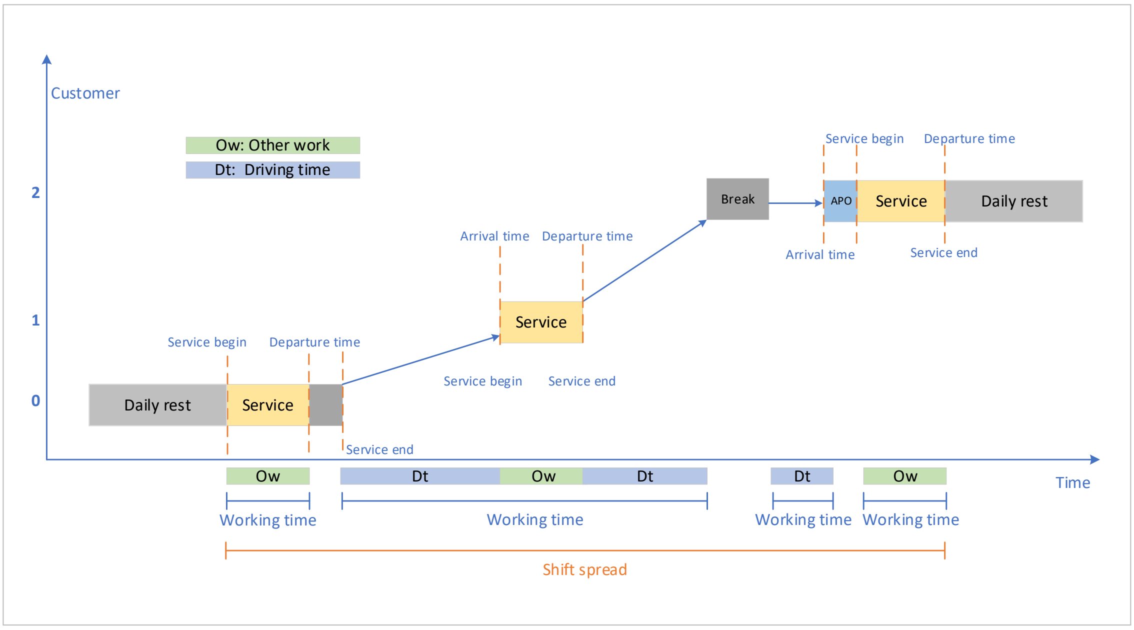


Figure 1-2. Shift spread.

Daily driving time. Cumulated driving time between rest periods or during a shift.

Daily working time. Cumulated working time between rest periods or during a shift.

Weekly driving time. Cumulated driving time during a week period.

Daily rest period. Daily period where the driver freely dispose of his time.

Reduced daily rest. A daily rest period that has a duration of at least nine hours but less than 11 hours.

1.1.2 Regulation (EC) No 561/2006 on driving hours

The Regulation (EC) No 561/2006 establishes rules on maximum driving hours and enforce the minimum duration and requirements for breaks and rest periods. It compels to freight cargo companies to organize the work of their employees in such a way that they are able to follow the set of dispositions of this rule. The transport company is responsible for any infringement committed by their employees. In addition, the regulation demands that every party involved in the transportation process, i.e. the transport undertakings, consignors, forwarders, tour operators, principal contractors, subcontractors, and even driver employment agencies ensure that driver schedules follow the legal requirements [8, article 10]. As a result, this regulation modifies in a strong way the operations of transport companies.

Following a brief description of the rules that concern this regulation.

Driving periods. Minimum requirements between driving times within a shift spread.

Full break. After 4.5 hours of continuous driving, a break of at least 45 minutes must be taken. Each time that a full break takes place; another period of 4.5 hours of driving time can take place.

Split driving break. A driver can split a full break into a break of at least 15 minutes followed by another of at least 30 minutes (in that order). Once the 30 minutes break takes place the driving time is reset to zero, and another period of 4.5 hours could take place.

A break of 30 minutes is considered as a full break, only if previously a break of at least 15 minutes but, strictly less than 45 minutes has been scheduled. Breaks of less than 30 minutes cannot become a full break. Figure 1-3 presents a split break. In order to have a valid split driving break, the first part of the split break could take place at any time t between 0 and 4.5 hours, and the second break will take place after $4.5h - t$ hours of driving. After, the second driving break, the driver can drive for another 4.5 hours; in other words, the driving time is reset to zero.

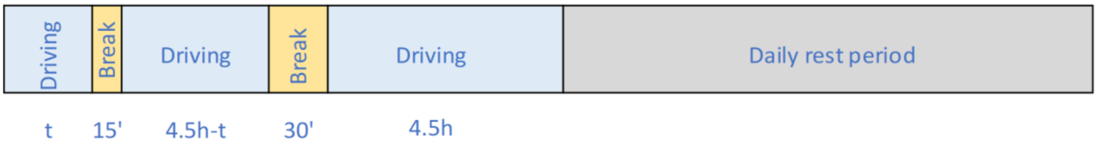


Figure 1-3. Driving periods and split breaks. [9].

Daily driving time. The total driving time during a day cannot exceed 9 hours.

Daily driving time extensions. The daily driving time could be extended up to 10 hours two times per week.

Daily rest period. The duration of a daily rest period is at least 11 hours. Within each period of 24 hours after the end of the previous daily rest period or weekly rest period a driver shall have taken a new daily rest period.

Daily rest reductions. Three times per week a daily rest period could be reduced up to 9 hours.

Split daily rest. A daily rest period could be taken in two periods, the first an uninterrupted period of at least 3 hours and the second an uninterrupted period of at least 9h.

Weekly driving times. The total driving time during a week duration shall be less of 56 hours. In addition, there is a maximum of 90 hours fortnightly driving.

Weekly rest periods. Every week should have a rest period of at least 45 consecutive hours after, at most, 144 hours from the end of your last weekly rest. Although, this period could be reduced to 24 hours, provided the reduction is settled account with an equivalent period of rest before the end of the third week following the week with the reduction, and at least one full rest is taken in any fortnight.

1.1.3 Directive 2002/15/EC on working hours

Directive 2002/15/EC establishes the minimum requirements in relation with the organisation of working time of persons involved in mobile road transport activities. This directive is a supplement of the driving rules from the regulation EC No 561/2006. Therefore, both regulations must be considered in the routing and scheduling problem. The working time comprises activities such as: driving, loading and unloading, assisting passengers boarding and disembarking from the vehicle, cleaning and maintenance, and all other work intended to ensure the safety of the vehicle, its cargo and passengers or to fulfil the legal or regulatory obligations directly linked to the specific transport operation under way [10]. During the periods of availability (POA) the driver cannot freely dispose of his time, and must be ready to resume normal work. ***Hence, they are not considered as working time, nor as a rest or a break, but they are included in the shift.***

The directive includes the following rules on working periods, weekly working times, and night work.

Working breaks. Break requirements between working times within a shift.

Continuous working time. The maximum accumulated working time without a break is at most 6h. A break should be at least 15 minutes long.

Working shift breaks. If the daily working day is between 6 and 9 hours in total, a break of at least 30 minutes should be scheduled. If the daily working day exceeds 9 hours in total, a break of at least 45 minutes should be scheduled. Breaks may be taken in periods of 15 minutes.

Weekly working time. The maximum weekly working time may not exceed 60 hours and over four months, an average of 48 hours a week must not be exceeded.

Night work. If night work is performed, the daily working time should not exceed 10 hours in each 24 hours' period, see Figure 1-4.

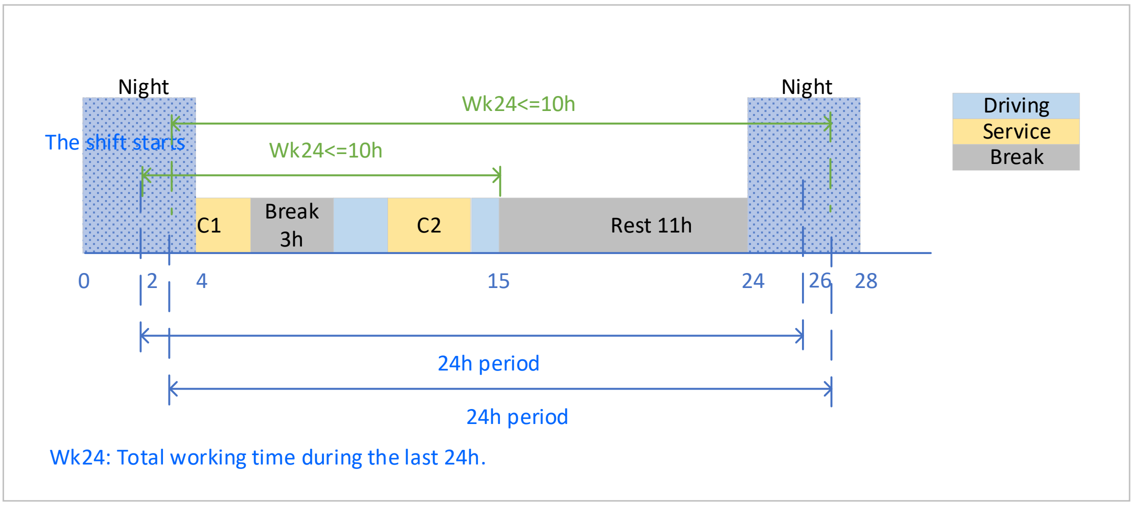


Figure 1-4. Night working constraint.

1.2 Related work

In general, in the Truck Driver Scheduling Problem (TDSP) a sequence of customer locations is given, some working time is processed at each location, multiple or single time windows apply on the starting time of the service at each location, and there is a transport time between the current customer to next. The objective is to find a feasible or optimal schedule of breaks and rests considering a particular regulation on driver's working hours. The importance of this problem is that it could be used within an integrated procedure for solving a vehicle routing problem, called Vehicle Routing and Truck Driver Scheduling Problem (VRTDSP), which as it was stated before, arise in real life applications, since most of the freight forward companies must design their vehicle routes considering the driver's working hours regulations.

When considering a solution method for the TDSP embedded into a vehicle routing procedure, it should have both properties: retrieve feasible solutions, when they exist and good speed performance. The ability of finding feasible solutions for a given trip, or sequence of customers, allows to explore different areas of the solution space, increasing the diversity of the solutions, and the overall quality of the procedure. On the other hand, the solution method should be fast, due to the vast number of sequences that are evaluated during the routing process.

1.2.1 The Truck Driver Scheduling Problem

Among the different works presented to cope with the TDSP, Archetti and Svelsbergh [11] consider a problem with a set of transportation request, each of them with a pick up and a delivery location, but time windows are imposed only at the pick up location. The pick up and delivery times are not addressed. The feasibility of a sequence of pick up and delivery transportation request is achieved considering the United States Hour Of Service. In particularly, after a certain amount of driving time *Tdrive* or duty time *Tduty* (driving + waiting) a compulsory rest of at least *Trest* hours should be scheduled. An algorithm with a complexity of $O(n^3)$ is proposed to check the feasibility of a driver schedule.

Later, given a sequence of customer locations that must be served within their time windows, Goel [12] presents an algorithm to find feasible schedules for working and driving hours of truck drivers, if they exist, considering the European Union Regulation (EC) No. 561/2006. In particularly, the rules that were considered are breaks for driving and daily rest periods each 24h. In addition, it investigates the possibility of splitting in two parts the driving breaks and the daily rests. Different modifications are applied on the method and they are compare with previous scheduling approaches embedded into vehicle routing procedures presented in [9] and [13]. Results shows that split breaks do not bring any benefits in the long distance transport; moreover, it is possible to reduce the computational effort while achieving the 90% of feasibility by reducing the number of partial schedules explored at each iteration.

Drexl and Prescott-Gagnon [14] besides computing a legal schedule for a given route, they simultaneously create a route and compute the legal schedule considering the Regulation (EC) No 561/2006 on driving hours. In particularly, this paper considers the following set of rules from the EC social legislation:

- Breaks for driving, with the possibility of splitting breaks.
- Maximum driving time per day.

- Extensions on driving time up to one hour two times per week.
- Weekly and fortnightly driving time.
- Daily rests and split daily rests.
- Weekly rest periods.

The problem is referred as Elementary Shortest Path Problem With resource constraints and Driver's Rules (ESPPWDR), which is considered as a variant of the Elementary Shortest Path Problem with Resource Constraint (ESPPRC) that takes into account the Driver's Rules. Different versions of a label setting algorithm are described, one is exact, meaning that the algorithm finds a legal schedule if one exists, and two heuristic approaches which do not guarantee to find a feasible schedule if it exists, but with faster computational times. Finally, the paper proposes the conjecture that the TDSP is a NP-Complete problem.

Goel and Kok [15] consider a sequence of λ locations where some stationary work should be developed, multiple disjunct time windows are imposed at each of them, and there is a driving time required to move from a given location to the next. The objective is to find a feasible schedule where a truck driver develops the working and driving activities within the corresponding time windows, while considering the United States Hours of Service regulations. They develop a scheduling method which guarantees to find feasible driver schedule if it exists. The method solves the problem in $O(\lambda^2)$ in the case of single time windows. In addition, this result is extended to the case of multiple time windows per customer, if the between time between them is at most the minimum rest duration.

Considering double manned vehicles and a given sequence of customer locations of size λ, Goel and Kok [16] present an algorithm that can find a feasible schedule if it exists in $O(\lambda^2)$. Some of the characteristics of the problem are time windows on the starting times of the service activities at each customer location and a planning horizon of one week. In this algorithm, they considered the EC social legislation for team drivers, in particular, minimum duration of a rest period, maximum elapsed time between the end of a rest period until the end of the next daily rest period, maximum driving time and extensions on maximum driving time.

An exact algorithm is presented in [17] to solve the TDS under Australian Standard rules and Basic Fatigue Management. Additionally, they present four heuristics, based on removing some of the most computationally expensive steps of the exact algorithm. The computational time diminish, albeit, they do not find all feasible solutions when they exists.

Using MIP based approaches Goel [18] presents a generic model for the TDSP problem, which is tuned to consider the EC social legislation and the United States Hours of Service. A sequence of customers is known in advance and multiple disjunctive time windows apply on the starting time of the service developed at each customer. The objective is to find feasible schedules with minimum duration. In the model, breaks and rest periods are taken only at customers or at suitable locations. There are considered the following set of rules from the EC social legislation: Daily rests and split daily rests, maximum daily driving time, breaks for driving and split driving breaks, and breaks for working time after 6h of work. In addition, a dynamic programming algorithm is developed to solve the problem and the computational time is smaller in comparison with the MIP solver. Other results also show that some schedules becomes infeasible when breaks/rest are only possible at customers or at suitable locations.

Considering other regulations, Goel et al. [19] present a linear model and two heuristics based on label setting algorithms to solve the TDSP under the Canadian Commercial Vehicle Drivers Hours of Service Regulation. The objective is to determine if a sequence of customer locations can be served within their time window while scheduling the compulsory breaks/rest of the Canadian driver regulations. In this paper, all parameters representing time are multiple of 15 minutes. As well, Goel in [20] formulates a MIP for the Canadian driver's regulation. The main differences with respect to the models presented in [19] are:

- The objective function is the minimization of the trip duration.
- Multiple time windows at each location are considered.
- Breaks/rest are taken only at customer locations.

Here, he solves the model using a commercial solver and compares it with two modified versions of the dynamic programming algorithm presented in [19]. Computational results show that the duration of the schedules could be notably reduced when besides finding feasible solutions, the objective function is the minimization of it. Although, these reductions imply increases in the computational effort.

Goel [21] propose a mixed integer programming model to minimize the duration of the truck driver schedules considering the Australian Heavy Vehicle Driver Fatigue Law. Breaks and rest periods could only take place at customers or at suitable parking lots and each customer has multiple disjunctive time windows. All parameters representing time are multiple of 15 minutes and the parking lots are dummy customers with zero service time. Cuts are included into the formulation in order to improve the computational running times. The MIP model is benchmarked against two dynamic programming algorithms. The first dynamic programming version is an adaptation of the algorithm presented in [20], where the objective is to find a feasible schedule if it exists. The second dynamic programming algorithm is from Goel [17], which minimizes the duration of the schedule, although, the computational effort increases. The experiments prove that the duration of the schedules could be reduced about 9%, if they are compared with the schedules of algorithms, which only search for a feasible solution if it exists.

A MIP model is introduced by Koç et al. [22] to solve the TDSP problem with idling [break] options considering the United States Hours of Service regulation. While a break/idle time the driver wants to keep the vehicle at adequate comfort level which implies fuel/energy consumption. The authors explore two cleaner idling options Electrified Parking Space (EPS) and Auxiliary Power Unit (APU) instead of the keeping the vehicle engine running, which is the alternative that pollutes the most . The driving idea is how to combine the compulsory rest and breaks imposed by the truck driver's legislation with the choice of idling options while minimizing the operational costs.

Finally, Sartori et al. [23] introduce the Truck Driver Scheduling Problem with Interdependent Routes considering the European Union social legislation. The objective is to develop feasible schedules for a given set of routes that complies with customer time windows and the EU social legislation. With respect to the EU regulation the model considers rules related to breaks for driving, breaks for working more than 6h, maximum daily driving time and maximum shift duration. Only one type of break of 45 minutes is considered, breaks only can take place at customer locations after the service time and the total driving time is restricted to one day. As in the TDSP problem the set of routes is given and their feasibility with respect to the daily driving time is checked in a pre-processing step. In order to solve the problem a Mixed Integer Linear Programing (MILP) formulation and a Label Propagation Algorithm (LPA) are developed. According to their numerical experiments the LPA is 20 times faster than the MILP.

As table 1.1 presents, most of the papers which consider the TDSP were published between 2009 and 2012 and the most recent work is from 2022 under the EU social legislation. Most of the papers consider the Regulation (EC) No 561/2006 on driving hours, among them Drexl et al. [14] describes the model that takes into account the complete regulation on driving hours, albeit, discards the regulation on working hours. Moreover, a common assumption is to consider a planning horizon of one week, because, most real-life operations are prepared for a short time framework (one week). Concerning to the Directive 2002/15/EC on working hours only Goel [18] considers the rule on working breaks. In particular, after 6 hours of working time a break must be scheduled, although, the constraints on working shift breaks are not considered. In this sense, there are no previous publications who work on the TDSP, which explicitly consider the night working rule nor the working shift breaks.

1.2.2 The Vehicle Routing and Truck Driver Scheduling Problem

Before the entrance in vigor of the current EC social legislation in 2007, Brandao and Mercer [24] present one of the seminal papers including driver's legislations, considering the United Kingdom regulation. This paper considers the Multip-Trip Vehicle Routing and Scheduling problem, with the following characteristics:

- More than one trip per vehicle.
- Time windows at customer locations
- Heterogeneous fleet of vehicles.
- Outsourcing options.
- Access to some customers restricted to some vehicles.
- Unloading times.
- The driver's schedules must respect the UK regulation.

Table 1.1. Papers related to the TDSP.

Papers	**Regulation (EC) No 561/2006 on driving hours**								**Directive 2002/15/EC on working hours**			
	Driving periods (full & split breaks)	*Daily driving time*	*Daily driving extensions*	*Daily rest period*	*Daily rest reductions*	*Split daily rest*	*Weekly driving times*	*Weekly rest periods*	*Working breaks*	*Weekly working time*	*Night work*	***Different regulation***
Archetti, C. and Svelsbergh, M. (2009) [11]												×
Goel, A. (2010) [12]	×	×		×		×						
Drexl, M. and Prescott-Gagnon, E. (2010) [14]	×	×	×	×		×	×	×				
Goel, A. and Kok, L. (2012) [15]												×
Goel, A. and Kok, L. (2012) [16]		×	×	×								×
Goel, A., Archetti, C., Savelsbergh, M. (2012) [17]												×
Goel, A. (2012) [18]	×	×		×		×			×			×
Goel, A. and Rousseau L-M. (2012) [19]												×
Goel, A. (2012) [20]												×
Goel, A. (2012) [21]												×
Koç, C., Bektas, T., Jabali, O., and Laporte, G. (2016) [22]												×
Sartori, C., Smet, P. and Vanden, G. (2022) [23]	×	×		×					×			

In particular, two rules from the UK regulation are compulsory in their schedules, first, a maximum legal driving time of 540 minutes and second, a driver cannot drive for more than 4.5 hours consecutively without having a break of 45min. In order to model breaks, they use the same approach of Rochat and Semet [25] (1994) in 1994, considering them as fictitious customers (dummy customers) with time windows, almost 17 years later, the same approach is used by Kok et al. [26] or Goel [18].

Moreover, there were some efforts to consider breaks/rests within the vehicle routing problems before the current EC social legislation. Savelsbergh, M. and Sol, M. [27] propose a model on the General Pickup and Delivery Problem. Among the problem characteristics, it considers a heterogeneous fleet of vehicles, 40% of demand arrives during the working day and vehicles outsourcing. This paper does not address the driver's legislation per se, albeit some considerations on breaks and daily rests are made. For instance, daily rests for the night at sleeping locations which includes the driver's home, weekly rest from friday night to monday morning and during the working day one break for lunch of 45 minutes that must be taken between 11 am and 2 pm.

One of the first papers considering a specific drivers' regulation set of rules is presented by Xu et al. [28]. They consider a problem where a set of pickup and delivery orders with multiple time windows at both delivery and pickup locations, must be covered by multiple carriers and multiple type of vehicles. Orders and carrier/vehicle types must satisfy a set of compatibility constraints, some orders can not be shipped together, carrier/vehicle types cannot cover all the orders and there are loading and unloading sequences of precedence that must be respected. In addition, the problem takes into account the Hours of Service regulation for each trip. The objective function considers different factors such as: total mileage, fixed charge, total waiting time, and total layover time of the driver. They formulate the problem as a set partitioning problem after solving the linear relaxation with a column generation procedure, the integer solution is found by solving the original formulation considering only the set of columns generated while solving the linear relaxation; in this step optimality is lost. One of their results is that under the characteristics of this problem, i.e. complex cost structure, Driver's rules and multiple time windows, they make the conjecture that only the trip scheduling problem is a NP-hard problem.

An insertion heuristic applied to vehicle routing and scheduling problems is presented by Campbell and Savelsbergh [29]. The main objective is to maintain the complexity of the algorithm in $O(n^3)$ or at most in $O(n^3 \log n)$, while increasing the difficulty of the problem by considering different constraints like: time windows, shift time limits, variable delivery quantities, fixed and variable delivery times, and multiple routes per vehicle. With respect to the shift limits, a particular drivers rule is not cited, albeit they follow the directive of the Department of Transportation, considering limits on duty and driving time per day, and a limit on the number of working hours every eight consecutive days.

Zapfel, G. and Bogl, M. [30] consider an integrated vehicle and scheduling problem within a planning horizon of one week. There is one depot and multiple demand points, where it is necessary to design a set of routes for the activities of pre-carriage (pickup of shipments) and post-carriage (delivery of shipments), while considering:

- Time windows at the demand points.

- A heterogeneous fleet of vehicles.
- Personnel assignment of leasing and own drivers.
- Outsourcing of freight carries.

Even though, they do not explicitly cite a given driver´s regulation, a series of constraints on drivers scheduling are taken into account. Such as, regular working time over a week, maximum driving time per day, breaks of 45 minutes after 4.5 hours of driving and daily rests.

Bartdoziej et al. [31] work on a combined vehicle and crew scheduling problem arising in the air cargo transportation. This is an allocation problem, where trucks and drivers should be assigned to a set of trips. Some of the problem characteristics are on-time requirements of the airline, the compatibility between trucks and trips, a pair of drivers is assigned to a truck and legal regulations on driver's rests. As well in this paper, they do not address a particular driver's regulation, although two legal regulation rules are considered, a regular rest of a fixed duration after a maximum driving time during the day, and one short break in the middle of the daily maximum driving time. Since, it is a multi-manned vehicle, the truck is in movement during the short break and it has to fully stop in order to take the regular daily rest.

A rich vehicle routing problem is solved by a column generation algorithm in [32]. The model considers time windows to develop the loading/unloading operations at both customers and depots, each customer has a set of orders, which can be split, outsourcing of courier services is possible, they also include different types of incompatibilities, i.e. between vehicles and items and locations or between items and locations, and finally rest periods due to driving time. This paper also does not address a certain driver's regulations, and the rule considered to schedule rests is, that *each driver has to rest for a period of duration T for every driving period of duration D*. Moreover, the rest period can be taken in smaller breaks, and periods of waiting time and loading/unloading are considered as breaks if they are taken without interruption for at least a period of time τ.

After the previous paper, most of the publications will explicitly state the particular driving regulation that is considered, avoiding confusions about the application context of rules, but also leading researchers to deploy flexible algorithms that can consider multiple set of regulations at the same time. Goel [9] is one of the first publications that considers the problem of scheduling breaks/rest according to EC social legislation after its entrance into effect in April 2007. In the scheduling problem are considered rules on, full breaks due to driving time, maximum driving time per day, and daily rests each 24 - hour period. Two scheduling procedures are proposed, a naive method, which schedules breaks and rests as soon as they are required and a multilabel method that is able to schedule daily rests before the driving time is exhausted, while exploring multiple schedules within a recursive method. A Large Neighborhood Search method is used to solve the vehicle routing problem with time windows and the EC social legislation. Computational results show that the multilabel method obtains better results than the naive method, taking advantage of scheduling rests in an earlier fashion.

A Dynamic Programming heuristic is proposed in [13] to solve the vehicle routing problem with time windows and the EC social legislation. This algorithm has two parameters, the number of states H at each node and E which is the number of expansions of a single state. As the route is constructed by the restricted dynamic programming algorithm, locally another heuristic procedure schedule the required breaks or rests periods. They present two different scheduling heuristics, which are a modification of the naive method presented in [9]. With respect to the EC social legislation they consider both, the (EC) No 561/2006 on driving hours and Directive 2002/15/EC on working hours, over a period of more than one week. However, it is missing the night working constraint from the directive on working hours. Numerical results show that considering the complete set of rules has a significant impact on the VRPTW solutions, in this sense, in practice they must be incorporated in solution methods for the VRPTW.

Prescott-Gagnon et al. [33] paper develops a Large Neighborhood Search algorithm (LNS) for a VRPTW under EC legislation. Neighborhoods are explored with a column generation heuristic, which uses a Tabu Search (TS) in order to generate the columns (routes). In the TS procedure, two types of moves are allowed insertion and deletion of a customer from a given route. A Label Setting algorithm is used in order to check feasibility after each insertion movement. In other words, they use a hybrid heuristic for solving the integrated problem of building vehicle routes that respect time windows and all driver rules from the EC social legislation within a period of less than one week. It is noteworthy that the night working rule is also not considered in this publication. Finally, they present a comparison against Goel [9] and Kok et al. [13] considering both, the basic set of rules and the extended set of rules; results show that their LNS algorithm outperforms the two previous approaches.

Again Kok et al. [25] work on the VRTDSP using a hybrid approach between an insertion heuristic and a linear model in order to solve it. The heuristic uses the Integer Linear Programming model to optimize the departures times of the sequence. Hence, the ILP evaluates a given sequence of customers, minimizing the total duty time, while considering time dependent travel times. Moreover, a continuous piecewise linear function is used in order to model the driving time between customers. In this trip evaluation, the (EC) No 561/2006 on driving hours is considered, in particular driving breaks (full and split), maximum driving time per day and maximum working time per day; at the end of the paper, one section shows how the linear model could be modified in order to include other extensions like, daily rest reductions up to nine hours three times per week, and the extension of daily driving time in one hour up to two times per week. Their experiments show that VRP routes can be used in practice, only if time dependant driving times and drivers' hours regulation are included into the formulations.

A multilevel variable neighborhood search heuristic is presented in [34] to solve an integrated multiperiod vehicle routing and scheduling problem. The model considers a planning horizon of one week, a heterogeneous fleet of vehicles, time windows on the service time at each customer, outsourcing options and working time regulations that apply to the internal drivers. With respect to the working time regulations, they consider breaks of 45 minutes after 4.5 hours of driving and a maximum working time during a week. The solution method uses three levels, in the first level demand points are aggregated, during the second level the weekly problem is decomposed into six daily problems, keeping as a coupling constraint the maximum working time per week; this two phases are developed to reduce the computational effort. Finally, in the third level the solution of the aggregated problem is scaled up to a solution for the original problem. The method is applied in the context of a fresh meat supply logistics system.

Derigs et al. [35] consider again an earlier work presented by Bartodziej et al. [30] in the context of the air cargo road feeder services, modifying different aspects of the previous resource allocation problem developed in [30]. In this version, the problem is formulated as a variant of a multiple-trip vehicle routing problem (VRPM), with two objectives, first to minimize the number of multi-trips and a second the minimization of the total cost. The trips are performed by different types of tractor/trailer-pairs, one or two drivers can be assigned to one trip, legal regulations are considered and there are on time requirements outlined by the airlines. They work under the EC social legislation; the planning horizon is up to two weeks albeit, they consider rules for one week. In order to check the feasibility of a given trip with respect to the drivers' regulation, they modify the procedure presented by Goel [9] and enlarge the set of rules considered within the procedure. In particular, they added the following rules: extension of a daily driving time, split of breaks and rest periods and daily rest reductions. To solve the problem, they propose two approaches based on a guided neighborhood search.

