Faculty of Electrical Engineering, Mathematics and Computer Science
Department of Applied Mathematics
Stochastic Operations Research

MASTER THESIS

STOCHASTIC OPTIMIZATION OF THE DIAL-A-RIDE PROBLEM

DEALING WITH VARIABLE TRAVEL TIMES AND IRREGULAR ARRIVAL OF REQUESTS IN THE PLANNING OF SPECIAL TRANSPORT SERVICES

Inge Tensen

Assessment committee:
Prof. dr. R.J. Boucherie (UT)
Ir. B.H. Hoeksma (HHM/TL)
Dr. J.C.W. van Ommeren (UT)
Dr. B. Manthey (UT)

BUREAU hhm

TimesLab onderzoek & innovatie

UNIVERSITY OF TWENTE.

May 1, 2015

Contents

Abstract

Special transport services are designed to meet the needs of people who are for various reasons unable to use conventional means of transport. Examples are elderly people or people with disabilities going to a day centre or children visiting special needs schools. In general the special transport used by these groups should be flexible and does not follow fixed routes or schedules. Planning of these transport services is difficult because of all the constraints that need to be taken into account. All persons that need to be transported have their own pickup and destination location, a time window at pickup and/or at arrival and a maximum ride time. Furthermore, in many cases the passenger may also have a wheelchair, may need a special driver or may be unable to ride in the same vehicle as certain other people.

In a mathematical context the planning problem of special transport services is called the ”dial-a-ride problem”. The dial-a-ride problem is a generalization of the pickup and delivery problem with time windows, in which people are transported instead of goods. Since the travel time between each two locations, as well as the arrival of requests, are both not exactly known in advance, stochastics are included to deal with these uncertainties. If the travel time is modelled as a random variable, the model makes sure that both the arrival within the time window as well as the maximum ride time, are satisfied with a minimum reliability.

Another uncertainty is the arrival of the requests. Part of the requests are static, they are known before the planning is made. The other part consists of dynamic requests that arrive during the day and need to be integrated into the static schedule. In this research a model is presented that takes the arrival of the dynamic requests into account while making the static schedule. In some cases it may be better to reserve specific vehicles for the dynamic requests, instead of planning them in the same vehicles as the static ones. To examine this switching point a machine scheduling problem is used.

1 Introduction

For most people in the Netherlands traveling is part of their daily routine. They can travel to work, school, family or any other destination in their own car, by public transport, by bike or on foot. However, for some people getting somewhere is less easy. People with disabilities or elderly people often cannot use public transport nor ride their own car. They are dependent on special transport services regulated by the authorities. Other users of these services are, for example, children visiting a special needs school or people going to a sheltered workshop. In most cases the special transport services are carried out by buses picking up people from their homes or care institutions and bringing them to their respective destinations.

All these different types of people cannot be transported at random times and all together. Each person needs to be picked up or dropped off within a certain time window. If a person needs to be transported to a school, day centre or sheltered workshop, the time window is defined by the opening hours of this destination. If a person is using a less regular type of transport, e.g. for visits to family or for medical consults, there is usually a time window at the pickup location of 15 minutes before and 15 minutes after the desired pickup time, to give some planning flexibility. Another time constraint is the maximum ride time of each person, which is usually about an hour.

The capacity of the vehicles is another important constraint. There are different kind of vehicles with different capacities. Some vehicles can only transport persons and others also have a wheelchair capacity. These wheelchairs also bring extra complications. Often the wheelchair that went in first, has to go out at the end because of space in the vehicle, following a first in last out discipline. There are also different types of wheelchairs, some can be folded and some take more space. Furthermore, some users of the system may need a special driver, which is only available in certain vehicles, or need an attendant.

At this moment most types of special transport services in the Netherlands are separately carried out by different parties/service providers. Among other things this means that the vehicles bringing children to a special needs school are other vehicles then the ones bringing elderly people to a day centre. Transportation could be performed more efficiently if different types of special transport would be combined. The most efficient schedule could be obtained by allowing the transportation of all types of users together in the same vehicle. This vehicle could then pickup people living in the same neighbourhood and bring them to their different destinations. In practice however, this is not always allowed. One can imagine that it is not desirable to transport sixteen year olds going to school together with mentally disabled persons. This makes it quite difficult to combine several types of users in the same vehicle.

Another way of combining types of special transport services is to transport different types of passengers after each other in the same vehicle. This still decreases the number of vehicles needed in total, without the problem of combining different kind of groups. In practice this is only possible, however, if these different groups would need transport at different times. Right now most transportations take place in the morning or afternoon, when schools, day centres and sheltered workshops are opening and closing. As a result, it is not possible to transport the different types of passengers one after the other with the same vehicle. However, this could be changed by shifting the opening and closing hours of one of the destinations. For some of the destinations this is not possible, e.g. for a school. But even when the destination itself would allow shifting the opening hours, e.g. a day centre, this is often undesirable, because a change in the opening hours would have great impact on the people using this facility and working there.

All these constraints make the planning of special transport services a complex problem, in which more constraints lead to a more inefficient solution. One can imagine that if all users can express all their wishes, making a good planning becomes really difficult. Therefore, it is essential to find the right balance between user satisfaction and reducing costs by optimizing the planning.

When a special needs school starts at 9 a.m. and a child is scheduled to arrive at the school two minutes before, chances are that the child will often be late. Traffic, weather conditions, roadwork and vehicle breakdowns are factors that can influence the travel time between two locations. If this travel time is seen as a constant value, the outcome of the planning in practice will be different than the planning on paper. In order for this child to arrive on time we need to take the uncertainties of the travel times into account. We can do this in two ways, by time-dependent travel times or stochastic travel times. Uncertainties like rush hour can be modelled with time-dependent travel times, which vary for each time of the day. Using stochastic travel times means the travel times have a certain mean and variance. In this variance several uncertain factors can be included, making the arrival time at a location vary each time. Of course a combination of both, where the stochastic travel times are also time-dependent, can be useful.

Another uncertainty lies in the arriving of the requests. Part of the requests will be known before the schedule is made; these are the regular requests that can be the same every day or every week, for example passengers going to a day centre or special needs school. We call these requests the static requests. We also have dynamic requests, which occur during the day and have to be added to the static schedule. Because these requests arrive during the day, the exact details of these requests are not known while making the static schedule. The problem of this is that it could occur that a dynamic request cannot be integrated in the current schedule anymore. In this case it would be necessary to add a new vehicle to the schedule; otherwise the dynamic request has to be rejected. It is obvious that neither of these options are desirable. If some information about the dynamic requests would be known beforehand, this information could be used to schedule the static requests in such a way that there is room left for the dynamic requests. This information could be for example that most dynamic requests occur at a certain time of day or that dynamic requests often concern the transport from or to a specific location or area.

In this report we research the stochastic optimization of the planning of special transport services. In the next section we first give an outline of the special transport service situation in the Netherlands. Section 3 gives an overview of the literature on the dial-a-ride problem, the mathematical designation of the problem. Based on the research already done, the research question of this report is formulated in Section 4. The deterministic model for this problem is formulated in Section 5. An Integer Linear Program (ILP) is given with a subset of all possible constraints that could be used to solve a small instance of the problem. To solve larger instances a planning heuristic is used, of which a short description is given.

