Aalto University

MASTER'S THESIS 2019

Developing an optimization model for VR's long-term locomotive allocation planning

Janne Eskola

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

Otaniemi, 14.2.2019

Supervisor: Professor Antti Punkka
Advisor: M.Sc Olli Suolanen

Aalto University
School of Science
Master's Programme in Mathematics and Operations Research

Aalto University

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Abstract

Author
Janne Eskola

Title
Developing an optimization model for VR's long-term locomotive allocation planning

School School of Science

Master's programme Mathematics and Operations Research

Major Systems and Operations Research **Code** SCI3055

Supervisor Professor Antti Punkka

Advisor M.Sc Olli Suolanen

Level Master's thesis **Date** 14.2.2019 **Pages** 6 + 58 **Language** English

Abstract

The VR Group's long-term process for locomotive allocation produces week-long routes for multiple locomotive types. The routes consist of various tasks, primarily driving trains and moving locomotives between stations. The goal of the planning is to create a plan minimizes operating costs while ensuring the availability of locomotives for all trains. This planning process is currently done manually, resulting in incomplete or inefficient plans. To overcome this problem, this thesis develops an optimization model for allocating locomotives in order to produce cost-efficient, robust and feasible solutions within one hour.

The problem is modeled as a multicommodity flow problem, in which two model variants (one using and another lacking driver costs) as well as three heuristic pre-processing algorithms were developed. These all were evaluated in terms of cost efficiency and robustness using VR's test data in order to select the best combination of model and pre-processing algorithm, which were then successfully implemented in VR's planning process. Testing of six model-algorithm combinations revealed that the model variant using driver costs performed better than that without driver costs. The most advanced pre-processing algorithm was selected, as it can be modified to produce the same outcome as either of the others.

The selected model-algorithm combination has successfully been implemented in VR's planning process. The model is able to produce cost-efficient, robust and feasible locomotive allocation plans in less than 15 minutes.

Keywords Integer optimization, Locomotive allocation, Multicommodity

A! Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Janne Eskola

Työn nimi
Optimointimallin kehittäminen VR:n pitkänaikavälin veturienkäytön suunnitteluun

Korkeakoulu Perustieteiden korkeakoulu

Maisteriohjelma Mathematics and Operations Research

Pääaine Systems and Operations Research **Code** SCI3055

Valvoja Professori Antti Punkka

Ohjaaja DI Olli Suolanen

Työn laji Diplomityö **Päiväys** 14.2.2019 **Sivuja** 6 + 58 **Kieli** englanti

Tiivistelmä

VR:n pitkän aikavälin veturienkäytön suunnittelussa laaditaan jokaiselle viikon mittaiselle suunnittelujaksolle reitit kaikille vetureille useilta eri veturisarjoilta. Veturien reitit koostuvat pääasiassa junien ajamisesta sekä siirroista asemien välillä. Suunnittelun tavoitteena on luoda suunnitelma, jossa varmistetaan kaikkiin juniin riittävä määrä vetureita ja minimioidaan operatiiviset kustannukset. Suunnitteluprosessi on tällä hetkellä manuaalinen ja johtaa usein vaillinaisiin suunnitelmiin tai epätehokkuuksiin veturien käytössä. Ratkaistakseen ongelman, työn tavoitteena on kehittää optimointimalli VR:n suunnitteluongelmaan, joka tuottaa toteuttamiskelpoisia, kustannustehokkaita ja robusteja uunnitelmia alle tunnissa.
Ongelma on mallinnettu monihyödykevirtausongelmana (Multicommodify flow). Saadaksemme ongelma ratkeamaan kohtuullisessa ajassa kehitettin mallista kaksi versiota ja kolme heuristista esikäsittelyä. Näitä molempia testattiin VR:n antamalla aineistolla ja paras yhdistelma valittiin osaksi VR:n veturien käytön suunnitteluprosessia. Testauksen myötä selvisi, että kuljettaja kustannusten huomiointi on kriittistä onnistumiselle sekä suunnitelman kustannustehokkuuden, että robustisuuden kannalta. Esikäsittely algorithmeistä valittiin edistynein, sillä se on mahdollista muokata tuottamaan sama sama lopputulos kuin kahdesta muusta.
Valittu malli-algoritmi yhdistelmä otettiin onnistuneesti osaksi VR:n suunnitteluprosessia ja sen avulla pystytään tuottamaan toteuttamiskelpoisia, kustannustehokkaita ja robusteja suunnitelmia alle 15 minuutissa.

Avainsanat Kokonaislukuoptimointi, Veturienkäytön optimointi, Monihyödykeongelma

Acknowledgements

I would first like to thank my thesis advisor M.Sc Olli Suolanen from the VR Group and Prof. Antti Punkka of the System analysis laboratory at Aalto University. They both offered me valuable support and allowed this paper to be my own work but steered me in the right the direction whenever I needed it.

I must express my very profound gratitude to my closest student friend Elmeri Lähevirta and to my partner Laura Lonka for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of writing this thesis. This accomplishment would not have been possible without them. Thank you.

Janne Eskola

Contents

1. Introduction

The railway industry is dependent on long-term planning due to limited track capacity. In order for trains to operate, they require accurate planning of personnel and train schedules as well as allocation of wagons and locomotives. Of these, a key planning problem is the planning of locomotive usage because the size of locomotive fleet is the least flexible resource due to the long acquisition time and capital-intensive nature which set boundaries for the growth. This locomotive allocation problem (LAP) refers to the planning of locomotive routes in order to operate all selected trains cost efficiently. To achieve this, each locomotive obtains its own route for a particular planning period. This route can involve driving trains, driving the locomotive between locations and performing other tasks. These tasks include train brake tests and activities for creating a pulling unit from multiple locomotives. Each of these activities has its own requirements and cannot be performed using every locomotive type. Moreover, locomotives are constrained by how quickly they can move from one activity to another. Additionally, the track capacity for driving a locomotive between stations is often limited. This restricts the options available for transferring the locomotive from its current location to the next. These multiple constraints make LAP complex to solve.

The VR Group (VR), a Finnish state-owned railway company, weekly operates roughly 3 000 trains, consisting of both freight and long-distance passenger trains. Additionally, VR also operates commuter traffic , though it does not need locomotives, as the trains are self-powered. Locomotive-driven trains are operated using a fleet of 350 locomotives, which are divided into six locomotive types, all of which have different attributes. The locomotive types differ in terms of the size of the fleet, energy source (diesel vs. electric), top speed and pulling power. These all affect the ability of the locomotive to perform different activities.

There are many feasible solutions for VR's LAP. They differ from each other in terms of their operating costs and robustness. Of these two factors, the costs are more significant, though the robustness of a plan is also important. In this thesis, robustness is defined as the ability to tolerate changes in train traffic with minimal alterations in locomotive planning. Robustness is particularly important in the long-term planning phase because there are likely to be many changes to traffic during the later planning stages. Therefore, it is essential that the number of planning decisions requiring other resources be kept as low as possible. the main indicator for robustness is considered by VR to be the amount of light travel. Light travel refers to driving only the locomotive from one location to another without pulling a train. Such light-travel journeys are needed, as they allow the same number of trains to be operated by a smaller locomotive fleet, thus increasing the utilization of locomotives. On the other hand, light travel is unwanted because both the locomotive and driver are not in productive use during light travel, which generates extra costs for the company. Moreover, the drivers are assigned shifts for light travels, which cannot be easily changed afterwards. Locking the drivers into unproductive shifts lowers the robustness of the plan and is avoided in the long-term planning phase because there is often high demand for extra traffic in later phases of the planning process.

Currently, the long-term planning phase is a time-consuming manual process. Although the planning is made in cycles of three weeks. the traffic typically varies between weeks, thus requiring that each week be planned as an independent entity. However, planning every week separately causes time pressure for locomotive planning. The tight time window limits the options for reacting to large, sudden changes, as all trains and their related tasks need to be planned manually. Both the tight time window and high amount of manual work often result in inefficient decisions and underuse of resources in the final long-term plan, thus lowering the robustness and cost efficiency.

One approach for addressing this problem would be to develop an optimization model in order to increase the effectiveness of locomotive planning under a strict time window. Optimization is an excellent tool for finding a good solution when there are many possible solutions. Although the locomotive allocation problem has been studied in literature, none of these optimization models are applicable as such to VR according to bachelor thesis by Eskola [2016]. The main reasons for this include the need to operate both passenger and freight trains using the same locomotives, the inability to rent locomotives between companies, and the sparse single-track rail network.

To address VR's long-term locomotive planning problem, this thesis develops an optimization model for producing cost efficient, feasible solutions in a reasonable amount of time.

The rest of this thesis is divided into eight chapters. Chapter 2 introduces VR's planning problem. Chapter 3 presents a literature survey on LAP optimization models. Chapter 4 formulates the problem into a mathematical form and constructs two alternative models. Chapter 5 constructs the needed data pre-processing algorithms. Chapter 6 presents the performance of different optimization models and data pre-processing algorithms in different scenarios. Chapter 7 compares the performance of the two models and the feasibility of utilizing the plans generated by these models. Chapter 8 evaluates the use of the optimization model in VR's planning process. Chapter 9 concludes by suggesting directions for future research.

2. Introduction of the problem

This chapter presents the core features of the VR's planning process and locomotive allocation problem and turns them into requirements for the model. The summary of the requirements is presented in Section 2.4. Based on these requirements we can assess the models in literature or develop a new model for the problem.

2.1 Background

Environment in which, VR operates is very isolated when comparing to most other European railway companies, which is due to several factors. First, Finland has very few railroad connections to other countries, as the only rail track connections are to Russia, moreover the locomotives do not cross the border between these two countries regularly. Secondly, VR also had a monopoly position for a long time in both passenger and freight train traffic, even though the freight traffic has been open for competition for 10 years the other operators are multiple magnitudes smaller. Furthermore, Finland has different track width than in Central Europe, preventing the use of same locomotives in Finland and in Central Europe. These three reasons cause that the fleet VR operates is fixed and renting the locomotives between other companies is not an option. This makes minimizing operational costs only viable option, when the other option would be to minimize the number of locomotives needed in the plan.

VR operates weekly around 1 000 locomotive driven passenger trains and 2 000 freight trains. Both traffics types are operated using the same locomotives to create synergies for locomotive and driver usage. The main driving factor is that passenger traffic is focused to daytime and freight traffic can be emphasized to night time. Dividing the train traffic this way smooths the utilization rate of locomotives more equally through the day. Using the same locomotives to operate both traffics aids to keep the fleet smaller and raise the utilization of locomotives, thus lowering the fixed costs.

To operate the train traffic VR owns locomotives of six locomotive types,half are powered by diesel and the other half by electricity. Both diesel and electrified fleets are roughly the same size, containing $150-200$ locomotives. Both electrified and diesel locomotives have one type allocated to specific circulations, that cannot be changed during the planning. This leaves 165 electrified locomotives and 166 diesel locomotives to be planned in long-term planning phase. The focus is on creating a plan for these 4 locomotive types in the thesis.

The driving using diesel locomotives is more expensive than with electrified locomotives. Therefore, long distances are preferably planned to electrified locomotives. Due to this difference in costs the diesel locomotives usually remain the whole week in relatively small area and very rarely travel across the Finland. On contrary to the electric locomotives can travel across Finland multiple times during the week, this makes the maintaining balance during the week most difficult task in the planning for electric locomotives. However, there are rail tracks that are not electrified and thus the trains there need to be always driven with diesel locomotives.

From the locomotive planning perspective, the passenger trains and the freight trains have a few key differences regarding the needs towards the locomotives. The two most critical factors are the maximum speed of a train and the weight of a train. Many of the passenger trains have such high maximum speed that they can only be pulled by one of VR's locomotive type and this covers already majority of the activities that this locomotive type performs during the week. Most of the passenger trains would be driven with electric locomotives solely based on the cost factor but the speed limits completely the option to plan them to diesel locomotives out. The second large difference between passenger and freight trains is the weight of the trains, on average the freight trains are heavier and require often two or three locomotives to pull the train. Because of these two factors the diesel locomotives work on average combined together by one or two other locomotives, whereas the electric locomotives perform most tasks as a single locomotive. The difference is due to the large proportion of passenger traffic being operated by electric locomotives and secondly by the fact that the pulling power of electric locomotives is much higher.