This report [36] shows the development of a Decision Support System (DSS) using a mathematical model as a core. The model is a Time Dependent Vehicle Routing and Scheduling Problem (TDVRSP), which minimizes the transportation costs and avoids driver schedule conflicts due to the Hours of Service (HOS) regulation. A step function with consecutive time intervals models the travel times, it considers the HOS and changes in the vehicle speed due to traffic congestions and road accidents. The model is solved by means of a simulated annealing, which is initialized with a greedy heuristic that returns the first tour, which visits all customers and then returns to the depot. Results show that for instances of 10 customers the computational time was under 7 minutes, and for larger size instance the average running time is slightly higher than quadratic in the number of customers.

In the context of long-haul transportation in the United States, Rancourt et al. [37] consider the Vehicle Routing Problem with Multiple Time Windows (VRPMTW) integrated with the scheduling process of rest periods for drivers according to the United States regulations. Among the characteristics of the problem, they consider a heterogeneous fleet of vehicles, the possibility of splitting a rest into two shorter periods spent in the sleeper berth and the scheduling process instead of finding a feasible solution, minimizes the duration of the trip. They developed three different scheduling algorithms which are embedded within a Tabu Search heuristic. Results show that taking a rest before depleting the allowable driving time brings better solutions; moreover, the option of split rests can improve the quality of the solutions.

Goel and Vidal [38] describe a hybrid genetic algorithm to solve the VRTDSP under different drivers' hour regulations, particularly in the United States, Canada, the European Union, and Australia. With respect to the EC social legislation they consider all rules except for the night working rule. They provide a comparison against the results of Prescott-Gagnon et al. [33] under two set of rules denoted *No Split* and *All*. Using a test Wilcoxon, they show that considering a p-value of 0.0001 the mean of the solutions are different, providing better results. However, their running times are 4.9 and 2.6 times greater than those achieved in [33]. In addition, other results indicate that the European Union rules bring the highest safety, while the most competitive in economic terms are the Canadian regulations.

A simulation-based methodology for regulatory impact assessment is proposed in [39]. The objective is to measure the impact of regulations on driving patterns under a horizon planning of one week. Thus, the author compares the Hours of Service regulation in the US of 2003 and its last change in 2013, as well as stricter constraints on driving periods per day, by reducing the maximum driving time per day to 10h and 9h. The model optimizes the vehicle routes with pickups and deliveries under hours of service regulations. In order to generate the set of solution routes, the Clarke and Wright heuristic is applied, modifying the algorithm in order to bring routes which comply with the HOS. This is done by checking the feasibility of the route with respect to the HOS at each iteration where the algorithm verifies if two routes can be merged. The experiments results show that increases in the operational costs are of the same order of magnitude as the monetized road safety benefits, also a positive total net benefit could be observed if the time limit is reduced to 10 or 9h.

The first exact algorithm to solve the VRTDSP is presented by Goel and Irnich [40], it is a branch and price algorithm adapted to solve a VRPTW which includes the hours of service regulations. Particularly, the auxiliary elementary shortest path problem with resource constraints (ESPRC) is solved by means of an heuristic label setting algorithm which relies on an auxiliary network that explicitly model all possible driver activities. Two regulations are considered the Hours of Service of the United States and the Social Legislation of the European Union. With respect to the rules from the Social legislation they do not consider rules related with the working time directive, daily driving time extensions and daily rest reductions. The algorithm finds optimal solutions for instances with 25 customers in the case of the HOS for U.S. and for 53 of 56 under the EU social legislation.

Koç et al. [41] introduce the Vehicle Routing and Truck Driver Scheduling Problem with Idling Options (VRTDSP-IO), which is an extension a previous work presented in [22] by embedding it in a routing problem. It is applied in the context of the Hours of service regulation in the U.S. The VRTDSP-IO is formulated as a set partitioning problem and it is solved by means of a matheuristic approach. Where the routing part is solved by an adaptive large neighborhood search (ALNS) and the scheduling problem is formulated as a linear program that is solved using CPLEX 12.6 optimizer. The multistart scheme implemented in the algorithm highly improves the quality of the solutions and the proposed algorithm can solve the VRTDSP-IO efficiently in a reasonable computational time.

Bowden and Ragsdale [42] extend the TDSP including fatigue monitoring by considering the Three Process Model Alertness (TPMA) within the formulation. The TPMA is a bio mathematical fatigue model, which in this paper it is composed by three primary processes: C the circadian influence on alertness, S the exponential decline in alertness with respect to the time awake, and U the afternoon dip in alertness. Their objective is to quantify and to understand the trade-off between routes length and alertness levels. Therefore, they propose a model that minimizes the route durations while considering the Hours of Service regulation in the U.S., time windows at customer locations and maintaining an acceptable level of alertness. While computing the TPMA according to the working and driving time scheduled in the route, the model becomes non-linear, and it is solved by using a metaheuristic algorithm.

Goel [43] works on the European Union social legislation, especially on two aspects that were not previously considering in VRTDSP problems: night working and the number of working days minimization. With regard to the night working, this paper only considers transport operations where drivers do not work during the night, in other words, the night working time is forbidden by covering night periods with rests. The solution algorithm is a modification of the method presented in [40], including the working time directive, daily driving time extensions and daily rest reductions. Experimental results show that many routes considering night work become feasible in practice and the running times of the algorithm to find a feasible solution are 4.5 times faster. Additionally, modifications on the objective function by the minimization of mileage and labor cost related to working days can reduce the total transportation costs by roughly 4%.

The previous work is extended by Tilk and Goel [44], by considering both, the European Union social legislation and the Hours of service from U.S., and from the algorithmic point of view improvements in the average computational time are achieved, trough the implementation of a bidirectional label setting algorithm instead of the former forward label setting version. The rules and assumptions considered with regard of the European Social legislation remains unaffected. In the case of the HOS an average computational time of 5 minutes is required to solve instances with 25 customers and for the EU regulation they can solve 152 out of 168 instances with 25 customers using an average computational time of about 17 minutes.

Table 1.2 presents a summary of the rules from the EU social legislation considered by different papers working on the VRTDSP. Goel [9] presents the first contribution working on the EU social legislation after its entry into force in 2004. All previous papers consider other specific regulations related to each particular country. With respect to the solution methods, using metaheuristic approaches the most complete methods in terms of the number of rules considered are presented in [13], [33] and [38], missing only the night working rule. On the other hand, using exact methods Goel [43] and Tilk and Goel [44], cope with all rules from the EU social legislation. Although, they made a simplification with respect to the night working rule by forbidding the night work. Therefore, there is not any metaheuristic or exact approach that solves the VRTDSP considering all the EU social legislation.

Table 1.2. Papers related to the VRTDSP.

Papers	Regulation (EC) No 561/2006 on driving hours								Directive 2002/15/EC on working hours			
	Driving periods (full & split breaks)	*Daily driving time*	*Daily driving extensions*	*Daily rest period*	*Daily rest reductions*	*Split daily rest*	*Weekly driving times*	*Weekly rest periods*	*Working breaks*	*Weekly working time*	*Night work*	***Different regulation***
Brandao, J. and Mercer, A. (1997) [24]												x
Savelsbergh, M. and Sol, M. (1998) [27]												x
Xu, H., Chen, Z., Rajagopal, S. and Arunapuram, S. (2003) [28]												x
Campbell, A. and Savelsbergh, M. (2004) [29]												x
Zapfel, G and Bogl, M. (2008) [30]												x
Bartodziej, P., Derigs, U., Malcherek, D. and Vogel, U. (2009) [31]												x
Ceselli, A., Righini, G. and Salani, M. (2009) [32]												x
Goel, A. (2009) [9]	x	x		x								
Kok, A.L., Meyer, C.M., Kopfer, H. and J. Schutten. (2010) [13]	x	x	x	x	x	x	x	x	x	x		
Prescott-Gagnon, Eric., Desaulniers, G., Drexl, M. and Rousseau, LM. (2010) [33]	x	x	x	x	x	x	x	x	x	x		
Kok, A.L., Hans, E.W., and J.M.J., Schutten. (2011) [26]	x	x		x								
Wen, M., Krapper, E., Larsen, J. and Stidsen, T. (2011) [34]												
Derigs, U., Kurowsky, R. and Vogel, U. (2011) [35]	x	x	x	x	x	x						
Min, H. (2011) [36]												x
Rancourt, M., Cordeau, J. and Laporte, G. (2013) [37]												x
Goel, A. and Vidal, T. (2014) [38]	x	x	x	x	x	x	x	x	x	x		
Goel, A. (2014) [39]												x
Goel, A. and Irnich, S. (2016) [40]	x	x		x								
Koç, Ç., Jabali, O. and Laporte, G. (2017) [41]												x
Bowden, Z. and Ragsdale, C. (2018) [42]												x
Goel, A. (2018) [43]	x	x	x	x	x	x	x	x	x	x	x	
Tilk, C. and Goel, A. (2020) [44]	x	x	x	x	x	x	x	x	x	x	x	

1.3 Scope of the thesis

This Thesis works on two interrelated combinatorial problems the Truck Drivers Scheduling Problem and the Vehicle Routing and Truck Drivers Scheduling Problem. First, we present the general assumptions considered in the models and later, the objectives of the thesis.

As in all papers working on both types of problems, instance data is considered deterministic. All the relevant information is known in advance i.e. time duration of driving or service/other work activities and time windows. In this sense, data instance does not change over the time, thus this thesis focus on offline-problems. Furthermore, we do not consider time-dependent parameters, such as the driving time between customers. Finally, similarly as the vast majority of the previous contributions, single time windows at each customer are considered.

With regard to the places where a break/rests could be scheduled, two options are considered. We present at chapter 2 models related with the assumption that breaks/rests are schedule only after an activity is finished i.e. after driving or after the service at the customer location. This assumption highly diminishes the number of options to explore, thus the complexity of the problem. Later, this assumption is modified and breaks/rests can be scheduled at any place or point in time, even in the middle of a service activity. Models working under this assumption are referred as preemptive versions. This change is required in order to make comparisons with the most complete models, in terms of the number of rules that they handle, since they work on this assumption.

This thesis only considers the provision provided by the EU social legislation. The models presented in it, can be modified to fit other particular regulations that in several cases are less complex than the EU social legislation. Moreover, our models deal with only one driver and there are not interdependencies between the routes of different truck drivers. It is noteworthy to say that in the multi-manned case (more than one driver is assigned to one vehicle) different rules from the EU social legislation may apply.

In the case of the TDSP most of the solution methods considering the EU social legislation neglect different provisions from the regulation; even the most complete contribution [14] does not consider the Directive 2002/15/EC on working hours and the daily rest reduction. Hence, a mathematical model that cope with the complete EU social legislation, not only brings a better understanding of the set of rules but also it can be used to develop tailored and faster solution algorithms.

On the other hand, with respect to the TDSP integrated into the VRTDSP, the most recent papers from Goel [43] and Tilk and Goel [44] simplify both the rule on working breaks during a shift and the night working rule. In particular, the night working rule entails different challenges depending on the solution method; in example, for label setting algorithms, the first challenge is to update the label resources in a sliding 24h time window and the second is to design efficient dominance rules to achieve optimality in a competitive computational time.

The main objective of the thesis is to define new models to evaluate a trip under the EU social legislation, using exact methods, including linear formulation and dynamic approaches, and a sub-optimal resolution scheme that could be included into a global search procedure to solve the integrated VRTDSP.

1.4 Conclusion

Fatigue is an important factor in Heavy Good Vehicle road accidents. As a result, the entry into force of the EU social legislation is intended to increase road safety, improve working conditions and coordinate competition conditions between road transport companies. In order to present the EU social legislation, some term definitions are given. Basically the EU social legislation is composed by the Regulation (EC) No 561/2006 on driving hours and Directive 2002/15/EC on working hours, both of them are presented. In addition, the chapter develops a review on recent contributions considering different driver's regulations.

The literature review surveys independently the scheduling of breaks and rests problem, the Truck Drivers Scheduling Problem and the integrated routing and scheduling problem, the Vehicle Routing and Truck Drivers Scheduling Problem. With respect to the TDSP, there are no previous publications that neither explicitly consider the night working rule nor the working shift breaks. In the case of the VRTDSP, there are not metaheuristic methods that solve the problem considering all rules from the EU social legislation; moreover, with regard to exact methods last contributions address all the rules, although they simplify the night working rule.

1.5 References

[1] Adminaité-Fodor, D. and Jost, G., "How to improve the safety of goods vehicles in the EU", European Transport Safety Council, Brussels, PIN Flash Report 39, 2020. [Online]. Available: https://etsc.eu/wp-content/uploads/PIN-FLASH39_FINAL.pdf

[2] Vitols, K, and Eckhard, V., "Driver fatigue in European road transport", European Transport Workers's Federation, Brussels, 2021. [Online]. Available: https://www.etf-europe.org/wp-content/uploads/2021/05/Driver-Fatigue-in-European-Road-Transport-Report.pdf

[3] European Road Safety Observatory, "European Commission (2020) Facts and Figures Pedestrians", Brussels, European Commission, Directorate General for Transport, 2020. [Online]. Available: https://ec.europa.eu/transport/road_safety/sites/default/files/facts_figures_buses_and_hgv_final_20210323.pdf

[4] Filtness, A., Anund, A., Maynard, S., Miller, k., Pilkington-Cheney, F., Dahlman, A. and Ihlstrom, J., "Bus driver fatigue", Transport Safety Research Group, Loughborough University, Loughborough, Final Report, 2019. [Online]. Available: http://content.tfl.gov.uk/bus-driver-fatigue-report.pdf

[5] Knipling, R., "Review of commercial driver fatigue research methodologies", Commissioned paper, National Research Council (NRC), Committee on National Statistics (CNSTAT), 2015. [Online]. Available:

https://sites.nationalacademies.org/cs/groups/dbassesite/documents/webpage/dbasse_171093.pdf

[6] European Commission, "Fatigue", Directorate General for Transport, 2018. [Online]. Available: https://ec.europa.eu/transport/road_safety/sites/default/files/pdf/ersosynthesis2018-fatigue.pdf

[7] European Transport Safety Council, "The role of driver fatigue in commercial road transport crashes", Directorate of Energy and Transport of the European Commission, Brussels, 2001. [Online]. Available: https://etsc.eu/wp-content/uploads/The-role-of-driver-fatigue-in-commercial-road-transport-crashes.pdf

[8] European Union. 2006. Regulation (EC) No. 561/2006 of the European Parliament and of the Council of 15 March 2006 on the harmonisation of certain social legislation relating to road transport and amending Council Regulations (EEC) 3821/85 and (EC) 2135/98 and repealing Council Regulation (EEC) 3820/85. Official Journal of the European Union L 102, 11.04.2006.

[9] Goel, A., "Vehicle scheduling and routing with drivers' working hours", *Transp. Sci.,* vol. 43, no. 1, pp. 17–26, 2009.

[10] European Union, Directive 2002/15/EC of the European Parliament and of the Council of 11 March 2002 on the organisation of the working time of persons performing mobile and road transport activities, 2002, [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32002L0015&from=EN

[11] Archetti, C. and Savelsbergh, M.W.P., "The trip scheduling problem", *Transp. Sci.*, vol. 43, pp. 417–431, 2009.

[12] Goel, A., "Truck driver scheduling in the European Union", *Transp. Sci.*, vol. 44, pp. 429–441, 2010.

[13] Kok, A. L., C. M. Meyer, H. Kopfer and J. M. J. Schutten., "A dynamic programming heuristic for the vehicle routing problem with time windows and European Community social legislation", *Transportation Sci*. vol. 44, no. 4, pp. 442–454, 2010.

[14] Drexl, M. and Prescott-Gagnon, E., "Labelling algorithms for the elementary shortest path problem with resource constraints considering EU drivers' rules", *Logist. Res.,* Vol. 2, no. 2, pp. 79-96, 2010.

[15] Goel, A. and Kok, A.L., "Truck driver scheduling in the United States", *Transp. Sci.*, vol. 46, pp. 317–326, 2012.

[16] Goel, A. and Kok, A.L., "Efficient scheduling of team truck drivers in the European Union", *Flex. Serv. Manuf. J.*, vol. 24, no. 1, pp. 96-81, 2012.

[17] Goel, A., Archetti, C., and Savelsbergh, M.W.P., "Truck driver scheduling in Australia", *Comput. Oper. Res.*, vol. 39, pp. 1122–1132, 2012.

[18] Goel, A., "The minimum duration truck driver scheduling problem", *EURO J. Transp. Logist.*, vol. 1, no. 4, pp. 285–306, 2012.

[19] Goel, A. and Rousseau L-M., "Truck driver scheduling in Canada", *J. Sched.,* vol.15, pp. 783–799, 2012.

[20] Goel, A., "The Canadian minimum duration truck driver scheduling problem", *Comput. Oper. Res.*, vol. 39, no. 10, pp. 2359–2367, 2012.

[21] Goel, A., "A mixed integer programming formulation and effective cuts for minimising schedule durations of Australian truck drivers", *J. Sched.*, vol. 15, no. 6, pp. 733–741, 2012.

[22] Koç, C., Bektas, T., Jabali, O. and Laporte, G., "A comparison of three idling options in long-haul truck scheduling", *Transport. Res. B-Meth.,* vol. 93, pp. 631-647, 2016.

[23] Sartori, C., Smet, P. and Vanden, G., "Scheduling truck drivers with interdependent routes under European Union regulations", *Eur. J. Oper. Res.,* vol. 298, no. 1, pp. 76-88, 2022.

[24] Brandao, J. and Mercer, A., "A tabu search algorithm for the multi-trip vehicle routing and scheduling problem", *Eur. J. Oper. Res.,* vol. 100, no. 1, pp. 180-191, 1997.

[25] Rochat, Y. and Semet, F., "A tabu search approach for delivering pet food and flour in Switzerland", *J. Oper. Res. Soc,* vol. 45, no. 11, pp. 1233-1246, 1994.

[26] Kok, A.L., Hans, E.W. and Schutten, J.M.J, "Optimizing departure times in vehicle routes", *Eur. J. Oper. Res.,* vol. 210, no. 3, pp. 579-587, 2011.

[27] Savelsbergh, M. and Sol, M., "Dynamic routing of independent vehicles", *Oper. Res.,* vol. 46, no. 4, pp. 474-490, 1998.

[28] Xu, H., Chen, Z., Rajagopal, S. and Arunapuram, S., "Solving a Practical Pickup and Delivery Problem", *Transp. Sci.*, vol. 37, no. 3, pp. 347–364, 2003.

[29] Campbell, A. and Savelsbergh, M., "Efficient insertion heuristics for Vehicle Routing and Scheduling problems", *Transp. Sci.*, vol. 38, no. 3, pp. 369–378, 2004.

[30] Zapfel, G. and Bogl, M., "Multi-period vehicle routing and crew scheduling with outsourcing options", *Int. J. Prod. Econ.*, vol. 113, no. 2, pp. 980–996, 2008.

[31] Bartodziej, P., Derigs, U., Malcherek, D. and Vogel, U., "Models and algorithms for solving combined vehicle and crew scheduling problems with rest constraints: an application to road feeder service planning in air cargo transportation", *OR Spectrum*, vol. 31, pp. 405–429, 2009.

[32] Ceselli, A., Righini, G. and Salani, M., "A column generation algorithm for a rich vehicle routing problem", *Transp. Sci.*, vol. 43, no. 1, pp. 56–69, 2009.

[33] Prescott-Gagnon, E., Desaulniers, G., Drexl, M. and Rousseau, L.M., "European Driver Rules in Vehicle Routing with Time Windows", *Transp. Sci.*, vol. 44, no. 4, pp. 455–473, 2010.

[34] Wen, M., Krapper, E., Larsen, J. and Stidsen, T., "A multilevel variable neighborhood search heuristic for a practical vehicle routing and driver scheduling problem", *Networks*, vol. 58, no. 4, pp. 311–322, 2011.

[35] Derigs, U., Kurowsky, R. and Vogel, U, "Solving a real-world vehicle routing problem with multiple use of tractors and trailers and EU-regulations for drivers arising in air cargo road feeder services", *Eur. J. Oper. Res.,* vol. 213, no. 1, pp. 309-319, 2011.

[36] Min, H., “Building a model-based decision support system for solving vehicle routing and driver scheduling problems under hours of service regulations”, University Transportation Centers program, Ohio, 2011. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/23718

[37] Rancourt, M., Cordeau, J. and Laporte, G., “Long-haul vehicle routing and scheduling with working hours rules”, *Transp. Sci.*, vol. 47, no. 1, pp. 81–107, 2013.

[38] Goel, A. and Vidal, T., “Hours of Service Regulations in Road Freight Transport: An Optimization-Based International Assessment”, *Transp. Sci.*, vol. 48, no. 3, pp. 391–412, 2013.

[39] Goel, A., “Hours of service regulations in the United States and the 2013 rule change”, *Transp. Policy.*, vol. 33, pp. 48–55, 2014.

[40] Goel, A. and Irnich, S., “An exact method for vehicle routing and truck driver scheduling problems”, *Transp. Sci.*, vol. 51, no. 2, pp.737–754, 2016.

[41] Koç, Ç., Jabali, O. and Laporte, G., “Long-haul vehicle routing and scheduling with idling options”, *J. Oper. Res. Soc,* vol. 69, no. 2, pp. 235-246, 2017.

[42] Bowden, Z. and Ragsdale, C., “The truck driver scheduling problem with fatigue monitoring”, *Decis. Support Syst.,* vol. 110, pp. 114-162, 2018.

[43] Goel, A., “Legal aspects in road transport optimization in Europe”, *Transport. Res. E-Log*, vol.114, pp. 144–162, 2018.

[44] Tilk, C. and Goel, A. “Bidirectional labeling for solving vehicle routing and truck driver scheduling problems”, *Eur. J. Oper. Res.*, vol. 283, pp. 108–124, 2020.

CHAPTER 2

TDSP under European driving rules without pre-emption

This chapter presents a novel Mixed Integer Linear Formulation and a Label Setting algorithm for the Truck Driver Scheduling problem considering the European driver regulations. Both models take into consideration different constraints enforced by the EU driver's rules, such as: driving breaks with split flexibility, working breaks, maximum driving time per day, extensions on driving times per day, regular daily rest and daily rest reductions. In contrast, with previous linear formulations, breaks due to working time, driving time extensions and daily rest reductions are included. One strong assumption is that breaks are only possible after an activity finishes. Therefore, to include a break in the middle of an activity it is necessary to add one *dummy* customer with zero service time. In this regard, these extra customers are like rest areas, which in many practical applications, in particular when motorways are used, are necessary since rest periods cannot be taken anywhere. A new set of instances are proposed in order to test the performance of the models and detail optimal solutions are provided.

2.1 Truck driver scheduling problem

Different approaches have been used to include breaks and rest periods within the vehicle scheduling, considering different regulation rules. Some of them are interested in verifying the feasibility of a given schedule. Archetti and Savelsbergh [1] proposed and algorithm with a complexity of $O(n^3)$ to check feasibility of a driver schedule. The sequence of pick-up and delivery request is done considering time windows at the pick-up locations and the regulation rules from United States, the United States Hours of Service. Later, Goel and Kok [2] under the same regulation rules developed a method, which guarantees to find feasible driver schedules if they exist. This method solves the problem in $O(n^2)$ in the case of single time windows. When multiple time windows have to be addressed, the same complexity could be obtained if the gap between subsequent time windows assigned to the same location is at least 10 hours, i.e. handling operations are only allowed between 8.00 A.M and 10.00 P.M.

Goel [3] presents a breadth-first search (BFS) algorithm to find feasible truck driver schedules if they exist under the working hours regulation for the EU. The experiments use sequences of customers varying from 3 to 12, each handling activity requires one hour of working and driving time between two subsequent locations is 4, 8, 12 or 16. From 15.6 million instances generated only 662,308 do not exceed the accumulated weekly driving or working time. The method is compared against algorithms developed in [4] and [5], the results show that previous methods fail to find feasible solutions half of the time when the size of the instances is large. Another result is that taking breaks in two parts does not bring a big difference for long distance hauling. Considering the set of regulation rules in the Australian standard rules and Basic fatigue management, Goel et al [6] presented an exact algorithm to solve the truck driver-scheduling problem. In addition, they presented four heuristics, based on removing some of the most computationally expensive steps of the exact method. As a result, the computational time diminishes, albeit, they did not guaranteed feasibility in all cases.

Among linear model formulations of the problem, Goel [7] developed a linear model complying with the Australian Heavy Vehicle Fatigue Law, in which the objective function is to minimize the total duration of the schedule. Drivers may only take rest periods before the service at a given location, albeit, they include parking lots or dummy locations with zero working time, to take rest periods after completing the service. In order to speed up the solution process, they strengthen the formulation by adding valid inequalities (cuts). The model is solved using a commercial solver, and the results are compared against those obtained by two dynamic programming models and a hybrid method.

Taking under consideration the Canadian Hours of Service, Goel et al [8] find the set of feasible schedules for the Canadian Truck Driver Scheduling Problem (CAN-TDSP) using two heuristics and one enumerative method, the three of them based on a dynamic programming approach. The two heuristic approaches present a good performance in terms of computational time, while the enumerative method is not competitive under these criteria. Later, Goel [9] presents a mixed integer programming formulation for the Canadian minimum duration truck driver-scheduling problem (CAN-MDTDSP). The model is solved using a commercial solver CPLEX 12 and its results are compare with a heuristic method which iteratively calls the dynamic programming algorithm presented in Goel et al [8]. Even though, CPLEX 12 offers better solutions while solving the mixed integer program, the iterative dynamic programming approach requires significantly smaller computational time.

The minimum duration truck driver-scheduling problem (MD-TDSP) is a generic model adapted to consider both, the EU and the US Hours of Service regulations in Goel [10]. The objective is to find feasible schedules with minimum duration, assuming that breaks and rest periods are taken only at customer locations or at suitable rest areas. The main decision variables are those regarding the type and the duration of rest periods at the different locations. Concerning to optional rules like taking rests in two parts, included in the EU regulation, they are not considered in the linear formulation, albeit, they are included in the dynamic programming model. The MD-TDSP is solved using a commercial solver CPLEX 12 and a dynamic programming approach. An interesting result is that including the flexibility of taking breaks and rest in two parts, improves the quality of the solutions and increase the number of feasible solutions. Finally, as expected the dynamic programming algorithm solves the problem in a small fraction of time than its counterpart the commercial solver.

2.2 Mixed Integer Linear Problem

A MILP formulation is proposed to schedule driver's shifts under the European Union regulations. To evaluate a sequence σ of customers that starts and ends at the depot, and that is fully defined by the following information at each customer i: the arrival time (A_i), the starting time of the service (St_i), the finishing time (Ft_i), the departure time (D_i), the break before the service ($B0_i$) and the break after the service ($B1_i$). The working time includes driving periods and service activities at the customer location or the depot (loading/unloading, cleaning the vehicle, etc). Other periods are called periods of availability (POA). Figure 2-1 presents a sequence evaluation based on the foregoing variables. A short respite is called break and a long period (more than 9h) rest. The MILP investigates the possibility to schedule breaks and rests only after an activity finishes. Break and rest periods durations are not restricted to a limited set of values. In addition, without loss of generality, the schedule starts and ends with a rest at the depot.

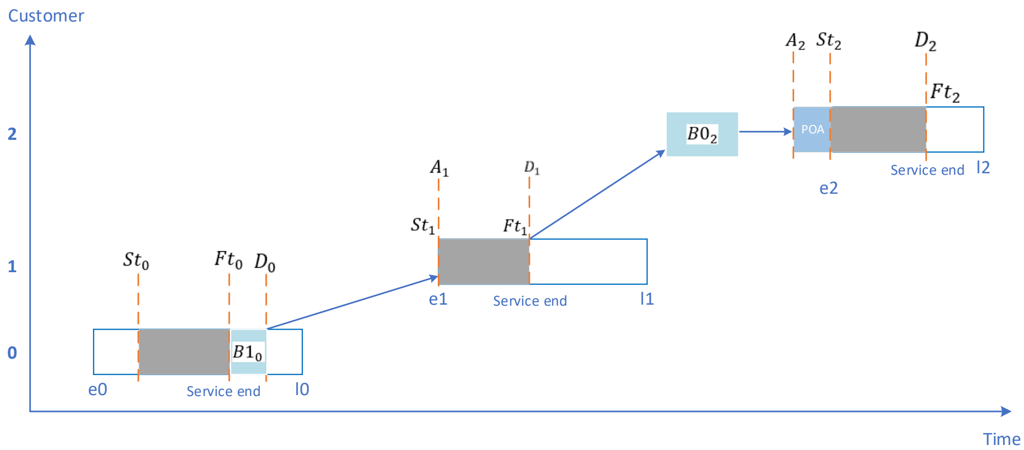


Figure 2-1. Sequence evaluation.

The following set of constraints from the EU regulation are considered:

R1. A driver cannot drive more than 4.5h without a break of more than 45min referred to as "full break".

R2. A full break of 45min can be split into two periods, a first break of at least 15min (and less than 45 min) and a second break of at least of 30min; the breaks should be taken in this order.

R3. A working period of 6h must encompasses a break of at least 15min.

R4. If the total working time of a shift is between 6h and 9h, a break of 30 min has to be scheduled.

R5. If the total working time during a shift is greater than 9h, a break of 45 min is required.