In Section 6 normally distributed stochastic travel times are implemented in the model. An approximation is used to derive constraints that make sure the time window and maximum ride time constraints are satisfied with a specified minimum reliability. In Section 7 the static and dynamic requests are combined in one planning. First a simplified model is introduced to leave space for the dynamic requests in the static schedule. In this simplified version there are only two locations and there are no wheelchairs or other special constraints. For this version the number of rejected dynamic requests can be calculated analytically. For the complete model an algorithm is used to combine both types of requests. To determine whether it is best to plan the dynamic requests in the same vehicles as the static ones or to reserve an empty vehicle for the dynamic requests, machine scheduling is used. For one dynamic job the probability that it can be scheduled is calculated analytically, both for one as well as for multiple machines. For more than one dynamic job this is calculated with simulation. Based on this probability it is then possible to determine the switching point when it is better to use a new vehicle.

The results of all models are presented in Section 8, followed by the conclusion of this research in Section 9 and the discussion in Section 10.

2 Special transport services in the Netherlands

There are several types of special transport services in the Netherlands, with various bodies which are responsible for the administration and practical execution. In Figure 1 one can see the distribution of turnover among the different types. WIA-transport is missing here, but has a smaller turnover than all other types of special transport. Since January 2015 the AWBZ is partly replaced by the Wlz, while other former AWBZ tasks have now become the responsibility of the local authorities within the framework of the Wmo. Next we describe all types of special transport services, given by MuConsult (2013) and CROW-KpVV (2014).

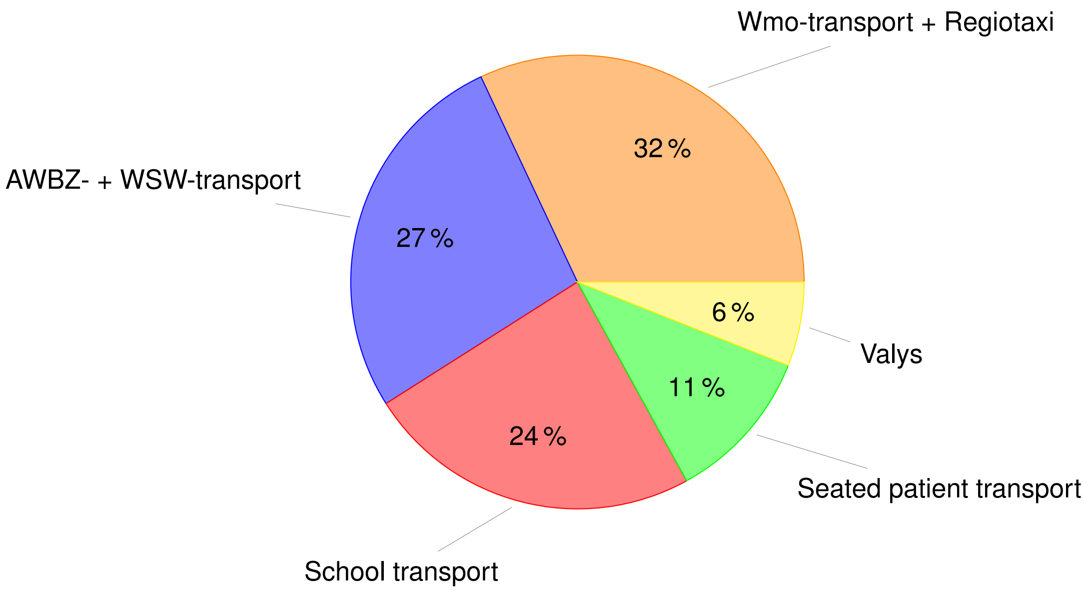

Figure 1: Distribution of turnover among different types of special transport services according to SEO-Economisch-Onderzoek (2011)

Wmo-transport

The aim of the Social Support Act (Wet maatschappelijke ondersteuning/Wmo) is to help people with an impairment participate in society and live as independently as possible. People who cannot travel independently anymore can use Wmo-transport for local and regional travels. Local authorities are responsible for the implementation of Wmo. In 2011 the number of people allowed to use Wmo-transport amounted to approximately 600,000, it is unknown how many of these people actually used this type of transport. Wmo-transport is demand-driven, which makes it difficult to predict and plan.

From January 2015 the extramural part of the AWBZ (General Exceptional Medical Expenses Act/ Algemene Wet Bijzondere Ziektekosten) has also been placed under the responsibility of the local authorities as part of the Wmo, which includes the transportation to day centres. This part of the Wmo is more predictable, although in practice a lot of changes occur at the last moment because the people who need to be transported are in general rather vulnerable.

Valys

Valys is transportation geared to social recreational long distance travels. The intentional goal is to transport people from their home to a public transport station. Only when persons are not able to use public transport, Valys is used as a door-to-door transportation. In practice however, almost all of the Valys travels (97%) are made from door-to-door. There are approximately 428,000 people allowed to use Valys transport, but it is unknown how many of these people actually use this type of transport. The Ministry of Health, Welfare and Sport (VWS) is responsible for this type of transport. Just like Wmo-transport, Valys is demand-driven, which makes it difficult to predict and plan.

Wlz-transport

The Long-Term Care Act (Wet langdurige zorg/Wlz) regulates intensive care for vulnerable elderly and disabled people. This law has replaced the AWBZ in January 2015. Usually people relying on the Wlz are living in an institution, but they can also live at home. Transportation to or from institutions is the responsibility of the institution itself.

School transport

When children are unable to go to a school in the vicinity of their home, because of a disability, or religious or ideological conviction, the local authorities are obliged to offer transport to a school that suits their personal situation. The implementation of school transport is the responsibility of the local authorities. There are 81,000 pupils using school transport. The times and locations of school transport are highly predictable, which makes it easy to plan. The pupils often need a high level of regularity, which means they need the same driver, bus and pickup time every day. Also the opening and closing times of the schools are strict, which makes this type of transport not very flexible.

WSW-transport

The Social Employment Law (Wet Sociale Werkvoorziening/WSW) helps people with disabilities to get work, for example in a sheltered workshop (85%). In 2011 approximately 102,000 people were working via WSW, of whom only a small part (approximately 20,000 people) were also using WSW-transport. The local authorities are responsible for WSW-transport. It is a highly predictable type of transport and therefore easy to plan.

WIA-transport

The Work and Income according to Labour Capacity Act (Wet werk en inkomen naar arbeidsvermogen/WIA) is meant for people with an occupational disability. WIA-transport regulates the transport of these people to work or education. There are approximately 60,000 users of WIA-transport. Just like WSW-transport, WIA-transport is predictable and easy to plan. The Employee Insurance Agency (Uitvoeringsinstituut Werknemersverzekeringen/UWV) is responsible for the implementation of WSW-transport.

Seated patient transport

Persons travelling for the purpose of a medical treatment can use seated patient transport. The health insurances are responsible for the implementation of this type of transport. In 2009 the number of users was estimated at 115,000. A small part of seated patient transport is periodically and therefore easy to plan, but the biggest part is less predictable.

Regiotaxi

The regiotaxi is a taxi service that can be used by everyone, including people without a referral for special transport. However, people with a referral travel with a discount. Usually regiotaxi is used in regions where public transport is minimal, so in this case it is used as a replacement of public transport services.