In addition to pulling the trains the locomotives are also used to perform *service operations*. Service operation means that the locomotive is ordered for a given railyard to organize the wagons to trains etc. Most of the service operations are performed with diesel locomotives whereas most of the trains are usually planned to electric locomotives. There are some railyards that are electrified in which the service operations are performed with electric locomotives.

The planning of locomotive allocation is made in three weeks cycles in the long-term planning phase. This three-week cycle is because the driver shifts must be published every three weeks and the personnel planning needs the information about the locomotive type used to drive the train and light travel routes, so they can be put to driver's shifts. The locomotive type is relevant for personnel planning because not all drivers have the qualification to drive all different locomotives. The traffic is different every week, mainly due to the track works and customers' needs especially in the raw wood freight. Thus, every week is planned separately. Each of the three weeks planning phases lasts for three weeks and takes place $5-7$ weeks before the start of the three-week cycle. However, the time to plan locomotive allocations using final data is only 6 days, as the three weeks require planning also from freight, railyard and driver planners. The 6 days are not divided equally between the weeks needed to plan. The time for planning of the first week is 4 days and the other two weeks must be finalized in 2 days. This means that there is very limited time window to analyze, plan and verify the final plan. The time window means that the run time of the optimization model needs to be such that is can be run multiple times in a day.

2.2 Problem constraints

The VR's planning problem has multiple important requirements to be considered in the long-term planning. Some of them are constraints from the physical world and the other come from the planning process. This section presents the key features the optimization must have to be utilized as part of VR's long-term planning process.

As stated VR has six different types of locomotives and four of them are planned during the long-term planning phase. Which locomotive type is assigned for each train is not known in advance, except for a few cases in which there is only one locomotive type that can operate the train. Thus, the different locomotive types cannot be planned separately, and all of them must be planned together. This yields to Constraint [1] for supporting multiple different locomotive types. Secondly, for all locomotive types there is limited number that can be used in the planning. For this a constraint is defined [2] for restricting the number of locomotives per type in planning.

The assigned locomotives must be suitable to operate the train. There are three key aspects for this. First, some trains cannot be operated with every locomotive type, clearest example from this is a train moving in unelectrified track section and it can only operated by diesel locomotive. Another example would be a high-speed passenger train that requires a locomotive that can travel $200km/h$, only 2 of VR's 6 locomotive types can reach this speed. The second aspect is that even when the locomotive type can be used to pull the train, it might need more than single locomotive. The most obvious reason for this is train weight but also the track elevation angle, length of the slope and speed of the train affect the number of required locomotives. For example, there might be a steep hill in the train's route and the pulling power of a single locomotive might not sufficient to pull the train up the steep hill alone. The only reliable method to test can certain $consist$, i.e. pulling unit, pull the train is to test it. VR has performed test drives on almost all track segments to determine the exact weight limits for all different consist. Therefore, as the number of needed locomotives can be greater than one, is must be possible to set a lower limit for assigned locomotives, this is our requirement [4]. The third aspect is setting upper limit for each train. In some locomotive types coupling one or more locomotives together raises the pulling capacity but lowers the maximum speed the train can travel. For example, coupling two electric locomotives together lowers their maximum speed from $200km/h$ to $160km/h$. If the train requires consist that can drive $200km/h$ allocating single locomotives is possible whereas allocating two is not. There is also a hard limit for each locomotive type on how many of them can physically be combined to consists. The hard limit is usually between $2-3$. Therefore, as there is always an upper limit on how many locomotives can be assigned to train, it must be possible to set an upper limit for each train how many locomotives can be assigned to it, this is our requirement [3]. The requirements [4] and [3] also fulfill the need for restricting the locomotive types that can be used to drive the train. Therefore, it is not created as separate requirement. For example, it can be stated that if locomotive type x is used to pull the train it needs to have at least 1 locomotive, but at most 0, thus the locomotive type cannot be used to pull the train.

The planning of suitable locomotives for each train is not enough, it must also be determined if there is a locomotive for the train. This can be broken into two requirements. First, there must be enough time for the locomotive(s) to travel from arriving train to departing train, this is Constraint [5]. The minimum time is time locomotives need to travel from the arriving train to the departing train minimum $turntime$. The minimum turntime can vary greatly depending on the arriving and the departing activity as well as about the location on which this is happening. At minimum turntime is 0 minutes, this happens for example when the arriving train is starting a service operation on the railyard and it's first duty is to move the wagons it is attached to. Also, if the arriving and departing trains are both passenger trains and the same passenger wagons are used to operate both trains the turntime is practically zero from the locomotive's perspective. The time between trains is only depend on the time that is needed for passengers to get in and out of the train. On the other end the turntime can be as long as $90 - 120$ minutes, this can happen for example when the arriving train is a passenger train that must be attached to power post. In this case locomotive must first move the wagons to power post and wait there when the wagons are attached to the power post (40 minutes). The next activity could be to form a consists with second locomotive, then the locomotive must be driven to locomotive yard to form the consist (30 minutes). After creating the consist the locomotives are driven to the departing train wagons, which can be kilometers away (20 minutes). Then the locomotives need to be connected to wagons and perform a brake test (30 minutes) and only after that the train is ready to depart. Thus, the total minimum turn time between this arriving train and departing train is 2 hours in this case. The last example also brought the requirement for consist busting [6] which means that the locomotives must be able to form and deform a consist. This is relatively common activity in major railway locations.

The second requirement is that there must also be a locomotive in the railyard for the train. In other words, it must be ensured that each arriving locomotive is assigned to exactly one departing task. Otherwise there might not be enough locomotives on a given time on a certain location. For example, if there is a location with one passenger train arriving and two departing freight trains, the arriving locomotive can only go to one of the departing freight trains not both. Thus, there must be an extra locomotive in the location before these events. Therefore, there is a need for a constraint to require that for every arriving locomotive there is exactly one departing allocation, this is Constraint [7]. In practice this means that it must always be known, which is the number of locomotives in each location per locomotive type and the number can never be negative to any of the locomotive types. The other option is to plan for each locomotive route that ensures that there is no double booking of locomotives. However, this is not required, as the information to which train the locomotive will go is not relevant at this stage of planning. Only that the railyard has enough locomotives to operate the traffic.

To achieve the sufficient number of locomotives in a location at given time, there is often a need to move just the locomotives between locations. Thus, the model must support option for locomotives to move between locations without a train, this is the requirement [8].

The planning process sets also two constraints. First one is for the computation time because there is only 2 days' time to finish the planning for two separate weeks. Thus, the optimizer must be able to obtain a solution quickly, as the plan needs to still be verified and most cases adjusted. Therefore, a goal is set for the reasonable computation time be at most 1 hour, this is Constraint [9]. With this computation time there is enough time to analyze the results and perform a second run if needed. The second requirement from planning process is that the plans for each week are assumed to be cyclic. In other words, this means that there should be as many locomotives in the end of the week as there where at the start of the week within each location and running train. This cyclicity of the plan is the requirement [10].

2.3 Cost factors

The costs are the most important indicator for good locomotive allocation plan. The most important cost factors come from locomotive energy consumption, maintenance costs and driver costs in light travel journeys. All of them are proportional to the distance travelled. The locomotive costs scale up with locomotive kilometers, but the driver costs scale up only to light travel kilometers. An example of the cost factors is shown in Figure 2.1. The trains naturally need driver also, but it is not costs that comes from locomotives allocation planning. The difference between locomotive and driver cost generation is because trains or light travels that need two or more locomotives generate costs for both locomotives. The train needs however only one driver and thus the costs for them are related to only train kilometers. This leads to two different objective function requirements: [11] for locomotive based costs and [12] for driver based costs. Both are equally important aspects in the final costs structure.

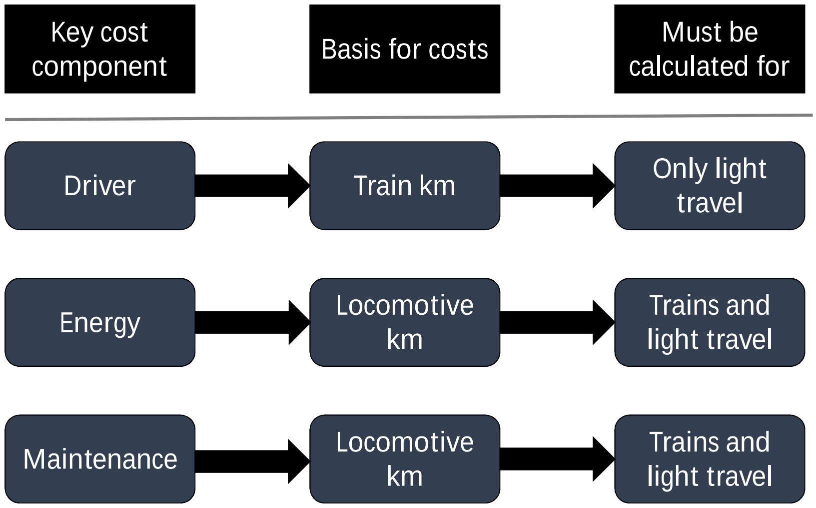

Figure 2.1. The breakdown of costs factors for light travel journeys and trains.

In addition to these costs, there are other costs mainly for minor operations that the locomotive performs or takes part of. They are however much smaller in scale than the costs regarding driving the trains or light travel. Two of the largest are consist busting and plugging the passenger trains to power post. The consist busting operation are mostly performed in large locations and there is usually local service driver who performs the consist busting operations. However, if there are too many, some of the consist busting operation must be planned to drivers, which leads to extra costs. In minor locations the driver must do the operation always, which increases working time and yields extra costs. The plugging of passenger wagons to utility pole is also operation that is normally performed only in large locations by the service driver. Thus, it does not usually lead to extra costs, but might if there are too many activities planned for the service driver. Because the costs from these two activity groups are relatively small and lead to extra costs rarely, thus they are not included as requirements to the model.

2.4 Summary of the requirements

Below the requirements for the model are summarized:

[1] There are multiple locomotive types

[2] Number of locomotives in each locomotive type is limited

[3] Maximum number of locomotives for each train can be set

[4] Minimum number of traction for each train can be set

[5] Minimum turntime between trains can be set

[6] Consist busting is allowed

[7] The number of locomotives in each station and at all times must be zero or greater

[8] Light travel is allowed

[9] Solution is found in a reasonable time

[10] The generated plan matches end of Sunday to start of Monday

[11] Costs of using locomotives are taken into account

[12] Light traveling is more expensive than balancing with excessive traction on trains

With these 12 requirements literature can be reviewed to find are there models sufficient to VR's planning problem.

3. Literature review

This chapter reviews literature to find most suitable optimization models for VR's LAP. To achieve this we review the optimization for LAPs, examine whether there is a suitable model in other fields of research and compare the most fitting models to the requirements presented in Section 2.4. Finally we decide if some of the models is suitable to be implemented to VR's LAP.

First, the most potential models are identified from the surveys performed on the LAP-field and study if there are potential other fields close to LAP that could be used. In the field of locomotive allocation problem there are two surveys published in the recent years. The surveys are done by Piu and Speranza [2014] and by Eskola [2016]. The bachelor thesis of Eskola is focused to the VR's planning problem and reviews in depth three potential approach to the problem. The survey done by Piu and Speranza is focused to the whole field and classification of models. In addition to these two, there is also an older survey by Cordeau et al. [1998]. However, it is over 20 years old and most of the studies are part of the survey by Piu and Speranza. Thus, the focus is in these two first surveys on the LAP-field.

There are fields that study similar problems. The closest of them is maybe the AAP, aircraft allocation problem. In LAP locomotives are planned to trains and in the AAP planes are planned to flights. The core in both problems is very similar. However, based on the survey of Marla et al. [2018] the aircraft model lack some of the key features such as the consist busting and light traveling. Additionally, the crew rostering is also optimized at the same time. Based on the master thesis of Porokka [2017] the crew rostering in the VR's environment is by itself extremely difficult task. Thus, the differences between AAP and LAP are too great for implementing the model from different field directly. Because of this the focus is solely to the field of locomotive allocation problems.

Based on survey by Piu and Speranza the models are divided into six categories. These are: solving type (optimization/heuristic), number of locomotive types, traffic type, model type, objective function and planning phase. These are also presented in Table 3.1. None of the models fit completely to our problems criteria, as the goal is to find an optimization model that minimizes the operation costs while supporting multiple locomotive types. Model also need to be developed for both passenger and freight traffic at long term planning phase. The modeling type of problem is not a limiting factor. If the attention is limited to those optimization models that allow multiple locomotives types to be planned and allow operating at least freight traffic while minimize the operating costs we end up with six candidates: Noble et al. [2001], Baceler and Garcia [2006], Ziarati et al. [2001], Powell and Bouzaiene-Ayari [2007], Ahuja et al. [2005], Vaidyanathan et al. [2008a]. In addition to these there are studies that are not in the study and match the criteria at least partly.