R6. The maximal driving time in a shift is 9h that could be extended to 10h twice per week.

R7. A shift cannot length more than 24 hours.

R8. The minimal rest period is 11h that could be reduced to 9h three times per week. Therefore, the sum of breaks, POA and working durations is bounded by 13h; 15h if the rest is reduced to 9h.

2.2.1 Data and variables

The initial set of data gives the sequence of locations to visit with the driving times and the working times at each location. Time window constraints are imposed on the starting time of service activities at customer locations. All sets, parameters and variables are given below.

Sets

A: Driving and service activities. The activities to schedule are numbered from 1 ... $|U|$, each depot and customer have two activities: service and driving (in this order).

$B = \{15min, 30min, 45min, 9h, 11h\}$: Types of break.

Parameters:

M: A big number.

cp_a: Duration of activity a.

δ_a: 1, if node a is driving; 0, otherwise.

e_a: Earliest starting time of the activity a.

l_a: Latest starting time of the activity a.

BD_b: Minimal break durations. $BD_b = \{0.25, 0.50, 0.75, 9, 11\}$

$wo_{a,c}$: Sum of processing time between activities a and c.

$$wo_{a,c} = \sum_{k=a}^{c} cp_u$$

$d_{a,c}$: Sum of driving time between activities a and c :

$$d_{a,c} = \sum_{k=\mathrm{a}|\delta_k=1}^{c} cp_u$$

CDT: Continuous driving time without a break [4.5h].

CWT: Continuous working time without a break [6h].

$WL9$: Minimal working time in a shift without at least $WSB1$ minutes of break [6h].

$WG9$: Minimal working time in a shift without at least $WSB2$ minutes of break [9h].

MDE: Extended driving time per shift [10h].

MDR: Regular driving time per shift [9h].

WBC: Minimal break duration after CWT hours of continuous working time [0.25h].

$WSB1$: Minimal break duration when working time during a shift is between 6h and 9h [0.5h].

$WSB2$: Minimal break duration when working time during a shift is greater than 9h [0.75h].

DDU: The maximal duration without a rest [24h].

DE: Maximum number of driving extensions per week [2].

RE: Maximum number of rest reductions per week [3].

Decision variables:

x_a: Starting time of activity a.

D_a: Duration of a break at activity a.

y_a^b: 1, if at activity a a break of type b takes place after the process. 0, otherwise.

F_u: 1, if at node u a full break takes place. 0, otherwise.

E_u: 1, if the driving time is extended during one hour. 0, otherwise.

Figure 2-2 gives an example of a sequence to evaluate which has two customers and includes the depot at both, first and last position. The set of activities A corresponds to the service and transport activities developed at each location. In this case, eight activities in total ($|A| = 8$), where the last activity, the transport activity at the last depot always has a duration of 0 hours $\left(cp_{|A|} = 0\right)$.

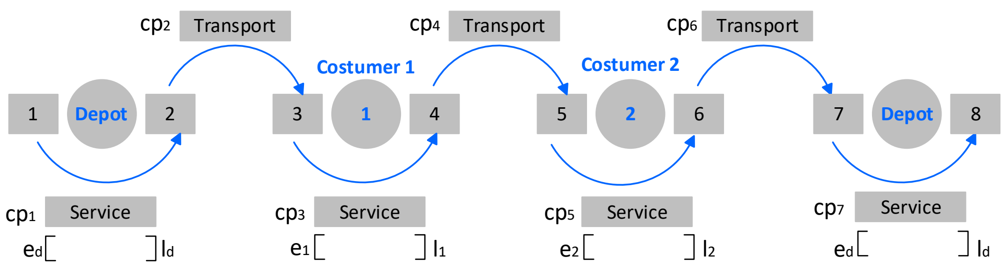


Figure 2-2. Example of a sequence with two customers.

2.2.2 Objective function and constraints

The objective function (1) is to minimize the completion time of the processing time of the last activity.

$$Min\ z = x_{|A|} + cp_{|A|} \tag{1}$$

It could be changed to minimize the makespan by subtracting the starting time of the first activity of service at the depot.

$$Min\ z = x_{|A|} + cp_{|A|} - x_0 \tag{1.a}$$

Constraint (2) enforces that the sequence finishes with a rest. We choose to insert the break at the node $|A| - 1$, just after the last activity with a $cp_a \geq 0$.

$$\sum_{b=5}^{|B|} y_{|A|-1}^{b} = 1 \qquad (2)$$

As Figure 2.3 shows, constraints (3) enforces that the starting time of activity $a+1$ can only start after the finishing time of activity a. i.e. after the starting time of a (x_a) plus the duration cp_a and plus the duration of break ($\sum_{b=1}^{|B|} BD_b \times y_a^b$) that are scheduled at customer a.

$$x_a + cp_a + \sum_{b=1}^{|B|} BD_b \times y_a^b \le x_{a+1} \qquad \forall\, a = 1,..,|A|-1 \qquad (3)$$

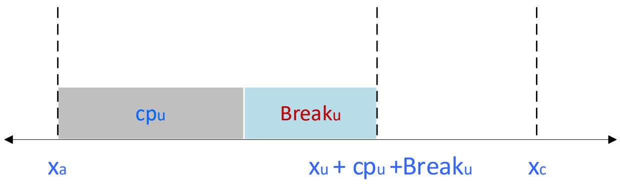


Figure 2-3. Sequence between successive activities.

Constraints (4) enforces that only one break is scheduled after an activity a finishes.

$$\sum_{b=1}^{|B|} y_a^b \le 1 \qquad \forall\, a = 1,..,|A| \qquad (4)$$

Constraints (5.1)-(5.3) compute the variable duration D_a. This variable is useful for compulsory cumulated break durations i.e. 45’ when working time during a shift exceeds 9h, since, this break could be satisfied for example with two breaks of 20 and 25 minutes. Constraints (5.1) and (5.2), define the upper and lower bounds of the variable D_a, once a break of type b takes place at activity a. Note that D_a is set to 0 if there is no break after the activity a.

$$\sum_{b=1}^{|B|} BD_b \times y_a^b \le D_a \qquad \forall\, a = 1,..,|A| \qquad (5.1)$$

$$D_a < \sum_{b=1}^{|B|-1} BD_{b+1} \times y_a^b \qquad \forall\, a = 1,..,|A| \qquad (5.2)$$

Constraints (5.3) guarantee the sequence between activities allowing a flexible break duration D_a in between, while, constraints (3) enforces to use the minimal break duration of the break taken at a.

$$x_a + cp_a + D_a \leq x_{a+1} \qquad \forall\, a = 1,..,|A| - 1 \qquad (5.3)$$

Constraints (6) guarantee a full break if between two driving activities there are more than CDT hours. Figure 2-4 depicts this situation.

$$\sum_{w=a}^{c-1} F_w \geq 1 \qquad \begin{matrix} \forall\, a = 1,..,|A| \\ \forall\, c = a+1,..,|A| \\ \delta_a = \delta_c = 1 \\ d_{ac-1} \leq CDT \;\wedge\; d_{ac-1} > CDT \end{matrix} \qquad (6)$$

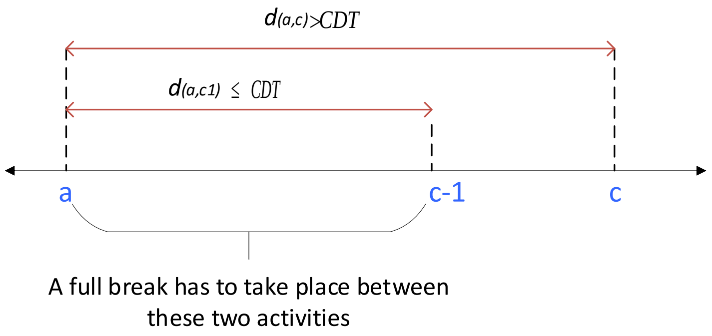


Figure 2-4. A full break takes place after CDT hours of driving.

Constraints (7.1)-(7.3) establish full break conditions. Constraints (7.1) enforces a full break if a break greater or equal to 45 minutes ($b \geq 3$) takes place after an activity a. Thus, any break duration greater than 45 minutes is a full break leading to $F_a = 1$. In example, constraints (7.1) enforces that $F_a = 1$ if $y_a^3 = 1 (y_a^3 \leq F_a)$, enforces that $F_a = 1$ if $y_a^4 = 1 (y_a^4 \leq F_a)$ and that $F_a = 1$ if $y_a^5 = 1 (y_a^5 \leq F_a)$. Constraints (7.2) bring the possibility of a full break ($F_a = 1$), if a break greater or equal than 30min takes place at activity a. Constraints (7.3) enforces that a break of type 1 (b_1) is not a full break.

$$\sum_{b=3}^{|B|} y_a^b \leq F_a \qquad \forall\, a = 1,..,|A| \qquad (7.1)$$

$$\sum_{b=2}^{|B|} y_a^b \geq F_a \qquad \forall\, a = 1,..,|A| \qquad (7.2)$$

$$1 - y_a^1 \geq F_a \qquad \forall\, a = 1,..,|A| \qquad (7.3)$$

The flexibility of split a full break in two periods, a first break of at least 15min (and less than 45 min) and a second break of at least of 30min is modeled by Constraints (8). There is not a full break at c if a break of type 2 is scheduled at c, and if there is not break less than 45 min between $[a, c[$ (as the first part of the split) and a full break is scheduled just before a. Note that $F_0 = 1$ by definition, since, the schedule starts and ends with a rest at the depot. Therefore, the third member of the right hand of Constraints (8) is equal to 0 when the schedule starts.

$$(1 - y_c^2) + \sum_{w=e}^{c-1} \sum_{b=1}^{2} y_w^b + (1 - F_{e-1}) \le F_c \qquad \begin{array}{c} \forall\ c = 1, \dots, |A| \\ \forall\ a = 1, \dots, c-1 \\ \forall\ e = a, \dots, c \\ d_{a+1,c} \le CDT\ \wedge\ d_{a,c} > CDT \\ \vee\ d_{0,c} \le CDT \end{array} \quad (8)$$

These inequalities forbid breaks of 30 min to be considered as a full break if previously the first part of a split break has not been scheduled, see Figure 2-5.

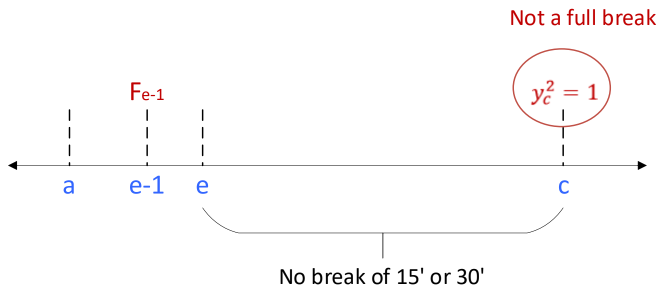


Figure 2-5. Split break conditions.

Constraints (9) guarantee a break of at least WBC hours if between two activities there are more than CWT hours of working time.

$$\sum_{b=1}^{|B|} \sum_{w=a}^{c} BD_b \times y_w^b \ge WBC \qquad \begin{array}{c} \forall\ a = 1, \dots, |A| \\ \forall\ c = a+1, \dots, |A| \\ wo_{a,c-1} \le CWT\ \wedge\ wo_{a,c} > CWT \end{array} \quad (9)$$

If the working time during a shift is greater than $WL9$ (=6h) but less or equal to $WG9$ (=9h), then a break of at least $WSB1$(=30min) hours must be scheduled. Constraints (10) model this rule.

$$\sum_{w=a}^{c-1} \sum_{b=1}^{|B|} BD_b \times y_w^b \ge WSB1 - M\left(2 - \sum_{b=4}^{|B|} y_{a-1}^b - \sum_{b=4}^{|B|} y_c^b\right)$$

$$\begin{array}{c} \forall\ a = 1, \dots, |A| \\ \forall\ c = a+1, \dots, |A| \\ wo_{a,c-1} > WL9\ \wedge\ wo_{a,c} \le WG9 \end{array} \quad (10)$$

The left hand of the inequality computes the total break time between activities a and $c-1$, when the working time between them is greater than $WL9$ and less or equal to $WG9$. After, this total break must be greater or equal to $WSB1$ if a shift starts at activity a, that is, a rest that took place during the precedent activity ($\sum_{b=4}^{|B|} y_{a-1}^b = 1$), and a shift ends at activity c ($\sum_{b=4}^{|B|} y_{a-1}^{b} = 1$). Figure 2-6 shows the foregoing situation, where the sum of the breaks between two rest taken at $a-1$ and c, must be greater than 30 minutes.

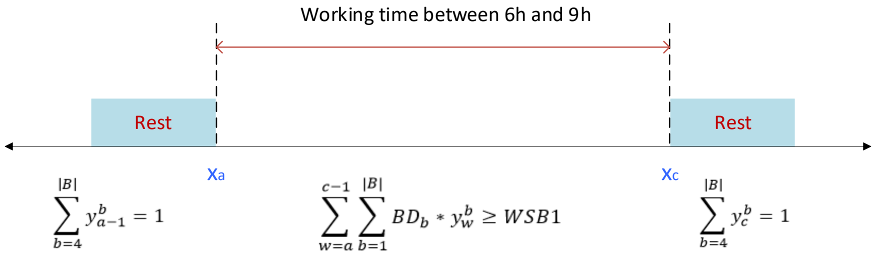


Figure 2-6. Total break greater to WSB1 if shift working time between 6h and 9h.

Constraints (11) ensure a break of at least $WSB2$ hours (i.e. 45min) if the working time during a shift is greater than $WG9$ (=9h). Constraints (11) and (10) behave alike. Although, constraints (11) introduce the variables D_w, which are break durations that are not subject to take the minimal value of a break, see constraints 5.1 and 5.2. The left hand of the inequality computes the total break duration between activities a and $c-1$, when the working time between them is greater than $WG9$. After, this total break duration ($\sum_{w=a}^{c-1} D_w$) must be greater or equal to $WSB2$ if a shift starts at activity a, that is, a rest that took place during the precedent activity ($\sum_{b=4}^{|B|} y_{a-1}^{b} = 1$), and a shift ends at activity c ($\sum_{b=4}^{|B|} y_{a-1}^{b} = 1$). Moreover, terms$D_w$ and $BD_b \times y_w^b$ are interchangeable, albeit, in (11) it is more likely to find solutions where different combinations of breaks durations fulfill the requirement of $WSB2$ hours of break.

$$\sum_{w=a}^{c-1} D_w \geq WSB2 - M\left(2 - \sum_{b=4}^{|B|} y_{a-1}^{b} - \sum_{b=4}^{|B|} y_c^{b}\right)$$

$$\forall\ a = 1, .., |A|$$
$$\forall\ c = a+1, .., |A| \quad (11)$$
$$wo_{a,c} \geq WG9$$

As in Constraints (8), note that $F_0 = 1$ by definition, since, the schedule starts and ends with a rest at the depot. Hence, the second member of the right hand of Constraints (10) and (11), becomes $M\left(1 - \sum_{b=4}^{|B|} y_c^{b}\right)$.

Constraints (12) and (13) enforces a rest if the driving time between two driving activities is greater than MDE (=10h) or MDR (=9h), respectively. That is to say, constraints (12) enforce that between two nodes a and $c-1$ where the distance exceeds MDE hours of driving, there is at least one rest for the night. Similarly, after the two possible weekly driving extensions DE(=2) are exhausted, Constraints (13) limit the daily driving time to MDR hours by one rest for the night.

$$\sum_{b=4}^{|B|} \sum_{w=a}^{c-1} y_w^b \geq 1 \qquad \forall\ a = 1, .., |A| \quad \forall\ c = a+1, .., |A| \quad (12)$$

$$\delta_a = \delta_c = 1$$
$$d_{a+1,c} \le MDE \wedge d_{a,c} > MDE$$

$$\sum_{b=4}^{|B|} \sum_{w=a}^{c-1} y_w^b + E_a \ge 1 \qquad \begin{gathered} \forall\, a = 1,..,|A| \\ \forall\, c = a+1,..,|A| \\ \delta_a = \delta_c = 1 \\ d_{a+1,c} \le MDR \wedge d_{a,c} > MDR \end{gathered} \tag{13}$$

Constraints (14) allow only DE driving extension per week.

$$\sum_{a=1}^{|A|} E_a \le DE \tag{14}$$

Every period of DDU(=24h) hours should have a rest period, Constraints (15) ensure this condition. The period of time between the start of activity a and the finish of activity c, including the rest at this position, should be less than DDU hours. This applies if there is not rest between this two activities $\sum_{b=4}^{|B|} \sum_{w=a}^{c-1} y_w^b = 0$. Figure 2-7 depicts this rule.

$$x_c + cp_c + \sum_{b=4}^{|B|} (BD_b \times y_c^b) - x_a \le DDU - M\left(\sum_{b=4}^{|B|} \sum_{w=a}^{c-1} y_w^b\right)$$

$$\begin{gathered} \forall\, a = 1,..,|A| \\ \forall\, c = a+1,..,|A| \end{gathered} \tag{15}$$

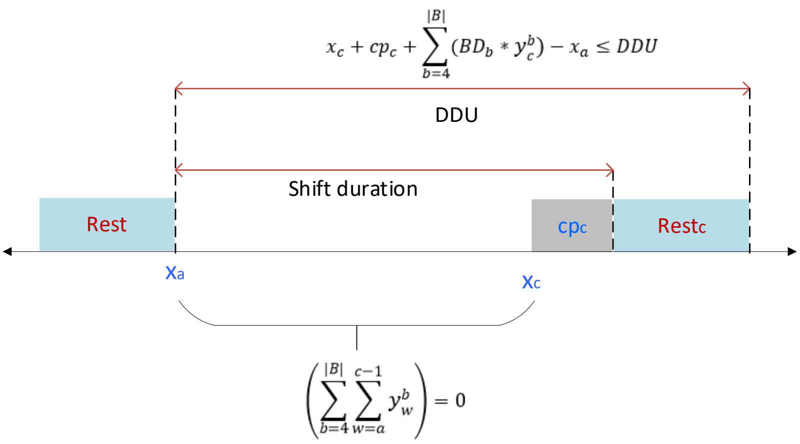


Figure 2-7. A rest should be scheduled each time interval of DDU hours.

Constraints (16) enforces that only RE(=3) times per week a reduced rest (rest of 9h) could be scheduled.

$$\sum_{a=1}^{|A|} y_a^4 \le RE \tag{16}$$

2.2.3 Numerical experiments

A new set of instances PGLT available at https://perso.isima.fr/~igpenaar/Roadef_2021/ the total number of instances is 41 and the size of the instances varies from 3 to 15 customers. Table 2-1 list the instances and the information about the number of clients, the total service time and the total driving time of each of them.

Table 2-1. Instances PGLT

Number	Clients	Total service time	Total driving time
1	3	2.00	10.00
2	4	2.00	8.00
3	2	2.00	8.00
4	5	3.00	8.00
5	6	3.00	20.00
6	11	5.00	16.00
7	11	5.00	16.00
8	11	5.00	16.00
9	11	7.00	15.00
10	11	7.00	15.00
11	11	7.00	15.00
12	11	7.00	15.00
13	4	3.50	11.00
14	7	5.00	21.00
15	10	6.50	31.00
16	10	6.50	31.00
17	10	6.50	31.00
18	11	5.00	16.00
19	11	5.00	16.00
20	11	5.00	16.00
21	11	5.00	16.00
22	15	21.00	23.00
23	15	21.00	23.00
24	10	9.00	14.50
25	10	10.50	19.50
26	10	17.00	7.50
27	10	21.00	7.50
28	10	30.00	8.50
29	4	5.60	4.50

Figure 2.8 presents the general data structure of the instances; it shows as an example instance 3 data information.

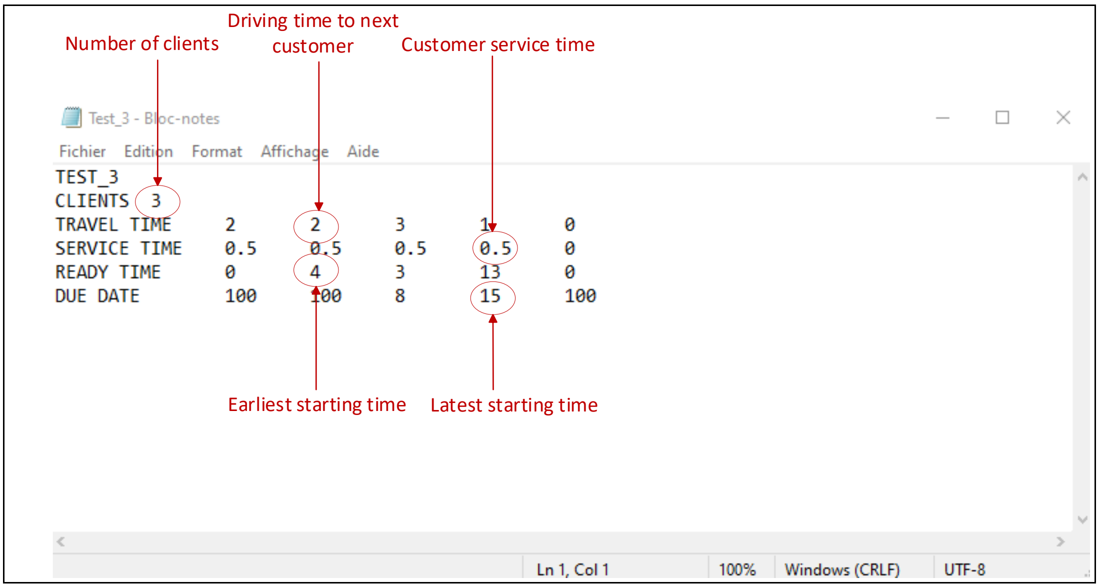


Figure 2-8. Instance 3.

All the experiments have been achieved on an Intel® Core™i5-8400 at 281 GHz under windows 10, using C++ and Gurobi 8.1.1. In addition, both detail solutions and their graphs are available at the same web page. Tables 2-2 and 2-3 presents the results of the linear model on the PGTL instances, minimizing the completion time and the makespan, respectively. The average completion time is 53.49h while the average makespan is 50.78h, with an average execution time of 309ms and 319ms, respectively.

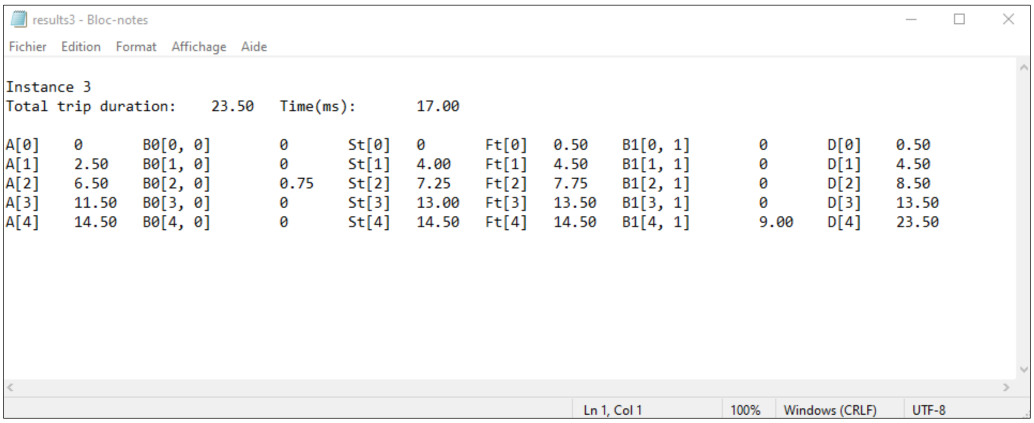
Table 2-2. Results instances PGLT-Completion time minimization

Number	Clients	Total service time	Total driving time	Completion time	CPU time(ms)
1	3	2.00	10.00	23.50	25.00
2	4	2.00	8.00	21.00	17.00
3	2	2.00	8.00	23.50	17.00
4	5	3.00	8.00	23.50	22.00
5	6	3.00	20.00	44.00	104.00
6	11	5.00	16.00	40.50	142.00
7	11	5.00	16.00	57.75	145.00
8	11	5.00	16.00	57.75	289.00
9	11	7.00	15.00	Unfeasible	Inf
10	11	7.00	15.00	45.50	89.00
11	11	7.00	15.00	45.50	92.00
12	11	7.00	15.00	45.50	85.00
13	4	3.50	11.00	33.25	24.00
14	7	5.00	21.00	55.25	192.00
15	10	6.50	31.00	79.25	2112.00
16	10	6.50	31.00	79.25	900.00
17	10	6.50	31.00	90.00	159.00
18	11	5.00	16.00	40.50	141.00
19	11	5.00	16.00	93.00	92.00
20	11	5.00	16.00	63.00	93.00
21	11	5.00	16.00	53.50	100.00
22	15	21.00	23.00	93.50	731.00
23	15	21.00	23.00	97.00	615.00
24	10	9.00	14.50	43.00	138.00
25	10	10.50	19.50	58.50	205.00
26	10	17.00	7.50	44.00	385.00
27	10	21.00	7.50	56.75	398.00
28	10	30.00	8.50	67.75	1317.00
29	4	5.60	4.50	22.35	34.00

Table 2-3. Results instances PGLT-Makespan minimization

Number	Clients	Total service time	Total driving time	Makespan	CPU time(ms)
1	3	2.00	10.00	23.50	36.00
2	4	2.00	8.00	19.75	40.00
3	2	2.00	8.00	20.50	29.00
4	5	3.00	8.00	23.50	27.00
5	6	3.00	20.00	44.00	136.00
6	11	5.00	16.00	40.50	163.00
7	11	5.00	16.00	40.50	152.00
8	11	5.00	16.00	57.75	166.00
9	11	7.00	15.00	Unfeasible	Inf
10	11	7.00	15.00	43.00	188.00
11	11	7.00	15.00	43.75	100.00
12	11	7.00	15.00	45.30	93.00
13	4	3.50	11.00	33.25	37.00
14	7	5.00	21.00	55.25	208.00
15	10	6.50	31.00	79.25	1816.00
16	10	6.50	31.00	79.25	791.00
17	10	6.50	31.00	89.50	277.00
18	11	5.00	16.00	40.50	131.00
19	11	5.00	16.00	85.50	102.00
20	11	5.00	16.00	55.50	103.00
21	11	5.00	16.00	40.50	148.00
22	15	21.00	23.00	84.50	688.00
23	15	21.00	23.00	84.50	441.00
24	10	9.00	14.50	43.00	228.00
25	10	10.50	19.50	58.50	311.00
26	10	17.00	7.50	44.00	519.00
27	10	21.00	7.50	56.75	822.00
28	10	30.00	8.50	67.75	1111.00
29	4	5.60	4.50	22.35	57.00

Figure 2.9 presents the solution for instance 3 while minimizing completion time.



Instance 3
Total trip duration: 23.50 Time(ms): 17.00

A[0]	0	B0[0, 0]	0	St[0]	0	Ft[0]	0.50	B1[0, 1]	0	D[0]	0.50
A[1]	2.50	B0[1, 0]	0	St[1]	4.00	Ft[1]	4.50	B1[1, 1]	0	D[1]	4.50
A[2]	6.50	B0[2, 0]	0.75	St[2]	7.25	Ft[2]	7.75	B1[2, 1]	0	D[2]	8.50
A[3]	11.50	B0[3, 0]	0	St[3]	13.00	Ft[3]	13.50	B1[3, 1]	0	D[3]	13.50
A[4]	14.50	B0[4, 0]	0	St[4]	14.50	Ft[4]	14.50	B1[4, 1]	9.00	D[4]	23.50

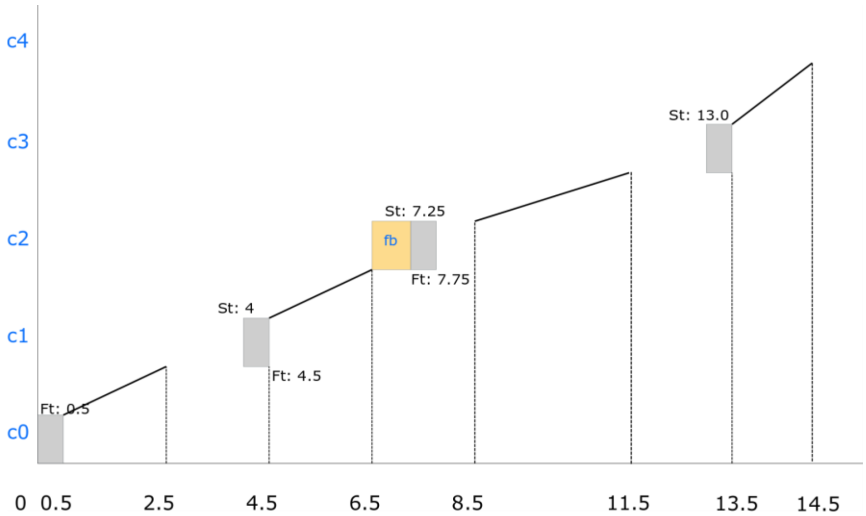


Figure 2-9. Solution instance 3.