Like said before, most types of special transport services are carried out separately. This is mainly caused by the fact that they fall under the responsibility of different bodies and are implemented by different service providers. Since January 2015 AWBZ-transport has been decentralized to the Wmo, which makes it the responsibility of the local authorities. This, together with financial budget cuts, make local authorities more and more aware of the necessity to develop a more coherent system for special transport services. One way to achieve this is by establishing a central control centre, which is already used in different ways in Scandinavia, Zealand and Rotterdam, see CROW-KpVV (2013).

Two other trends seen lately is that more and more people are encouraged and trained to use public transport and that drivers of special transport vehicles are more often volunteers. But even though more people are going to use public transport and costs are reduced by the use of volunteers, a large, vulnerable group of people will remain dependent on special transport services, so a flexible and efficient planning remains really important.

3 Literature review

In a mathematical context the planning problem of special transport services is called the ”dial-a-ride problem”. The dial-a-ride problem is a generalization of the pickup and delivery problem with time windows, in which people are transported instead of goods.

The overview below first describes the research done on the deterministic version of the problem, followed by a paragraph about the papers treating the stochastic dial-a-ride problem.

3.1 Deterministic dial-a-ride problem

The dial-a-ride problem was first introduced by Wilson et al. (1971). Since then a broad diversity of papers was written about this problem. Most papers deal with the static case of the problem, where all requests are known beforehand. Interest for the dynamic version of the problem has grown significantly in the last decade. We give an overview of a selection of all papers written about both the static and the dynamic version of the problem.

An often used heuristic for the static dial-a-ride problem is a tabu search. This algorithm searches for the best solution in the neighbourhood at each iteration. To avoid cycling, solutions can be declared ’tabu’ for a number of iterations. In Cordeau and Laporte (2003a) a tabu search heuristic is used with the neighbourhood defined as all solutions that can be obtained by removing a request from one route and transferring it to another route. A combination of a mixed-integer program and a tabu search heuristic is used by Melachrinoudis et al. (2007). Other examples of algorithms used for the static problem are an insertion heuristic (Jaw et al. (1986), Diana and Dessouky (2004), Luo and Schonfeld (2007)), a sacrificing algorithm (Healy and Moll (1995)), a grouping genetic algorithm (Rekiek et al. (2006)) and an adaptive large neighbourhood search (Pisinger and Ropke (2007)). Although most of these papers consider time windows and maximum riding time constraints, the authors often neglect other realistic constraints. Most papers use homogeneous vehicles and do not consider a difference between customers with or without wheelchair. One of the papers that assumes heterogeneous vehicles and more realistic constraints is Xiang et al. (2006). In addition, they also fix the maximum working hours and break times for the drivers. Similar realistic constraints are formulated in Parragh et al. (2012). In this paper a column generation algorithm is combined with variable neighbourhood search to solve the static heterogeneous dial-a-ride problem with driver-related constraints. An overview of additional research on the static dial-a-ride problem is given by Berbeglia et al. (2007).

Although most of the papers on the dial-a-ride problem consider the static version, there is also a considerable amount of research carried out for the dynamic counterpart, especially in the last decade. Also for this problem the tabu search heuristic is often used. Because this heuristic is highly efficient, but can also have a high running time, Attanasio et al. (2004) use parallel computing to speed up the algorithm. For the dynamic version it is also possible to use an insertion heuristic, which is done by Madsen et al. (1995) and by Coslovich et al. (2006). Another approach was chosen by Teodorovic and Radivojevic (2000), who used fuzzy logic to formulate and solve the problem. An overview of additional research on the dynamic version of the problem is given by Berbeglia et al. (2010) and Pillac et al. (2013).

Because a large number of requests are known in advance in the dynamic dial-a-ride problem as well, solution methods for the static version are also used for the dynamic version. Therefore there are several papers covering both versions. Horn (2002) describes a software system to bridge the gap between static and dynamic approaches. Also in Cordeau and Laporte (2007) several models for the static as well as the dynamic dial-a-ride problem are presented. An overview of both the static and dynamic version is given by Cordeau and Laporte (2003b).

3.2 Stochastic dial-a-ride problem

All these papers mentioned above solve the deterministic problem. In this report we are looking at the stochastic case. The first important way stochastics can be included in the dial-a-ride problem is by assuming time-dependent or stochastic travel times. In this way uncertainties like traffic or weather conditions can be modelled. Another stochastic factor is the dynamic requests that have to be taken into account while making the static schedule. The following paragraphs describe the research done in the area of the stochastic dial-a-ride problem.

A lot of research was done on time-dependent travel times in the vehicle routing problem. In Ichoua et al. (2003) a parallel tabu search is used once more to solve the vehicle dispatching problem, which differs from our problem mainly because the goods transported here all have to be dropped off at the same location. The same holds for Taniguchi and Shimamoto (2004). Haghani and Jung (2005) use a genetic algorithm to solve the problem with a continuous travel time function. But because this article is about the transportation of goods, instead of people, several constraints are absent. In Kok et al. (2010) the vehicle routing problem with time windows is extended with time-dependent travel times as well as driving hours regulations.

In the case of stochastic travel times the travel times follow a certain distribution with known mean and variance. In this way unexpected changes due to for example weather conditions or accidents can be modelled. This is done by Li et al. (2010). In this paper the stochastic constraints resulting from the stochastic travel times are transformed into deterministic constraints by using an approximation function of the normal distribution. After this, a tabu search-based heuristic is used to solve the problem.

Of course, these two could also be combined in a stochastic dial-a-ride or vehicle routing problem with time-dependent travel times. In this case the mean and variance of the travel times are time-dependent. This is done by Fu (2002), one of the first papers on the stochastic dial-a-ride problem. In this paper the travel time of a vehicle from one stop to another is defined by a stochastic process. A parallel insertion algorithm is extended to make a schedule that minimizes the total cost including client inconvenience and makes sure all requests are being served on time with a pre-specified minimum probability. This paper treats the static version of the problem. A paper using time-dependent stochastic travel times on the dynamic side is Schilde et al. (2014). The solution method in this paper first forms an initial solution, which can be improved using one of four metaheuristic methods.

Xiang et al. (2008) also use stochastic and time-dependent travel times. Additionally their model is able to cope with unexpected changes, such as vehicle breakdowns and cancelations of requests. This is yet another example of the introduction of stochastics in the dial-a-ride problem.

A last way stochastics can play a role in the dial-a-ride problem is through the dynamic requests. Research in this area is scarce. Schilde et al. (2011) address a similar problem by using the knowledge that each person also requires a return transport that can be predicted with historical data. A stochastic variable neighbourhood search and a multiple scenario approach are used to test whether a stochastic method performs better than a deterministic one. In the stochastic variable neighbourhood search the data of the return transports is only used to check whether the new solution performs better than the previous one. In the multiple plan approach a local search is performed with sampled requests based on the available data.

Based on the research done in this area we formulate the main and sub-questions of this research in the next section.

4 Research question

Ideally a model of the dial-a-ride problem would take into account all stochastic uncertainties mentioned in the previous section. For practical reasons, however, a selection will be made in this research. As described in the previous section, quite some research has already been done on time-dependent travel times. Most of the papers about this subject divide the day in intervals and define different travel speeds in each interval. This increases the state space of the problem, but will not significantly increase the complexity of the problem. For this reason and because a lot of research has already been done on the subject, time-dependency is not part of this research.