Table 3.1. The classification of LAPs in Piu et al. [2015]

Class			
Solving type	Optimization	Heuristic	
Nr. of locomotive types	Single	Multi	
Traffic type	Passenger	Freight	
Model type	Commodity flow	Allocation	
Objective function	Operational costs	Nr. of locomotives	Other
Planning phase	Day of operation	Operational	Strategic

The studies outside this article are Teichmann et al. [2015] that was studied in the bachelor thesis of Eskola, which is too simple and lacks many key features as the multiple locomotive types and light traveling and studies of Caprara et al. [2007] and Rouillon et al. [2006]. Both meet many of the requirement but are only focusing on the passenger trains, this is a large downside, as the passenger and freight traffic operate on different rules regarding the turntimes and consist busting. Also one of the key differences is that the passenger trains seldom need the light travel and most cases it is not included into solving process and secondly there is not an option to assign more than single locomotive to a train - simply because passenger trains never need more than one, for freight trains it is very common to have more than one locomotive. The model of Jaumard et al. [2014] matches the most of the criteria. It is an optimization model supporting multiple locomotive types and minimizing the costs. It was also selected as two of the most promising candidates for VR's planning problem in the Eskola. Additionally there is an article in the Handbook of Operations Research Applications at Railroads the author Vaidyanathan and Ahuja [2015]. The section of locomotive planning section however is not tested with real data and thus not directly applicable. It is also based heavily to authors previous models that would thus be better candidates for further research.

Based on the analysis from the surveys and looking from the studies outside the surveys and other fields, we take a further look to two studies that look most promising based on our criteria. These will be the Jaumard et al. [2014] and Ahuja et al. [2005]. From these we will evaluate the compatibly to requirements in the previous section and present shortly the background.

The model of Ahuja et al. [2005] was done in collaboration with Canadian railways. The model does fit well with the requirements. The modeling is done as multicommodity flow. We present multicommodity flow more in depth at the next chapter. The requirement [1] for supporting multiple locomotive types is in model. It is satisfied as multicommodity flow models naturally support multiple locomotives types. The requirement [2] for limiting the available number of locomotives is also supported. The number of used locomotives is calculated at single moment and limited there. As the model requires that the amount of locomotives is constant at all times, this limits the number of locomotives in the plan. The requirement [3] for limiting the maximum number of locomotives in a train is supported. This is easy to limit for each arc separately. The same is true for requirement [4] to set a boundary for minimum number of pulling locomotives in a train. This is however done by calculating the pulling power of locomotives, which cannot be the case in Finland, as different track sections have different pulling power requirements. This is not impossible to model differently but would require changes in the model. The requirement for [5] is supported in the model quite well. The consist busting operations are not modelled as such. Instead each arriving train has an arrival node, from which the locomotives move to the actual location node. The minimum turn time and consist busting times are taken into account between the arriving node and actual location node. For example, if a train A would arrive at 11:00 o'clock the arrival node is created to 11:00 o'clock. However, if there is a turntime of 30 minutes and 15 minutes reservation for consist busting operation the locomotives are freed to actual location node only after those at 11:45

o'clock. If there would have been a train departing 11:35 o'clock. There would have been an instant travel arc between the arrival node of train A and departure train B. The model must then decide does it use the instant travel arc or does it free the locomotives to the location. This complicates the problem quite much and probably is the reason for computation time issues. However, the requirement [6] for consist busting is supported in the model. The requirement [7] for balance in each location is also supported. The number of locomotives must be greater or equal to zero at all arcs at all times. Secondly the number of arriving locomotives per type must equal to the number of departing locomotives. This satisfies the requirement. The requirement [8] for light travel is supported by adding a predefined set of light travel arcs to the model. This does not allow free movement of locomotives at all times, unless we add them to every minute, but is definitely good enough. The requirement [9] for computation time is not satisfied by the model. The model was not able to find an optimal solution even in the 72 hours computation time. The requirement [10] for cyclicity is satisfied. The model links the arcs back to the first nodes. The requirement [11] for locomotive based costs is satisfied. The also makes a difference between pulling locomotives and transported ones. The requirement [12] for driver based costs is satisfied with the model. The use of light travel arcs triggers an extra cost. Overall the model matches the requirements well. The only non-satisfied is the computation time. However, that is the most crucial as the tight time window of 2 days planning time does not allow over 72 hours calculation time.

Regarding the performance, it is often difficult to estimate the actual performance before testing it. In this case the test sets were however quite close to the ones in Finland. The test sets contained roughly 3000 trains, that is quite the same as the number in Finland. The number of locomotives was however also 3000, which is much more than the fleet size in VRs possession. However, the number of arcs is more relevant because it determines the size of the network and thus the number of constraints. Also, the number of light travel arcs was much smaller roughly 100. In VRs plan the number of planned light travel arcs is closer to 500 in a week and the set of possible candidates even larger. That would indicate that the problem could be even harder to solve, as the resulting network is much larger. This puts even larger risks to computation time, which is the greatest downside in this model.

The authors discovered a way to overcome this weakness. They noticed that over 90% of the trains were identical on each weekday. When they changed the problem to single day, they were able to obtain a optimal solution. From this single day solution, they expanded it to all seven days with heuristics and maintained relatively good level of plan. Finland has less than 80% of the trains the same on each weekday. This might not seem much less but could be the critical difference.

As a summary the model of Ahuja et al. is a promising candidate. However, the performance is not good enough to be trusted. The trick they used to overcome this is not as well applicable to VR's problem and the model itself would require adjustments especially in the modeling of minimum and maximum number of locomotives.

The other interesting model developed by Jaumard et al. [2014] was made in collaboration with Canadian railway company. The model does fit well with the requirements. The modeling is done as multicommodity flow as with the model in the previous article. As such the requirement [1] for supporting multiple locomotive types is supported. The requirement [2] for limiting the available number of locomotives is also supported. However, the model counts the number of locomotives arriving to sink nodes. This is problem as the problem should be cyclic and converting the equations to circular problem, would also require changes in this. This complicates the adoption of model as this would need to be changed also if the problem is modelled as a cyclic. The requirement [3] for limiting the maximum number of locomotives in a train is supported. This is easy to limit for each arc separately. The same is true for requirement [4] to set a boundary for minimum number of pulling locomotives in a train. The requirement [5] for minimum turntime is modelled in the model. This is done in two stages and the first step they use a concept of $train\ string$. These mean that set of trains are combined to be driven with single consist and checked that the turntime is ok within them. This is practically the turntime calculation outside of the optimization. These train strings are iterated between the solutions to find a global minimum. For each train string they add a turntime including time for consist busting to the end of the string. This is rather efficient type of dealing with turntime. However, the requirement [6] for consist busting is supported in the model. The model goes even further and tries to minimize the amount of consist busting. The requirement [7] for balance in each location is also supported. The number of locomotives must always be greater or equal to zero at all arcs. Secondly the number of arriving locomotives per type must equal to the number of departing locomotives. This satisfies the requirement. The requirement [8] for light travel is not part of the study but is still supported by adding a predefined set of light travel arcs to the model. This does always not allow free movement of locomotives, unless we add them to every minute, but is good enough. The requirement [9] for computation time is on verge of satisfaction. The test sets were able to find a solution between $1 - 4$ hours, which is tolerable limit. The requirement [10] for cyclicity is not satisfied. The model uses source and sink nodes. The requirement [11] for locomotive based costs is satisfied. The also makes a difference between pulling locomotives and transported ones. The requirement [12] for driver based costs are not satisfied with the model. The use of light travel arcs does not trigger an extra cost. Overall the model matches the requirements quite well. The largest downsides are the requirements [2], [5] and [12]. Partly also the requirements [9] and [10].

The researchers were able to obtain optimal solutions in reasonable computing times. The test sets were quite similar in size compared to VR. The number of trains was between $1000 - 2000$ and number of locomotives around 1000 that is higher. The number of light travels also raises worries. It can quickly raise the size of the network. The exact performance in Finnish environment however remains a question mark, as in general the problem size is only one factor affecting the computation times.

The model takes into account the maintenance interval. This is not needed in the VR's long term planning and thus complicates the problem unnecessarily. It might be possible to drop out the feature, but it might have other affects to model.

To summarize this model, it is not suitable as such to VR's problem. It misses too many requirements and changing would require much work. The computation time however might not be an issue, as the maintenance feature could be dropped off and thus simplifying the model. The exact performance is still unknown because too many requirements were missing. From performance point of view the key open question would be the adding of driver costs.

As a summary from the whole literature review. There are no models that could be used as such to VR's planning problem. None of the models satisfy all requirements. The model of Ahuja et al. [2005] satisfied all but the computation time requirement. But it is too critical to be passed, as the planning process sets tight time window. However, all the requirements for model were included in some of the models. This helps the development of own model, which we do in the next chapter.

4. Formal definition to VR's planning problem

In previous chapter we concluded that there none of the models in literature can be applied directly for VR's planning problem. Thus we develop our own in this chapter. As we know little about the performance and based on literature review it is the largest risk, we develop two model variants. This chapter formulates the problem into mathematical format and develops the two model variants for the problem.

Before starting the development, we take a look on what information we need for the model. In the VR's locomotive allocation problem all trains have 8 attributes:

1. Departing time.
2. Departing location.
3. Arriving time.
4. Arriving location.
5. Distance of the train travel.
6. Possible locomotive types that can pull the train.
7. Minimum number of locomotives to pull the train per locomotive type.
8. Maximum number of locomotives that can be assigned to train per locomotive type.

4.1 Suitability of allocation and multicommodity flow models to VR's LAP

In literature there are two major ways to approach this problem. The first approach is the modeling the problem as a commodity flow problem and the second is modeling the problem as an allocation problem. The key difference between these two types is how the next activity for a locomotive is modelled. In the allocation method, each train will get the number of locomotives assigned to it. The next activity is linked directly to the activity and thus all locomotives are planned to the same next allocation. On the contrary, in the multicommodity flow problem the trains are modelled as arcs and each arc gets the number of assigned locomotives as in allocation method. However, after arriving to a location the locomotives are freed to a stock of that location, from which they can be freely distributed to the next task.

The allocation method has an advantage on the turntime calculation. It is a natural part of the pre-processing as the turntime can be easily checked because the next activity is exactly specified. However, this method is not very well suited to modeling of consist busting operations. It is mostly used in passenger traffic modeling in which each train only has one locomotive.

The multicommodity flow problem is the more used approach. Compared to the allocation method, it is better suited for taking care of light traveling and consist busting. The turntime calculations can be taken into account in the multicommodity flow problem, but based on the Ahuja et al. [2005] and Vaidyanathan et al. [2008b], they might be a computational risk. Furthermore, the minimum and maximum number of locomotives per train is easily supported by this method.

We select the multicommodity flow as it offers better support for more key requirements than the allocation method. The ability to perform consist busting and light travel, as well as the ease of defining minimum and maximum traction for multiple locomotive types outweights the potential ease of turntime modeling of the allocation method.

4.2 Construction of the multicommodity network

Now we start constructing the multicommodity flow network. It consists from two things: nodes and arcs. In the LAP, each time locomotive is moving either in time or "space", there needs to be an arc. Each point in which three (not two) or more arcs meet, there needs to be a node. The nodes have two attributes: location and time.

We model each train as an arc. Each arc has the attributes that the train possess, and each arc is connected to two nodes. One that is the departure of the corresponding train and the other that is the arrival of the train. In addition to train arcs, we have also two other types of arcs. The second type of arcs are the light travel arcs. They are similar to trains in all other ways, but they are not needed to be driven. Thus, using them is optional and but generates more costs, as they require a driver in addition to locomotive based costs. The third type of arcs is parking arcs that represent the railyard. These differ from train and light travel arcs in few ways. First of all, they only move through time, not "space". The trains and light travel arcs move through both time and space. The parking arcs are always between two consecutive nodes in the same location and contain the number of locomotives that are in the railyard during the duration of the arc. Similarly to the train and light travel arcs, the parking arcs also have the start and end nodes.