The total trip duration is 23.5h after adding the rest period at the end of the sequence. The total execution time 17ms. The sequence begins at the depot with a service starting at 0 ($St_0 = 0$), after 0.5h of service the departure time is $D_0 = 0.5h$. Followed by a driving period of 2h, the arrival time at customer 1 is $A_1 = 2.5h$. Since the arrival time is before the earliest starting time of the service at customer 1 ($e_1 = 4h$), there is a waiting time of 1.5h, this is represented by a discontinuity in the graph. After the service time at customer 1 of 0.5h and the driving time of 2h, the arrival time at customer 2 is $A_2 = 6.5h$. At this point the continuous driving time is 4.0h, then, before going any further it is necessary to take the first full break, in order to reset the driving time, this full break is taken just before the service time at customer 2 ($B0_2 = 0.75h$). Note that this break could have been taken after the service and the optimal total trip duration does not change. There are two waiting times between customers 2 and 3, one of 0.75h after the finishing time of the service at customer 2 $Ft_2 = 7.75h$, then departure time is $D_2 = 8.5h$, the second waiting time of two hours is before the service time at customer 3, since the arrival time at customer 3 after three hours of driving is $A_3 = 11.5h$, and this is again due to the starting time of the time window at this customer $e_3 = 13h$. The starting time at customer 3 is $St_3 = 13h$, and the finishing time and departure time after 0.5h of service is 13.5h.

Followed by a driving period of 1h the sequence arrive at the final depot at $A_4 = 14.5h$, and sequence finishes with the compulsory rest of 9h.

2.2.4 Conclusion

A new MILP model for the Truck Driver Scheduling Problem under the European Union Regulation is presented. Some rules from the EU-regulation framework, that have not been previously taken into account in earlier linear formulations, such as: working breaks, extensions on driving time per day and daily rest reductions, are included in the linear formulation that is presented in this chapter. The model is solved using an optimization package Gurobi 8.1.1 and it is tested under a new set of instances, for which detail solutions are provided. Results show that the average computational time to solve instances up to 32 activities is about 319ms and 309ms, using makespan and completion time minimization, respectively. The maximum running times for both types of objective functions are 2112ms for completion time and 1816ms for makespan, moreover, both of them are found at instance 15.

2.3 Label setting algorithm

2.3.1 Introduction

In this section a label setting algorithm is presented for the truck driving scheduling problem under the European Union regulations. As for the MILP presented at the first part of the chapter, the method considers a sequence σ of customers that starts and ends at the depot, each of them has time windows and two activities: service and driving, where a given activity could have a duration of zero units of time. Figure 2-10 presents this arrangement. The objective is to find the optimal schedule of breaks and/or rests complying with the European Union Regulation, which minimizes the completion time of the sequence.

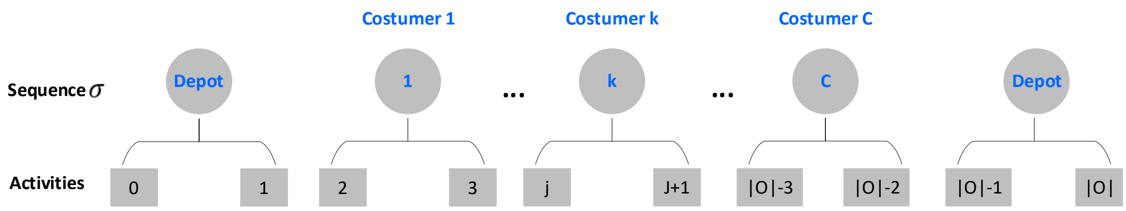


Figure 2-10. Arrangement customers and activities in the sequence.

Let C be the set of customer locations, which belong to the sequence σ, and A the set of activities, service and driving, related with each customer in C. Each activity $i \in A$ have a processing time p_i and time window $[ew_i, lw_i]$. Figure 2-11 gives an example of the setup of a sequence to evaluate, which has two customers and includes the depot at both, first and last position.

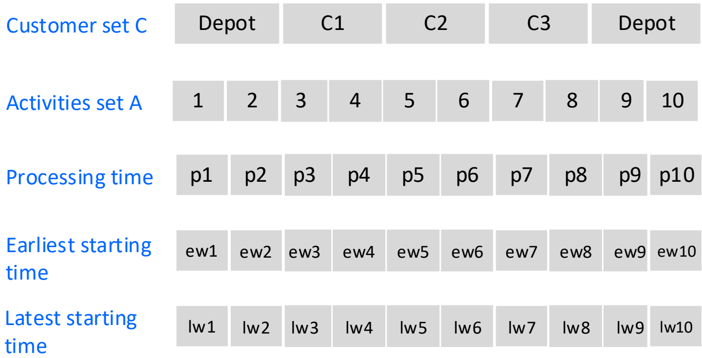


Figure 2-11. Example of an instance data.

The algorithm starts with the service activity at the depot, using an initial solution set to 0 ($\lambda_0 \leftarrow 0$). The movement from one given activity to the next, is done by adding the durations of the activity and the break/rest, in this order. Since this model does not allow breaks in the middle of an activity, all the activity duration is considered at once. Therefore, a label located at the list of solutions of activity i will have all their extensions at the list of solutions of activity $i+1$.

At each activity $i \in A$, it determines a set of feasible solutions or labels $(S, i+1)$ with a minimum completion time, by applying the *Extend* function to each label $\lambda_k \in (S, i)$. The process repeats until it reaches node de, which is the depot at the end of the sequence. Procedure 2-1 presents the description of the algorithm.

Procedure 2-1. Label Setting

```
procedure LabelSetting
  input:
  A: Set of activity nodes related to the customers that compose the sequence
    of visit σ;
  (S,i): A list of labels on node i;
  p_i: Processing time [driving/service] of activity i;
  ew_1: Earliest starting time window at the depot;
  lw_1: Latest starting time window at the depot;
  output:
  λ_k ∈ (S,de) with minimal completion time;
   INITIALIZATION: λ_0 ← 0; λ_0.shiftDelay ← (lw_1 − ew_1);
   (S,1) ← (S,1) ∪ {λ_0};
  foreach i ∈ C\de do
  | foreach λ_k ∈ (S,i) do
  | | (S,i+1) ← (S,i+1) ∪ {extend(λ_k, p, (S,i+1)}
  | end
  end
```

Procedure 2-1. Label Setting algorithm.

2.3.2 Basic attributes of a label

At any node, a label could be represented by a set of attributes Q. These are the attributes and their description.

$terminal\_node$, finishing node at which the label is attached.

$SLtoEF$, shift latest start for earliest finish. Feasible latest starting time of the first activity of the current shift, which allows arriving at earliest time at the terminal node of the following label.

EF, earliest finishing time of terminal node activity.

$shiftDelay$, maximum delay of the starting time of a given shift.

$DrPe$, cumulated driving time between two full breaks.

$DrSh$, current driving time during the shift.

$WkCo$, cumulated working time without a break.

$WkSh$, current working time during the shift.

$BrDu$, break duration at the current node.

$BrSh$, cumulated break time during the shift.

$BrSplDR$, True (Y), first part of a split driving break. False (N), otherwise.

$DrExt$, number of driving extensions during the week.

$ReRed$, number of reduced daily rest during the week.

2.3.3 Extension function

In order to move between two consecutive activities, each label saved at the list of labels (S, i) will create a new label or extension for every type of break. The extension process adds to the current earliest finishing time of the label at the current node i (EF), the processing time of the activity, plus the break duration and the waiting time if any; Giving the EF of the new solution at node $i + 1$, as Figure 2-12 depicts. If the extension is feasible and it is not dominated by the current set of solutions in node $i + 1$, this brand new extension is inserted in the list of labels $(S, i + 1)$.

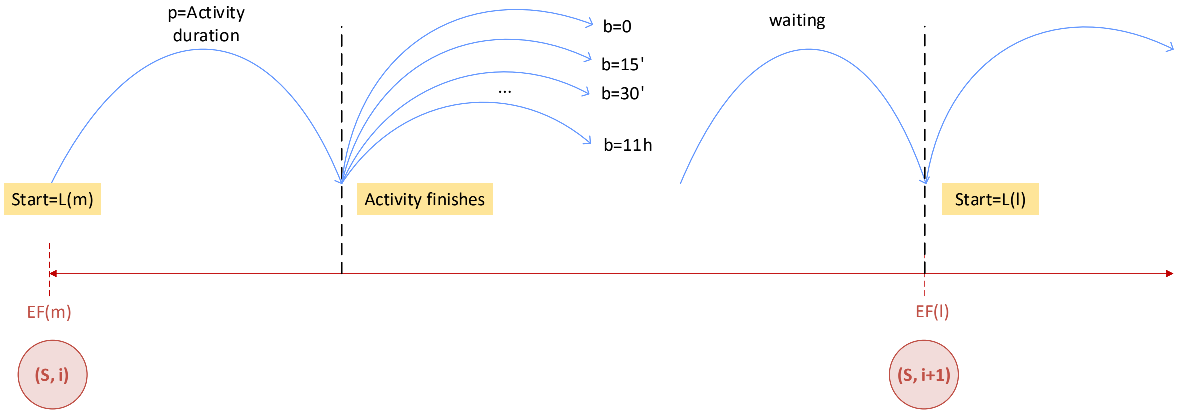


Figure 2-12. Function extend.

Function extend is applied to create a new set of solutions from a label $\lambda_k \in (S, i)$ towards node $i + 1$ when the extension is possible, see Procedure 2-2. One extension or solution φ_l is created when possible, for each type of break in the set of breaks $BD = \{0, 0.25h, 0.50h, 0.75h, 3h, 9h, 11h\}$. A solution is rejected when it does not pass the feasibility test or when it is out of the service time windows of a given customer. Only the extensions that are not dominated are included in the set of feasible solutions F_k. Finally, all feasible and not dominated solutions are included in the set of solutions of node the next node $(S, i+1)$.

Procedure 2-2. Extend

```
procedure Extend
  input:
  λ_k ∈ (S,i) Label to extend;
  p_i: Processing time [driving/service] of activity i;
  (S,i+1): A list of labels on node i+1;
  output:
  F_k: Set of labels extended from λ_k;
   foreach b ∈ BD do
   |  New extension φ_l;
   |  update(λ_k, φ_l, b); // Updating label φ_l attributes
   |  feasible ← feasibility(φ_l, p); // Feasibility test
   |  if feasible ≔ false then
   |  | b ← b + 1;
   |  else
   |  | dominate ← dominance((S, i + 1), φ_l); //Applying dominance rules on label φ_l
   |  | if dominate ≔ True then
   |  |   | b ← b + 1;
   |  | else
   |  | | F_k ← F_k ∪ {φ_l};
   |  | end
   |  end
   end
```

Procedure 2-2. Function extend.

Procedure 2-3 presents the update process of a new extension. The extended label φ_l inherits all the attribute levels from its parent label λ_k and afterwards they are updated depending on the processing time and the break duration. Accordingly to the type of break b, the attributes are reset to zero, increase their value, or change they state from false to true or the opposite, i.e. $\varphi_l.BrSplDr = True$ changes to $False$ if the break duration is greater or equal to 45 minutes. In like manner, all attributes related with driving or working are incremented by the processing time p, depending on the nature of the activity driving or working. Additionally, in this function is called the procedure $righShift$, which computes the attributes $\varphi_l.LStoEF$ and $\varphi_l.shiftDelay$.

Procedure 2-3. UpDate

```
procedure UpDate
  input:
  λ_k ∈ (S,i) Label to extend;
  φ_l: Label extended;
  p: Processing time [driving/service];
  output:
  Updated label φ_l;
φ_l.terminal − node ← λ_k.terminal − node + 1;
φ_l.father ← λ_k.id;
φ_l.BrDu ← b;
φ_l.ReRed ← λ_k.ReRed;
if b < B11h then φ_l.ReRed ← λ_k.ReRed + 1;
rightShift(λ_k, φ_l, b); //Computing LStoEF from φ_l
if driving activity then
  | φ_l.DrPe ← λ_k.DrPe;
  | φ_l.DrSh ← λ_k.DrSh;
  | if b ≥ B45m then φ_l.DrPe ← 0;
  | else if b ≥ B30m and λ_k.BrSplDr == True then
  | | φ_l.DrPe ← 0;
  | end
  | if b ≥ B9h then
  | | φ_l.DrPe ← 0;
  | | φ_l.DrSh ← 0;
  | end
  | φ_l.DrPe ← φ_k.DrPe + p;
  | φ_l.DrSh ← φ_k.DrSh + p;
end
φ_l.WkCo ← λ_k.WkCo;
φ_l.WkSh ← λ_k.WkSh;
φ_l.BrSh ← λ_k.BrSh;
if b ≥ B9h then
| φ_l.BrSh ← 0;
| φ_l.WkSh ← 0;
| φ_l.WkCo ← 0;
else if b ≥ B15m then
| φ_l.BrSh ← λ_k.BrSh + b;
| φ_l.WkCo ← 0;
end
if b ≥ B45m then φ_l.BrSplDr ← False;
else if B15m ≤ b < B45m then
| φ_l.BrSplDr ← True;
end
φ_l.WkCo ← φ_l.WkCo + p;
φ_l.WkSh ← φ_l.WkSh + p;
```

Procedure 2-3. Function UpDate.

Computing the latest starting time of a shift or the movement to the right of all the current operations within a shift, is performed at each extension by the procedure $rightShift$, which is presented in Procedure 2-4. At each node is compared the earliest arrival time ER of the label with the earliest starting time of the service ew; if the arrival time is before the ew of the node, then it is possible to delay the starting time of the shift. In the case where the activity is in the middle of the shift, the maximum delay is bounded by the minimum value between the current slack/waiting [$ew - ER$] and the current maximum delay of the shift [$\lambda_k.ShiftDelay$]. Later, the movement is valid if the

ER plus the delay is before the latest starting time (lw). In the second case when the activity is at the beginning of the shift and there is slack [$ew - ER > 0$], it starts at the earliest possible time (ew). In both cases, the attributes $\varphi_l.LStoEF$ and $\varphi_l.shiftDelay$ are updated accordingly.

Procedure 2-4. RightShift

```
procedure RightShift
 input:
λ_k ∈ (S,i) Label to extend;
 φ_l: Label extended;
 b: Type of break;
 p: Processing time [driving/service];
 ew: Earliest starting time window at current node;
 lw: Latest starting time window at current node;
 output:
 φ_l.LStoEF;
 ER ← λ_k.EF + b;
 if b < B9h then
 | if ER < ew then
 | | φ_l.EF ← ew + p;
 | | delay ≔ MIN{λ_k.shiftDelay; ew − ER};
 | | φ_l.LStoEF ← λ_k.LStoEF + delay;
 | | φ_l.ShiftDelay ← λ_k.shiftDelay − delay;
 | else
 | | φ_l.EF ← ER + p;
 | | φ_l.LStoEF ← λ_k.LStoEF;
 | | φ_l.shiftDelay ← λ_k.shiftDelay;
 | end
 | LR ← φ_l.EF − p + φ_l.shiftDelay;
 | if LR > lw then
 | | φ_l.shiftDelay ← φ_l.shiftDelay − [LR − lw];
 | else
 | | continue; //Infeasible extension
 | end
 else
 | //Daily rest [New shift]
 | if ER < ew then
 | | φ_l.EF ← ew + p;
 | | φ_l.LStoEF ← ew;
 | | φ_l.shiftDelay ← lw − ew;
 | else
 | | φ_l.EF ← ER + p;
 | | φ_l.LStoEF ← ER;
 | | φ_l.ShiftDelay ← lw − ER;
 | end
 end
```

Procedure 2-4. Function rightShift.

2.3.4 Feasibility

During the extension process, every new label goes through a feasibility test. In the procedure, a feasible solution should satisfy the European Union Regulation for Truck Drivers. It starts checking the rules on continuous driving time and the total driving time over the shift. After, rules on continuous working time and maximum shift duration are verified, the procedure continues with rules on compulsory minimal break duration according to working time during the shift. Finally, the feasibility process finishes checking rules on driving time extensions and daily rest reduction. Procedure 2-5 depicts the procedure.

Procedure 2-5. feasibility

```
procedure feasibility
  input:
  $\lambda_k \in (S,i)$ Label to extend;
  $\varphi_l$ Label extended;
  $p$: Processing time assigned to the current extension;
  output:
  state: $True$, if the solution is feasible. $False$, otherwise;
  $state \leftarrow True$;
  if $\lambda_k.DrPe + p > 4.5h$ then $state \leftarrow False$;
  if $\lambda_k.DrSh + p > 10h$ then $state \leftarrow False$;
  else if $\lambda_k.DrSh + p > 9h$ and $\varphi_l.DrExt \geq DE$ then
  | $state \leftarrow False$;
  end
  if $\lambda_k.WkCo + p > 6h$ then $state \leftarrow False$;
  if $\lambda_k.WkSh + p > 15h$ then $state \leftarrow False$;
  else if $\lambda_k.WkSh + p > 13h$ and $\varphi_l.ReRed \geq RE$ then
  | $state \leftarrow False$;
  end
  if $\varphi_l.BrDu > B9h$ then
  | if $\lambda_k.WkSh + p > 9h$ and $\lambda_k.BrSh < B45m$ then
  | | $state \leftarrow False$;
  | else if $6h > \lambda_k.WkSh < 9h$ and $\lambda_k.BrSh < B30m$ then
  | | $state \leftarrow False$;
  | end
  end
  if $\varphi_l.DrExt > DE$ then $state \leftarrow False$;
  if $\varphi_l.ReRed > RE$ then $state \leftarrow False$;
  end
```

Procedure 2-5. Function feasibility.

2.3.5 Dominance

Throughout the execution of the extension process, only non-dominated solutions should be considered, therefore, after verifying their feasibility, a dominance procedure that compares two labels $\{\lambda_1, \lambda_2\} \in (S,i)$, determines if label λ_1 dominates label λ_2. Procedure 2-6 describes this procedure. As in [11], the comparison of the attributes is done one by one over the set of labels at the current node i. Let, $R_i^1$and R_i^2 the attributes vectors of labels λ_1 and λ_2, respectively. Then, label λ_1 dominates λ_2 if $R_i^1 \leq R_i^2\ \forall\ i \in A$. In the case of the resource$BrSh$, an additional condition $\lambda_1.BrSh < B45m$.

Procedure 2-6. dominance

procedure dominance
  input:
  λ_1 First label to compare;
  λ_2 Second label to compare;
  output:
  dom: $True$, if λ_1 dominates λ_2. $False$, Otherwise;
  $dom \leftarrow True$;
  if $\lambda_1.LStoEF < \lambda_2.LStoEF$ **then** $dom \leftarrow False$;
  if $\lambda_1.BrSh < B45m$ **and** $\lambda_1.BrSh < \lambda_2.BrSh$ **then**
  | $dom \leftarrow False$;
  end
  if $\lambda_1.EF < \lambda_2.EF$ **then** $dom \leftarrow False$;
  if $\lambda_1.LStoEF + \lambda_1.shiftDelay < \lambda_2.LStoEF + \lambda_2.shiftDelay$ **then** $dom \leftarrow False$;
  if $\lambda_1.BrSplDr == False$ **and** $\lambda_2.BrSplDr == True$ **then** $dom \leftarrow False$;
  if $\lambda_1.DrPe > \lambda_2.DrPe$ **then** $dom \leftarrow False$;
  if $\lambda_1.ReRed > \lambda_2.ReRed$ **then** $dom \leftarrow False$;
  if $\lambda_1.ExtDr > \lambda_2.ExtDr$ **then** $dom \leftarrow False$;
  if $\lambda_1.WkCo > \lambda_2.WkCo$ **then** $dom \leftarrow False$;
  if $\lambda_1.WkSh > \lambda_2.WkSh$ **then** $dom \leftarrow False$;
  if $\lambda_1.DrSh > \lambda_2.DrSh$ **then** $dom \leftarrow False$;
  end

Procedure 2-6. Function dominance.

2.3.6 Numerical experiments for the label setting algorithm

In order to evaluate the performance of the label setting algorithm, it is run over the set of instances PGLT and compared with the MILP model presented in section 2.2. First, Figure 2-13 shows the details of the solution obtained by the label setting algorithm for instance 3. The completion time achieved is the same as using the linear model, although, there are some differences between the schedules. The linear model starts the schedule at time 0 and schedules only one break at the beginning of the service of customer 2. On the other hand, the label setting algorithm starts the schedule at time 3 and schedules two breaks one after the service at customer 2, and the other before the service at customer 3.

results3 - Bloc-notes

Fichier Edition Format Affichage Aide

Instance 3
Total trip duration: 23.5

A[0]	3	B0[0, 0]	0	St[0]	3	Ft[0]	3.5	B1[0, 1]	0	D[0]	3.5
A[1]	5.5	B0[1, 0]	0	St[1]	5.5	Ft[1]	6	B1[1, 1]	0	D[1]	6
A[2]	8	B0[2, 0]	0	St[2]	8	Ft[2]	8.5	B1[2, 1]	0.75	D[2]	9.25
A[3]	12.25	B0[3, 0]	0.75	St[3]	13	Ft[3]	13.5	B1[3, 1]	0	D[3]	13
A[4]	14	B0[4, 0]	0	St[4]	14.5	Ft[4]	14.5	B1[4, 1]	9	D[4]	23.5

Ln 18, Col 1 100% Windows (CRLF) UTF-8

Figure 2-13. Solution instance 3.

Table 2-4 presents the results of the label setting algorithm on the PGLT instances. As expected, the completion times achieved are equal to the results of the MILP with completion time minimization. To some extent, these results confirms that the label setting algorithm provides optimal solutions, although, with a significant difference in the running times. The average running time for the MILP was of 309ms, while, for the label setting algorithm it was of 6.1ms, roughly speaking 50 times faster.

Table 2-4. Results label setting algorithm for instances PGLT

Number	Clients	Total service time	Total driving time	Completion time	CPU time(ms)
1	3	2.00	10.00	23.50	3.00
2	4	2.00	8.00	21.00	2.00
3	2	2.00	8.00	23.50	2.00
4	5	3.00	8.00	23.50	1.00
5	6	3.00	20.00	44.00	3.00
6	11	5.00	16.00	40.50	3.00
7	11	5.00	16.00	57.75	13.00
8	11	5.00	16.00	57.75	5.00
9	11	7.00	15.00	Infeasible	1.00
10	11	7.00	15.00	45.50	3.00
11	11	7.00	15.00	45.50	4.00
12	11	7.00	15.00	45.50	3.00
13	4	3.50	11.00	33.25	3.00
14	7	5.00	21.00	55.25	6.00
15	10	6.50	31.00	79.25	6.00
16	10	6.50	31.00	79.25	6.00
17	10	6.50	31.00	90.00	3.00
18	11	5.00	16.00	40.50	4.00
19	11	5.00	16.00	93.00	7.00
20	11	5.00	16.00	63.00	7.00
21	11	5.00	16.00	53.50	10.00
22	15	21.00	23.00	93.50	21.00
23	15	21.00	23.00	97.00	34.00
24	10	9.00	14.50	43.00	4.00
25	10	10.50	19.50	58.50	5.00
26	10	17.00	7.50	44.00	4.00
27	10	21.00	7.50	56.75	5.00
28	10	30.00	8.50	67.75	6.00
29	4	5.60	4.50	22.35	3.00

2.5 Conclusion

Using the same assumptions of the MILP, a label setting algorithm is proposed in order to solve the problem in more efficient way. Both solution methods achieved the same completion time values over the test bed set of instances, albeit, the label setting algorithm solve them 50 times faster on average than the MILP model. The previous models are going to be used to develop more comprehensive formulations, in order to compare them with state of the art models in the literature. Notably, considering pre-emption or scheduling breaks in the middle of a driving activity.

2.6 References

[1] Archetti, C. and Savelsbergh, M.W.P., "The trip scheduling problem", *Transp. Sci.*, vol. 43, pp. 417–431, 2009.

[2] Goel, A. and Kok, A.L., "Truck driver scheduling in the United States", *Transp. Sci.*, vol. 46, pp. 317–326, 2012.

[3] Goel, A., "Truck driver scheduling in the European Union", *Transp. Sci.*, vol. 44, pp. 429–441, 2010.

[4] Goel, A., "Vehicle scheduling and routing with drivers' working hours", *Transportation Sci.*, vol. 43, no. 1, pp. 17–26, 2009.

[5] Kok, A. L., C. M. Meyer, H. Kopfer, J. M. J. Schutten., "A dynamic programming heuristic for the vehicle routing problem with time windows and European Community social legislation", *Transportation Sci*. vol. 44, no. 4, pp. 442–454, 2010.

[6] Goel, A., Archetti, C., and Savelsbergh, M.W.P., "Truck driver scheduling in Australia", *Comput. Oper. Res*., vol. 39, pp. 1122–1132, 2012.

[7] Goel, A., "A mixed integer programming formulation and effective cuts for minimising schedule durations of Australian truck drivers", *J. Sched.*, vol. 15, no. 6, pp. 733–741, 2012.

[8] Goel, A. and Rousseau L-M., "Truck driver scheduling in Canada", *J. Sched.*, vol.15, pp. 783–799, 2012.

[9] Goel, A., "The Canadian minimum duration truck driver scheduling problem", *Comput. Oper. Res.*, vol. 39, no. 10, pp. 2359–2367, 2012.

[10] Goel, A., "The minimum duration truck driver scheduling problem", *EURO J. Transp. Logist.*, vol. 1, no. 4, pp. 285–306, 2012.

[11] Feillet D, Dejax P, Gendreau M and Gueguen C., "An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems". *Networks.*, vol. 44, no. 3, pp. 16–229, 2004.

CHAPTER 3

TDSP under European driving rules with pre-emption

A new MILP model and a Label Setting algorithm are presented in this chapter. In the models presented in chapter 2 breaks/rest could only take place when an activity is completely done, the models of this chapter remove this assumption allowing taking breaks in the middle of an activity. Additionally, two new rules are considered, split daily rests and night work. Therefore, these models include all weekly rules to schedule when the sequence of visits is fixed. The set of test instances is enhanced in order to verify the behavior of the models under the new assumption and rules. Moreover, based on a set of numerical experiments, we discuss the effect of the night working rule on the feasibility of the schedules and we make a comparison against a simplification of this rule.

3.1 Introduction

In the literature all previous MILP formulations work under the assumption of taking breaks at customers or suitable locations, meaning that they work under non pre-emptive assumptions, and do not consider different rules from the EU regulation, particularly the night rule. Hence, there is missing a thorough formulation that allows a better understanding of the problem and in particular of the EC Social legislation. Moreover, it could be a tool in order to develop handcrafted algorithms to solve the problem. On the other hand, there are different solution methods that work under pre-emption assumptions for driving activities while taking into account all the rules from the EC social legislation, including a simplification of the night working rule.

Goel in [1] presents two algorithms to solve the scheduling problem. One naïve method that schedules rest/breaks as soon as a rule requires them, and a recursive algorithm, which explores different alternatives at the same time. Whereas, the routing problem is solved by means of a large neighbourhood search algorithm. The algorithm works under pre-emption assumptions, although, there are missing some of the EU rules like: working breaks, daily rest reductions, daily driving time extensions and the night working rule.

Kok et al [2] present a dynamic programming heuristic to solve the VRPTW complying with all restrictions on driving and working hours in the EU driver's rules, except, for the night working rule. The routing problem is solved using a restricted dynamic program, which uses two parameters E and H; E restricts the number of state expansions of a single state, and H specifies the maximum number of states to be taken to the next iteration. In addition, they propose two methods for the scheduling breaks problems, the naïve label setting method presented in [1], and an extended method.

Results show that the naïve (basic) method outperforms state of the art heuristics for the VRPTW with the EU Social legislation.

Later, Prescott-Gagnon et al. [3] develop a Large Neighborhood Search algorithm (LNS) for a VRPTW under the EC social legislation. It is noteworthy that the night working rule is also not considered in this publication. Moreover, they present a comparison against [1] and Kok et al. [2] considering both, the basic set of rules and the extended set of rules; results show that their LNS algorithm outperforms the two previous approaches. Goel and Vidal [4] describe a hybrid genetic algorithm to solve the VRTDSP under different drivers' hour regulations, particularly in the United States, Canada, the European Union, and Australia. With respect to the EC social legislation, they consider all rules except for the night working rule. As well, they provide a comparison against the results of Prescott-Gagnon et al. [3] under two set of rules denoted *No Split* and *All*. Using a test Wilcoxon, they show that considering a p-value of 0.0001 the mean of the solutions are different, providing better results. However, their running times are 4.9 and 2.6 times greater, respectively. In addition, other results indicate that the European Union rules bring the highest safety, while the most competitive in economic terms are the Canadian regulations.

Considering both, the regulation from United States the Hours of Service (HOS) legislation and the European Union legislation, Goel and Irnich [5] developed a branch and price algorithm to solve the Vehicle Routing and Truck Driver Scheduling Problem (VRTDSP). It is the first algorithm that solves to optimality the VRTDSP. Although, with respect to the rules from the EC Social legislation, they do not consider rules related with the working time directive, daily driving time extensions and daily rest reductions. They consider 56 instances with 25 customers, in the case of the HOS they solve all of them and for the EU legislation they solve 53 instances.