On the other hand, stochastic travel times are expected to have more impact on the problem. If we use stochastic travel times it may become uncertain whether the transport will arrive within the right time window. However, we try to minimize the probability of arriving outside this time window. The question arises whether this will lead to more of the actual constraints satisfied.

As noted in the literature review some research has already been done on stochastic travel times in de dial-a-ride problem. In Fu (2002) stochastic travel time constraints are added and transformed into deterministic constraints by using an approximation of the normal distribution. However, the deterministic constraints derived in this article are not correct in all cases. In our research we prove when these constraints can be used and we add stochastic travel times in the maximum ride time constraint.

The dynamic requests are also of interest for this research. We want to take the dynamic requests into account while making the static schedule in such a way that the probability of having to use a new vehicle for a dynamic request is minimized. Of course, an essential requirement would be that the static schedule must be efficient and does not result in an undue increase of the vehicles used in comparison to the static model.

Little research has been done on taking into account the dynamic requests when making the static schedule. In Schilde et al. (2011) a similar thing is done, although knowledge about arrival of dynamic requests is only used to test a solution and not to improve it. In our research we also want to improve a solution using the knowledge about the arrival process of the dynamic requests.

All unexpected changes, such as vehicle breakdowns or cancelations of requests can be modelled in either the stochastic travel times or in the stochastic arrival process of the dynamic requests. This is why we do not model these changes explicitly.

Based on these uncertainties the following research question and sub questions were formulated:

How can we stochastically optimize the dial-a-ride problem?

Sub question 1:
- How can we model stochastic travel times?
- Does the addition of stochastic travel times lead to a more realistic schedule where more actual constraints are satisfied?

Sub question 2:
- Can we take the dynamic requests into account while making the static schedule?
- If so, does this addition lead to a more efficient schedule for the static as well as the dynamic requests?

Sub question 3:
- Can we combine stochastic travel times and taking into account the dynamic requests?

5 Deterministic model formulation

In this section we start with a formulation of the deterministic dial-a-ride problem. An integer linear program (ILP) is formulated to find an optimal solution of small instances of the problem. In this ILP some constraints are not included, like which people are allowed to ride in the same vehicle. Since in reality there are a lot of variables and constraints, we use an algorithm to find an approximation of the optimal solution. For this we use an insertion algorithm described in Span et al. (2013).

5.1 Problem formulation

In the dial-a-ride problem there are n requests that have to be served by a fleet of m heterogeneous vehicles given in the set $M = \{1, ..., m\}$. The stops are defined as $N = \{0, ..., 2n + 1\}$. Each request consists of a pickup location and an arrival location. The set of all pickup locations is $P = \{1, ..., n\}$ and the set of all arrival locations is $D = \{n + 1, ..., 2n\}$. The stops 0 and $2n + 1$ are the start and end location of the vehicles. For now we assume that all vehicles start and end at the same location.

So every request $r \in \{1, ..., n\}$ consists of a pickup location i and an arrival location $n+i$. Furthermore, every request is related to a service time at the arrival location to get in the vehicle, d_i, and a service time to get out of the vehicle at the destination, d_{n+i}. Also the number of passengers that need to be transported q_i^p and the number of wheelchairs that need to be transported q_i^w are defined for each location. For now we assume that a request can only concern either a walking passenger or a passenger in a non-folding wheelchair, while in reality also folding wheelchairs or rollators are used. All these variables are zero for stops 0 and $2n + 1$. For $i \in \{1, ..., n\}$ we have $q_i^p \geq 0$, $q_i^w \geq 0$, $q_{n+i}^p = -q_i^p$ and $q_{n+i}^w = -q_i^w$. Each request is also related to two time windows in which the request has to be served, one for the pickup and one for the arrival location. The time window of stop i is denoted by $[e_i, l_i]$. If a request has no desired time window for pickup or arrival, a wide time window is applied.

Each vehicle k has a capacity c_k^p, which indicates the number of persons without wheelchairs that can be transported by the vehicle. Other variables are the wheelchair capacity of a vehicle c_k^w and the maximum time a passenger is allowed to ride in the vehicle, which is denoted by R_{max}. This maximum ride time is the same for all requests.

For every pair of stops i and j the time to travel from i to j is defined as t_{ij} and the costs associated with this trip are c_{ij}. Within these costs several objectives can be included, such as the total travel time, the travel time the vehicle is not idle, the travel time costs and difference between the ride time of a passenger and the direct travel time. Also weighted combinations of these aspects could be used in the objective function. The notation c_{ij} is used for simplification.

5.2 ILP formulation

To formulate the ILP we define a few extra variables. We define B_i as the beginning of service at stop $i \in P \cup D$, Q_i^p as the number of persons and Q_i^w as the number of wheelchairs in the vehicle directly after leaving stop i. The decision variable x_{ij}^k indicates whether vehicle k drives from stop

i to stop j. The ILP can be formulated as follows:

$$\min \sum_{i,j \in N} \sum_{k \in M} c_{ij} x_{ij}^k \tag{5.1}$$

s.t.

$$\sum_{i \in N} \sum_{k \in M} x_{ij}^k = 1 \qquad \forall j \in P \cup D \tag{5.2}$$

$$\sum_{j \in N} \sum_{k \in M} x_{ij}^k = 1 \qquad \forall i \in P \cup D \tag{5.3}$$

$$\sum_{j \in N} x_{0j}^k = 1 \qquad \forall k \in M \tag{5.4}$$

$$\sum_{i \in N} x_{i(2n+1)}^k = 1 \qquad \forall k \in M \tag{5.5}$$

$$\sum_{j \in N} x_{mj}^k - \sum_{i \in N} x_{im}^k = 0 \qquad \forall m \in P \cup D \tag{5.6}$$

$$\sum_{j \in N} x_{ij}^k - \sum_{j \in N} x_{n+i,j}^k = 0 \qquad \forall i \in P, k \in M \tag{5.7}$$

$$B_{n+i} \geq B_i \qquad \forall i \in P \tag{5.8}$$

$$B_j \geq B_i + d_i + t_{ij} - M_{ij}(1 - \sum_k x_{ij}^k) \qquad \forall i, j \in N \tag{5.9}$$

$$Q_j^p \geq Q_i^p + q_j^p - V_{ij}(1 - \sum_k x_{ij}^k) \quad \forall i, j \in N, j \neq 2n+1 \tag{5.10}$$

$$Q_j^w \geq Q_i^w + q_j^w - W_{ij}(1 - \sum_k x_{ij}^k) \quad \forall i, j \in N, j \neq 2n+1 \tag{5.11}$$

$$\sum_{j \in N} x_{ij}^k Q_i^p \leq c_k^p \qquad \forall i \in N, k \in M \tag{5.12}$$

$$\sum_{j \in N} x_{ij}^k Q_i^w \leq c_k^w \qquad \forall i \in N, k \in M \tag{5.13}$$

$$e_i \leq B_i \leq l_i - d_i \qquad \forall i \in P \cup D \tag{5.14}$$

$$B_{n+i} \leq B_i + d_i + R_{max} \qquad \forall i \in P \tag{5.15}$$

$$x_{ij}^k \in \{0, 1\} \qquad \forall i, j \in N, k \in M$$

The objective function (5.1) minimizes the total costs of the trips. Constraints (5.2) and (5.3) make sure that every stop is visited exactly once and Constraints (5.4) and (5.5) make sure the vehicles start and end at the depot locations. Constraints (5.6) make sure each location is served by the same vehicle and Constraints (5.7) make sure each pickup and arrival pair is served by the same vehicle. Constraints (5.8) make sure each pickup location is visited before the corresponding arrival location. The starting times at each node are calculated by Constraints (5.9) and the number of persons and wheelchairs in the vehicle are calculated by (5.10) and (5.11) respectively. Constraints (5.12) and (5.13) make sure the number of persons and wheelchairs in the vehicles never exceed the capacity. Each stop also has a time window, which is satisfied because of Constraints (5.14). Note that the whole service needs to be performed within the specified time window, which is why we subtract the service time from the right hand side. Constraints (5.15) make sure the maximum ride time is not exceeded.