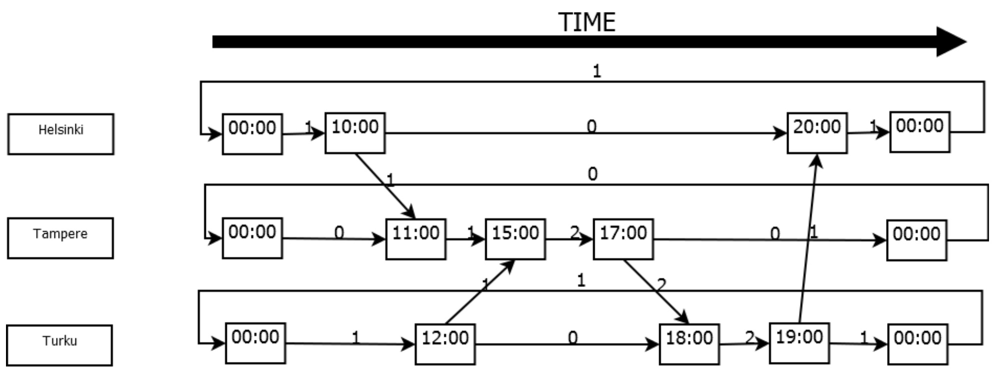

Figure 4.1. Example of commodity flow network in Finland. There are three locations, inside which the parking arcs connect the nodes to next one within location. The arcs crossing from location to location are the trains that need to be driven. Number above the arcs represent the number of locomotives in the arc.

These three types of arcs and the nodes form the space time network to which we construct our model. A visual example of the network is found in Figure 4.1. These types of problems are multicommodity flows and more information about them is available in the books of Bazaraa et al. [1990], Assad [1978] and Ahuja et al. [1993] that address the multicommodity flows.

Now we form a formal definition. We denote the nodes with a set of N, in which each node u has the attributes time $t(u)$ and location $l(u)$. There are no subsets in this set The arcs create a set A, which consists of three subsets. The subset A_T consists of all the trains arcs, the subset A_L consists of all the light travel arcs, and the subset A_P consists of the parking arcs. Additionally there is a subset A_M that contains all arcs that wrap from the end of the week to start of the week. Each arc has a departure node and an arrival node. Additionally, each arc has a distance and three vectors of length k, where $k \in K$ is the amount of different locomotive types. The vector element k in first vector is denoted by $Possible_k$, which is a binary number that represents whether the locomotive type k can be used for driving the train. The second vector has elements $Minimum_k$ that tells the number of locomotives of type k that are needed to drive the train. The third vector has elements $Maximum_k$ that tell the maximum number of locomotives of type k that can be assigned to the train. The list of attributes for both arcs and nodes is represented in Table 4.1. These two sets of A and N form the network $G(N, A)$.

Table 4.1. The attributes of nodes and arcs in the optimization model.

Node	Type	Arc	Type
$Location$	String	$Departure\ node$	Node
$Time$	Number	$Arrival\ node$	Node
		$Distance$	Number
		$Possible_k$	Vector, Binary
		$Minimum_k$	Vector, Number
		$Maximum_k$	Vector, Number

Additionally, we have the costs. For each locomotive type k, we have own kilometre-based cost c_k that contains the variable costs. The driver costs are denoted by c_{Driver} that defines the cost of a driver. This cost applies only to light travel arcs, where there is need for additional driver resource. For each train, we can assume there is already a driver and thus it has not an impact to the result.

There are no arcs going backwards in time, except those crossing the midnight between the Sunday and the Monday. Here, for each location, the last node in time is connected to the first one in time of the same location, in order to make the problem circular. In practice this means that each location must have the same number of locomotives at the start than at the end of the week. Trains crossing the midnight between the Sunday and Monday are connected like the next Monday would be the first. In practice this means that we subtract 7 days from the arriving time of all arcs crossing the midnight.

For each train, we must decide the driving locomotive type and the number of assigned locomotives. Thus, we define for variables that each $a \in A_T \cup A_L$ has a binary variable vector x, where $a(x_k) \in 0, 1$ states whether the locomotive type k is used as the pulling locomotive type. Additionally, define a vector y, where $y_k \geq 0$ gives the number of assigned locomotives of type k for the arc.

4.3 Defining the model variant using driver costs

This section shows the construction of the model variant using driver ($Variant\ D$) costs based on the requirements in Section 2.4.

We use the notation of (u, w) to denote an arc from node u to node w. This is used because some constraints demand to be precise about whether the arrival and departure nodes are the same with one or more arcs in the constraint.

We start the construction of model from the requirement [7]. The requirement stated that the number of locomotives must be always non-negative in each location and each arriving locomotive must be assigned to exactly one departing task. In terms of multicommodity flow, this means that no commodities can appear or disappear. Each locomotive type is it's own commodity in the model. This is a fundamental part of all commodity flows, with a small exception of sink or source nodes. In our case, the problem is however circular and thus there are no sink or source nodes. This means that everything that arrives to a node in the network must also depart from it. In other words, this means that the number of arriving locomotives of type k either from train arcs, light travel arcs or parking arcs must equal to the number of departing locomotives of type k. This must hold for every node in the network and therefore, we get

$$\sum_{(u,w)\in A} y_k(u, w) = \sum_{(u,w)\in A} y_k(w, v) \ \ \forall w \in N \wedge k \in K, \tag{4.1}$$

where u and v are any possible nodes. In this formula (u, w) represents all arcs arriving to the node w and The (w, v) represents all arcs departing from the node w. This satisfies the requirement [7], as each arriving locomotive is immediately assigned to a departing arc.

As a second constraint, we define a constraint that was not explicitly stated in the requirements but rises from the variables we selected. This is that all trains can have only one locomotive type assigned as pulling. In addition, they must have at least one pulling locomotive type. This means that there must always be 1 locomotive type assigned as pulling for all arcs in A_T. This yields the following

$$\sum_{k} x_k(u,v) = 1 \ \ \forall (u,v) \in A_T. \tag{4.2}$$

This constraint comes from the way we defined the variables and is not part of the original requirements it is more a real-world constraint that comes from the properties of VR's locomotive types. In some cases, in real-world, there can be multiple different types of locomotives pulling a train, but this is not considered in this model.

In addition to the previous constraint, we must ensure that the selected locomotive type is also a possible one. For example, we cannot assign an electric locomotive type to unelectrified tracks or a locomotive that has top speed of $120\ km/h$ for a train that travels $200\ km/h$. Thus, the selected locomotive type must be a possible one for the given arc. This must be true for all arcs and thus we get following constraint,

$$x_k(u,v) \leq Possible_k(u,v) \ \ \forall (u,v) \in A. \tag{4.3}$$

Then we can start going through the rest of the requirements. The requirement [1] is to support multiple locomotive types. We have defined the model to support k different locomotive types and thus this requirement is satisfied. The next requirement [2] is that the number of each locomotive type is limited. To limit the number of locomotives available in the plan, we count them in the arcs crossing the midnight between Sunday and Monday and add a restriction there for each locomotive type. Thus, we get an constraint

$$\sum_{(u,w)\in A_M} y_k(u,v) = Type_k \ \ \forall k \in K, \tag{4.4}$$

where $A_M \in A$ is the set of all arcs crossing the midnight between Sunday and Monday, and $Type_k$ is the number of locomotives of locomotive types k. This constraint satisfies the requirement [1], as the number of locomotives that can be used is limited per locomotive type.

The requirement [3] is that each train has a maximum number of locomotives that can be assigned to the train. To satisfy this we define an constraint,

$$y_k(u,v) \leq Maximum_k(u,v) \ \ \forall(u,v) \in A_T \wedge k \in K. \tag{4.5}$$

The requirement [4] states that each train must have enough locomotives to pull the train. As we have defined that the pulling locomotive types is stated in the variable x, we can define an constraint

$$y_k(u,v) \geq x_k(u,v) \cdot Minimum_k(u,v) \ \ \forall(u,v) \in A_T \wedge k \in K. \tag{4.6}$$

If the locomotive type k is assigned as pulling, the constraint simplifies to a format that the number of locomotives assigned to the train must be at least the minimum number of locomotives of type k to pull the train. On the other hand, if the locomotive type k is not selected, the right-hand side is 0 and the constraint is always satisfied.

The requirement [5] for minimum turntime between trains is handled in the next chapter. The requirement [6] for allowing consist busting is satisfied by the model. The locomotives are handled as separate entities and only the number of them is tracked. This allows the model to freely create consists and dismantle them. The requirement [7] for having the number of locomotives in each location to be always greater than zero. In this model, the number of locomotives in a location is stored in the parking arcs. For them, we have defined the variable y that is always non-negative. Together with Constraint (4.1) this satisfies the requirement [7] because the number of locomotives is always at least 0. The requirement [8] for light travel is also taken into account with the model, as they have their own subgroup A_L.

The requirement [9] for reasonable computation time is tested in the later chapters. The model also satisfies the requirement [10] for continuity from Sunday to Monday because the model is cyclic, which means that the arcs from the end of the week are connected to corresponding ones in the start of the week.

Before going to last two requirements we need to complete the model variant with two extra requirement that are not in the original list. If not denied, the model allows planning of locomotive $transports$. Transport means that the locomotive is attached to a train as a wagon and is not used to pull the train. This can be done as an operational decision, but transports are not planned in the long-term planning phase for trains and thus must be denied. The constraint for transport denial is

$$y_k(u,v) \leq Maximum_k(u,v) \cdot x_k(u,v) \;\; \forall (u,v) \in A_T \cup A_L. \tag{4.7}$$

This means that if the locomotive type k is not selected as pulling, the number of locomotives can be at most 0. If the locomotive type k is selected as pulling, the number of locomotives that can be assigned is the $Maximum_k(u,v)$

The second extra requirement is that if a light travel arc has assigned locomotives, it needs to have a pulling locomotive type assigned to the arc. This is separately needed for light travel arcs as Constraint (4.2) is only defined for train arcs. Without this extra requirement the objective function would not work properly. However, the previous constraint takes care of this as well. If a light travel arc (u,v) has a locomotive of type k assigned to it, Constraint (4.7) is not true, unless the type k is defined as pulling type. This allows the light travel arcs to have multiple pulling locomotive types. However, this is not a problem as the light travel arcs do not have timetables yet and their exact times are to be defined later. In this planning phase the timetable can be requested for each locomotive type separately and we can take into account the driver cost for each of these separate light travel arcs in the model.

Now we can construct the objective function to minimize the costs. There are two requirements for this part. First is the requirement [11] that states that the locomotive based costs must be taken into account. These mainly consist of energy and maintenance costs. The energy consumption is roughly linear to the distance travelled. The same is true for maintenance, as the maintenance intervals are based on the kilometer limits. Thus, we can define constraint

$$TrainCosts = \sum_{(u,w)\in A_T \cup A_L} \sum_{k\in K} y_k(u,v) \cdot Distance(u,v) \cdot c_k,$$

where $TrainCosts$ is the sum of locomotive kilometers travelled multiplied by the locomotive type specific cost factor c_k.

The second requirement [12] is that the light traveling must be more expensive than using existing trains. This is modelled with the driver costs that is assigned for each new light travel arc per locomotive type. Recall that we allowed more than one locomotive type be marked as pulling type for light travel arcs. Thus, we get

$$Driver = \sum_{(u,w)\in A_L} \sum_{k\in K} x_k(u,v) \cdot C_{Driver},$$

where $DriverCosts$ is the sum of all new light travel arc kilometers per locomotive types. The driver costs are actually much more complicated to calculate, as the drivers are paid the basic salary in any case. However, the billing is done based on driver kilometers and thus modeling them as kilometer-based is good enough way of modeling the costs.

The total costs are the sum of these two, which is minimized during the optimization.

$$Minimize(TrainCosts + DriverCosts).$$

Now our Variant D is ready. The model itself satisfies all the requirements, except the requirement [5] that is addressed in the next chapter and requirement [9] that can be only tested with real planning data.

4.4 Defining the model variant without driver cost

Because of the requirement [9] for computation time, we create an another model variant ($Variant\ N$) that is easier to solve. The calculation of driver costs independently from the locomotive-based kilometer costs complicates the problem, as there are more variables that affect the total costs. This model variant does not take driver costs into account, but makes the light traveling simply twice as expensive as the train kilometers. In practice it might give good enough solutions.

In this case, the constraints are equal to the previous one and the only difference comes in the objective function.

As for the objective function the part for $TrainCosts$ stays identical. Thus the first requirement [11] is satisfied as in the first model.