Goel [6] modifies the previous work, considering two new aspects in VRTDSP problems: night working and the number of working days minimization. In addition, this contribution includes the missing rules in [5]. With regard to the night working rule, this paper considers transport operations where drivers do not work during the night, in other words, the night working time is forbidden by covering night periods with breaks/rests or waiting time. Later, Tilk and Goel [7] extends the previous work by considering both, the European Union social legislation and the Hours of service from U.S. Furthermore, from the algorithmic point of view they found improvements in the average computational time, trough the implementation of a bidirectional label setting algorithm instead of the former forward label setting version. The rules and assumptions considered with regard of the EC Social legislation remained unaffected.

As a result, the most recent dynamic programming models to generate compliant (legal or feasible) schedules under EC social legislation are those presented by [6] and [7]. Even though, they consider all the rules from the EC social legislation, they propose a simplified version of the night working rule, assuming that a rest/break or waiting time are taken every night, and do not offer optimal solutions for the TDSP.

3.2 Mixed Integer Linear Problem

A MILP formulation is proposed to schedule driver's shifts under the European Union regulations considering pre-emption assumptions. The model evaluates a sequence σ of customers that starts and ends at the depot. The following set of constraints from the EU regulation are considered:

R1. A driver cannot drive a total amount of 4.5h without a break of more than 45min referred to as "full break".

R2. A full break of 45min can be split into two periods, a first break of at least 15min (and less than 45 min) and a second break of at least of 30min; the breaks should be taken in this order.

R3. If the total working time of a shift is less than 6h, a break of at least 15 minutes has to be scheduled.

R4. If the total working time of a shift is between 6h and 9h, a break of 30 min has to be scheduled.

R5. If the total working time during a shift is greater than 9h, a break of 45 min is required.

R6. The maximal driving time in a shift is 9h that could be extended to 10h twice per week.

R7. A shift cannot exceed 24 hours.

R8. The minimal rest period is 11h that could be reduced to 9h three times per week. Therefore, the sum of breaks, POA and working durations is bounded by 13h; 15h if the rest is reduced to 9h.

R9. A rest can be taken into two periods, the first an uninterrupted period of at least 3h, and the second a rest of at least 9h.

R10. If the shift includes night working time, then the total working time in the shift is limited to 10h.

To the best of our knowledge, different rules have not been previously included in linear formulations, such as: split rests, working breaks and night work. In addition, previous works have proposed a different assumption on the night work rule, but the night rule of the EU regulation [6] and [7].

3.2.1 Data and variables

Hereafter, we present a general description of the linear model. The initial set of data gives the sequence of locations to visit σ, with driving and service times at each location. Depending on the problem, service times are loading and/or unloaded operations to service customers with time-windows consideration for services.

Each activity $a \in A$ is modeled through a set of nodes $u \in U$, where at each node a processing time and a break/rest can be assigned in this order. Some parameters are required to model an activity a:

- A set of nodes of length n_a.
- $fu(a)$ is the first node related to activity a.
- lu(a) is the last node of activity a.

The model take advantage of a set of U of nodes where $U = \sum_{a=1}^{a=n} n_a$. Figure 3-1 depicts this modeling data structure for a sequence with three customers.

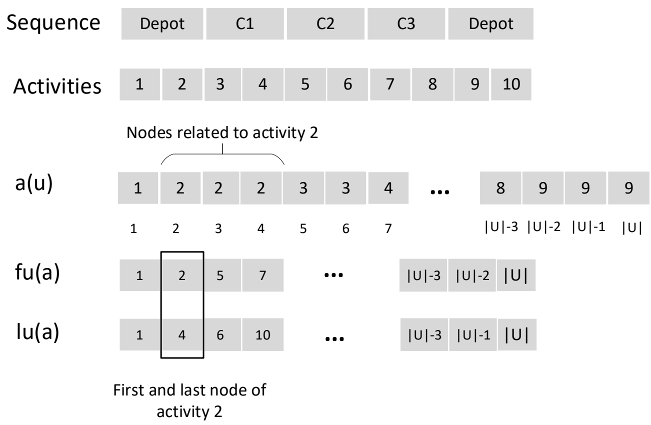


Figure 3-1. Modeling data structure.

In addition, the model considers the European Regulation definition which defines night intervals from 0am to 4am for good vehicles and from 1am to 5am for passenger vehicles. The model addresses the good vehicle transportation problem and defines a set intervals $I = \{[0,4], [24,28], \dots, [144,148]\}$ to model the periodic night time periods, the set of types of breaks $B = \{0, 0.25h, 0.5h, 0.75h, 3h, 9h, 11h\}$. In this version, breaks are restricted to have only values from this set. Breaks that are in between two of these break durations are rounded down to the smallest minimal break duration. For example, if there is enough time to schedule a break of 0.7h, it is considered as a break of 0.5h followed by a POA of 0.2h.

All the parameters of the model are listed and the linear constraints are given just after. Note that all durations are in hours. Figure 3-2 gives an example of a sequence to evaluate, which has three customers and includes the depot at both, first and last position. The set of activities A corresponds to the service and transport activities developed at each location. As well, a fixed number of nodes assigned to each activity composes the set of nodes U. In order to retrieve the activity a related with a node u, also the first and the last node u related with an activity a, three vectors are defined $a(u), fu(a)$ and $lu(a)$, respectively.

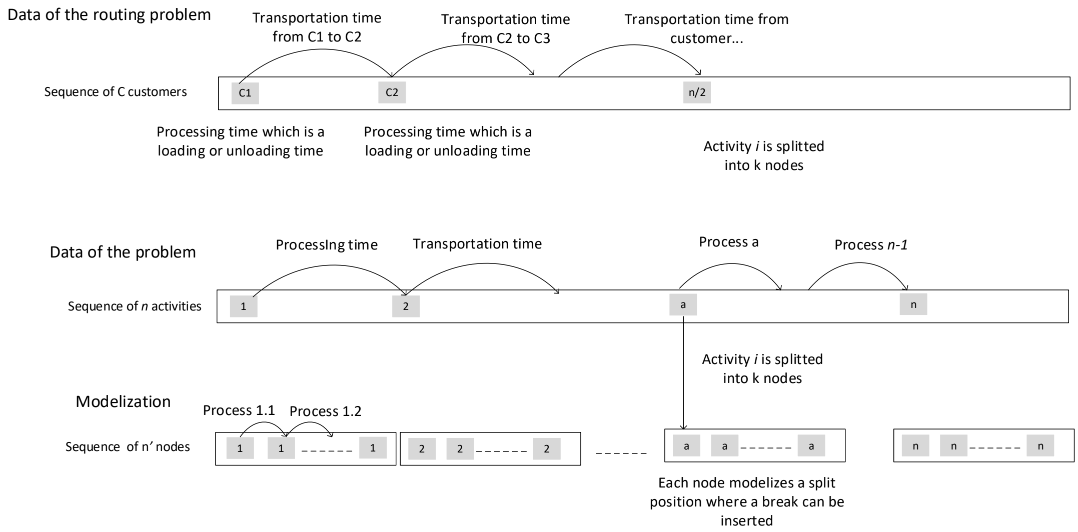


Fig. 3-2. Modelling approach

Data:

M: A big number.

$n(a)$: number of nodes used to modelized the activity a.

$fu(a)$: First node related to activity a.

lu(a): Last node of activity a.

cp_a: Duration of activity a.

δ_a: 1, if activity a is related to a driving activity and 0 otherwise.

δ_u: 1, if node u is related to a driving node and 0 otherwise.

$a(u)$: Activity related with the node u.

e_a: Earliest starting time of the activity a.

l_a: Latest starting time of the activity a.

$BD_b = \{0.25, 0.50, 0.75, 3, 9, 11\}$: Minimal break durations.

$LB_i = \{0, 24, \dots, 144\}$: Upper bound of the night interval i.

$UB_i = \{4, 28, \dots, 148\}$: Upper bound of the night interval i.

$WL9$: Break duration if working time is less than 9h [30min].

$WG9$: Break duration if working time is greater than 9h [45min].

CWT: Continuous working time without a break [6h].

$WSB1$: Maximal working time in a shift without at least $WL9$ ($=$ 30) minutes of break [6h].

$WSB2$: the sum of working times in a shift is upper bounded by 9h without at least $WG9$ ($=$ 45) minutes of break [9h].

CDT: Continuous driving time without a break [4.5h].

MDR: Regular driving time per shift [9h].

MDE: Extended driving time per shift [10h].

DE: Maximum number of driving extensions per week [2 times].

RE: Maximum number of rest reductions per week [3 times].

DDU: The maximal duration without a rest [24h].

MWN: Maximum working time if night work is performed [10h].

Decision variables:

x_u: Starting time of process at node u.

y_u^b: 1, if at node u a break of type b takes place after the process. 0, otherwise.

F_u: 1, if at node u a full break takes place. 0, otherwise.

E_u: 1, if the driving time is extended during one hour. 0, otherwise.

H_u: 1, if a rest of 9h is taken without a previous rest of 3h and 0 otherwise.

b_{uv}: 1 if working time between nodes u and v is greater than 6h and 0 otherwise.

c_{uv}: 1, if working time between nodes u and v is greater than 9h and 0 otherwise.

bn_{ui}: 1, if the activity u starts before the night interval I_i and 0, otherwise.

an_{ui}: 1, if the activity u starts after the night interval I_i and 0 otherwise.

nwn_{uv}: 1, if at least one node k in $[u, v]$ is a working node and one node k is overlapping one night i and 0 otherwise.

$leq24stEnd_{uv}$: 1, if time between the starting time of node u and the finishing time of node v is less or equal to 24h and 0 otherwise.

$leq24stSt_{uv}$: 1, if time between the starting time of node u and the starting time of node v is less or equal to 24h and 0 otherwise.

p_u: Processing time at node u.

wo_{uv}: sum of all working time between nodes u and v.

d_{uv}: sum of all driving times between nodes u and v.

3.2.2 Objective function and constraints

The objective function (1) is to minimize the completion time of the processing time of the last node.

$$Min\ z = x_{|U|} + p_{|U|} \tag{1}$$

Precedence constraint

Constraints (2) and (3) enforce the sequence between activities and nodes, respectively as Figure 3-3 shows.

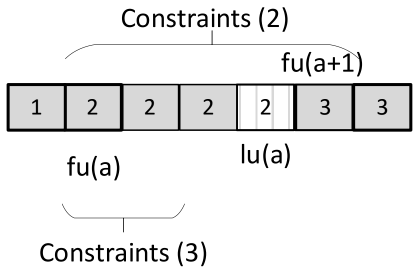


Figure 3-3. Constraints 2 applies over activities and Constraints 3 applies for each pair of nodes.

This constraint enforces that the starting time $x_{fu(a+1)}$ of the node $fu(a+1)$ (which concerns the activity $a+1$) is greater than the starting time of the node $fu(a)$ plus the processing time of activity a (i.e. cp_a) plus the sum of the breaks ($\sum_{u=fu(a)}^{lu(a)} \sum_{b=1}^{|B|} BD_b \times y_u^b$) scheduled between the node $fu(a)$ and the last node $lu(a)$ which refers to the last node of activity a.

$$x_{fu(a)} + cp_a + \sum_{u=fu(a)}^{lu(a)} \sum_{b=1}^{|B|} BD_b \times y_u^b \leq x_{fu(a+1)} \qquad \forall\ a = 1, \dots, |A| - 1 \tag{2}$$

The constraint (3) is dedicated to successive nodes of the same activity. This constraint enforces that the starting time of node $u+1$ is greater that the starting time of node u plus the duration of u (p_u) plus the break ($\sum_{b=1}^{|B|} BD_b \times y_u^b$) that is potentially scheduled at node u.

$$x_u + p_u + \sum_{b=1}^{|B|} BD_b \times y_u^b \leq x_{u+1} \qquad \forall\ u = 1, .., |U| - 1 \tag{3}$$

Break constraints per node

Only one type of break could take place at each node is enforced by constraints (4).

$$\sum_{b=1}^{|B|} y_u^b \leq 1 \qquad \forall\, a = 1,..,|A| \qquad (4)$$

Time windows per activity

The starting time of a node u for a service activity a must be between the earliest starting time $e_{a(u)}$ and the latest starting time $l_{a(u)} + cp_{a(u)}$, as Figure 3-4 depicts. In the graph this constraint must hold for all nodes u from a service activity ($\delta_{a(u)} = 0$).

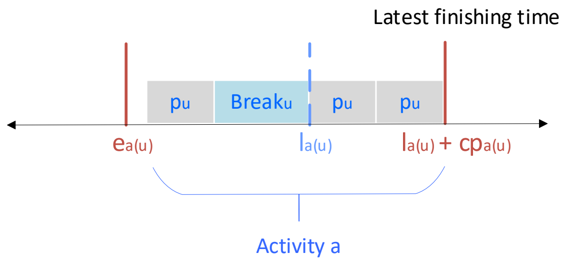


Figure 3-4. Nodes of an activity a have to start their process between the time window $[e_{a(u)}, l_{a(u)} + cp_{a(u)}]$.

Constraints (5) ensure that the starting time of a service node u is greather than e_a and lower than $l_a + cp_{a(u)}$ where $l_{a(u)} + cp_{a(u)}$ refers to as the latest starting time.

$$e_{a(u)} \leq x_u \leq l_{a(u)} + cp_{a(u)} \qquad \forall\, u = 1,..,|U| - 1\,; \quad \delta_{a(u)} = 0 \qquad (5)$$

Maximal driving time constraint

Constraints (6) enforces that the driving time d_{uv} (this driving time is the sum of driving times from u to $v - 1$) between node u and node v must not exceed 4.5 hours, see Figure 3-5. It means that a full break must be scheduled after 4.5 hours of continuous driving time. If no break is scheduled from node u (starting time of u) to starting time of node $v - 1$ then the sum of break $\sum_{w=u}^{v-1} F_w = 0$ and the constraint 6 holds to enforce that $d_{uv} \leq 4.5$. If a break is scheduled between u to $v - 1$ then $\sum_{w=u}^{v-1} F_w > 1$ and the constraint hold trivially.

$$d_{uv} \leq CDT + M \sum_{w=u}^{v-1} F_w \qquad \forall\, u,v\, \in U, u \leq v \qquad (6)$$

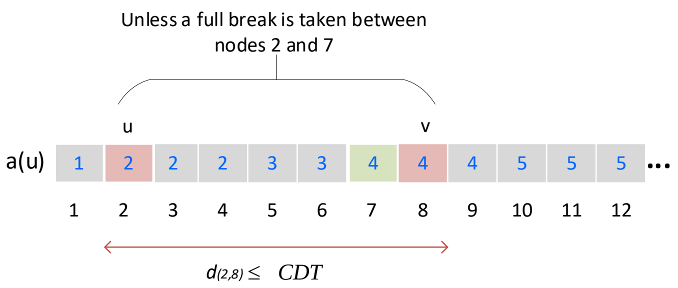


Figure 3-5. Compulsory full break after CDT hours of driving.

Full break definition

A full break is a break with a duration greater or equal to 45 min. Constraints (7)-(9) enforce $F_u = 1$ it a full break is achieved at node u and 0 otherwise.

If a break of either 45 min, 3h, 9h or 11h are scheduled at node u, $\sum_{b=3}^{|B|} y_u^b = 1$ then constraint (7) enforces that $F_u \geq 1$ and if a break greater or equal 45min ($b \geq 3$) is scheduled at node u.

If no break of duration greater or equal to 45min are schedule i.e. $\sum_{b=3}^{|B|} y_u^b = 0$ then constraint C7 holds trivially and the constraint –C8) enforces that $F_u = 0$. The constraint C9 enforces that if a break of 15min is scheduled then it is not a full break.

$$\sum_{b=3}^{|B|} y_u^b \leq F_u \qquad \forall\, u = 1,..,|U| \qquad (7)$$

$$\sum_{b=2}^{|B|} y_u^b \geq F_u \qquad \forall\, u = 1,..,|U| \qquad (8)$$

$$1 - y_u^1 \geq F_u \qquad \forall\, u = 1,..,|U| \qquad (9)$$

Special situation where a break of 30min can be the last part of a full break that has been split into 15 min first and 30 second.

Constraint (10) addresses the specific situation where a full break of 30min is scheduled at v.

$$\left(1 - \sum_{b=1}^{2} y_v^b\right) + \sum_{w=u}^{v-1} \sum_{b=1}^{2} y_w^b + (1 - F_{u-1}) \geq F_v$$

$$\forall\, v = 1,..,|U|; \forall\, u = 1, \dots, v; \qquad (10)$$

Between two nodes u and v, a break of 30 minutes at node v ($y_v^2 = 1$) can be the second part of a split break, if two conditions hold. First, a break of 15 minutes or 30 minutes takes place in the interval $[u, v[$ ($\sum_{w=u}^{v-1} \sum_{b=1}^{2} y_w^b = 1$). Second, a full break has been scheduled at node $u - 1$ ($F_{u-1} = 1$). The constraint can be rewritten

$$\sum_{w=u}^{v-1} \sum_{b=1}^{2} y_w^b \geq F_v$$

meaning that potentially the break of 30' scheduled at node v can be either a full break or not.

On the contrary, after a break of 30 minutes at node v ($y_v^2 = 1$), it is not considered as a full break, if there is not a break of 15 or 30 minutes between $[u, v[$ (as the first part of the split) ($\sum_{w=u}^{v-1}\sum_{b=1}^{2} y_w^b = 0$) and a full break is scheduled just before ($F_{u-1} = 1$). The constraint can be rewritten : $0 \geq F_v$

This inequalities (10) forbids breaks of 30 min to be considered as full break if there is not first part of split break. Note that $F_0 = 1$ by definition. See Figure 3-6.

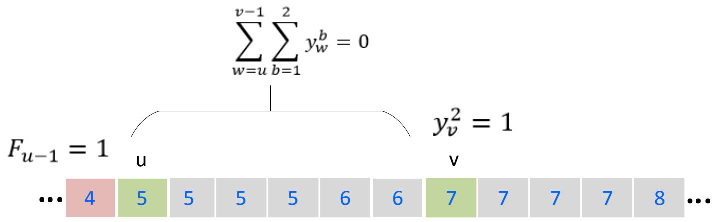


Figure 3-6. A 30 minutes break is not a full break.

Continuous working time break

The working time between two nodes u and v has to be less than six hours ($wo_{uv} \leq CWT = 6h$) if there is no break in between them($\sum_{w=u}^{v-1}\sum_{b=1}^{|B|} BD_b \times y_w^b = 0$).

$$wo_{uv} \leq CWT + M \sum_{w=u}^{v-1}\sum_{b=1}^{|B|} BD_b \times y_w^b$$

$$\forall\, u, v = 1, .., |U|, u \leq v \qquad (11)$$

If $\sum_{w=u}^{v-1}\sum_{b=1}^{|B|} BD_b \times y_w^b = 0$ the constraint can be rewritten $wo_{uv} \leq CWT$.

30 minutes of break for 6hours of Working time

If the working time during a shift is greater than 6h then the total break duration of the shift must be greater than 30 minutes, this is enforced by constraint (12). If nodes u and v denote the beginning and the end of a shift, that is to say, two consecutive daily rest take place at nodes v and $u-1$ ($\sum_{b=5}^{|B|} y_v^b - y_{u-1}^b = 2$) and the total working time between u and v is greater than 6h ($b_{uv} = 1$) then the total break duration of the shift must be greater than $WL9$ (=30minutes) ($\sum_{w=u}^{v-1}\sum_{b=1}^{|B|} BD_b \times y_w^b \geq WL9$). See Figure 3-7.

$$M\left(2 - \sum_{b=5}^{|B|} y_v^b - y_{u-1}^b\right) + \sum_{w=u}^{v-1}\sum_{b=1}^{|B|} BD_b \times y_w^b + M(1 - b_{uv}) \geq WL9$$

$$\forall\, u, v = 1, .., |U|, u \leq v \qquad (12)$$

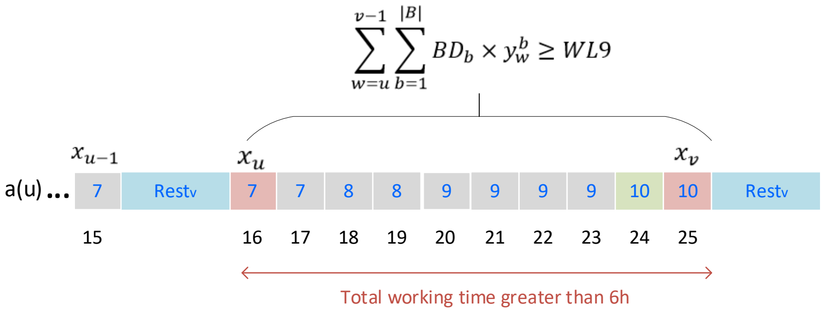


Figure 3-7. Break shift of 30 minutes if the working time during the shift is greater than WL9.

If $\sum_{b=5}^{|B|} y_v^b - y_{u-1}^b = 2$ and $b_{uv} = 1$ then the constraint is rewritten:

$$\sum_{w=u}^{v-1}\sum_{b=1}^{|B|} BD_b \times y_w^b \geq WL9$$

<u>45 minutes of break for more than 9 hours of Working time</u>

In addition, if a shift is defined between nodes u and v, that is to say, two consecutive daily rest take place at nodes v and $u-1$ ($\sum_{b=5}^{|B|} y_v^b - y_{u-1}^b = 2$) and the total working time between u and v is greater than 9h ($c_{uv} = 1$) then the total break duration within the shift must be greater than $WG9$ (= 45 minutes) ($\sum_{w=u}^{v-1}\sum_{b=1}^{|B|} BD_b \times y_w^b \geq WG9$).

$$M\left(2 - \sum_{b=5}^{|B|} y_v^b - y_{u-1}^b\right) + \sum_{w=u}^{v-1}\sum_{b=1}^{|B|} BD_b \times y_w^b + M(1 - c_{uv}) \geq WG9$$

$$\forall\, u, v = 1, \ldots, |U|, u \leq v \qquad (13)$$

Constraints (14) and (15) compute the binary indicators b_{uv} and c_{uv}, which are set to one when the working time exceed their respective limits. Hence, if $wo_{uv} \geq WSB1 (= 6h)$ then $b_{uv} = 1$ and if $wo_{uv} \geq WSB2 (= 9h)$ then $c_{uv} = 1$.

$$M \times b_{uv} \geq wo_{uv} - WSB1 \qquad \forall\, u = 1, \ldots, |U|; u \leq v \qquad (14)$$

$$M \times c_{uv} \geq wo_{uv} - WSB2 \qquad \forall\, u = 1, \ldots, |U|; u \leq v \qquad (15)$$

<u>Daily driving limit and daily driving extensions</u>

The maximum driving time during a shift is $MDR (= 9h)$ and it could be extended for one additional hour $DE (= 2)$ times per week, this driving limit is set by constraint (16) (Figure 3-8)

$$d_{uv} \leq MDR + E_u + M\left(1 - \sum_{b=5}^{|B|} y_{u-1}^b + \sum_{w=u}^{v-1}\sum_{b=5}^{|B|} y_w^b\right)$$

$$\forall\, u, v = 1, \ldots, |U|, u \leq v \qquad (16)$$

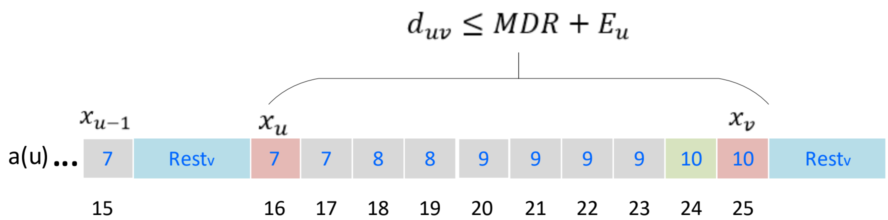


Figure 3-8. Maximum driving time during a shift.

When a shift is defined between nodes u and v ($\sum_{b=5}^{|B|} y_{u-1}^b + \sum_{w=u}^{v-1} \sum_{b=5}^{|B|} y_w^b = 1$), then the constraints can be rewritten

$$d_{uv} \le MDR + E_u$$

and enforces that the driving time d_{uv} must be less or equal to $MDR(= 9h)$ plus the extension E_u. In addition, note that $y_0^b = 1$ by definition.

<u>The driving time can be extended of 1 hour per day two times per week</u>

Constraint (17) ensures that in one week there are at most $DE(= 2)$ driving extensions of one hour.

$$\sum_{u=1}^{|N|} E_u \le DE \tag{17}$$

<u>Shift duration</u>

Every interval $DDU(= 24h)$ must have one rest period.

$$x_v + p_v + \sum_{b=1}^{|B|} BD_b \times y_v^b - x_u \le DDU + M \sum_{w=u}^{v-1} \sum_{b=5}^{|B|} y_w^b$$

$$\forall\, u, v = 1, .., |U|, u < v \tag{18}$$

As Figure 3-9 shows, constraint (18) enforces that between the finishing time of the activity at node v $(x_v + p_v)$ plus the break at this node $(\sum_{b=1}^{|B|} BD_b \times y_v^b)$ and the starting time of node u there is less than DDU hours, if there is not a rest between u and $v - 1$ ($\sum_{w=u}^{v-1} \sum_{b=5}^{|B|} y_w^b = 0$).

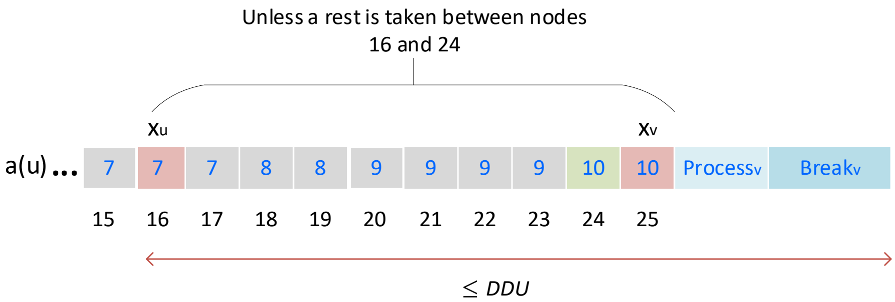


Figure 3-9. Compulsory rest each DDU hours.

Processing time

The processing time of the activity a (cp_a) is equal to the sum of the processing times p_u of all nodes u related with the activity. Constraint (19) compute the processing time for nodes u, between $fu(a), \dots, lu(a) - 1$.

$$p_u = x_{u+1} - \left[x_u + \sum_{b=1}^{|B|} BD_b \times y_u^b \right]$$

$$\forall\, a = 1, .., |A|; \quad \forall\, u = fu(a), \dots, lu(a) - 1 \qquad (19)$$

The processing time at node u is the difference between the starting time of the next node x_{u+1} and the starting time of node u plus the break at this position, as Figure 3-10 depicts.

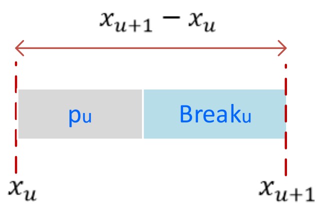


Figure 3-10. Processing time from two consecutive nodes of the same activity.

Processing time

Constraint (20) computes the processing time at the last node of the activity as the difference between the total processing time of the activity cp_a and the accumulated processing time of the precedent nodes u of the activity. This constraint ensures that the processing time of the activity is equal to the sum of the processing times of each node u related to the activity a :

$$\sum_{u=fu(a)}^{lu(a)} p_u = cp_a$$

$$\forall\, a = 1, .., |A| \qquad (20)$$

Driving and working time between nodes

Constraints (21) and (22) compute the driving time and the working time between two nodes u and v respectively. Constraint (21) adds the processing time of nodes that satisfy the condition $\delta_u = 1$, meaning nodes related with a driving activity.

$$d_{uv} = \sum_{w=u|\delta_u=1}^{v} p_w$$

$$\forall\, u, v = 1, .., |U|, u < v \qquad (21)$$

Since both activities, driving and service are working time, constraint (22) adds the processing time and driving time of nodes between u and v.

$$wo_{uv} = \sum_{w=u}^{v} p_w$$
$$\forall\, u, v = 1, \ldots, |U|, u < v \tag{22}$$

Split daily rest

A daily rest could be split in two parts, the first part a continuous rest of 3 hours, followed by a second interrupted rest of 9 hours, Figure 3-11 shows this case, between nodes 20 and 25.

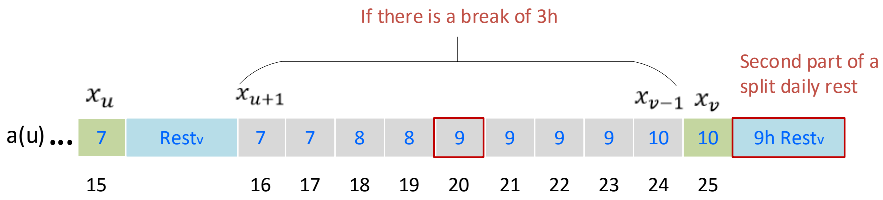


Figure 3-11. Split daily rest.

Split rest of 3h plus 9 hours

Constraints (23) establish the conditions for a split daily rest.

$$y_v^5 + \sum_{b=5}^{|B|} y_u^b - \sum_{w=u+1}^{v-1} y_w^4 \le H_v + 1$$
$$\forall\, u = 1, \ldots, |U|; u \le v \tag{23}$$

If a rest of 9h takes place at node v ($y_v^5 = 1$) and there is not a 3h break between $]u, v[$ ($\sum_{w=u+1}^{v-1} \sum_{b=4}^{|B|} y_w^b = 0$) the constraint can be rewritten:

$$\sum_{b=5}^{|B|} y_u^b \le H_v$$

If a break of 9h or 11h is scheduled at node u then $\sum_{b=5}^{|B|} y_u^b = 1$ and the constraint enforces that $H_v = 1$. $H_v = 1$ means that a break of either 9h is scheduled at node v and it is not a split rest of 3h plus 9 hours.