Constraints (5.12) and (5.13) are nonlinear. We make them linear by adding two new variables

Y_{ij}^k and Z_{ij}^k and the following constraints:

$$
\begin{aligned}
Y_{ij}^k &\leq n x_{ij}^k \quad && \forall i,j \in N, k \in M \\
Y_{ij}^k &\leq Q_i^p \quad && \forall i,j \in N, k \in M \\
Y_{ij}^k &\geq Q_i^p - n(1 - x_{ij}^k) \quad && \forall i,j \in N, k \in M \\
Y_{ij}^k &\geq 0 \quad && \forall i,j \in N, k \in M \\
Z_{ij}^k &\leq n x_{ij}^k \quad && \forall i,j \in N, k \in M \\
Z_{ij}^k &\leq Q_i^w \quad && \forall i,j \in N, k \in M \\
Z_{ij}^k &\geq Q_i^w - n(1 - x_{ij}^k) \quad && \forall i,j \in N, k \in M \\
Z_{ij}^k &\geq 0 \quad && \forall i,j \in N, k \in M
\end{aligned}
$$

The constraints replacing Constraints (5.12) and (5.13) become:

$$
\begin{aligned}
\sum_{j \in N} Y_{ij}^k &\leq c_k^p \quad && \forall i \in N, k \in M \\
\sum_{j \in N} Z_{ij}^k &\leq c_k^w \quad && \forall i \in N, k \in M
\end{aligned}
$$

5.3 Planning heuristic

The ILP formulation can be used to find the exact solution of the problem with up to approximately 20 requests and 2 vehicles. However, in reality the problem size is much larger. This is why we use a planning heuristic to approximate the optimal solution. We use an insertion algorithm described by Span et al. (2013). We shortly describe the idea of this heuristic.

An initialization step is first used to initialize the input and sort the requests according to their times and locations. In the next step the requests are added to the planning one by one by searching the optimal place of insertion in the planning. After each new request added to the planning, three optimization steps are performed to improve the current planning. The first step is a 'merge' algorithm, that tries to combine requests of two vehicles in a new vehicle. The second step is a 'transfer' algorithm, aimed at transferring the requests from one vehicle to another. The last optimization step is a 'shuffle' algorithm. In this step the requests for a specific vehicle are shuffled to find the optimal order of the requests.

6 Stochastic travel times

In this section we add stochastic travel times to the deterministic model of the previous section. This means that t_{ij} is a random variable. The travel time has a distribution with mean μ_{ij} and standard deviation σ_{ij}. We assume that waiting is only allowed after service at a stop. We make this assumption to be able to work with the arrival time A_i instead of the beginning of service B_i. We want to make a schedule that ensures both the time window constraint (5.14) as well as the maximum ride time constraint (5.15) are satisfied with pre-specified minimum reliabilities of α and β respectively. So we have to make sure that:

$$P(e_i \leq A_i \leq l_i - d_i) \geq \alpha, \tag{6.1}$$

$$P(A_{n+i} \leq A_i + d_i + R_{max}) \geq \beta. \tag{6.2}$$

We assume that the travel time is stochastic and the service time is deterministic. Because the travel time is always followed by a service time, the variation of the service time could be approximated by including it in the variation of the travel time.

We follow the approach from Fu (2002) to transform the stochastic constraints in deterministic ones again. However, the results derived by Fu (2002) are not valid in all cases. We prove in which cases these results are applicable and derive the results in other cases.

Because the travel time is stochastic, both the arrival time and departure time at each stop are random variables with a distribution depending on the distributions of the travel times. We assume that the travel times are normally distributed. We make the following assumption about the arrival and departure time at each stop:

Assumption 6.1 *The arrival time and departure time at each stop is normally distributed with their means and standard deviations depending only on the distribution of the departure time at its preceding stop and the travel time from the preceding stop.*

Based on Assumption 6.1 we can rewrite Equations (6.1) and (6.2) as:

$$\Phi\left(\frac{l_i - d_i - \mathbb{E}[A_i]}{\sqrt{Var(A_i)}}\right) - \Phi\left(\frac{e_i - \mathbb{E}[A_i]}{\sqrt{Var(A_i)}}\right) \geq \alpha, \tag{6.3}$$

$$\Phi\left(\frac{d_i + R_{max} - \mathbb{E}[A_{n+i}] + \mathbb{E}[A_i]}{\sqrt{Var(A_{n+i}) + Var(A_i) - 2Cov(A_{n+i}, A_i)}}\right) \geq \beta, \tag{6.4}$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. Note that since A_i and A_{n+i} are dependent of each other we get a covariance of the two variables in the second case.

We simplify Equations (6.3) and (6.4) even more using an approximation of the standard normal distribution. The approximation we use here is from Karian and Dudewicz (1991) and is the same as the one used in Fu (2002):

$$\Phi(x) = \begin{cases} \frac{x(4.4-x)}{10} + 0.5 & \text{if } 0 \leq x \leq 2.2 \\ 0.99 & \text{if } 2.2 < x < 2.6 \\ 1 & \text{if } x \geq 2.6. \end{cases}$$

To simplify it even more we do not distinguish between $2.2 < x < 2.6$ and $x \geq 2.6$, but we assume $\Phi(x) = 1$ in both cases. So instead we use:

$$\Phi(x) = \begin{cases} \frac{x(4.4-x)}{10} + 0.5 & \text{if } 0 \leq x \leq 2.2 \\ 1 & \text{if } x > 2.2. \end{cases} \tag{6.5}$$

We now rewrite Equations (6.3) and (6.4) using this approximation. To use this approximation we need to have $x \geq 0$. In order to make sure this is the case we make the following assumption:

Assumption 6.2 *We assume the schedule is always feasible in an expected sense, i.e.,* $e_i \leq \mathbb{E}[A_i] \leq l_i - d_i$ *and* $\mathbb{E}[A_{n+i}] - \mathbb{E}[A_i] \leq d_i + R_{max}\ \forall i$.

In the next subsection we rewrite the time window constraint and after that we rewrite the maximum ride time constraint.