The other requirement [12] is that the light traveling must be more expensive than using existing trains. This is modelled by making the light travel kilometers twice as expensive. Thus our driver costs are the sum of locomotive kilometers travelled in the light travel arcs multiplied by the locomotive type specific cost factor.

$$LightTravelCost = \sum_{(u,w) \in A_L} \sum_{k \in K} y_k(u,v) \cdot Distance(u,v) \cdot c_k.$$

The total costs are the sum of these two, which is minimized during the optimization.

$$Minimize(TrainCosts + LightTravelCost).$$

Thus, our Variant N is ready. The model itself satisfies all the requirements, except the requirement [5] that is done in the next chapter, and partly the requirement [12] that is modelled not by train kilometers in the light travel arcs, but with locomotive kilometers.

5. Development of data pre-processing algorithms

As we left the requirement [5] out of the scope of the model variants we need a pre-processing algorithm to satisfy the requirement. Additionally, we need to define procedure for adding the light travel arcs and parking arcs to the data. This chapter constructs three different pre-processing algorithm versions to ensure that the the requirements are met, as we do not know the best way to take the requirement into account while still achieving results in reasonable time.

5.1 Development of minimal pre-processing

The lightest form of pre-processing consists of adding a set of light travel arcs, adding standard turntimes to each train, and creating the parking arcs to each location.

As stated before, the light travel arcs are created to ensure both the feasibility and the efficiency of the plan, otherwise the locomotive fleet might not be sufficient in numbers to operate the given set of trains. The locomotives need to be able to move between locations quite freely, so the number of light travel arcs are going to be quite high, even though only a small partition of them will be used in the plan.

We create the light travel arcs in two phases. The first phase consists of creating arcs related to the minor locations and the second phase consists of creation of light travel arcs between the major locations. In the first phase, we generate two lists, one containing all the arcs that end to a minor location and an another that contains all the arcs that start from a minor location. For those arcs that depart from a minor location light travel arcs are generated before each departing train. The time reserved between a train and a light travel arc is defined by a parameter (usually

$10-20$ minutes). Each minor location has also predefined options to which locations the light travel arcs are created, how long the travel time is, how long the driving distance is, and what are the possible locomotive types that can use the light travel arc. The same process is applied to the train arcs that arrive to a minor location. In the second phase, we generate the light travel arcs between the major locations based on a time interval parameter (usually 2 hours). This means that after every 2 hours we add a new set of light travel arcs from each major location to the neighboring major locations. As in the first phase, for each major station the possible light travel route destinations are listed and contain the same information about the travel time, distance and the possible locomotive types. After these two phases we have created all needed light travel arcs.

The turntime is added to end of each arc based on a standard turntime. Additionally, we have separate a turntime that is added only for the light travel arcs. In the beginning of each train there is a time reservation for the train related activities such as a brake testing etc. These activities we receive from VR and do not need a logic for adding them before trains.

In the final phase of the pre-processing we add the parking arcs. For each location all events that involve locomotive (=each end and start of an arc) are sorted by the time of the event and between two consecutive events we create a parking arc. After this we have done all phases that are required for minimal pre-processing.

5.2 Development of passenger pre-processing

The minimal pre-processing has only one standard turntime for every train to train connection. However, there are many passenger to passenger train connections, in which the turntime is significantly shorter and the locomotives are wanted to be kept attached to the passenger wagons. To tackle these cases, we define new phase to the pre-processing algorithm, before the creation of the light travel arcs.

The passenger pre-processing is based on the passenger wagon circulation. We assume that if the duration between two passenger trains that operate on the same wagons is less than the time required to perform all operations related in the switching of the locomotive to an another one, the train to train connection is worth of connecting to single arc. Making this choice leads us to combine most trains using the MVO-class wagons, as they have relatively high time requirement for switching the locomotive. The MVO-class means that the locomotive can be used for both pulling and pushing the train and thus never needs to be moved to the other end of the train. Even if there is a possibility to switch the locomotive that is done very rarely because the wagons need to be always kept electrified either via locomotive or via power post. Thus switching the locomotives generates much extra work for personnel that is wanted to be kept at minimum.

To link the trains with the wagon circulation we have a table that defines the wagon circulation and a parameter for each location and wagon class that tells the maximum time the trains can have in between in order to be combined to a single arc. The combining process is done by traversing through the wagon circulation train by train and after each train the time between the arriving and the departing train is checked and if it is shorter than the specific parameter the arcs are combined. In this phase, we also check the possible traction of both arcs and if necessary change them to such that all locomotive requirements are met.

5.3 Development of LiFo pre-processing

In addition to passenger train turntimes there are few similar cases for freight trains in smaller locations in which the standard turntime does not properly hold. These are cases in which the locomotive starts service operation immediately after arriving to the minor location and then departs immediately after service operation has ended. Combining these into single arcs prevents also the possibility of making a consists busting operation that is difficult to execute in minor locations, as they need to be done by driver alone. Secondly, this simplifies the problem and thus could lead to a better performance. However, this can also limit the options that optimizer has in moving locomotives between major locations.

The process is repeated for all minor locations and performed after the added passenger pre-processing phase. We define this pre-processing algorithm as following:

1. Select a set of arcs arriving to minor locations and sort them based on the end time
2. Start with the arc that has the earliest end time

3. Search the first departing arc after the arrival of the selected arc in the arrival location.

4. Check that the found arc has at least one same possible pulling locomotive type and range for possible number of locomotives than the arc selected in step 1. If this is not satisfied, then find a next departing arc after the current and repeat until suitable arc is found or end of the arcs list is reached.

5. Check if the found arc is within the location's specific maximum gap. If it is, then combine the arcs. Otherwise select the next arriving arc and go back to 2.

The result should be that we have very few arcs ending or starting in minor locations after this phase. Note that there is no need to perform this for departing arcs, as all possible pairs are found in this phase. For those few arcs that are not combined in this phase, the normal pre-processing creates needed light travel arcs.

6. Testing of model variants and pre-processing algorithms in different scenarios

In this chapter, we create the test scenarios, apply different pre-processing algorithms to them, optimize the test sets with both model variants, and report the obtained results. This is done to find out the most suitable model-algorithm combination in the next chapter. In this chapter, we focus on comparing the differences in produced plans between each model-algorithm combination within each test scenario, not yet comparing the overall performance of the combinations across the scenarios. The test scenarios that are based on a real set of activities that were planned for one week during Autumn of 2018 in the long-term planning phase. In the results we report the total costs of the plan, run time of optimization, small breakdown of the cost factors and discuss the other aspects of robustness. The main interest is in the amount of light travel, as it is the most important cost factor during this phase of the planning and a key component for the robustness.

The optimization models are solved using CPLEX that is run on a basic laptop that has Intel Core i7-6600 CPU using two $2.60\ GHz$ cores and $8Gt$ of RAM memory. The CPLEX version that is used is 12.8. on Windows 10 operating system. During the tests we do not use any other programs in the computer.

6.1 Creation of the test sets

We generate multiple test sets from the data provided by VR. The main things we want to investigate are the differences in performance, cost efficiency, and robustness between the plans produced by the two model variants and the three pre-processing algorithms. To investigate the performance, we divide the sets into two main categories, the ones that are one day-long and to the others that are week-long. In order to test other aspects, we create multiple different test scenarios, which each contain the 6 different test sets, as we have three different preprocessing algorithms for both week-long and day-long categories.

Naturally one test scenario is the planning of all four locomotive types at once. However, to examine the effects of different pre-processing algorithms and model variants have to the plan in different situations, we create three other test scenarios. First, we create a test scenario using only a single electric locomotive type, this allows us to examine the effects to light traveling and consist busting in more detail because they have highest light travel rate and highest consist busting rate of the electric locomotive type. As for second scenario we select only the two electric locomotive types, this allows us to examine especially the effect of passenger pre-prepossessing and tight turns, as most of the passenger trains are operated using these locomotives. As the last scenario we select both diesel locomotive types, this allows us to examine especially the effects of LiFo pre-processing because most of the minor location service operations are done using the diesel locomotives. The test layout is presented in Figure 6.1.

For brevity, we refer on these two chapters the different pre-processing algorithms in the tables as $pure$, pax, and $LiFo$ respectively meaning minimal pre-processing, passenger pre-processing, and LiFo pre-processing.

We use the number of nodes as the most representative for the size of the problem because the heaviest calculation is maintaining the balance within each node. Thus, it is better approximation for size than the number of arcs, as there is always at least three arcs connected to a node.

The first test scenario contains the sets for single electric locomotive type, this is the most common plan case according to VR. The details are shown in Table 6.1. The passenger pre-processing decreases the number of nodes by $4-5\%$ in this scenario and the LiFo pre-processing decreases the number of nodes by $29-31\%$ from the original size. For every set we add set of light travel arcs that is 2-3 times the size of the number of the train arcs.

The second test scenario contains the sets for two electric locomotive types, this is a planning case relative seldom according to VR. The details are shown in Table 6.2. The passenger pre-processing decreases the number of nodes by $18-19\%$ in this scenario and the LiFo pre-processing decreases

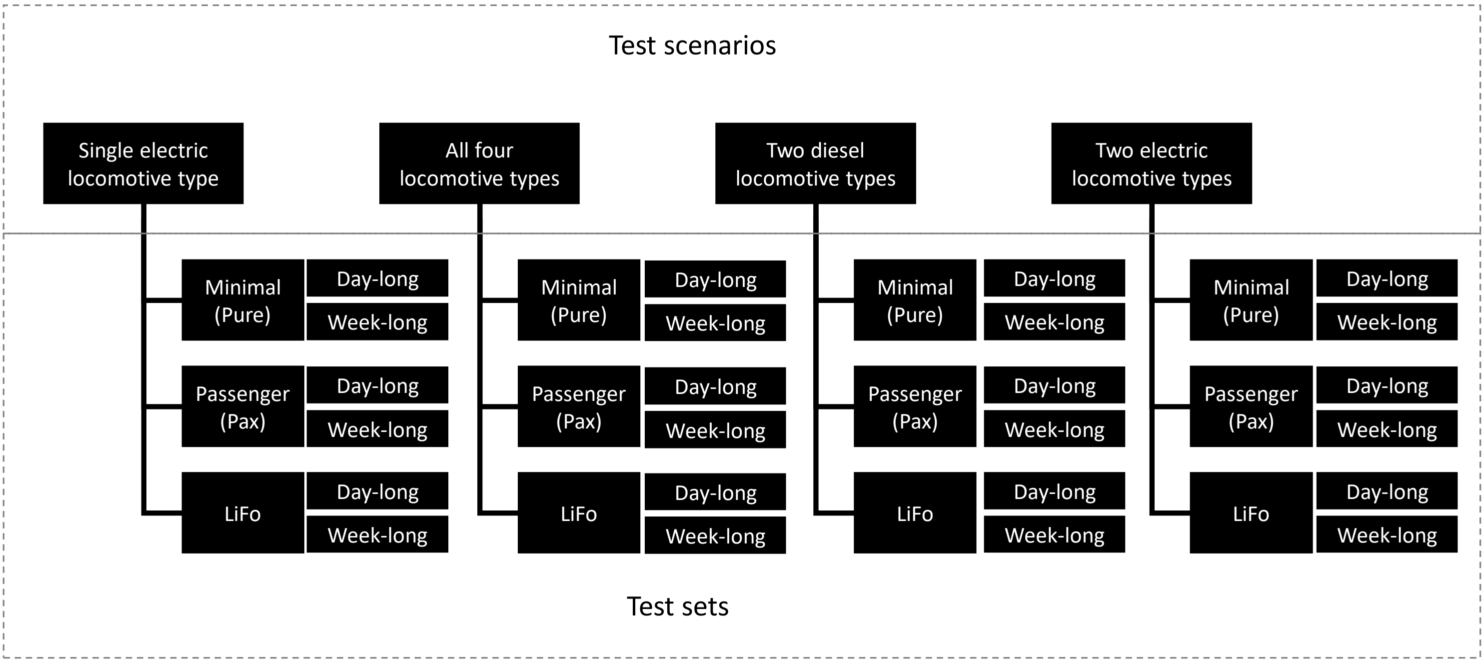

Figure 6.1. The test layout of test scenarios and sets.