Number of time a split rest of 3h plus 9 hours is scheduled

Within a week, there is a limit of RE (= 3) reduced daily rests. Constraints (24) limits the number of 9h rests ($H_u = 1$) that are limited to RE.

$$\sum_{u=1}^{|U|} H_u \le RE \tag{24}$$

Night working rule : constraint (25)– (34)

EU night rule constraints (25)– (34): if there is night work performed, the shift working time over the last 24h should be up to 10h.

Constraints (25)-(26) enforce activities nodes u to be processed exclusively before(LB_i). Therefore, if an activity starts before the lower bound of the night interval i (LB_i), it has to finish before it. See Figure 3-12.

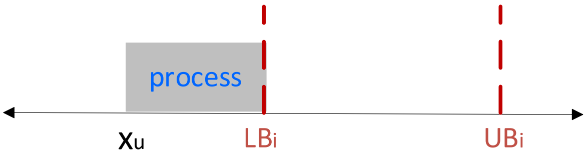


Figure 3-12. An activity node u starts and finishes before (LB_i).

Constraints (25) and (26) set to one the variable $bn_{ui} = 1$ if the node u finishes its activity $(x_u + p_u)$ before the night $(x_u + p_u \leq LB_i)$.

$$M \times bn_{ui} \geq LB_i - x_u$$
$$\forall\, u = 1,..,|U|; \forall\, i = 1,..,|I| \qquad (25)$$
$$M \times [1 - bn_{ui}] \geq x_u + p_u - LB_i$$
$$\forall\, u = 1,..,|U|; \forall\, i = 1,..,|I| \qquad (26)$$

If $x_u + p_u < LB_i$ then $x_u < LB_i$ and the constraint (25) enforces that $bn_{ui} = 1$ since $x_u < LB_i$.

Constraints (27)-(28) enforce activities nodes u to be processed exclusively after (UB_i). Hence, an activity has to start and finish after the upper bound of the night interval i (UB_i).

Constraint (27) enforces the variable an_{ui} to take the value of one, if the activity node u finishes its process after the night interval $x_u + p_u \geq UB_i$.

$$M \times an_{ui} \geq x_u + p_u - UB_i$$
$$\forall\, u = 1,..,|U|; \forall\, i = 1,..,|I| \qquad (27)$$

When $an_{ui} = 1$ then constraint (28) guarantee the condition $x_u \geq UB_i$.

$$M \times [1 - an_{ui}] \geq UB_i - x_u$$
$$\forall\, u = 1,..,|U|; \forall\, i = 1,..,|I| \qquad (28)$$

By consequence, if the activity starts its process after the night interval i $(an_{ui} = 1)$, then the activity has to start$x_u \geq UB_i$ (C28) and finish $x_u + p_u \geq UB_i$ (C27) after the night interval i. See Figure 3-13.

Figure 3-13. An activity node u starts and finishes after (UB_i).

There is night work when three conditions hold. First, there is a process time greater than cero $(p_k > 0)$. Second, an activity starts the process after LB_i meaning $bn_{ki} = 0$ $(x_u \geq LB_i)$ and third, the activity finishes before UB_i, which means $an_{ki} = 0$ $(x_u + p_u \leq UB_i)$. (Figure 3-14).

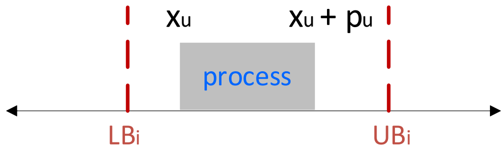


Figure 3-14. An activity node u performed during the night.

Constraint (29) guarantees the conditions for night work. It sets to one the variable nwn_{uv} when between nodes u, v there is at least one working node $k \in [u, v]$ performing some process $(p_k > 0)$ during a night interval $(LB_i \leq x_u + p_u \leq UB_i)$.

$$nwn_{uv} \geq \frac{p_k}{cp_{a(k)}} - bn_{ki} - an_{ki}$$

$$\forall\, u, v = 1, .., |U|; \forall\, k \in [u, v]; \tag{29}$$

$$cp_{a(k)} > 0; \forall\, i = 1, .., |I|$$

Constraints (30) and (31) determines if between the starting time of node u and the finishing time of node v there are $DDU (= 24h)$ hours or less, setting $leq24stEnd_{uv}$ equal to one. As Figure 3-15 Depicts.

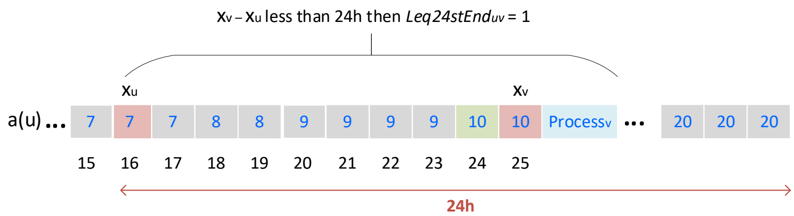


Figure 3-15. Elapsed time between the starting time of activity u and the finishing time of activity v less than DDU hours.

$$M \times leq24stEnd_{uv} > 24 - [x_v + p_v - x_u]$$

$$\forall\, u = 1, .., |U|; u \leq v \tag{30}$$

$$M \times [1 - leq24stEnd_{uv}] \geq [x_v + p_v - x_u] - 24$$

$$\forall\, u = 1, .., |U|; u \leq v \tag{31}$$

Constraints (32) set to one the variable $leq24stSt_{uv}$ if the time between the starting times of nodes u and v is less than $DDU (= 24h)$ hours. Figure 3-16 presents this case.

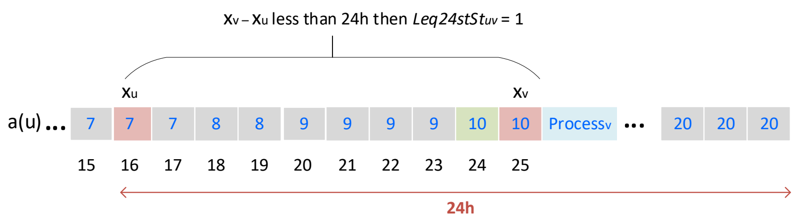


Figure 3-16. Elapsed time between the starting times of activities u and v less than DDU hours.

$$M \times leq24stSt_{uv} > 24 - [x_v - x_u]$$
$$\forall\, u = 1,..,|U|; u \leq v \qquad (32)$$

Constraints (33) enforce the working time between u and v to be less than MWN ($= 10h$) hours, if between the starting time of u and the finishing time of v some night work ($nwn_{uv} = 1$) has been performed and there are less than DDU hours of elapsed time between the starting time of u and the finishing time of v ($leq24stEnd_{uv} = 1$).

$$wo_{uv} \leq MNW + M \times [2 - leq24stEnd_{uv} - nwn_{uv}]$$
$$\forall\, u = 1,..,|U|;\, u \leq v \qquad (33)$$

Constraint (34) is the complement of constraint (33). The DDU hours of elapsed time between the starting time of activity u and the finishing time of activity v could finish while processing activity v, i.e. $x_v + p_v - x_u \geq DDU$ and $x_v - x_u < DDU$. Therefore, the working time between the end of the DDU hours period and the finishing time of the activity v must be removed from the working time between nodes u and v as Figure 3-17 depicts.

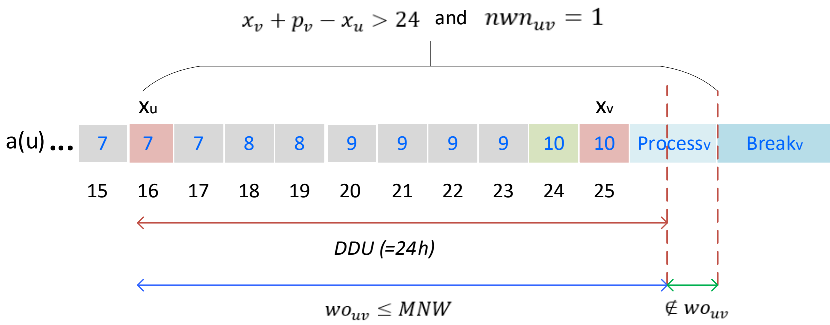


Figure 3-17. The period of DDU hours finishes in the middle of activity v.

Hence, if there is night work performed between activities u and v and the $DDU (= 24h)$ hours period finishes in the middle of the activity v (meaning that $leq24stSt_{uv} = 1$ and $leq24stEnd_{uv} = 0$). Constraints (34) guarantees the working time during the DDU hours between u and v is less than $MNW (= 10h)$ hours. Note that $x_v + p_v - x_u - 24$ is the amount of working time of process v out of the interval of DDU hours.

$$wo_{uv} - [x_v + p_v - x_u - 24] \leq MNW + M \times [2 - nwn_{uv} - leq24stSt_{uv} + leq24stEnd_{uv}]$$
$$\forall\, u, v = 1,..,|U|; u \leq v \qquad (34)$$

The running times of the foregoing formulation highly depend on the number of nodes assigned to each activity. That is to say, with less number of nodes better running times, albeit optimality could be lost. Overall, computational experiments show that the MILP formulation is too slow, thus it is necessary to develop tailored solution methods that guaranteed a better performance.

3.2.3 Assumptions and improvements

Following the EU rules, we made some assumptions. Each activity can be split into an unlimited number of pieces of any duration. Several breaks or rests can be processed without any working time in between. Also, break and rest periods durations are not restricted to a limited set of values.

We compute an upper bound on the number of nodes per activity depending on its type (service or transport) and duration. If a driving activity is less than 4.5h then optimality requires 1 split of the activity for a rest or no split and 1 additional split for the night, in some cases to cover the night it is necessary to schedule more than one rest or break consecutively. It is easy to state that any break can be scheduled before or after the activity without loss of generality. Similar reasoning gives Table 3-1 for different values of service and driving activities. The number of nodes is the number of nodes for split plus one. For long activities, some split nodes model rest and break for a shift included in the activity.

Table 3-1. Maximum number of nodes by type of activity and duration

Activity	*Duration*	*Maximum number of nodes*
Service	$< 6h$	3
	$< 13h$	4
	Otherwise	$\frac{cp_a}{3} + 3$
Transport	$< 4.5h$	3
	$< 9h$	4
	Otherwise	$\frac{cp_a}{3} + 3$

We identified the longest possible sequence of breaks in an optimal solution: a break $(< 9h)$ → a rest $(\geq 9h)$ → the first part of split break $(3h)$ → the first part of split break $(< 45min)$. The first break is imposed by respite at of the first shift. The two last breaks may initiate split rests and breaks. We can show that such series of breaks and rest do not have to be scheduled in optimal solution only at the beginning of an activity. Therefore, the number of nodes per activity is increased by 3 and the processing times of the three first nodes of an activity are set to zero.

For R5 the cumulated duration of break can be reached by two breaks of 20 and 25 minutes. Therefore the duration of break could be optimally set with values different than lower bounds of break durations. An additional continuous variable D_u is ranged by break duration interval of the type of break scheduled at node u. However, minimal respite durations to satisfy constraints usually provide optimal solutions; one example (Instance 41) has been built where optimality is not reach in this case.

Finally, the number of constraints is reduced by limiting the pairs of nodes u and v in the domain of the constraints for which the working or driving time between their activities ensures that the constraint does not apply. For instance, constraint (6) should exclude the nodes for which the cumulated driving time between $a(u) + 1$ and $a(v) - 1$ is greater than 4.5h.

3.3 Discrete label setting algorithm

3.3.1 Principles

Since the performance of the MILP model presented in section 4 is directly related with the number of nodes per activity, it is necessary to develop an alternative method to solve the problem. Among different solutions methods, label setting algorithms are one of the most used methods in the literature to solve the TDSP. Developing a label setting algorithm that considers the night working rule entails two challenges. First, updating the resources in a sliding 24h time window. Second, designing efficient dominance rules to achieve optimality in a competitive computational time.

When using a planning horizon of one week most of the rules apply during a shift, thus the resources related to them reset their values to zero when a shift ends, that is to say, when a rest/break of 9h or more takes place. However, this is not the case for the resources related to the night constraint. As Figure 3-18 depicts, the processing time continuously changes at each 24h interval, meaning that if the 24h interval is shifted by one minute to the right, the processing time of the interval changes depending if there is process at the beginning and at the end of the interval. Similarly, the condition whether or not night working time is performed during the last 24h is updated for each 24h time window.

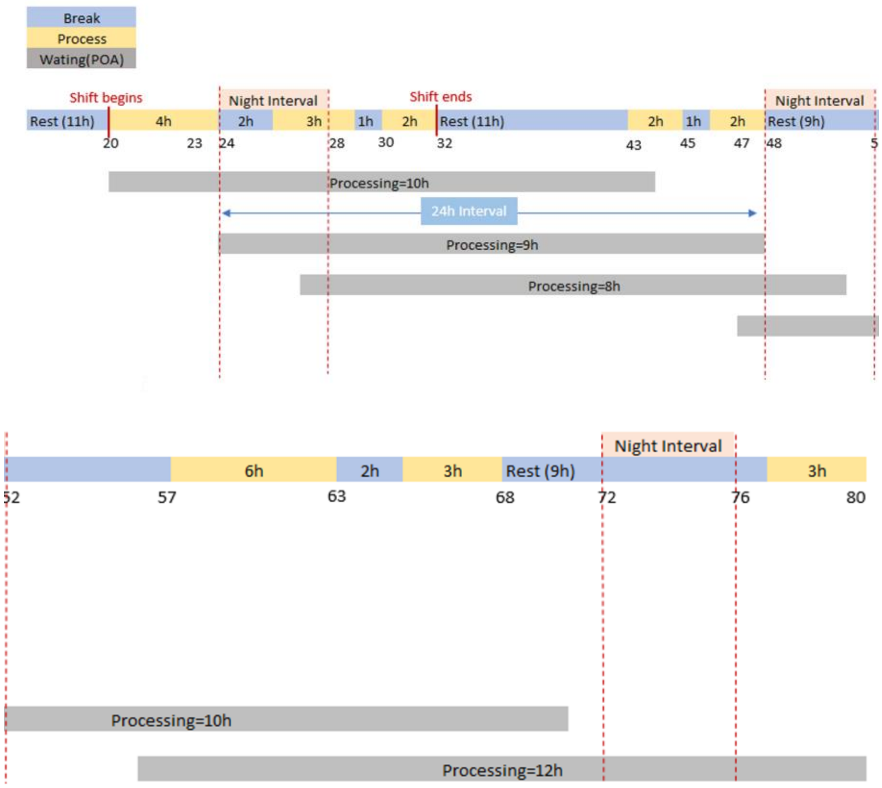


Figure 3-18. Night working constraint.

During the extension process, several partial solutions are generated, in order to reduce this number and improve the computational time performance it is necessary to develop efficient dominance rules. In particular, when two partial schedules are compared at a given point in time, due to the night working constraint the resource usage level is not enough to determine if one schedule dominates over another. Figure 3-19 shows two partial schedules for which the night working constraint applies and both of them have the same resource level consumption, including the processing time over the 24h period. Even if they have the same resource usage level, any extension that increases the processing time from *schedule a* is infeasible, since the total working time during the last 24h is greater than 10h, thus any extension from *label a* is dominated by an extension from *label b*. Therefore, not only the quantity of the processing time along the 24h interval, but its distribution along the interval determine the dominance relationship between labels.

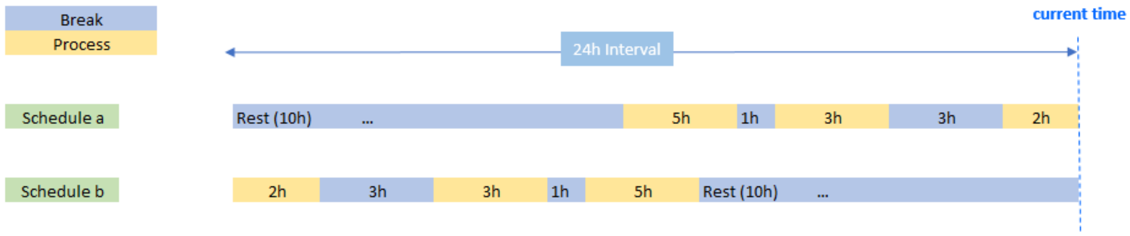


Figure 3-19. Comparing two partial schedules.

A discrete label setting algorithm is presented in order to solve the Truck Drivers Scheduling Problem that minimizes the completion time of a given sequence of customers, while considering all the rules from the EC social legislation for a weekly planning period, including the night working constraint. In contrast to previous contributions, which focus on finding a feasible solution, our break scheduling method provides the optimal solution if a solution exists. In addition, breaks could take place at any moment, which means service and driving activities are pre-emptive.

In addition, let R to be the number of resources related with each driver activity that are used to represent a driver state. Figure 3-1 gives an example of the setup of a sequence to evaluate which has two customers and includes the depot at both, first and last position. Finally, the objective is to find the optimal schedule of breaks/rests complying with the set of rules from the EC social legislation presented in section 3, which minimizes the completion time of the sequence.

In this sense, a route σ is a path defined between a set of activities or nodes A, where 0 is the service activity at the depot and n is a dummy activity with zero duration at the final depot. And a set of arcs T that is composed by the processing times p_i related with each activity i.

Definition 1. A label is a partial path or schedule λ_i, which is a sequence of states or operations: driving, service, breaks/rests and POA, that have been developed between the origin node 0 and the activity node i.

When searching for a path or a schedule of driver operations for the initial sequence of customers σ, partial schedules from 0 and an activity node i are extended to create new schedules. Every label resident at activity node i stores all resources variables at this node for its path.

Definition 2. A schedule λ_i from the origin node 0 to the activity node i represents a resource vector $\lambda_i = \{l_0, l_1, l_2, \ldots, l_{|R|}\}$ related to the quantity of each resource $r \in R$ used by the schedule resident at the node i.

The algorithm has one general assumption, in which there exists an interval of size δ such that each break duration, earliest starting time, latest completion time, service time and driving time between customers is a multiple of δ. We can state that the label setting algorithm will find the optimal solution when using this δ. There are two types of extensions, process or idle each of them followed by a break. In the first case, process (driving/service), the length of the processing time is determined by the maximum process time until a rule could be broken, the activity is finished or the beginning of the night is reached. An extension with idle is used to postpone the beginning of the shift or the beginning of an activity, if the current time is before the earliest starting time of the next customer of the sequence (waiting). Each extension with idle has a length duration of δ. After each type of extension, one label for each type of break duration is generated.

The algorithm starts at node 0 using an initial solution $\lambda_0 \leftarrow 0$. At each node $i \in A$, it determines a set of feasible solutions or labels (S, i) with a minimum completion time. Hence, for each label $\lambda_i \in (S, i)$ until $(S, i) \neq \emptyset$, the procedure computes a set of feasible labels F_i by calling the *extend* function; depending on the state of the activity, in process or finished, the set of labels result of the extension procedure are saved at their respective set of solutions. Finally, the current label λ_i is removed from the set of solutions at node i. The process repeats until it reaches node n at the final depot. Procedure 3-1 presents the description of the algorithm.

Procedure 3-1. DiscreteLabelSetting

```
procedure DiscreteLabelSetting
  input:
  δ: Size of the idle extension;
  output:
  λn ∈ (S,n) with minimal completion time;
λ0 ← 0;
(S,0) ← (S,0) ∪ {λ0};
foreach i ∈ A \ n do
| While (S,i) ≠ ∅ do
| | Label λi ∈ (S,i);    //Label to expand.
| | Fi = extend(λi,δ);    //Set of feasible labels after the extension.
| | if activity finished == true then
| | | (S,i+1) ← (S,i+1) ∪ Fi;    //Label stored at the next activity.
| | else
| | | (S,i) ← (S,i) ∪ Fi;    //Label stored at the current activity.
| | end
| | (S,i) ← (S,i)\λi;    //Label to expand.
| end
end
```

Procedure 3-1. Discrete label setting algorithm.

3.3.2 Basic resources of a label

The following are the set of resources R used to describe a schedule λ_i resident at node i.

$l_{terminal\_node}$, finishing node at which the label is attached.

l_{ER}, earliest readiness time at terminal node activity.

l_q, cumulative processing time of the activity.

l_{LStoER}, latest start to earliest readiness. Starting time of the current shift, i.e. the starting time of the first break or process after a rest.

l_{DrPe}, current driving time without a full break.

l_{DrSh}, current driving time during the shift.

l_{WkCo}, current working time without a break of at least 15 minutes.

l_{WkSh}, current working time during the shift.

l_{BrDu}, break duration at the current node.

l_{BrSh}, cumulated break time during the shift.

$l_{BrSplDr}$, True (Y), first part of a split driving break has been scheduled. False (N), otherwise.

l_{ReSpl}, True (Y), first part of a split daily rest has been scheduled. False (N), otherwise.

l_p, process [Service/Driving] time of the label. Used to retrieve the optimal solution at the end of the schedule.

l_{DrExt}, number of driving extensions of one hour during the week [0, 1, 2].

l_{ReRed}, number of reduced daily rest during the week [0, 1, 2, 3].

l_{act24}, array with the activities $\in \{1 = process, 0 = idle/break\}$ during the last 24h.

l_{wk24}, cumulated working time [driving/working activities] during the last 24h.

l_{fwrd}, amount of working time at the beginning of the 24h interval until a break/rest is scheduled.

l_{lnwp}, starting time of the last period where night work was performed.

Attribute l_{act24} is an array that keeps trace of the activities that have been performed during the last 24h. It is a binary sequence, where one represents a driving or service activity and zero stands for idle or break time. The information is stored from the most recent event to the last in the 24h time window, moreover, the vector is divided in slots of δ size, as Figure 3-20 shows.

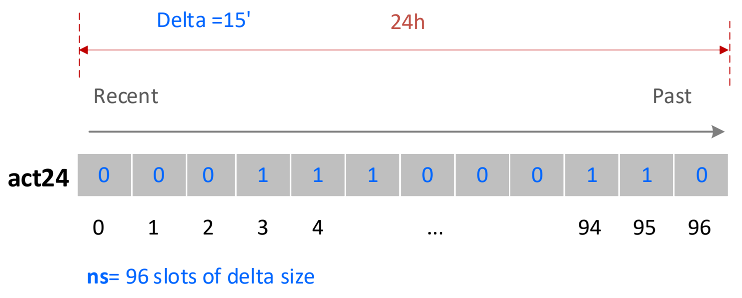


Figure 3-20. Array l_{act24}.

This attribute is used in the dominance procedure and to compute l_{wk24} which stores the cumulated working time during the last 24h.

The label keeps trace of the working time at the beginning of the 24h interval using the resource l_{fwrd}. This resource increases the size of the processing time of the extension if the night constraint applies and the schedule has reached the limit of 10 hours of working time during the last 24 hours period (l_{wk24}). See Figure 3-21.

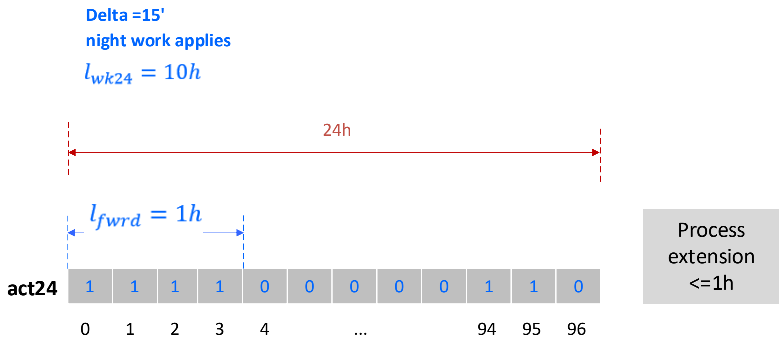


Figure 3-21. Resource l_{fwrd}.

3.3.3 Extension function

For each label λ_i saved at the list of labels (S, i) the procedure evaluates two options for the extension: process and idle, each of them followed by a break from the set of breaks, as Figure 3-22 depicts. The extension compose by only a break is the particular case when $p = 0$. For instance, this type of extension is required when an activity has zero duration or the beginning of a night interval has been reached and the algorithm explore the possibility of do not work during the night. Extensions with idle time have a fixed duration of δ units of time.

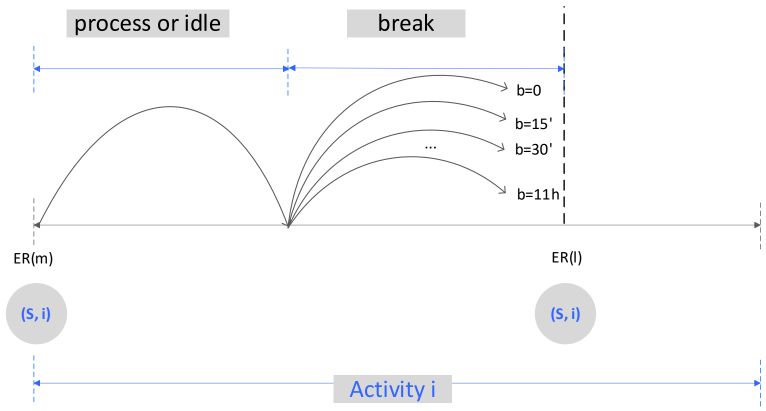


Figure 3-22. Extension options.

The algorithm computes the processing time of the extensions by calling the function *compute*. In order to keep a feasible schedule the size of the processing time must guarantee that no rule is broken. As in Goel and Irnich [5], since there is not a resource interval for all the resources, a feasible extension restricts the resources to be less or equal to the maximum allowed values imposed by the EC social legislation. Hence, the maximum processing time of a label λ_i until a rule could be broken p_{λ_i}, is defined as:

$$p_{\lambda_i} \coloneqq min\{p_i - l_q, CWT - l_{WkCo}, CDT - l_{DrPe}, RDS, RSR, NIR\}$$

Where,

$p_i - l_q$, remaining time to finish the activity.

$CWT - l_{WkCo}$, working time to schedule a break of at least 15 minutes, due to continuous working time.

$CDT - l_{DrPe}$, remaining driving time for a full break due to continuous driving time.

RDS, driving time until a rest due to daily driving time is compulsory.

If $l_{DrExt} \leq DE$ then $RDS = MDE - l_{DrSh}$, otherwise $RDS = MDR - l_{DrSh}$.

RSR, working time until a rest due to maximum shift duration is compulsory.

If l_{ReSpl} is true or $l_{ReRed} \leq RE$ then $RSR = MWE - [l_{ER} - l_{LStoER}]$, otherwise $RSR = MWR - [l_{ER} - l_{LStoER}]$.

NIR, working time until maximum working time if night working has been performed.

If some night work has been performed during the last 24 hours, then $NIR = MWN + l_{fwrd} - l_{wk24}$.

Considering a feasible label λ_i resident at node i, each type of extension with process (p_{λ_i}) or idle (δ) followed by a type of break b, along the arc $p_i \in T$, is performed according to a set of resource-extension functions $f^r(\lambda_i|b)$, in order to generate a new feasible schedule. First, according to the type of activity: driving, service or idle, we define the processing time due to driving and working as p_d and p_w, respectively.

$$p_d \coloneqq \begin{cases} p_{\lambda_i}, & \text{if driving activity.} \\ 0, & \text{if service activity.} \\ 0, & \text{if idle time.} \end{cases}$$

$$p_w \coloneqq \begin{cases} 0, & \text{if driving activity.} \\ p_{\lambda_i}, & \text{if service activity.} \\ 0, & \text{if idle time.} \end{cases}$$

Second, depending on the type of break b that follows the extension, the resource-extension functions $f^r(\lambda_i|b)$ update the resources values of the extended label λ_i', as Table 3-2 presents. When the extension is done with idle time, the resource l_{ER}' is computed as $l_{ER}' = l_{ER} + \delta + b$; all the other entries in Table 3-2 remain unchanged.

The case of a break of 0.5h appears in two different columns depending on the state of the resource $l_{BrSplDr}$ of label λ_i. If a split break has been previously schedule $(l_{BrSplDr} = True)$, it is a full break and resources of λ_i' are updated according to column 3, otherwise it is the first part of a split break and column 2 is used.

Since the night resources do not depend on the type of break, and they are updated for each sliding 24h time window, they do not explicitly appear on Table 3-2, and devoted resource-extension functions are required. Most of the night resources rely on l_{act24} thus is the first resource-extension function to present by means of one example.

Table 3-2. Resource-extension functions.