6.1 Time window constraint

We want to rewrite Equation (6.3) using the approximation of the standard normal distribution given by Equation (6.5). With Assumption 6.2 we know that $l_i - d_i - \mathbb{E}[A_i] \geq 0$, which means we can use the approximation for the first Φ-function. Unfortunately the assumption makes $e_i - \mathbb{E}[A_i] \leq 0$, but we can easily solve this by rewriting the second Φ-function:

$$\Phi\left(\frac{e_i - \mathbb{E}[A_i]}{\sqrt{Var(A_i)}}\right) = 1 - \Phi\left(\frac{\mathbb{E}[A_i] - e_i}{\sqrt{Var(A_i)}}\right).$$

Because of the different values of $\Phi(x)$ for x smaller or greater than 2.2 for both Φ-functions, we distinguish four different cases in which we can rewrite Equation (6.3) differently. In Figure 2 these cases are presented graphically, where the dotted arrows mean there are multiple possibilities.

Case 1: $l_i - d_i - 2.2\sqrt{Var(A_i)} \leq \mathbb{E}[A_i] \leq e_i + 2.2\sqrt{Var(A_i)}$,

Case 2: $\mathbb{E}[A_i] \leq l_i - d_i - 2.2\sqrt{Var(A_i)}$ and $\mathbb{E}[A_i] \leq e_i + 2.2\sqrt{Var(A_i)}$,

Case 3: $\mathbb{E}[A_i] \geq l_i - d_i - 2.2\sqrt{Var(A_i)}$ and $\mathbb{E}[A_i] \geq e_i + 2.2\sqrt{Var(A_i)}$,

Case 4: $l_i - d_i - 2.2\sqrt{Var(A_i)} \geq \mathbb{E}[A_i] \geq e_i + 2.2\sqrt{Var(A_i)}$.

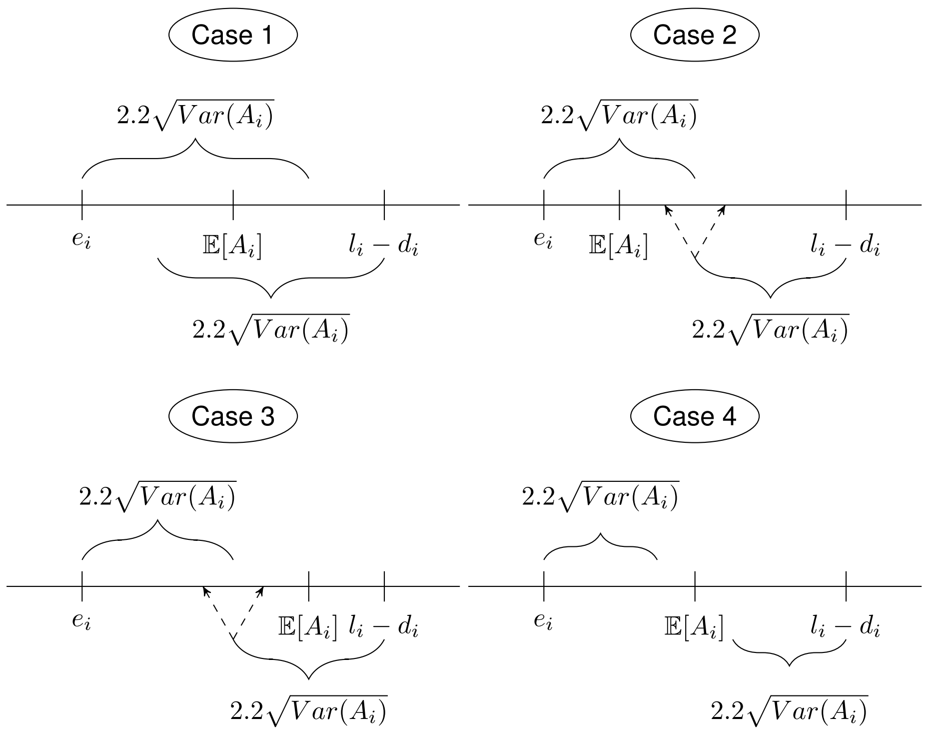

Figure 2: Different cases

In Case 1 the variance of A_i is large in relation to the length of the time window and in Case 4 it is small. Intuitively one would say that if the variance is small enough and the expectation of the arrival time is approximately in the middle of the time window, the probability is close to one that the vehicle will arrive within the time window. As we prove later on, this is indeed the case in Case 4. In Case 2 the expected arrival time is quite close to the left side of the time window, which means the probability of arriving too late is really small. In Case 3 it is the other way around. We can now rewrite Equation (6.3) in these four different cases, which we divide in four different lemmas.

Lemma 6.3 *Using the approximation of* (6.5) *we can rewrite Constraint* (6.3) *in Case 1 as:*

$$\frac{1}{2}(e_i + l_i - d_i) - \sqrt{2.2(l_i - d_i - e_i)\sqrt{Var(A_i)} - \frac{1}{4}(l_i - d_i - e_i)^2 - 5Var(A_i)\alpha} \leq \mathbb{E}[A_i]$$
$$\leq \frac{1}{2}(e_i + l_i - d_i) + \sqrt{2.2(l_i - d_i - e_i)\sqrt{Var(A_i)} - \frac{1}{4}(l_i - d_i - e_i)^2 - 5Var(A_i)\alpha}. \quad (6.6)$$

Proof In Case 1 we have:

$$l_i - d_i - 2.2\sqrt{Var(A_i)} \leq \mathbb{E}[A_i] \leq e_i + 2.2\sqrt{Var(A_i)},$$

which means that both $\frac{l_i - d_i - \mathbb{E}[A_i]}{\sqrt{Var(A_i)}}$ and $\frac{\mathbb{E}[A_i] - e_i}{\sqrt{Var(A_i)}}$ are in the first case of (6.5). Using the first case of this approximation function, Equation (6.3) can be written as:

$$\frac{1}{10}\frac{l_i - d_i - \mathbb{E}[A_i]}{\sqrt{Var(A_i)}}\left(4.4 - \frac{l_i - d_i - \mathbb{E}[A_i]}{\sqrt{Var(A_i)}}\right) + \frac{1}{2} + \frac{1}{10}\frac{\mathbb{E}[A_i] - e_i}{\sqrt{Var(A_i)}}\left(4.4 - \frac{\mathbb{E}[A_i] - e_i}{\sqrt{Var(A_i)}}\right) + \frac{1}{2} - 1 \geq \alpha$$
$$\Rightarrow -\frac{\mathbb{E}[A_i]^2}{5Var(A_i)} + \frac{(e_i + l_i - d_i)\mathbb{E}[A_i]}{5Var(A_i)} + \frac{4.4(l_i - d_i - e_i)\sqrt{Var(A_i)} - l_i^2 - e_i^2 - d_i^2 + 2l_i d_i - 10Var(A_i)\alpha}{10Var(A_i)} \geq 0. \quad (6.7)$$

Setting this equal to zero and using the abc-formula gives two zeros:

$$\mathbb{E}[A_i] = \frac{1}{2}(e_i + l_i - d_i) \pm \sqrt{2.2(l_i - d_i - e_i)\sqrt{Var(A_i)} - \frac{1}{4}(l_i - d_i - e_i)^2 - 5Var(A_i)\alpha}.$$