Table 6.1. Information on test sets containing planning data for single electric locomotive type test scenario

	Day-long set			Week-long set		
.	Pure	Pax	Full	Pure	Pax	Full
Trains	224	224	224	1 432	1 432	1 432
Arcs	224	209	163	1 432	1 342	997
Lt-arcs	594	568	434	3 827	3 745	2 677
Nodes	1 303	1 236	931	8 347	8 050	5 692
Compression	100 %	95 %	71 %	100 %	96 %	68 %

the number of nodes by $41 - 47\%$ from the original size. For every set we add set of light travel arcs that is roughly 2 times the size of the number of the train arcs.

Table 6.2. Information on test sets containing planning data for two diesel locomotive types test scenario

	Day-long set			Week-long set		
.	Pure	Pax	Full	Pure	Pax	Full
Trains	431	431	431	2 824	2 824	2 824
Arcs	431	335	260	2 824	2 167	1 587
Lt-arcs	822	689	486	5 418	4 528	2 874
Nodes	2 055	1 675	1 204	13 462	10 882	7 133
Compression	100 %	82 %	59 %	100 %	81 %	53 %

The third test scenario contains the sets for two diesel locomotive types, this is the second most common planning case according to VR. The details are shown in Table 6.3. The passenger pre-processing does nothing in this set, as there are only a few passenger trains in the set. The LiFo pre-processing decreases the number of nodes by $30 - 31\%$ from the original size. For every set we add set of light travel arcs that is 2-3 times the size of the number of the train arcs.

Table 6.3. Information on test sets containing planning data for two electric locomotive types test scenario

	Day-long set			Week-long set		
.	Pure	Pax	Full	Pure	Pax	Full
Trains	204	204	204	1 117	1 117	1 117
Arcs	204	204	138	1 117	1 117	726
Lt-arcs	666	666	488	3 962	3 962	2 921
Nodes	1 399	1 399	974	8 008	8 008	5 522
Compression	100 %	100 %	70 %	100 %	100 %	69 %

The last test scenario contains the sets for all four locomotive types, this the quite common planning case according to VR. The details are shown in Table 6.4. The passenger pre-processing decreases the number of nodes by $12 - 14\%$ in this scenario and the LiFo pre-processing decreases the number of nodes by $43 - 48\%$ from the original size. For every set we add set of light travel arcs that is almost twice the size of the number of the train arcs.

Table 6.4. Information on test sets containing planning data for all four locomotive types test scenario

	Day-long set			Week-long set		
.	Pure	Pax	Full	Pure	Pax	Full
Trains	635	635	635	3 941	3 941	3 941
Arcs	635	539	392	3 941	3 284	2 297
Lt-arcs	1 200	1 067	662	7 379	6 479	3 707
Nodes	3 050	2 675	1 738	18 641	16 093	9 776
Compression	100 %	88 %	57 %	100 %	86 %	52 %

Now we have all test sets created and presented and we can move to testing the developed model variants and compare the effects of the different pre-processing algorithms to the results.

6.2 Optimization of the test sets

In this section we report results from the produced plans in different test scenarios. For Variant N, we report the costs calculated with the driver costs to make the results comparable to the other optimization model. The objective function of Variant D is selected, as it better represents the real costs that we aim to minimize.

For each test set we report the run time of the model, the costs, number of train kilometers, number of locomotive kilometers and total distance of used light travel arcs as well the total distance for locomotives that used the light travel arcs. These are reported for each pre-processing algorithm separately. Additionally we discuss the other aspects of robustness for the selected scenarios. The number of tight turns is evaluated in major locations when the locomotive count turns to zero. The consist busting operations are examined mainly in the minor locations when there are at least two locomotives at the same time present in the location.

Because we are interested in obtaining a solution in reasonable time, we limit the calculation time to 1 hour if the model does not seem to reach the optimal solution otherwise. If the optimizer has found solution by then it might still be an optimal, but the optimality is not proven. However, the lack of optimality is not a problem if the produced plan is still a cost efficient plan.

6.2.1 Single electric locomotive type

The test scenario for single locomotive type proved to be the second easiest problem to solve. The exact results are shown in Table 6.5. All the test sets generated based on this scenario are solvable within reasonable time by both model variants. For both model variants the LiFo pre-possessed test set is the fastest to solve and the one using minimal pre-processing is the slowest.

In terms of costs, the LiFo pre-possessed achieved the worst result with both model variants. Compared to the minimal pre-processing the passenger pre-processing diminished the costs in Variant N and in the week-long set for the Variant D. The explanation for this is that the passenger pre-processing allows some turntimes that are less than the standard turntime. Otherwise the explanation for the rise in costs comes is the increase in the light traveling kilometers.

This scenario was crafted to evaluate the performance on the consist busting aspect. In terms of that there are differences between models and pre-processing algorithms. Variant D produces a plan with significantly lower amount of consist busting operations in the minor locations. On the algorithm side the LiFo pre-processing generates the least consist busting operations, whereas the minimal and passenger pre-processing end up in stalemate on this category.

Table 6.5. The results of single locomotive type dataset. The Lt* means Light travel and represent the kilometer count of light travel that needs a driver. The Loc Lt* km is larger than this because there can be more than one locomotive traveling at the same time.

Variant N

Day-long	Pure	Pax	Full	Pax	Full
Time	9	8	7	-13 %	-22 %
Arcs	1 303	1 236	931	-5 %	-29 %
Cost	156 050	155 578	165 216	0 %	6 %
Train km	36 922	36 922	36 922	0 %	0 %
Lt* km	3 152	2 968	4 488	-6 %	42 %
Loc km	47 814	47 902	49 088	0 %	3 %
Loc Lt*km	3 674	3 914	5 578	7 %	52 %

Week-long	Pure	Pax	Full	Pax	Full
Time	88	80	63	-9 %	-29 %
Arcs	8 347	8 050	5 692	-4 %	-32 %
Cost	854 384	852 985	903 318	0 %	6 %
Train km	233 161	233 161	233 161	0 %	0 %
Lt* km	4 220	3 982	9 966	-6 %	136 %
Loc km	279 168	279 019	287 818	0 %	3 %
Loc Lt*km	4 932	4 672	13 308	-5 %	170 %

Variant D

Day-long	Pure	Pax	Full	Pax	Full
Time	13	10	8	-19 %	-35 %
Arcs	1 303	1 236	931	-5 %	-29 %
Cost	152 851	154 086	161 730	1 %	6 %
Train km	36 922	36 922	36 922	0 %	0 %
Lt* km	2 284	2 316	2 832	1 %	24 %
Loc km	47 905	48 274	50 134	1 %	5 %
Loc Lt*km	3 380	3 262	4 104	-3 %	21 %

Week-long	Pure	Pax	Full	Pax	Full
Time	146	131	80	-10 %	-45 %
Arcs	8 347	8 050	5 692	-4 %	-32 %
Cost	851 579	850 452	897 338	0 %	5 %
Train km	233 161	233 161	233 161	0 %	0 %
Lt* km	3 128	2 946	8 204	-6 %	162 %
Loc km	279 689	279 556	288 174	0 %	3 %
Loc Lt*km	4 302	4 280	12 472	-1 %	190 %

6.2.2 Two electric locomotive types

The test scenario for two electric locomotive types proved to be the second hardest problem to solve. The exact results are shown in Table 6.6. Every of the day-long test sets in this scenario is solvable within reasonable time by both model variants. However, the week-long test set is only solvable by Variant N, but Variant D produces still a better plan within the one hour calculation time. For both model variants the LiFo pre-possessed test set is the fastest to solve and the one using minimal pre-processing is the slowest. Thus, we tested if the LiFo pre-processed test set could be solved in longer calculation time but we terminated the calculation after 6 hours.

In terms of costs, the LiFo pre-possessed achieved the worst result with both model variants. Compared to the minimal pre-processing the passenger pre-processing increased the costs in all sets except in the week-long set with driver costs, where the produced solutions were equal in terms of cost efficiency. The largest explanation for the rise in costs comes is the increase in the light traveling kilometers. However, in this scenario the utilization rate of cheaper locomotive type also plays a minor role, but this only explains 10% of the cost increase between the LiFo and the minimal pre-processing, which means the effect is $0,4\%$ at the most in this scenario.

This scenario was crafted to evaluate the performance on the tight turns aspect. In terms of that there are no differences between models, nevertheless between pre-processing algorithms there are. The minimal pre-processing produces the plan containing the largest number of tight turns. This is estimated by searching the moments, in which the number of locomotive type turns to zero in a major location and examining these bottle necks more closely. The other two pre-processing algorithms perform better but there is no significant difference between them in terms of number of tight turns.

Table 6.6. The results of two electric locomotive types dataset. The Lt* means Light travel and represent the kilometer count of light travel that needs a driver. The Loc Lt* km is larger than this because there can be more than one locomotive traveling at the same time.

Variant N

Day-long	Pure	Pax	Full	Pax	Full
Time	12	10	10	-18 %	-18 %
Arcs	2 055	1 675	1 204	-18 %	-41 %
Cost	267 911	268 132	277 938	0 %	4 %
Train km	79 855	79 855	79 855	0 %	0 %
Lt* km	3 416	3 314	4 406	-3 %	29 %
Loc km	93 038	93 079	94 401	0 %	1 %
Loc Lt*km	4 266	4 176	5 508	-2 %	29 %

Week-long	Pure	Pax	Full	Pax	Full
Time	436	343	131	-21 %	-70 %
Arcs	13 462	10 882	7 133	-19 %	-47 %
Cost	1 685 627	1 690 462	1 756 798	0 %	4 %
Train km	527 000	527 000	527 000	0 %	0 %
Lt* km	13 428	13 902	20 292	4 %	51 %
Loc km	596 858	597 234	609 777	0 %	2 %
Loc Lt*km	15 006	15 770	24 740	5 %	65 %

Variant D

Day-long	Pure	Pax	Full	Pax	Full
Time	18	17	9	-9 %	-50 %
Arcs	2 055	1 675	1 204	-18 %	-41 %
Cost	262 365	263 894	273 882	1 %	4 %
Train km	79 855	79 855	79 855	0 %	0 %
Lt* km	1 800	1 840	2 504	2 %	39 %
Loc km	92 501	92 668	95 029	0 %	3 %
Loc Lt*km	2 898	2 972	3 788	3 %	31 %

Week-long	Pure	Pax	Full	Pax	Full
Time	3 720	3 720	3 720	0 %	0 %
Arcs	13 462	10 882	7 133	-19 %	-47 %
Cost	1 671 809	1 671 809	1 734 573	0 %	4 %
Train km	527 000	527 000	527 000	0 %	0 %
Lt* km	6 996	7 120	13 012	2 %	86 %
Loc km	597 622	598 053	609 929	0 %	2 %
Loc Lt*km	10 456	11 110	21 000	6 %	101 %

6.2.3 Two diesel locomotive types

The test scenario for two diesel locomotive types proved to be the easiest problem to solve. The exact results are shown in Table 6.7. Both model variant can solve every test set in this scenario within reasonable time. The LiFo pre-possessed test set is the fastest to solve for both model variants but this time the passenger pre-processed test set was the slowest in most cases. This is rather surprising, as the passenger pre-processing perform no arc combinations in this test scenario. For the week-long test sets the effect to calculation time is over 400% when it is optimized by Variant D. This explanation for this effect is unknown this is the only test scenario in which this occurred.

In terms of costs, the LiFo pre-possessed achieved the worst result with both model variants. As the passenger pre-processing does not perform any combinations in this scenario, the results between minimal and passenger pre-processed sets are identical. Apart from a small difference in day-long test set, which is explained by that there is a difference with single light travel arc that could have been chosen either way and Variant N sees no difference between these two options. Between LiFo and minimal pre-processed sets the largest explanation for the rise in costs comes from the increase in the light traveling kilometers. However, in this scenario the LiFo preprocessing also rises the amount of locomotive kilometers, which is caused by the combination of arcs in the minor locations.

This scenario was crafted to evaluate the performance on the tight turns and consist busting aspects. In terms of these, there are differences between both model variants and pre-processing algorithms. Variant D performs less consist busting operations than the other variant but between the number of tight turns there is no difference in this test scenario. Regarding the pre-processing algorithms, the minimal and passenger pre-processing end up in a stalemate in this test scenario. Nevertheless, LiFo pre-processing produces the plan with the least consist busting operations and performs equally in terms of number of tight turns compared to the other two algorithms.

Table 6.7. The results of two diesel locomotive types. The Lt* means Light travel and represent the kilometer count of light travel that needs a driver. The Loc Lt* km is larger than this because there can be more than one locomotive traveling at the same time.