	Resource-extension functions $f^r(\lambda_i \vert b)$		
λ_i'	$b \le 0.5h$	$0.5h \le b \le 3h$	$b \ge 9h$
$l'_{terminal\_node}$	i, if $l_q + p_{\lambda_i} < p_i$	i, if $l_q + p_{\lambda_i} < p_i$	i, if $l_q + p_{\lambda_i} < p_i$
	$i + 1$, Otherwise.	$i + 1$, Otherwise.	$i + 1$, Otherwise.
l'_{ER}	$l_{ER} + p_{\lambda_i} + b$	$l_{ER} + p_{\lambda_i} + b$	$l_{ER} + p_{\lambda_i} + b$
l'_q	$l_q + p_{\lambda_i}$, if $l_q + p_{\lambda_i} < p_i$	$l_q + p_{\lambda_i}$, if $l_q + p_{\lambda_i} < p_i$	$l_q + p_{\lambda_i}$, if $l_q + p_{\lambda_i} < p_i$
	0, Otherwise.	0, Otherwise.	0, Otherwise.
l'_{LStoER}	l_{LStoER}	l_{LStoER}	l_{ER}
l'_{DrPe}	$l_{DrPe} + p_d$	0	0
l'_{DrSh}	$l_{DrSh} + p_d$	$l_{DrSh} + p_d$	0
l'_{WkCo}	l_{WkCo}, if $b = 0$	0	0
	0, Otherwise.		
l'_{WkSh}	$l_{WkSh} + p_{\lambda_i}$	$l_{WkSh} + p_{\lambda_i}$	0
l'_{BrDu}	$l_{BrDu} + b$	$l_{BrDu} + b$	0
l'_{BrSh}	$l_{BrSh} + b$	$l_{BrSh} + b$	0
$l'_{BrSplDr}$	False, if $b = 0$	0	0
	True, Otherwise.		
l'_{ReSpl}	l_{ReSpl}	True, if $b = 3h$	0
		l_{ReSpl}, Otherwise.	0
l'_p	p_{λ_i}	p_{λ_i}	p_{λ_i}
l'_{DrExt}	l_{DrExt}	l_{DrExt}	l_{DrExt}+1, if $l'_{DrSh} > 9h$
			l_{DrExt}, Otherwise.
l'_{ReRed}	l_{ReRed}	l_{ReRed}	l_{ReRed}+1, if $l_{ER} + l'_p - l_{LStoER} > 13h$
			l_{ReRed}, Otherwise.

The resource-extension function takes the sequence of the current extension i.e. process/idle plus break, and transforms it into a sequence of 0, 1 assigning this sequence to the first positions in the array l'_{act24}. The rest of the information within it comes from l_{act24} of the "father" label λ_i or the previous/precedent label of the current extension.

Figure 3-23 shows the update process of the attribute l'_{act24} of label λ'_i. In this example, the extension has a process of 15 minutes ($p_{\lambda_i} = 15min$) followed by a break of 45 minutes; since the size of δ is 15 minutes, only the first four positions of the array l_{act24} change. The first three positions are equal to 0, because they are related with the break and position three is equal to 1, since it is a process activity. From positions four to the end of the array, the information remains unchanged.

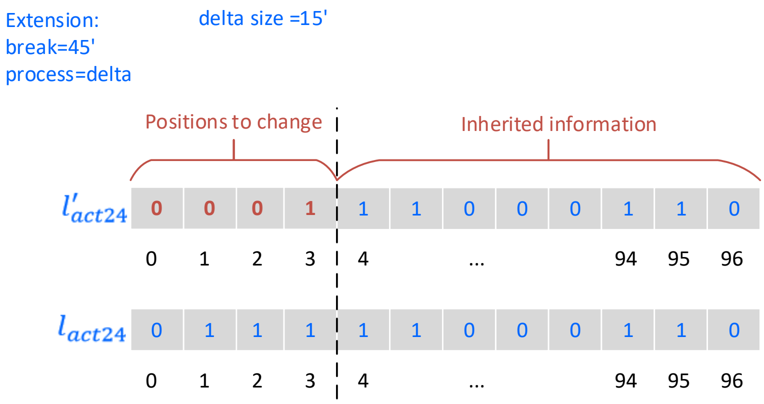


Figure 3-23. Extending attribute l'_{act24}.

Based on the resource l'_{act24} resources l'_{wk24} and l'_{fwrd} are computed. Resource l'_{wk24} is the sum of all entries with value 1 in the vector l'_{act24} multiplied by its size (δ).

$$l'_{wk24} = \delta \sum_{i=0}^{24/\delta} l'_{act24_i}$$

In addition, l'_{fwrd} is the sum of the last positions with value 1 of the array l'_{act24} until a break, an entry with value 0 is found.

$$l'_{fwrd} = \delta \sum_{i=0}^{24/\delta \vee l'_{act24_i}=0} l'_{act24_{[(\frac{24}{\delta})-i]}}$$

Finally, the last night working period of the schedule λ'_i is defined as,

$$l'_{lnwp} := \begin{cases} MIN(l_{ER} + p_{\lambda_i}; UB_i), & LB_i \leq l_{ER} \leq UB_i \text{ and } p_{\lambda_i} > 0. \\ l_{lnwp} & Otherwise \end{cases}$$

In general, if $\delta < 0.25h$ extensions with idle are useful at all times, in order to avoid rules infractions. However, when $\delta \geq 0.25h$ the only case where idle is dominant is at the beginning of the shift, therefore extending with idle is only possible in this case. The case in which the activity process p_i is greater than zero and at the same time both, the process time p_{λ_i} and the break are equal to zero is forbidden in the algorithm; since, this case means an extension without any movement. The extension process adds to the earliest readiness time l_{ER}, the length of the movement [process/idle] and the break duration, in order to find l'_{ER}

A solution is rejected when it breaks the night working constraint, it does not pass the feasibility test or when it is out of the service time windows of a given customer. Only the extensions that are not dominated are included in the set of feasible solutions F_k. Procedure 3-2 presents this procedure.

Procedure 3-2. Extend

```
procedure Extend
  input:
  B: Set of breaks. B = {0, 0.25h, 0.50h, 0.75h, 3h, 9h, 11h}
  m: A label stored in labels(a), which is going to be extended;
  δ: Size of the idle extension;
  output:
  F_k: A Set of valid labels extended from label m;
Compute p;  //As a in section 5.3.
P = {(process, 0), (process, p), (idle, δ)};
foreach p ∈ P/{idle|m.between2Shift == true and δ ≥ 0.25} do
| foreach b ∈ B/{b = 0|p.second = 0 and activity process > 0} do
| | Label l;   //Creating a new label.
| | l.ER ← m.ER + p.second + b;
| | l.lnwp ← m.lnwp;
| | if Binf[k] ≤ m.ER < Bsup[k] and p.second > 0 then
| | | l.lnwp ← m.ER;
| | end
| | update act24;
| | limit24 =MAX(m.ER − 24; 0); //Starting time of the elapsed 24h period.
| | l.wk24=computeWk24(l);
| | if l.lnwp ≥ limit24 and l.wk24 > 10h then insert = false;
| | UpDate(l, m, b, p); //Updating label l attributes.
| | feasible ← feasibility(l, p); //Feasibility test.
| | if feasible == false then b ← b + 1;
| | else
| | | dominate ← dominace(l); //Applying dominance rules on label l.
| | | if dominate == true) then
| | | | b ← b + 1;
| | | else
| | | | F_k ← F_k ∪ {l};
| | | end
| | end
| end
end
```

Procedure 3-2. Function extend.

3.3.4 Feasibility

During the extension process, every new label goes through a feasibility test. In the procedure, a feasible solution should satisfy rules R1-R10 listed in section 3. It starts checking the rules on continuous driving time and the total driving time over the shift. After, rules on continuous working time and maximum shift duration are verified, the procedure continues with rules on compulsory minimal break duration according to working time during the shift. Finally, the feasibility process finishes checking rules on driving time extensions and daily rest reduction. Procedure 3-3 depicts the procedure.

Procedure 3-3. Feasibility

procedure feasibility
input:
l: Label extended;
p: Processing time;
output:
$state$: true, if a solution is feasible. false, otherwise;

$state \leftarrow true$;
if $l.DrPe > 4.5h$ **then** $state \leftarrow false$;
if $l.DrSh > 10h$ **then** $state \leftarrow false$;
else if $l.DrSh > 9h$ **and** $l.DrExt > 1$ **then** $state \leftarrow false$;
if $l.WkCo > 6h$ **then** $state \leftarrow false$;
if $l.WkSh > 15h$ **then** $state \leftarrow false$;
else if $l.WkSh > 13h$ **and** $l.ReRed > 2$ **then** $state \leftarrow false$;
if $l.BrDu > B9h$ **then**
| **if** $l.WkSh > 9h$ **and** $l.BrSh < 45min$ **then** $state \leftarrow false$;
| **else if** $6h \leq l.WkSh \leq 9h$ **and** $l.BrSh < 30min$ **then** $state \leftarrow false$;
end
if $l.DrExt > 2$ **then** $state \leftarrow false$;
if $l.ReRed > 3$ **then** $state \leftarrow false$;

Procedure 3-3. Function feasibility.

3.3.5 Dominance

Procedure 3-4 describes the dominance procedure.

Procedure 3-4. dominance

procedure dominance
input:
λ_1: First label to compare;
λ_2: Second label to compare;
output:
dom: True, if λ_1 dominates λ_2. False, otherwise;

$dom \leftarrow True$;
if $\lambda_1.q < \lambda_2.q$ **then** $dom \leftarrow false$;
if $\lambda_1.BrSh < B45m$ **and** $\lambda_1.BrSh < \lambda_2.BrSh$ **then** $dom \leftarrow false$;
if $\lambda_1.DrPe > \lambda_2.DrPe$ **then** $dom \leftarrow false$;
if $\lambda_1.WkCo > \lambda_2.WkCo$ **then** $dom \leftarrow false$;
if $\lambda_1.WkSh > \lambda_2.WkSh$ **then** $dom \leftarrow false$;
if $\lambda_1.DrSh > \lambda_2.DrSh$ **then** $dom \leftarrow false$;
if $\lambda_1.ReRed > \lambda_2.ReRed$ **then** $dom \leftarrow false$;
if $\lambda_1.DrExt > \lambda_2.DrExt$ **then** $dom \leftarrow false$;
if $\lambda_1.BrSplDr < \lambda_2.BrSplDr$ **then** $dom \leftarrow false$;
if $\lambda_1.ReSpl < \lambda_2.ReSpl$ **then** $dom \leftarrow false$;
if $\lambda_1.LStoER < \lambda_2.LStoER$ **then** $dom \leftarrow false$;
$limit24 = \text{MAX}(\lambda_1.ER - 24; 0)$;
if $\lambda_1.lnwp > limit24$ **and** $\lambda_1.lnwp > \lambda_2.lnwp$ **then** $dom \leftarrow false$;
$Sum\lambda_1 = Sum\lambda_1 \leftarrow 0$;
for $u = 1$ ***to*** ns **do**
| $Sum\lambda_1 = Sum\lambda_1 + \lambda_1.act24[u]$;
| $Sum\lambda_2 = Sum\lambda_2 + \lambda_2.act24[u]$;
| **if** $Sum\lambda_1 > Sum\lambda_2$ **then** $dom \leftarrow false$;
end

Procedure 3-4. Function dominance.

Throughout the execution of the extension process, only non-dominated solutions should be considered, therefore, after verifying their feasibility, a dominance procedure that compares two labels $\{\lambda_1, \lambda_2\} \in (S, i)$, determines if label λ_1 dominates label λ_2. As in [8], the comparison of the attributes is done one by one over the set of labels at the current node i. Let, R_i^1 and R_i^2 the attributes vectors of labels λ_1 and λ_2, respectively. Then, label λ_1 dominates λ_2 if $R_i^1 \leq R_i^2 \; \forall \; i \in A$. In the case of the resource $BrSh$, an additional condition $\lambda_1.BrSh < B45m$.

3.4 Computational experiments

3.4.1 Instances definition

This section presents some computational results on two set of instances PGLT and GOEL. Peña-arenas et al. [9] proposed the set of instances PGLT, which has 40 instances and the size of the instances varies between 3 and 15 customers. In addition, a second set GOEL with 157 instances is used. This set is the result of deriving one instance from each route of the solutions of the routing problem provided by Goel (2018). In this sense, we set the parameters of the instances as multiples of δ and generate one instance from each route of the set of solutions found by Goel (2018).

All the experiments have been achieved on an Intel® Core™i5-8400 at 2.81 GHz under Windows 10, using C++ and Gurobi 8.1.1. A limit on the computation time has been imposed after two hours. The set parameters of the instances are multiples of a δ size of 15 minutes. Additionally, the number of nodes per activity used for the MILP model are computed based on the LS solution.

3.4.2 MILP solutions

Tables 3-3 and 3-4 present the results of the MILP model for PGLT and GOEL instances. For PGLT instances, only Test_15 did not finish under the time limit of 2 hours, although the GAP with respect to the best bound is 1%. The average running time of the MILP model under this set of instances is 183.97 seconds, including the two hours from Test_15. In the case of GOEL instances, the average running time per instance is 124.33 seconds. In addition, instances TDS_R110_1, TDS_R209_1 and TDS_R209_2 reach the computational time limit of 7200 seconds, with a relative GAP of less than 0.8%.

Table 3-3. Results MILP for instances PGLT.

Instance	Completion (min)	Time (s)
TEST_1	1635	0.19
TEST_2	1260	0.11
TEST_3	1410	0.12
TEST_4	2040	0.17
TEST_5	2955	4.33
TEST_6	2910	16.62
TEST_7	3720	15.82
TEST_8	3720	11.73
TEST_9	0	0.01
TEST_10	0	0.00
TEST_11	0	0.00

TEST_12	0	0.01
TEST_13	2250	1.52
TEST_14	3720	7.38
TEST_15	5220	>7200.00
TEST_16	0	0.01
TEST_17	0	0.00
TEST_18	2910	21.51
TEST_19	5580	9.35
TEST_20	3780	0.82
TEST_21	3210	2.74
TEST_22	5610	5.47
TEST_23	5820	3.13
TEST_24	2820	3.74
TEST_25	3795	7.07
TEST_26	2865	4.58
TEST_27	3105	9.87
TEST_28	4290	22.51
TEST_29	1575	0.24
TEST_30	1485	0.11
TEST_31	2280	1.54
TEST_32	2280	2.40
TEST_33	2280	1.71
TEST_34	2400	0.59
TEST_35	1545	0.16
TEST_36	1665	0.34
TEST_37	0	0.01
TEST_38	2100	0.24
TEST_39	3000	0.71
TEST_40	3345	0.29
Average		183.97

Table 3-4. Results MILP for instances GOEL.

Instance	**Completion (min)**	**Time(s)**
TDS_C101_1	7095	6
TDS_C101_2	5625	12
TDS_C101_3	7185	24
TDS_C102_1	6465	5
TDS_C102_2	5340	62
TDS_C102_3	6180	17
TDS_C103_1	6465	4
TDS_C103_2	4530	6
TDS_C103_3	6180	17
TDS_C104_1	5925	1
TDS_C104_2	4650	10
TDS_C104_3	6180	9
TDS_C105_1	5385	10
TDS_C105_2	6990	35
TDS_C105_3	6915	4
TDS_C106_1	5595	6
TDS_C106_2	7125	3
TDS_C106_3	7020	29
TDS_C107_1	5250	27
TDS_C107_2	6660	5
TDS_C107_3	6750	40
TDS_C108_1	5460	7
TDS_C108_2	6600	7

TDS_C108_3	6555	1
TDS_C109_1	6030	2
TDS_C109_2	6120	23
TDS_C109_3	4680	10
TDS_C201_1	4140	1
TDS_C201_2	8505	7
TDS_C202_1	4140	5
TDS_C202_2	8505	8
TDS_C203_1	4140	5
TDS_C203_2	8505	17
TDS_C204_1	4140	5
TDS_C204_2	8475	37
TDS_C205_1	3930	1
TDS_C205_2	8415	27
TDS_C206_1	3870	1
TDS_C206_2	8235	36
TDS_C207_1	4095	1
TDS_C207_2	7620	12
TDS_C208_1	3525	1
TDS_C208_2	8580	123
TDS_R101_1	6885	7
TDS_R101_2	6525	4
TDS_R101_3	6825	4
TDS_R101_4	7470	4
TDS_R101_5	6690	5
TDS_R102_1	6945	8
TDS_R102_2	6690	3
TDS_R102_3	7590	4
TDS_R102_4	7110	12
TDS_R103_1	6690	4
TDS_R103_2	7995	62
TDS_R103_3	8445	6
TDS_R104_1	6690	3
TDS_R104_2	4785	2
TDS_R104_3	8445	4
TDS_R105_1	6630	17
TDS_R105_2	7245	8
TDS_R105_3	4710	3
TDS_R105_4	6690	92
TDS_R106_1	5865	6
TDS_R106_2	8055	45
TDS_R106_3	6690	2
TDS_R107_1	5535	4
TDS_R107_2	7305	6
TDS_R107_3	6795	30
TDS_R108_1	6315	5
TDS_R108_2	8085	49
TDS_R109_1	6615	29
TDS_R109_2	5880	3
TDS_R109_3	6600	33
TDS_R110_1	6870	>7200
TDS_R110_2	6045	5
TDS_R111_1	6720	11
TDS_R111_2	5565	49
TDS_R111_3	8055	286
TDS_R201_1	4590	3
TDS_R201_2	6885	7
TDS_R201_3	7440	1

TDS_R201_4	8220	52
TDS_R202_1	7500	4
TDS_R202_2	6840	8
TDS_R202_3	8310	15
TDS_R202_4	4590	4
TDS_R203_1	9105	8
TDS_R203_2	8280	1
TDS_R203_3	6030	3
TDS_R204_1	6675	4
TDS_R204_2	8160	9
TDS_R205_1	6990	4
TDS_R205_2	6465	20
TDS_R205_3	4500	4
TDS_R205_4	7425	6
TDS_R206_1	8175	9
TDS_R206_2	7050	3
TDS_R206_3	6750	1719
TDS_R207_1	7380	2
TDS_R207_2	5235	4
TDS_R207_3	8010	792
TDS_R209_1	6840	>7200
TDS_R209_2	6585	>7200
TDS_R209_3	6720	34
TDS_R210_1	5610	7
TDS_R210_2	8685	37
TDS_R210_3	7110	78
TDS_R211_1	5580	16
TDS_R211_2	7365	41
TDS_RC101_1	6645	18
TDS_RC101_2	6975	13
TDS_RC101_3	7245	10
TDS_RC102_1	7245	10
TDS_RC102_2	6975	12
TDS_RC102_3	6975	19
TDS_RC103_1	7110	22
TDS_RC103_2	7245	10
TDS_RC103_3	6975	12
TDS_RC104_1	7350	9
TDS_RC104_2	6975	12
TDS_RC104_3	6240	10
TDS_RC105_1	7395	8
TDS_RC105_2	7305	39
TDS_RC105_3	7350	4
TDS_RC106_1	5865	3
TDS_RC106_2	6675	3
TDS_RC106_3	5985	12
TDS_RC107_1	5655	18
TDS_RC107_2	4365	4
TDS_RC107_3	5580	17
TDS_RC108_1	6225	7
TDS_RC108_2	5760	7
TDS_RC108_3	5595	8
TDS_RC201_1	7230	6
TDS_RC201_2	8670	9
TDS_RC201_3	7425	8
TDS_RC202_1	8670	12
TDS_RC202_2	7425	9
TDS_RC202_3	7410	5

TDS_RC203_1	8070	84
TDS_RC203_2	8670	19
TDS_RC203_3	7410	5
TDS_RC204_1	8745	5
TDS_RC204_2	7410	17
TDS_RC204_3	7005	4
TDS_RC205_1	8670	11
TDS_RC205_2	7470	29
TDS_RC205_3	7110	3
TDS_RC206_1	7095	16
TDS_RC206_2	7170	6
TDS_RC206_3	7380	6
TDS_RC207_1	6735	33
TDS_RC207_2	6600	4
TDS_RC207_3	5565	14
TDS_RC208_1	5895	10
TDS_RC208_2	5745	2
TDS_RC208_3	5745	6
Average		124.33

3.4.3 Label setting solutions

Table 3-5 and 3-6 reports the results of the LS for PGLT and GOEL instances, respectively. In both tables, columns two and three present the completion and the running times per instance. For PGLT instances, the average running time per instance is 6.51 seconds and seven instances become infeasible due to the night constraint. Whereas, 11.85, 169.13 and 1 seconds are the average, the maximum and the minimum running times for GOEL instances.

Table 3-5. Results LS for instances PGLT.

	LS (delta=15min)	
Instance	**Completion (min)**	**Time (s)**
TEST_1	1635	2.22
TEST_2	1260	0.07
TEST_3	1410	0.11
TEST_4	2040	0.05
TEST_5	2955	17.03
TEST_6	2910	4.67
TEST_7	3720	11.60
TEST_8	3720	11.56
TEST_9	0	0.08
TEST_10	0	0.08
TEST_11	0	0.04
TEST_12	0	0.03
TEST_13	2250	4.25
TEST_14	3720	20.42
TEST_15	5220	50.98
TEST_16	0	0.71
TEST_17	0	0.72
TEST_18	2910	4.70
TEST_19	5580	5.03
TEST_20	3780	5.80

TEST_21	3210	9.19
TEST_22	5610	6.83
TEST_23	5820	7.68
TEST_24	2820	12.38
TEST_25	3795	15.28
TEST_26	2865	9.59
TEST_27	3105	8.43
TEST_28	4290	19.88
TEST_29	1575	2.92
TEST_30	1485	0.91
TEST_31	2280	3.40
TEST_32	2280	3.03
TEST_33	2280	5.46
TEST_34	2400	1.55
TEST_35	1545	0.91
TEST_36	1665	1.99
TEST_37	0	0.41
TEST_38	2100	0.25
TEST_39	3000	1.11
TEST_40	3345	9.06
Average		6.51

Table 3-6. Results LS for instances GOEL.

Instance	**LS (delta=15min) Completion (min)**	**Time (s)**
TDS_C101_1	7095	1.03
TDS_C101_2	5625	4.79
TDS_C101_3	7185	1.96
TDS_C102_1	6465	1.07
TDS_C102_2	5340	6.52
TDS_C102_3	6180	4.06
TDS_C103_1	6465	1.33
TDS_C103_2	4530	25.59
TDS_C103_3	6180	4.06
TDS_C104_1	5925	1.50
TDS_C104_2	4650	17.68
TDS_C104_3	6180	1.69
TDS_C105_1	5385	7.10
TDS_C105_2	6990	3.33
TDS_C105_3	6915	1.50
TDS_C106_1	5595	4.79
TDS_C106_2	7125	0.85
TDS_C106_3	7020	1.74
TDS_C107_1	5250	10.29
TDS_C107_2	6660	1.60
TDS_C107_3	6750	2.88
TDS_C108_1	5460	9.02
TDS_C108_2	6600	2.16
TDS_C108_3	6555	1.58
TDS_C109_1	6030	1.77
TDS_C109_2	6120	2.43
TDS_C109_3	4680	8.57
TDS_C201_1	4140	5.97
TDS_C201_2	8505	6.65
TDS_C202_1	4140	3.36

TDS_C202_2	8505	10.63
TDS_C203_1	4140	3.35
TDS_C203_2	8505	10.70
TDS_C204_1	4140	3.35
TDS_C204_2	8475	14.59
TDS_C205_1	3930	8.40
TDS_C205_2	8415	131.02
TDS_C206_1	3870	5.67
TDS_C206_2	8235	169.13
TDS_C207_1	4095	5.55
TDS_C207_2	7620	10.59
TDS_C208_1	3525	4.82
TDS_C208_2	8580	10.77
TDS_R101_1	6885	2.76
TDS_R101_2	6525	2.19
TDS_R101_3	6825	0.56
TDS_R101_4	7470	4.25
TDS_R101_5	6690	1.04
TDS_R102_1	6945	3.37
TDS_R102_2	6690	1.72
TDS_R102_3	7590	3.91
TDS_R102_4	7110	2.66
TDS_R103_1	6690	2.58
TDS_R103_2	7995	5.34
TDS_R103_3	8445	3.06
TDS_R104_1	6690	4.72
TDS_R104_2	4785	8.70
TDS_R104_3	8445	3.32
TDS_R105_1	6630	3.00
TDS_R105_2	7245	4.95
TDS_R105_3	4710	11.91
TDS_R105_4	6690	2.43
TDS_R106_1	5865	51.42
TDS_R106_2	8055	9.54
TDS_R106_3	6690	5.02
TDS_R107_1	5535	5.75
TDS_R107_2	7305	7.45
TDS_R107_3	6795	18.66
TDS_R108_1	6315	9.78
TDS_R108_2	8085	7.73
TDS_R109_1	6615	6.78
TDS_R109_2	5880	28.74
TDS_R109_3	6600	6.97
TDS_R110_1	6870	41.45
TDS_R110_2	6045	69.20
TDS_R111_1	6720	6.77
TDS_R111_2	5565	27.31
TDS_R111_3	8055	9.17
TDS_R201_1	4590	1.44
TDS_R201_2	6885	2.73
TDS_R201_3	7440	1.23
TDS_R201_4	8220	25.18
TDS_R202_1	7500	3.14
TDS_R202_2	6840	2.15
TDS_R202_3	8310	5.81
TDS_R202_4	4590	2.08
TDS_R203_1	9105	4.49
TDS_R203_2	8280	0.74

TDS_R203_3	6030	4.24
TDS_R204_1	6675	10.15
TDS_R204_2	8160	21.53
TDS_R205_1	6990	3.76
TDS_R205_2	6465	43.01
TDS_R205_3	4500	3.82
TDS_R205_4	7425	4.01
TDS_R206_1	8175	4.82
TDS_R206_2	7050	67.17
TDS_R206_3	6750	5.29
TDS_R207_1	7380	6.50
TDS_R207_2	5235	6.29
TDS_R207_3	8010	12.74
TDS_R209_1	6840	6.71
TDS_R209_2	6585	2.81
TDS_R209_3	6720	45.70
TDS_R210_1	5610	6.67
TDS_R210_2	8685	6.34
TDS_R210_3	7110	33.63
TDS_R211_1	5580	129.86
TDS_R211_2	7365	67.80
TDS_RC101_1	6645	4.72
TDS_RC101_2	6975	20.73
TDS_RC101_3	7245	10.60
TDS_RC102_1	7245	10.59
TDS_RC102_2	6975	3.77
TDS_RC102_3	6975	14.61
TDS_RC103_1	7110	7.57
TDS_RC103_2	7245	18.64
TDS_RC103_3	6975	3.71
TDS_RC104_1	7350	25.34
TDS_RC104_2	6975	4.22
TDS_RC104_3	6240	7.64
TDS_RC105_1	7395	4.79
TDS_RC105_2	7305	34.10
TDS_RC105_3	7350	10.20
TDS_RC106_1	5865	2.58
TDS_RC106_2	6675	2.07
TDS_RC106_3	5985	3.97
TDS_RC107_1	5655	11.29
TDS_RC107_2	4365	8.19
TDS_RC107_3	5580	3.82
TDS_RC108_1	6225	3.37
TDS_RC108_2	5760	4.25
TDS_RC108_3	5595	2.02
TDS_RC201_1	7230	3.38
TDS_RC201_2	8670	4.33
TDS_RC201_3	7425	5.66
TDS_RC202_1	8670	6.23
TDS_RC202_2	7425	5.50
TDS_RC202_3	7410	11.73
TDS_RC203_1	8070	4.13
TDS_RC203_2	8670	14.54
TDS_RC203_3	7410	11.70
TDS_RC204_1	8745	14.35
TDS_RC204_2	7410	12.31
TDS_RC204_3	7005	6.21
TDS_RC205_1	8670	10.69

TDS_RC205_2	7470	2.87
TDS_RC205_3	7110	1.64
TDS_RC206_1	7095	8.74
TDS_RC206_2	7170	18.25
TDS_RC206_3	7380	3.43
TDS_RC207_1	6735	5.95
TDS_RC207_2	6600	4.67
TDS_RC207_3	5565	1.94
TDS_RC208_1	5895	8.52
TDS_RC208_2	5745	1.86
TDS_RC208_3	5745	8.09
Average		11.88

3.4.4 Comparison between MILP and LS

The first objective is to compare the performance of the LS and the MILP model, over the two sets of instances. Table 3-7 presents the running times in seconds and the speed factor of each model for the sets of instances PGLT and GOEL. For the PGLT instances, both models bring the same solutions, albeit for GOEL instances the MILP model does not find the optimal solution for test TDS_R209_1 using the initial setting. The optimal solution is obtained after increasing the number of nodes per activity in the MILP mode, with a huge increase in the running time. Thus, in order to do not distort the analysis, this running time was not considered when computing the statistics. As stated before the MILP model is too slow, requiring on average 3 minutes to solve an instance from PGLT set, and about 2 minutes in the case of GOEL set. The label setting clearly outperforms the MILP model solving the set of instances PGLT and GOEL, 28 and 10 times faster, respectively.

Table 3-7. Comparison between models.

Model	PGLT Avg. running time (s)	PGLT Speed factor	GOEL Avg. running time (s)	GOEL Speed factor
MILP	183.97	28.26	124.33	10.46
LS	6.51	1.00	11.89	1.00

The second experiment compares the LS under the assumption of forbidding the night work as presented in Goel (2018). Goel's rule is referred to as GRULE, and the night working rule from the EC social legislation (EURULE). Table 3-8 reports for GOEL instances.

Table 3-8. Results LS under GRULE for instances Goel.