Since the derivative of (6.7) at the left zero is positive and the derivative at the right zero is negative, Equation (6.7) becomes:

$$\frac{1}{2}(e_i + l_i - d_i) - \sqrt{2.2(l_i - d_i - e_i)\sqrt{Var(A_i)} - \frac{1}{4}(l_i - d_i - e_i)^2 - 5Var(A_i)\alpha} \leq \mathbb{E}[A_i]$$
$$\leq \frac{1}{2}(e_i + l_i - d_i) + \sqrt{2.2(l_i - d_i - e_i)\sqrt{Var(A_i)} - \frac{1}{4}(l_i - d_i - e_i)^2 - 5Var(A_i)\alpha}.$$

■

Lemma 6.4 *Using the approximation of* (6.5) *we can rewrite Constraint* (6.3) *in Case 2 as:*

$$e_i + (2.2 - \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)} \leq \mathbb{E}[A_i]. \quad (6.8)$$

Proof In Case 2 we have:

$$\mathbb{E}[A_i] \leq l_i - d_i - 2.2\sqrt{Var(A_i)} \text{ and } \mathbb{E}[A_i] \leq e_i + 2.2\sqrt{Var(A_i)},$$

which means that $\frac{l_i - d_i - \mathbb{E}[A_i]}{\sqrt{Var(A_i)}}$ is in the second case of (6.5) and $\frac{\mathbb{E}[A_i] - e_i}{\sqrt{Var(A_i)}}$ is in the first case. This means the first Φ-function is approximately equal to one, so Equation (6.3) can be written as:

$$\frac{1}{10}\frac{\mathbb{E}[A_i] - e_i}{\sqrt{Var(A_i)}}\left(4.4 - \frac{\mathbb{E}[A_i] - e_i}{\sqrt{Var(A_i)}}\right) + \frac{1}{2} \geq \alpha$$
$$\Rightarrow \frac{-\mathbb{E}[A_i]^2}{10Var(A_i)} + \frac{2.2\sqrt{Var(A_i)}\mathbb{E}[A_i] + e_i\mathbb{E}[A_i]}{5Var(A_i)} + \frac{-4.4\sqrt{Var(A_i)}e_i - e_i^2 + (5 - 10\alpha)Var(A_i)}{10Var(A_i)} \geq 0. \quad (6.9)$$

Setting this equal to zero and using the abc-formula gives two zeros:

$$\mathbb{E}[A_i] \quad = \quad 2.2\sqrt{Var(A_i)} + e_i \pm \sqrt{9.84 - 10\alpha}\sqrt{Var(A_i)}.$$

Since the derivative of (6.9) at the left zero is positive and the derivative at the right zero is negative, Equation (6.9) becomes:

$$e_i + (2.2 - \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)} \leq \mathbb{E}[A_i]$$
$$\leq e_i + (2.2 + \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)}.$$

Since in this case we already have

$$\mathbb{E}[A_i] \leq e_i + 2.2\sqrt{Var(A_i)} \leq e_i + (2.2 + \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)},$$

we only need the left boundary for $\mathbb{E}[A_i]$:

$$e_i + (2.2 - \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)} \leq \mathbb{E}[A_i].$$

■

Lemma 6.5 *Using the approximation of* (6.5) *we can rewrite Constraint* (6.3) *in Case 3 as:*

$$\mathbb{E}[A_i] \leq l_i - d_i - (2.2 - \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)}. \tag{6.10}$$

Proof In Case 3 we have:

$$\mathbb{E}[A_i] \geq l_i - d_i - 2.2\sqrt{Var(A_i)} \text{ and } \mathbb{E}[A_i] \geq e_i + 2.2\sqrt{Var(A_i)},$$

which means that $\frac{l_i - d_i - \mathbb{E}[A_i]}{\sqrt{Var(A_i)}}$ is in the first case of (6.5) and $\frac{\mathbb{E}[A_i] - e_i}{\sqrt{Var(A_i)}}$ is in the second case. This means the second Φ-function is approximately equal to one, so Equation (6.3) can be written as:

$$\frac{1}{10}\frac{l_i - d_i - \mathbb{E}[A_i]}{\sqrt{Var(A_i)}}\left(4.4 - \frac{l_i - d_i - \mathbb{E}[A_i]}{\sqrt{Var(A_i)}}\right) + \frac{1}{2} \geq \alpha$$
$$\Rightarrow \frac{-\mathbb{E}[A_i]^2}{10Var(A_i)} + \frac{(l_i - d_i - 2.2\sqrt{Var(A_i)})\mathbb{E}[A_i]}{5Var(A_i)} + \frac{4.4(l_i - d_i)\sqrt{Var(A_i)} - (l_i - d_i)^2 + (5 - 10\alpha)Var(A_i)}{10Var(A_i)} \geq 0 \tag{6.11}$$

Setting this equal to zero and using the abc-formula gives two zeros:

$$\mathbb{E}[A_i] \quad = \quad -2.2\sqrt{Var(A_i)} + l_i - d_i \pm \sqrt{9.84 - 10\alpha}\sqrt{Var(A_i)}.$$

Since the derivative of (6.11) at the left zero is positive and the derivative at the right zero is negative, Equation (6.11) becomes:

$$l_i - d_i - (2.2 + \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)} \leq \mathbb{E}[A_i]$$
$$\leq l_i - d_i - (2.2 - \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)}.$$

Since in this case we already have

$$\mathbb{E}[A_i] \geq l_i - d_i - 2.2\sqrt{Var(A_i)} \geq l_i - d_i - (2.2 + \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)},$$

we only need the right boundary for $\mathbb{E}[A_i]$:

$$\mathbb{E}[A_i] \leq l_i - d_i - (2.2 - \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)}.$$

■

Lemma 6.6 *Using the approximation of* (6.5) *we have in Case 4:*

$$P(e_i \leq A_i \leq l_i) \approx 1 \geq \alpha. \tag{6.12}$$

Proof In Case 4 we have:

$$l_i - d_i - 2.2\sqrt{Var(A_i)} \geq \mathbb{E}[A_i] \geq e_i + 2.2\sqrt{Var(A_i)},$$

which means that $\frac{l_i - d_i - \mathbb{E}[A_i]}{\sqrt{Var(A_i)}}$ and $\frac{\mathbb{E}[A_i] - e_i}{\sqrt{Var(A_i)}}$ are both in the second case of (6.5). This means both Φ-functions are approximately equal to one, so Equation (6.3) becomes:

$$1 \geq \alpha.$$

Since α is a value between zero and one, this is always true. This means that we have:

$$P(e_i \leq A_i \leq l_i) \approx 1 \geq \alpha.$$

■

We now formulate the following three theorems:

Theorem 6.7

$$\textit{If} \quad e_i + 2.2\sqrt{Var(A_i)} \leq l_i - d_i - 2.2\sqrt{Var(A_i)}, \tag{6.13}$$

then we can rewrite Constraint (6.3) *using the approximation of* (6.5) *as:*

$$e_i + (2.2 - \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)} \leq \mathbb{E}[A_i] \leq l_i - d_i - (2.2 - \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)}. \tag{6.14}$$

Proof Since we have (6.13), we are either in Case 2, Case 3 or Case 4. If we are in Case 4, according to Lemma 6.6 Condition (6.3) is met for all values of $\mathbb{E}[A_i]$. If we are in Case 2, we have a lower bound for $\mathbb{E}[A_i]$ given by Lemma 6.4 and if we are in Case 3, we have an upper bound given by Lemma 6.5. Together, this results in the boundaries of (6.14). ■