Variant N

Day-long	Pure	Pax	Full	Pax	Full
Time	8	12	6	52 %	-23 %
Arcs	1 399	1 399	974	0 %	-30 %
Cost	283 680	284 056	303 186	0 %	7 %
Train km	25 622	25 622	25 622	0 %	0 %
Lt* km	1 042	1 136	1 892	9 %	82 %
Loc km	36 864	36 864	39 158	0 %	6 %
Loc Lt*km	1 530	1 530	2 824	0 %	85 %

Week-long	Pure	Pax	Full	Pax	Full
Time	128	118	54	-8 %	-58 %
Arcs	8 008	8 008	5 522	0 %	-31 %
Cost	1 135 941	1 135 701	1 255 323	0 %	11 %
Train km	99 316	99 316	99 316	0 %	0 %
Lt* km	1 536	1 476	7 576	-4 %	393 %
Loc km	146 607	146 607	160 693	0 %	10 %
Loc Lt*km	2 324	2 324	9 392	0 %	304 %

Variant D

Day-long	Pure	Pax	Full	Pax	Full
Time	8	9	7	12 %	-10 %
Arcs	1 399	1 399	974	0 %	-30 %
Cost	283 680	283 680	303 186	0 %	7 %
Train km	25 622	25 622	25 622	0 %	0 %
Lt* km	1 042	1 042	1 892	0 %	82 %
Loc km	36 864	36 864	39 158	0 %	6 %
Loc Lt*km	1 530	1 530	2 824	0 %	85 %

Week-long	Pure	Pax	Full	Pax	Full
Time	892	4 479	164	402 %	-82 %
Arcs	8 008	8 008	5 522	0 %	-31 %
Cost	1 135 450	1 135 450	1 251 344	0 %	10 %
Train km	99 316	99 316	99 316	0 %	0 %
Lt* km	1 250	1 250	6 286	0 %	403 %
Loc km	147 098	147 098	161 788	0 %	10 %
Loc Lt*km	1 986	1 986	8 708	0 %	338 %

6.2.4 Four locomotive types

The test scenario for all four locomotive types proved to be the hardest problem to solve. The exact results are shown in Table 6.8. Both model variant can solve every one of the day-long test sets within reasonable time. However, the week-long test set are not solvable optimally by neither of the model variants but both variants produce a plan within the one-hour calculation time. The LiFo pre-possessed test set is the fastest to solve for both variants, whereas the slowest set to be solved is the one using minimal pre-processing.

In terms of costs, the LiFo pre-possessed achieved the worst result with both model variants. The passenger pre-processing increased the costs in both sets optimized by Variant D when compared to the minimal pre-processing, however when the sets were optimized by Variant N, the costs passenger pre-processing achieved lower costs. The largest explanation for the rise in costs comes from the increase in the light traveling kilometers. However, in this scenario the utilization rate of cheaper locomotive type also plays a minor role but this effect only explains $5 - 10\%$ of the cost differences between the LiFo and the minimal pre-processing, which means the effect is $0,3 - 0.6\%$ at the most in this set.

Table 6.8. The results of four locomotive types dataset. The Lt* means Light travel and represent the kilometer count of light travel that needs a driver. The Loc Lt* km is larger than this because there can be more than one locomotive traveling at the same time.

Variant N

Day-long	Pure	Pax	Full	Pax	Full
Time	58	27	11	-53 %	-81 %
Arcs	3 050	2 675	1 738	-12 %	-43 %
Cost	546 738	546 241	577 330	0 %	6 %
Train km	105 477	105 477	105 477	0 %	0 %
Lt* km	4 146	3 876	5 708	-7 %	38 %
Loc km	129 424	129 202	133 635	0 %	3 %
Loc Lt*km	5 226	4 896	7 848	-6 %	50 %

Week-long	Pure	Pax	Full	Pax	Full
Time	3 720	3 720	3 720	0 %	0 %
Arcs	18 640	16 092	9 775	-14 %	-48 %
Cost	2 824 643	2 819 463	2 998 859	0 %	6 %
Train km	626 316	626 316	626 316	0 %	0 %
Lt* km	16 030	15 408	27 738	-4 %	73 %
Loc km	745 677	743 889	770 437	0 %	3 %
Loc Lt*km	19 110	18 414	33 348	-4 %	75 %

Variant D

Day-long	Pure	Pax	Full	Pax	Full
Time	120	96	37	-20 %	-69 %
Arcs	3 050	2 675	1 738	-12 %	-43 %
Cost	540 942	542 550	574 083	0 %	6 %
Train km	105 477	105 477	105 477	0 %	0 %
Lt* km	2 332	2 584	4 372	11 %	87 %
Loc km	128 990	129 072	133 786	0 %	4 %
Loc Lt*km	3 818	4 082	6 564	7 %	72 %

Week-long	Pure	Pax	Full	Pax	Full
Time	3 720	3 720	3 720	0 %	0 %
Arcs	18 640	16 092	9 775	-14 %	-48 %
Cost	2 799 420	2 811 084	2 976 503	0 %	6 %
Train km	626 316	626 316	626 316	0 %	0 %
Lt* km	7 736	8 668	18 226	12 %	136 %
Loc km	743 641	744 603	769 979	0 %	4 %
Loc Lt*km	11 902	12 808	27 798	8 %	134 %

7. Comparison of the optimization model variants and pre-processing algorithms

Chapter 6 used four test scenarios to compare the two model variants and pre-processing algorithms in terms of cost-efficiency and robustness. The third criterion of computational performance was found to be satisfactory, as it was able to produce a plan within the one-hour time limit, and therefore need not be considered. In order to determine the best combination of model variant and pre-processing algorithm, this chapter first selects the most suitable model variant and then a pre-processing algorithm is identified for the selected model.

7.1 Comparison between the two model variants

The differences in the costs efficiency of the two model variants are presented in Figure 7.1. The figure shows that in each of the test sets, Variant D performs better or at least as well as Variant N. The smallest difference (0% and 0.3%)was observed in the diesel test scenario, while the largest difference (1.1% to 2.9%) was found in the test scenario for two electric locomotive types. Thus, Variant D performs better in terms of cost-efficiency, as it always produces better or at least as good a solution as the variant without driver costs.

The second criterion is the robustness of the plan, which consists of three factors: the number of light traveling kilometers, consist busting operations (especially in minor locations), and tight turns. The amount of light travel is the easiest criterion to compare and is significantly lower in Variant D. The results of this comparison are shown in Figure 7.2. The difference (0%-22%)is the smallest in the diesel test scenario, with the largest difference ($76\% - 110\%$) being observed in the scenario for two electric locomotives types. Thus, Variant D performs better or at least as well as the other

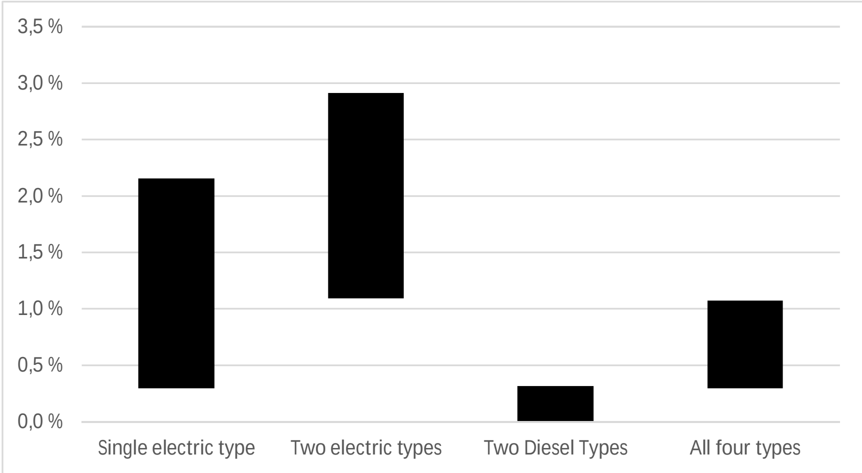

Figure 7.1. The comparison of costs differences between the optimization models. The black box shows the relative change range of costs in different scenarios. The results are reported as an increase from the model variant using drivers costs.

variant in every test scenario and therefore is better in terms of the amount of light travel.

The amount of consist busting was examined in the test scenarios for the single electric locomotive type and for two diesel locomotive types. Based on the findings presented in Sections 6.2.1 and 6.2.3, the Variant D performs consistently better or at least as well in every test set. The difference is lowest when using the LiFo pre-processing, as the number of consist busting operations is small for all test sets in the evaluated minor locations. However, when using other pre-processing algorithms, the difference tends to favor Variant D.

The number of tight turns was examined in the test scenarios for both two electric locomotive types and two diesel locomotive types. In the test scenario for two electric locomotive types in Section 6.2.2, the focus was on the tight turns occurring at major locations, in which no large differences were observed between the two model variants regardless of the pre-processing algorithm used. Likewise, in the diesel test scenario in Section 6.2.3 no clear difference was seen in the number of tight turns at minor locations. Therefore, it can be concluded that both models perform equally well in this aspect.

At this point, it is relevant to discuss the possible effect of changing the

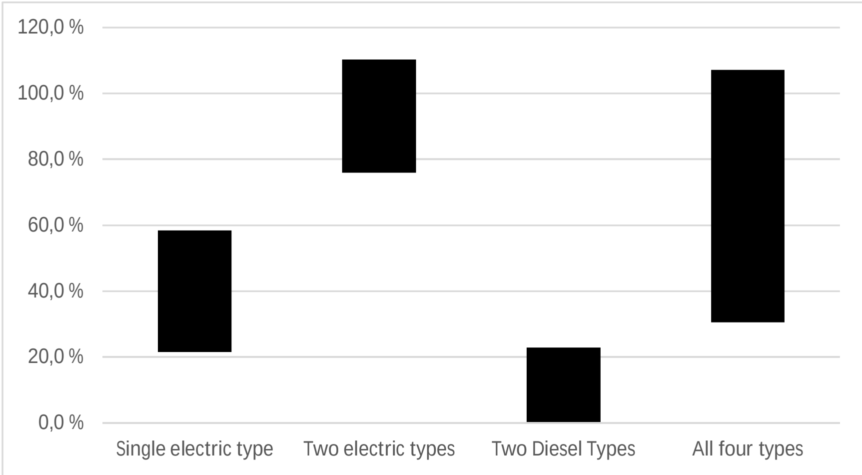

Figure 7.2. The comparison of amount of used light travel arc km differences between the model variants. The black box shows the relative change range of costs in different scenarios. The results are reported as an increase from the model variant using drivers costs.

size of the penalty parameter for light travel in Variant N. In the tests, the parameter was 1 but could have been any other value as well. However, changing the penalty parameter would not change the outcome of the above comparisons. The reason for this is that the largest difference was found between the results produced by the model variants for light traveling in terms of both costs and robustness. In general, the difference between the light travel kilometers for locomotives is less than 5%, while the difference in the amount of light travel arc kilometers used was over 20%. Adjusting the penalty parameter would have no effect on the comparison results. This can be best seen in the case when two locomotives arrive at a location and both continue from there via light travel arc or arcs. Variant N easily directs them into two separate directions, whereas the other variant directs them into the same direction, thus saving driver costs. An example of this is presented in Figure 7.3. In this example, changing the cost parameter for light travel penalty does not affect the solution produced by model variants. The number of locomotive light travel kilometers is somewhat higher in Option B; however, when adding the driver cost, it becomes the cheaper one. The same example explains the reason for the higher consist busting rate in Variant N. Thus, changing the value of the penalty parameter has no effect on these results.

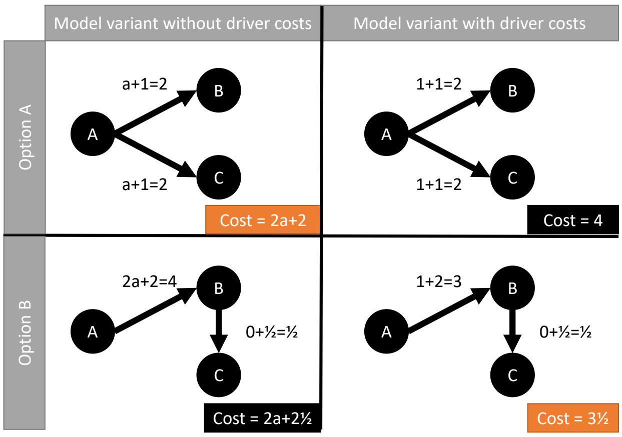

Figure 7.3. The example of model variants producing different result. The a is the penalty parameter for light travel.