Instance	LS-EURULE Completion (min)	LS-EURULE Time (s)	LS-GRULE Completion (min)	LS-GRULE time (s)
TDS_C101_1	7095	1.03	7095	0.10
TDS_C101_2	5625	4.79	5640	0.11
TDS_C101_3	7185	1.96	7185	0.11
TDS_C102_1	6465	1.07	6705	0.09
TDS_C102_2	5340	6.52	5340	0.11
TDS_C102_3	6180	4.06	6180	0.15

TDS_C103_1	6465	1.33	6705	0.11
TDS_C103_2	4530	25.59	4530	0.17
TDS_C103_3	6180	4.06	6180	0.15
TDS_C104_1	5925	1.50	5925	0.11
TDS_C104_2	4650	17.68	4650	0.20
TDS_C104_3	6180	1.69	6180	0.16
TDS_C105_1	5385	7.10	5625	0.11
TDS_C105_2	6990	3.33	6990	0.13
TDS_C105_3	6915	1.50	6915	0.10
TDS_C106_1	5595	4.79	5640	0.08
TDS_C106_2	7125	0.85	7125	0.07
TDS_C106_3	7020	1.74	7020	0.11
TDS_C107_1	5250	10.29	5490	0.16
TDS_C107_2	6660	1.60	6720	0.12
TDS_C107_3	6750	2.88	6810	0.17
TDS_C108_1	5460	9.02	5460	0.14
TDS_C108_2	6600	2.16	6810	0.18
TDS_C108_3	6555	1.58	6720	0.13
TDS_C109_1	6030	1.77	6030	0.16
TDS_C109_2	6120	2.43	6120	0.23
TDS_C109_3	4680	8.57	4680	0.19
TDS_C201_1	4140	5.97	4140	0.07
TDS_C201_2	8505	6.65	8505	0.10
TDS_C202_1	4140	3.36	4140	0.18
TDS_C202_2	8505	10.63	8505	0.12
TDS_C203_1	4140	3.35	4140	0.18
TDS_C203_2	8505	10.70	8505	0.18
TDS_C204_1	4140	3.35	4140	0.18
TDS_C204_2	8475	14.59	8475	0.31
TDS_C205_1	3930	8.40	4050	0.10
TDS_C205_2	8415	131.02	8415	0.24
TDS_C206_1	3870	5.67	4050	0.11
TDS_C206_2	8235	169.13	8235	0.32
TDS_C207_1	4095	5.55	4140	0.08
TDS_C207_2	7620	10.59	7620	0.43
TDS_C208_1	3525	4.82	3735	0.15
TDS_C208_2	8580	10.77	8580	0.27
TDS_R101_1	6885	2.76	6990	0.08
TDS_R101_2	6525	2.19	6735	0.12
TDS_R101_3	6825	0.56	6825	0.07
TDS_R101_4	7470	4.25	7470	0.10
TDS_R101_5	6690	1.04	6720	0.09
TDS_R102_1	6945	3.37	7050	0.09
TDS_R102_2	6690	1.72	6720	0.09
TDS_R102_3	7590	3.91	7590	0.15
TDS_R102_4	7110	2.66	7110	0.16
TDS_R103_1	6690	2.58	6720	0.13
TDS_R103_2	7995	5.34	7995	0.20
TDS_R103_3	8445	3.06	8445	0.17
TDS_R104_1	6690	4.72	6720	0.18
TDS_R104_2	4785	8.70	4785	0.18
TDS_R104_3	8445	3.32	8445	0.16
TDS_R105_1	6630	3.00	6840	0.12
TDS_R105_2	7245	4.95	7245	0.14
TDS_R105_3	4710	11.91	4710	0.14
TDS_R105_4	6690	2.43	6690	0.10
TDS_R106_1	5865	51.42	5865	0.20
TDS_R106_2	8055	9.54	8055	0.18

TDS_R106_3	6690	5.02	6930	0.15
TDS_R107_1	5535	5.75	5565	0.16
TDS_R107_2	7305	7.45	7305	0.22
TDS_R107_3	6795	18.66	6930	0.29
TDS_R108_1	6315	9.78	6555	0.32
TDS_R108_2	8085	7.73	8085	0.30
TDS_R109_1	6615	6.78	6615	0.25
TDS_R109_2	5880	28.74	5880	0.21
TDS_R109_3	6600	6.97	6600	0.13
TDS_R110_1	6870	41.45	6870	0.32
TDS_R110_2	6045	69.20	6045	0.30
TDS_R111_1	6720	6.77	6720	0.14
TDS_R111_2	5565	27.31	5565	0.46
TDS_R111_3	8055	9.17	8055	0.22
TDS_R201_1	4590	1.44	4590	0.08
TDS_R201_2	6885	2.73	6885	0.12
TDS_R201_3	7440	1.23	7440	0.11
TDS_R201_4	8220	25.18	8310	0.29
TDS_R202_1	7500	3.14	7500	0.11
TDS_R202_2	6840	2.15	6840	0.14
TDS_R202_3	8310	5.81	8430	0.18
TDS_R202_4	4590	2.08	4590	0.12
TDS_R203_1	9105	4.49	9105	0.18
TDS_R203_2	8280	0.74	8280	0.08
TDS_R203_3	6030	4.24	6030	0.19
TDS_R204_1	6675	10.15	6720	0.31
TDS_R204_2	8160	21.53	8400	0.46
TDS_R205_1	6990	3.76	7110	0.13
TDS_R205_2	6465	43.01	6705	0.20
TDS_R205_3	4500	3.82	4500	0.15
TDS_R205_4	7425	4.01	7425	0.15
TDS_R206_1	8175	4.82	8175	0.19
TDS_R206_2	7050	67.17	7170	0.24
TDS_R206_3	6750	5.29	6750	0.18
TDS_R207_1	7380	6.50	7380	0.17
TDS_R207_2	5235	6.29	5475	0.25
TDS_R207_3	8010	12.74	8010	0.28
TDS_R209_1	6840	6.71	6870	0.16
TDS_R209_2	6585	2.81	6585	0.12
TDS_R209_3	6720	45.70	6720	0.28
TDS_R210_1	5610	6.67	5610	0.17
TDS_R210_2	8685	6.34	8685	0.19
TDS_R210_3	7110	33.63	7170	0.25
TDS_R211_1	5580	129.86	5580	0.93
TDS_R211_2	7365	67.80	7365	0.73
TDS_RC101_1	6645	4.72	6645	0.15
TDS_RC101_2	6975	20.73	6975	0.17
TDS_RC101_3	7245	10.60	7245	0.12
TDS_RC102_1	7245	10.59	7245	0.12
TDS_RC102_2	6975	3.77	6975	0.18
TDS_RC102_3	6975	14.61	6975	0.18
TDS_RC103_1	7110	7.57	7110	0.17
TDS_RC103_2	7245	18.64	7245	0.22
TDS_RC103_3	6975	3.71	6975	0.18
TDS_RC104_1	7350	25.34	7350	0.22
TDS_RC104_2	6975	4.22	6975	0.22
TDS_RC104_3	6240	7.64	6240	0.23
TDS_RC105_1	7395	4.79	7395	0.13

TDS_RC105_2	7305	34.10	7305	0.15
TDS_RC105_3	7350	10.20	7350	0.13
TDS_RC106_1	5865	2.58	5865	0.14
TDS_RC106_2	6675	2.07	6675	0.16
TDS_RC106_3	5985	3.97	5985	0.13
TDS_RC107_1	5655	11.29	5655	0.17
TDS_RC107_2	4365	8.19	4365	0.27
TDS_RC107_3	5580	3.82	5640	0.17
TDS_RC108_1	6225	3.37	6225	0.10
TDS_RC108_2	5760	4.25	5760	0.19
TDS_RC108_3	5595	2.02	5595	0.17
TDS_RC201_1	7230	3.38	7230	0.16
TDS_RC201_2	8670	4.33	8670	0.12
TDS_RC201_3	7425	5.66	7425	0.15
TDS_RC202_1	8670	6.23	8670	0.13
TDS_RC202_2	7425	5.50	7425	0.16
TDS_RC202_3	7410	11.73	7410	0.19
TDS_RC203_1	8070	4.13	8070	0.18
TDS_RC203_2	8670	14.54	8670	0.22
TDS_RC203_3	7410	11.70	7410	0.19
TDS_RC204_1	8745	14.35	8745	0.21
TDS_RC204_2	7410	12.31	7410	0.23
TDS_RC204_3	7005	6.21	7140	0.20
TDS_RC205_1	8670	10.69	8670	0.17
TDS_RC205_2	7470	2.87	7470	0.17
TDS_RC205_3	7110	1.64	7110	0.15
TDS_RC206_1	7095	8.74	7095	0.19
TDS_RC206_2	7170	18.25	7170	0.17
TDS_RC206_3	7380	3.43	7380	0.10
TDS_RC207_1	6735	5.95	6735	0.18
TDS_RC207_2	6600	4.67	6660	0.13
TDS_RC207_3	5565	1.94	5565	0.14
TDS_RC208_1	5895	8.52	5895	0.20
TDS_RC208_2	5745	1.86	5745	0.16
TDS_RC208_3	5745	8.09	5745	0.24
Average	6687.61	11.85	6719.62	0.18

Table 3-9 presents the results for this comparison. On average, the EURULE yields a reduction of about 32 hours in the completion time of the schedules. Although, there is a high impact on the running times, where the average time to solve an instance using the GRULE is only 0.18 seconds, while using the EURULE it sharply increases to 11.85 seconds.

Table 3-9. Comparison assumptions on night working rule

	Avg. Completion (min)	*Avg. Time (s)*
GRULE	6719.62	0.18
EURULE	6687.61	11.85

In order to assess the effect of the night constraint on the schedules and the performance of the LS, the third experiment is to run the LS without (NO-Night) and with the working night constraint (EURULE) over the set PGLT. (See Table 3-10).

Table 3-10. Results LS NO-Night and EURULE for instances PGTL.

	NO-Night		EURULE	
Instance	Completion	Time (s)	Completion	Time (s)
TEST_1	1410	0.10	1635	2.22
TEST_2	1260	0.04	1260	0.07
TEST_3	1410	0.04	1410	0.11
TEST_4	1410	0.04	2040	0.05
TEST_5	2640	0.21	2955	17.03
TEST_6	2430	0.10	2910	4.67
TEST_7	3465	0.13	3720	11.60
TEST_8	3465	0.12	3720	11.56
TEST_9	2730	0.04	0	0.08
TEST_10	2730	0.04	0	0.08
TEST_11	2730	0.03	0	0.04
TEST_12	2730	0.04	0	0.03
TEST_13	1995	0.07	2250	4.25
TEST_14	3270	0.20	3720	20.42
TEST_15	4665	0.38	5220	50.98
TEST_16	4665	0.19	0	0.71
TEST_17	5340	0.15	0	0.72
TEST_18	2430	0.07	2910	4.70
TEST_19	5580	0.06	5580	5.03
TEST_20	3780	0.06	3780	5.80
TEST_21	3210	0.10	3210	9.19
TEST_22	5610	0.16	5610	6.83
TEST_23	5820	0.16	5820	7.68
TEST_24	2580	0.16	2820	12.38
TEST_25	3510	0.25	3795	15.28
TEST_26	2640	0.18	2865	9.59
TEST_27	2880	0.20	3105	8.43
TEST_28	4065	0.29	4290	19.88
TEST_29	1335	0.07	1575	2.92
TEST_30	1245	0.05	1485	0.91
TEST_31	2025	0.08	2280	3.40
TEST_32	2025	0.08	2280	3.03
TEST_33	2025	0.08	2280	5.46
TEST_34	2400	0.04	2400	1.55
TEST_35	1305	0.05	1545	0.91
TEST_36	1440	0.06	1665	1.99
TEST_37	4530	0.07	0	0.41
TEST_38	1380	0.03	2100	0.25
TEST_39	2400	0.04	3000	1.11
TEST_40	3345	0.10	3345	9.06
Average		0.11		6.51

Table 3-11 summarizes the results for this experiment, computing the average completion and the running times over the set of feasible instances. For this set of instances, the average completion time is the same in both cases. However, the LS without considering the night working rule is 72 times faster than the LS under this constraint. Finally, the 17.5% of the sequences without considering the night working rule become infeasible if the night working rule applies.

Table 3-11. Night effect on the LS performance.

	Avg. Completion (h)	Avg. Time (s)	Infeasible
NO-Night	2989.62	0.13	0
EURULE	2989.62	9.51	7

3.5 Conclusion

Several contributions in the past have worked on the TDSP, albeit most of them have neglected or forbidden the night working rule. In this paper, we present a label setting algorithm to solve the TDSP which minimizes the completion time of a given sequence of customers, while considering all the rules from the EC social legislation for a weekly planning period, including the night working constraint. The LS is compared against a MILP model, both solution methods bring the same solutions, however the LS requires significantly smaller computational times. Results shows that forbidding or neglecting the night working drastically diminishes the running times of the algorithm, nevertheless the EC night working rule yield schedules with equal or better completion times.

In particular, it is possible to find schedules with the same completion time with or without considering the night working rule, if there is a feasible solution. With regard to the experiments results, when solution methods do not consider the night working rule are likely to find infeasible solutions. Finally, even though the LS exhibits a good performance in both, quality of the solution and running times, smaller computational times are required in order to solve the combined VRTDSP.

3.6 References

[1] Goel, A., “Vehicle scheduling and routing with driver’s working hours”, *Transport. Sci.*, vol.43, no. 1, pp. 17–26, 2009.

[2] Kok, A.L., Meyer, C.M., Kopfer, J.M., and Schutten, J.M., “A dynamic programming heuristic for the vehicle routing problem with time windows and European Community Social Legislation”, *Transport. Sci.*, vol.44, no. 4, pp. 429–553, 2010.

[3] Prescott-Gagnon, E., Desaulniers, G., Drexl, M. and Rousseau, L.M., “European Driver Rules in Vehicle Routing with Time Windows”, *Transp. Sci.*, vol. 44, no. 4, pp. 455–473, 2010.

[4] Goel, A. and Vidal, T., “Hours of Service Regulations in Road Freight Transport: An Optimization-Based International Assessment”, *Transp. Sci.*, vol. 48, no. 3, pp. 391–412, 2013.

[5] Goel, A. and Irnich, S., “An exact method for vehicle routing and truck driver scheduling problem”, *Transport. Sci.*, vol.51, no. 2, pp. 737–754, 2016.

[6] Goel, A., “Legal aspects in road transport optimization in Europe”, *Transport. Res. E-Log*, vol.114, pp. 144–162, 2018.

[7] Tilk, C. and Goel, A., “Bidirectional labeling for solving vehicle routing and truck driver scheduling problems”, *Eur. J. Oper. Res.*, vol. 283, pp. 108–124, 2020.

[8] Feillet, D, Dejax, P, Gendreau, M. and Gueguen, C., “An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems”. *Networks.*, vol. 44, no. 3, pp. 16–229, 2004.

[9] Peña-Arenas. I., Garaix, T., Lacomme P. and Tchernev, N., “A mixed integer programming formulation for the truck drivers scheduling problem considering the European Union drivers rules”, 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), 2021, pp. 101-106, doi: 10.1109/CASE49439.2021.9551634.

CONCLUSION

To This document addressed the Truck Drivers Scheduling Problem under the European Community Social Legislation. In the past the problem has been treated alone, as the problem of scheduling breaks/rests given a fixed sequence of customers or as a part of a vehicle routing problem, the Vehicle Routing and Truck Drivers Scheduling Problem. This thesis provides different types of solution methods, mainly linear formulations and label setting algorithms under non-preemption and preemption assumptions.

This thesis used two different assumptions while developing the solutions methods: non-preemption and preemption. Since the first objective was to understand of the rules, then the starting point was to develop a linear formulation. In the literature all the linear formulations use non preemption assumptions, that is to say, breaks or rests can only take place at customer or specific locations. Thus, the first models presented in the document follow this assumption, which is useful since, it is more straight-forward and diminishes the complexity of the problem. However, the most recent approaches use preemption assumptions on driving activities, therefore the second part of the models proposed in the thesis work under this assumption for both, driving and service activities.

There are diverse linear formulations related to different driver working hours regulations, in example: The US Hours of Service, the Canadian Commercial Vehicle Drivers Hours of Service, etc. In the case of the EC Social Legislation, the MILP formulations work on the basic set of rules. Therefore, all of them exclude the directive 2002/15/EC on working hours, in particular, the night working hour rule. In addition to linear formulations, different heuristic and metaheuristics have been proposed, albeit, most of them strive to find a feasible or legal solution. As the MILP models, they start with the most simple models, those which only tackle the basic set of rules, evolving to a more complete models. However, even the most recent approaches do not consider the night working rule.

The methods presented in the thesis filled in two different gaps in the literature. First, a linear formulation which takes into account all the rules from the EC Social Legislation that apply for a planning horizon of one week. This formulation brings a better understanding of the EC Social Legislation and helps to develop better solution methods, since they retrieve optimal solutions and bring a measure of the running times required to solve the problem. Second, an optimal label setting algorithm, which in contrast with all previous solution methods, considers the night working rule while solving the TDSP.

The first part of the thesis presented a linear formulation and a label setting algorithm working under non preemption assumptions. The linear formulation extends previous MILP implementations while considering: breaks due to working time, driving time extensions and daily rest reductions are included. As expected the computational experiments showed that the optimal label setting algorithm outperforms the MILP formulation. Moreover, a new set of instances is proposed in order to validate the models and full detail solutions are provided, which could be used in future researches.

The second part of the thesis overrides the non-preemption assumption, developing a MILP formulation and an optimal label setting algorithm. As a result, the models are more complex with respect to the non-preemptive versions since, they work on an enhanced solution space. Moreover, they include the night working constraint. Therefore, in the case of the label setting algorithm, new attributes and better dominance rules are proposed in order to improve its performance. The label setting algorithm attain optimal solutions in reasonable running times, although improvements are required in order to boost the performance of the algorithm. Finally, new tests are included in the proposed set of instances and detail solutions for all of them are provided.

These works offer different prospects and possible extensions.

Even though the version of the label setting algorithm presented in this document obtain optimal results in competitive running times, it is necessary to improve this performance if the algorithm is going to be used as a subroutine in a vehicle routing problem. One way to improve the running times is to modify the algorithm to retrieve feasible or legal solutions instead of the optimal. Although, there is one key research point that is going to be missed; in the literature all the scheduling algorithms that have been used to solve the VRTDSP are heuristics, thus it would be more than interesting to check what would be the results if the an optimal scheduling algorithm is used embedded in the vehicle routing problem.

The main objective of solving the TDSP is to develop a solution method that could be used as a subroutine inside an integrated vehicle routing schema. Thus, the natural evolution of the problem is to solve the VRTDSP. Several exact and approximate approaches have been developed to solve the integrated routing and scheduling problem, albeit none of them have considered the night working rule from the EC social legislation.

The first possible extension consists in considering heterogeneous fleet of vehicles with different fixed and variable costs meaning that for the same trip, the breaks of different vehicles have to be scheduled at different locations. The major problem leads in the definition of one efficient local search trying to define time efficient operators. Note that all routing problems (including the IRP, the HVR, etc.) should be addressed considering the regulations on break and that extensions should have significant impact on the operators to use.

The second extension should address stochastic routing problem to compute solutions with breaks trying to favor robust solutions as regards fluctuations in the transportation times. Computation of robust solutions should require computation of solutions where break are not schedule a the latest possible date or location, but at a date and location that define an acceptable compromise between the solution cost and the consequence of transportation time fluctuation.

REFERENCES

Adminaité-Fodor, D. and Jost, G., "How to improve the safety of goods vehicles in the EU", European Transport Safety Council, Brussels, PIN Flash Report 39, 2020. [Online]. Available: https://etsc.eu/wp-content/uploads/PIN-FLASH39_FINAL.pdf

Archetti, C. and Savelsbergh, M.W.P., "The trip scheduling problem", *Transp. Sci.*, vol. 43, pp. 417–431, 2009.

Archetti, C. and Savelsbergh, M.W.P., "The trip scheduling problem", *Transp. Sci.*, vol. 43, pp. 417–431, 2009.

Bartodziej, P., Derigs, U., Malcherek, D. and Vogel, U., "Models and algorithms for solving combined vehicle and crew scheduling problems with rest constraints: an application to road feeder service planning in air cargo transportation", *OR Spectrum*, vol. 31, pp. 405–429, 2009.

Bowden, Z. and Ragsdale, C., "The truck driver scheduling problem with fatigue monitoring", *Decis. Support Syst.,* vol. 110, pp. 114-162, 2018.

Brandao, J. and Mercer, A., "A tabu search algorithm for the multi-trip vehicle routing and scheduling problem", *Eur. J. Oper. Res.,* vol. 100, no. 1, pp. 180-191, 1997.

Campbell, A. and Savelsbergh, M., "Efficient insertion heuristics for Vehicle Routing and Scheduling problems", *Transp. Sci.*, vol. 38, no. 3, pp. 369–378, 2004.

Ceselli, A., Righini, G. and Salani, M., "A column generation algorithm for a rich vehicle routing problem", *Transp. Sci.*, vol. 43, no. 1, pp. 56–69, 2009.

Derigs, U., Kurowsky, R. and Vogel, U, "Solving a real-world vehicle routing problem with multiple use of tractors and trailers and EU-regulations for drivers arising in air cargo road feeder services", *Eur. J. Oper. Res.,* vol. 213, no. 1, pp. 309-319, 2011.

Drexl, M. and Prescott-Gagnon, E., Labelling algorithms for the elementary shortest path problem with resource constraints considering EU drivers' rules", *Logist. Res.,* Vol. 2, no. 2, pp. 79-96, 2010.

European Commission, "Fatigue", Directorate General for Transport, 2018. [Online]. Available: https://ec.europa.eu/transport/road_safety/sites/default/files/pdf/ersosynthesis2018-fatigue.pdf

European Road Safety Observatory, "European Commission (2020) Facts and Figures Pedestrians", Brussels, European Commission, Directorate General for Transport, 2020. [Online]. Available: https://ec.europa.eu/transport/road_safety/sites/default/files/facts_figures_buses_and_hgv_final_20210323.pdf

European Transport Safety Council, “The role of driver fatigue in commercial road transport crashes”, Directorate of Energy and Transport of the European Commission, Brussels, 2001. [Online]. Available: https://etsc.eu/wp-content/uploads/The-role-of-driver-fatigue-in-commercial-road-transport-crashes.pdf

European Union, Directive 2002/15/EC of the European Parliament and of the Council of 11 March 2002 on the organisation of the working time of persons performing mobile and road transport activities, 2002, [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32002L0015&from=EN

European Union. 2006. Regulation (EC) No. 561/2006 of the European Parliament and of the Council of 15 March 2006 on the harmonisation of certain social legislation relating to road transport and amending Council Regulations (EEC) 3821/85 and (EC) 2135/98 and repealing Council Regulation (EEC) 3820/85. Official Journal of the European Union L 102, 11.04.2006.

Feillet D, Dejax P, Gendreau M and Gueguen C., “An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems”. *Networks.,* Vol. 44, no. 3, pp. 16–229, 2004.

Filtness, A., Anund, A., Maynard, S., Miller, k., Pilkington-Cheney, F., Dahlman, A. and Ihlstrom, J., “Bus driver fatigue”, Transport Safety Research Group, Loughborough University, Loughborough, Final Report, 2019. [Online]. Available: http://content.tfl.gov.uk/bus-driver-fatigue-report.pdf

Goel, A. and Irnich, S., “An exact method for vehicle routing and truck driver scheduling problem”, *Transport. Sci.*, vol.51, no. 2, pp. 737–754, 2016.

Goel, A. and Kok, A.L., “Efficient scheduling of team truck drivers in the European Union”, *Flex. Serv. Manuf. J.*, vol. 24, no. 1, pp. 96-81, 2012.

Goel, A. and Rousseau L-M., “Truck driver scheduling in Canada”, *J. Sched.,* vol.15, pp. 783–799, 2012.

Goel, A. and Vidal, T., “Hours of Service Regulations in Road Freight Transport: An Optimization-Based International Assessment”, *Transp. Sci.*, vol. 48, no. 3, pp. 391–412, 2013.

Goel, A. Vehicle scheduling and routing with drivers’ working hours. *Transp. Sci.,* vol. 43, no. 1, pp. 17–26, 2009.

Goel, A. Vehicle scheduling and routing with drivers’ working hours. *Transportation Sci.,* vol. 43, no. 1, pp. 17–26, 2009.

Goel, A., “A mixed integer programming formulation and effective cuts for minimising schedule durations of Australian truck drivers”, *J. Sched.*, vol. 15, no. 6, pp. 733–741, 2012.

Goel, A., “Hours of service regulations in the United States and the 2013 rule change”, *Transp. Policy.*, vol. 33, pp. 48–55, 2014.

Goel, A., “Legal aspects in road transport optimization in Europe”, *Transport. Res. E-Log*, vol.114, pp. 144–162, 2018.

Goel, A., “The minimum duration truck driver scheduling problem”, *EURO J. Transp. Logist.*, vol. 1, no. 4, pp. 285–306, 2012.

Goel, A., “Truck driver scheduling in the European Union”, *Transp. Sci.*, vol. 44, pp. 429–441, 2010.

Goel, A., “Vehicle scheduling and routing with driver’s working hours”, *Transport. Sci.*, vol.43, no. 1, pp. 17–26, 2009.

Goel, A., Archetti, C., and Savelsbergh, M.W.P., “Truck driver scheduling in Australia”, *Comput. Oper. Res*., vol. 39, pp. 1122–1132, 2012.

Goel, A., The Canadian minimum duration truck driver scheduling problem. *Comput. Oper. Res.*, vol. 39, no. 10, pp. 2359–2367, 2012.

Knipling, R., “Review of commercial driver fatigue research methodologies”, Commissioned paper, National Research Council (NRC), Committee on National Statistics (CNSTAT), 2015. [Online]. Available: https://sites.nationalacademies.org/cs/groups/dbassesite/documents/webpage/dbasse_171093.pdf

Koç, C., Bektas, T., Jabali, O. and Laporte, G., “A comparison of three idling options in long-haul truck scheduling”, *Transport. Res. B-Meth.,* vol. 93, pp. 631-647, 2016.

Koç, Ç., Jabali, O. and Laporte, G., “Long-haul vehicle routing and scheduling with idling options”, *J. Oper. Res. Soc,* vol. 69, no. 2, pp. 235-246, 2017.

Kok, A. L., C. M. Meyer, H. Kopfer and J. M. J. Schutten. “A dynamic programming heuristic for the vehicle routing problem with time windows and European Community social legislation”, *Transportation Sci*. vol. 44, no. 4, pp. 442–454, 2010.

Kok, A.L., Hans, E.W. and Schutten, J.M.J, “Optimizing departure times in vehicle routes”, *Eur. J. Oper. Res.,* vol. 210, no. 3, pp. 579-587, 2011.

Min, H., “Building a model-based decision support system for solving vehicle routing and driver scheduling problems under hours of service regulations”, University Transportation Centers program, Ohio, 2011. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/23718

[9] Peña-Arenas. I., Garaix, T., Lacomme P. and Tchernev, N., "A mixed integer programming formulation for the truck drivers scheduling problem considering the European Union drivers rules”, 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), 2021, pp. 101-106, doi: 10.1109/CASE49439.2021.9551634.

Prescott-Gagnon, E., Desaulniers, G., Drexl, M. and Rousseau, L.M., “European Driver Rules in Vehicle Routing with Time Windows”, *Transp. Sci.*, vol. 44, no. 4, pp. 455–473, 2010.

Rancourt, M., Cordeau, J. and Laporte, G., “Long-haul vehicle routing and scheduling with working hours rules”, *Transp. Sci.*, vol. 47, no. 1, pp. 81–107, 2013.

Rochat, Y. and Semet, F., “A tabu search approach for delivering pet food and flour in Switzerland”, *J. Oper. Res. Soc,* vol. 45, no. 11, pp. 1233-1246, 1994.

Sartori, C., Smet, P. and Vanden, G., "Scheduling truck drivers with interdependent routes under European Union regulations", *Eur. J. Oper. Res.,* vol. 298, no. 1, pp. 76-88, 2022.

Savelsbergh, M. and Sol, M., "Dynamic routing of independent vehicles", *Oper. Res.,* vol. 46, no. 4, pp. 474-490, 1998.

Tilk, C. and Goel, A. "Bidirectional labeling for solving vehicle routing and truck driver scheduling problems", *Eur. J. Oper. Res.*, vol. 283, pp. 108–124, 2020.

Vitols, K, and Eckhard, V., "Driver fatigue in European road transport", European Transport Workers's Federation, Brussels, 2021. [Online]. Available: https://www.etf-europe.org/wp-content/uploads/2021/05/Driver-Fatigue-in-European-Road-Transport-Report.pdf

Wen, M., Krapper, E., Larsen, J. and Stidsen, T., "A multilevel variable neighborhood search heuristic for a practical vehicle routing and driver scheduling problem", *Networks*, vol. 58, no. 4, pp. 311–322, 2011.

Xu, H., Chen, Z., Rajagopal, S. and Arunapuram, S., "Solving a Practical Pickup and Delivery Problem", *Transp. Sci.*, vol. 37, no. 3, pp. 347–364, 2003.

Zapfel, G. and Bogl, M., "Multi-period vehicle routing and crew scheduling with outsourcing options", *Int. J. Prod. Econ.*, vol. 113, no. 2, pp. 980–996, 2008.