To summarize, Variant D is clearly better suited for production use because it produces a plan with lower costs and outperforms the other model variant in terms of robustness. The amount of light travel and consist busting is smaller in the plans produced by the selected model variant, while the number of tight turns is roughly the same for both model variants.

7.2 Comparison between the three pre-processing algorithms

In this section, the plans produced by the selected model variant are used to compare the pre-processing algorithms in terms of cost-efficiency and robustness. Comparison of the three algorithms based on cost-efficiency is presented in Figure 7.4. The figure shows that the minimal pre-processing algorithm produces the best results, whereas the LiFo pre-processing algorithm produces the worst results of the three.

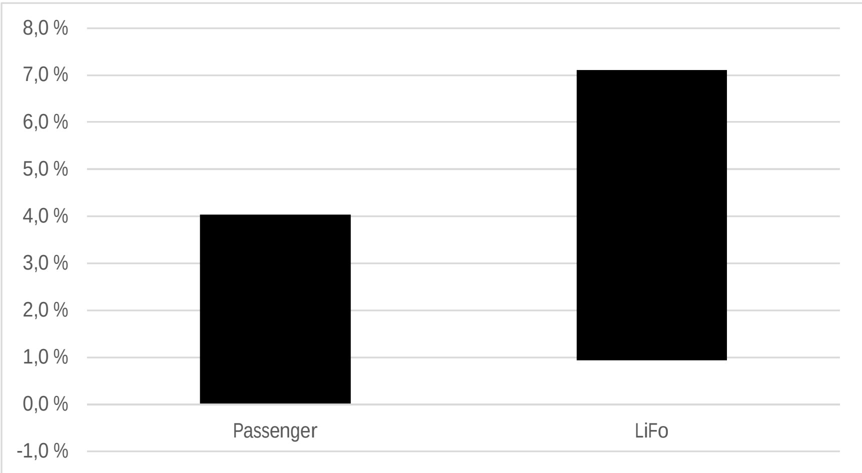

Figure 7.4. The comparison of costs differences between the pre-processings algorithms. The change is shown as a relative change to the of minimal pre-processing. The black box shows the range of change in all sets optimized using the model variant that containded the driver costs.

As in Section 7.1, The same indicators are used for robustness. The results for light travel comparison are shown in Figure 7.5. For this criterion, both minimal pre-processing and passenger pre-processing algorithms can produce the best results, depending on the test scenario, whereas the LiFo pre-processing is found to be the worst algorithm in all scenarios. Therefore, two algorithms can be chosen as the best candidate for this criterion.

The amount of consist busting was examined in the test scenarios of single electric locomotive type and the scenario for two diesel locomotive types. The findings in Sections 6.2.1 and 6.2.3 demonstrate that the LiFo pre-processing algorithm performs significantly better in both test scenarios than the other two algorithms. The passenger pre-processing and minimal

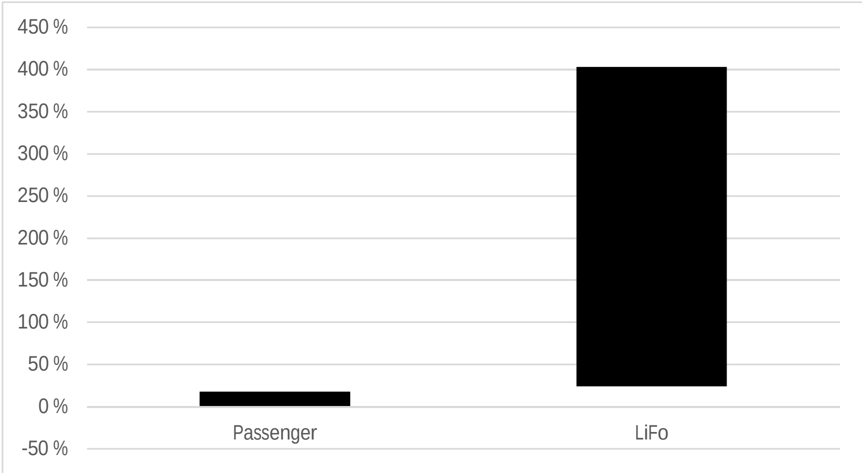

Figure 7.5. The comparison of amount in used light travel arc km between the pre-processing algorithms. The change is shown as a relative change to the of minimal pre-processing. The black box shows the range of change in all sets optimized using the model variant that contained the driver costs.

pre-processing algorithms differ little regarding this aspect. Thus, the LiFo-preprocessing algorithm offers superior performance in terms of the number of consist busting operations.

The number of tight turns was examined in both the test scenarios for the two electric locomotive types and the two diesel locomotive types. Based on the findings in Sections 6.2.2 and 6.2.3, the minimal pre-processing performs significantly worse in both test scenarios than the other two algorithms. For the remaining two algorithms, little difference was found. Thus, the two best algorithms according to this criterion are LiFo and passenger pre-processing algorithms.

It is impossible to select the best pre-processing algorithm in terms of robustness, as all algorithms differ in their strengths and weaknesses. Nevertheless, as the minimal pre-processing is actually a particular case of passenger pre-processing, and the passenger pre-preprocessing is a particular case of the LiFo pre-processing, any of the algorithms can be easily implemented, thus eliminating the need to select a single pre-processing algorithm.

7.3 Conclusion

In summary, Variant D is selected, as it performs better in terms of both cost efficiency and robustness. Unfortunately, it is not possible to select one pre-processing algorithm over the others, as they all excel and at the same time show weaknesses in at least one aspect of the selection criteria. This is not problematic, as the minimal and passenger pre-processing algorithms are just particular cases of the LiFo pre-processing algorithm. This allows the planners the freedom to adjust the parameters and select the best algorithm for a given situation.

8. Practical experiences

This chapter discuss the success of selected model-algorithm combination in VR's planning process. The selected optimization model and pre-processing algorithm were taken in VR's planning process with success. In addition to adjustment of the pre-processing parameters, the number of available locomotives was limited to less than the real fleet size. The long-term planning is made usually using an assumption that roughly 10 percent of the locomotives are reserved for maintenance and depending on locomotive type, $2 - 20\%$ for extra traffic not known at the time of the long-term planning. The adjustment of available locomotives proved to speed up the calculation significantly. Every set VR's planned during the first six weeks was solved in less than 15 minutes. The most common planning set of single electric locomotive type was solved in $2 - 5$ minutes for each week in the three-week test period.

The model was able to satisfy all the set requirements. The plan was feasible, cost efficient, robust and generated in a reasonable time. After the model variants and pre-processings algorithms were developed, the largest uncertainties were with the light travel and turntimes but planners deemed both to be working fine in VR's planning process. The light travels produced pleased the planners in most cases. However, there were a few times when the light travel was needed to redirect to some other direction, but these cases were caused by the home location of a driver. If the driver could have driven the suggested light travel route, the plan would have been fine. The greatest proof for success is that the developed model was taken as part of the planning process.

The model was also suitable for strategic planning as the model makes it easy to create a viable locomotive allocation plan for different train traffic sets. Additionally, it was found out that the model fits quite well to empty wagon planning problem as such, thus creating even more potential benefits. These were additional benefits that were not an objective of this thesis.

9. Conclusion

The aim of this thesis was to develop an optimization model for addressing VR's long-term locomotive allocation planning problem. To reach the goal, the thesis defined VR's planning problem, reviewed literature, developed an optimization model and tested it using VR's planning data. As there was uncertainty concerning the performance of the optimization model, two optimization model variants and three different pre-processing algorithms were developed to reach the goal. The different candidates were tested using VR's planning data in multiple test scenarios to determine the best combination in terms of performance, feasibility and cost efficiency. The best combination was chosen to be implemented into VR's planning process to further verify model.

The results of the practical experiences confirm that the optimization model produced efficient and feasible solutions within 2-15 minutes. This gives the planners more time to finalize the plan and enables them to react to large, sudden changes in train traffic. Most significantly, the optimization model shortened the time it would take to make a new plan from scratch. The time for this operation was reduced from several days to a few hours. The capability of quickly obtaining good solutions also affected the strategic analysis, as the model makes it easier and much faster than previously to analyze the effects of different train traffic scenarios.

The most important factors for success were found to be the proper modeling of different turn times between trains using the pre-processing algorithm and adding the driver cost for light travel arcs. These two components heavily influenced not only the cost efficiency but also the feasibility and robustness of the plan. Additionally, it was found during the testing that the sorting of data before starting the optimization has a huge impact on computation time. This was most evident in the optimization of diesel locomotives, where the change in sorting order increased the solving time from 15 minutes to 75 minutes. This effect should be investigated further.

Even though the results proved to be excellent, the model could still be improved. Although all the defined requirements were met, new aspects arose during the thesis. The most important one would be to create an activity for consist busting, the activity of combining locomotives together or taking them apart. this addition would make it possible to define both time and cost for the operation, as well as to limit the time and place where they can be performed and the number of simultaneous activities. Modeling of maintenance and fueling cycles would come as the next priority after that for further studies.

Bibliography

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice hall, 1993.

R. K. Ahuja, J. Liu, J. B. Orlin, D. Sharma, and L. A. Shughart. Solving real-life locomotive-scheduling problems. *Transportation Science*, 39(4):503–517, 2005.

A. A. Assad. Multicommodity network flows—a survey. *Networks*, 8(1):37–91, 1978.

A. Baceler and A. Garcia. An optimization approach to the locomotive scheduling problem in brazilian railways. *Rail Achieving Growth: Conference Proceedings*, 2006.

M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali. *Linear Programming and Network Flows*. John Wiley, 2nd edition, 1990.

A. Caprara, L. Kroon, M. Monaci, M. Peeters, P. Toth, C. Barnhart, and G. Laporte. Passenger railway optimization. *Transportation*, 14:129–187, 2007.

J. F Cordeau, P. Toth, and D. Vigo. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32(4):380–404, 1998.

J. Eskola. Optimization of locomotive allocation in finland. Bachelor thesis Aalto University, 2016.

B. Jaumard, H. Tian, and P. Finnie. Locomotive assignment under consist busting and maintenance constraints. In *2014 Joint Rail Conference, JRC 2014*. American Society of Mechanical Engineers (ASME), 2014.

L. Marla, V. Vaze, and C. Barnhart. Robust optimization: Lessons learned from aircraft routing. *Computers and Operations Research*, 98:165–184, 2018.

D. H. Noble, M. Al-Amin, and R. G. J. Mills. Production of locomotive rosters for a multi-class multi-locomotive problem. *Journal of the Operational Research Society*, 52:1191–1200, 2001.

F. Piu and M. G. Speranza. The locomotive assignment problem: A survey on optimization models. *International Transactions in Operational Research*, 21 (3):327–352, 2014.

F. Piu, V. Prem Kumar, M. Bierlaire, and M. G. Speranza. Introducing a preliminary consists selection in the locomotive assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, 82:217–237, 2015.

A. Porokka. Train driver rostering in finland considering driver satisfaction. Master's thesis, Aalto University, 2017.

W.B. Powell and B. Bouzaiene-Ayari. Approximate dynamic programming for rail operations. *Internationales Begegnungs- und Forschungszentrum für Informatik*, 7, 2007.

S. Rouillon, G. Desaulniers, and F. Soumis. An extended branch-and-bound method for locomotive assignment. *Transportation Research Part B: Methodological*, 40(5):404–423, 2006.

D. Teichmann, M. Dorda, K. Golc, and H. Bínová. Locomotive assignment problem with heterogeneous vehicle fleet and hiring external locomotives. *Mathematical Problems in Engineering*, 2015, 2015.

B. Vaidyanathan and R. K. Ahuja. *Locomotive scheduling problem*, volume 222 of *International Series in Operations Research and Management Science*. Springer Science, 2015.

B. Vaidyanathan, R. K. Ahuja, J. Liu, and L. A. Shughart. Real-life locomotive planning: New formulations and computational results. *Transportation Research Part B: Methodological*, 42(2):147–168, 2008a.

B. Vaidyanathan, R. K. Ahuja, J. Liu, and L. A. Shughart. Real-life locomotive planning: New formulations and computational results. *Transportation Research Part B: Methodological*, 42(2):147–168, 2008b.

K. Ziarati, F. Soumis, J. Desrosiers, and A. Saintonge S. Gelinas. Locomotive assignment with heterogeneous consists at cn north america. *European Journal of Operational Research*, 97:281–292, 2001.