Theorem 6.8

$$\textit{If} \qquad l_i - d_i - 2.2\sqrt{Var(A_i)} \leq e_i + 2.2\sqrt{Var(A_i)} \tag{6.15}$$

and

$$\mathbb{E}[A_i] \leq l_i - d_i - 2.2\sqrt{Var(A_i)} \textit{ or } \mathbb{E}[A_i] \geq e_i + 2.2\sqrt{Var(A_i)}, \tag{6.16}$$

then we can rewrite Constraint (6.3) *using the approximation of* (6.5) *as:*

$$e_i + (2.2 - \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)} \leq \mathbb{E}[A_i] \leq l_i - d_i - (2.2 - \sqrt{9.84 - 10\alpha})\sqrt{Var(A_i)}. \tag{6.17}$$

Proof Since we have (6.15) and (6.16), we are either in Case 2 or Case 3. If we are in Case 2, we have a lower bound for $\mathbb{E}[A_i]$ given by Lemma 6.4 and if we are in Case 3, we have an upper bound given by Lemma 6.5. Together, this results in the boundaries of (6.17). ■

Theorem 6.9

$$\textit{If} \qquad l_i - d_i - 2.2\sqrt{Var(A_i)} \leq e_i + 2.2\sqrt{Var(A_i)} \tag{6.18}$$

and

$$l_i - d_i - 2.2\sqrt{Var(A_i)} \leq \mathbb{E}[A_i] \leq e_i + 2.2\sqrt{Var(A_i)}, \tag{6.19}$$

then we can rewrite Constraint (6.3) *using the approximation of* (6.5) *as:*

$$\frac{1}{2}(e_i + l_i - d_i) - \sqrt{2.2(l_i - d_i - e_i)\sqrt{Var(A_i)} - \frac{1}{4}(l_i - d_i - e_i)^2 - 5Var(A_i)\alpha} \leq \mathbb{E}[A_i]$$
$$\leq \frac{1}{2}(e_i + l_i - d_i) + \sqrt{2.2(l_i - d_i - e_i)\sqrt{Var(A_i)} - \frac{1}{4}(l_i - d_i - e_i)^2 - 5Var(A_i)\alpha}. \quad (6.20)$$

Proof Since we have (6.18) and (6.19) we are in Case 1. This means we can use Lemma 6.3 which results in (6.20). ■

If the variance is small in relation to the length of the time window we could schedule the arrival in a part of the time window such that the probability of arriving outside the time window is almost zero. If we schedule the arrival earlier than this middle part of the time window, the probability of arriving too late is almost zero, which means we only have to take the left boundary into account. If we schedule the arrival later we only need the right boundary. This results in the boundaries for $\mathbb{E}[A_i]$ of Theorem 6.7. These are the same boundaries as the ones derived in Fu (2002).

If the variance is larger in relation to the length of the time window, we can only use the same boundaries if either the probability of arriving too early or the probability of arriving too late is almost zero. This is the case in Theorem 6.8, where we are either at the left side or the right side of the time window.

If we have to take into account both the probability of arriving too early and the probability of arriving too late, the boundaries for $\mathbb{E}[A_i]$ become less attractive. We see this in Theorem 6.9.

We can use Theorem 6.7 if we have $\sqrt{Var(A_i)} \leq \frac{1}{4.4}(l_i - d_i - e_i)$ and we can use Theorem 6.8 or Theorem 6.9 if $\sqrt{Var(A_i)} > \frac{1}{4.4}(l_i - d_i - e_i)$. In this second case the boundary for $\mathbb{E}[A_i]$ is more difficult since the boundary depends on the value of $\mathbb{E}[A_i]$ itself. This is why we would like to only use the first case, where the variance is relatively small compared to het length of the time window.

Since the standard deviation of $\mathbb{E}[A_i]$ is equal to $\sqrt{Var(A_i)}$, the maximum value of the standard deviation in this case is $\sigma = \frac{1}{4.4}(l_i - d_i - e_i)$. Since we made the assumption that A_i is normally distributed, we know that 95% of the values of A_i lie between $\mathbb{E}[A_i] - 2\sigma \geq \mathbb{E}[A_i] - \frac{1}{2.2}(l_i - d_i - e_i)$ and $\mathbb{E}[A_i] + 2\sigma \leq \mathbb{E}[A_i] + \frac{1}{2.2}(l_i - d_i - e_i)$. In reality the time window length is often 15 minutes and the service time is often 1 minute, which gives $\frac{1}{2.2}(l_i - d_i - e_i) = \frac{14}{2.2} \approx 6.36$. This means that in this case 95% arrives within 6.36 minutes from the planned arrival time. Note that this is the maximum deviation, so it can also be smaller. This seems a reasonable deviation in practice. This is why we choose our variance such that we can use Theorem 6.7.

In the ILP of Section 5.2 we replace Constraint (5.14) by Constraint (6.14) to achieve a reliability of α when we have a variance satisfying $\sqrt{Var(A_i)} \leq \frac{1}{4.4}(l_i - d_i - e_i)$.

6.2 Maximum ride time constraint

We continue rewriting Equation (6.4). Using Assumption 6.2 we know that $d_i + R_{max} - \mathbb{E}[A_{n+i}] + \mathbb{E}[A_i] \geq 0$, which means we can use the approximation of (6.5). We have two cases now:

Case 1: $\mathbb{E}[A_i] \leq \mathbb{E}[A_{n+i}] - d_i - R_{max} + 2.2\sqrt{q}$,
Case 2: $\mathbb{E}[A_i] \geq \mathbb{E}[A_{n+i}] - d_i - R_{max} + 2.2\sqrt{q}$,
where $q = Var(A_{n+i}) + Var(A_i) - 2Cov(A_{n+i}, A_i)$.

We now formulate the following two lemmas:

Lemma 6.10 *Using the approximation of* (6.5) *we can rewrite Constraint* (6.4) *in Case 1 as:*

$$\mathbb{E}[A_{n+i}] - \mathbb{E}[A_i] \leq d_i + R_{max} + \left(\sqrt{9.84 - 10\beta} - 2.2\right)\sqrt{q}. \tag{6.21}$$

Proof In Case 1 we have:

$$\mathbb{E}[A_i] \leq \mathbb{E}[A_{n+i}] - d_i - R_{max} + 2.2\sqrt{q},$$

which means that we are in the first case of (6.5). Using the first case of this approximation function, Equation (6.4) can be written as:

$$\begin{aligned}
& \frac{1}{10}\frac{d_i + R_{max} - \mathbb{E}[A_{n+i}] + \mathbb{E}[A_i]}{\sqrt{q}}\left(4.4 - \frac{d_i + R_{max} - \mathbb{E}[A_{n+i}] + \mathbb{E}[A_i]}{\sqrt{q}}\right) + \frac{1}{2} \geq \beta \\
\Rightarrow \quad & \frac{-\mathbb{E}[A_i]^2}{10q} + \frac{(2.2\sqrt{q} + \mathbb{E}[A_{n+i}] - d_i - R_{max})\mathbb{E}[A_i]}{5q} \\
+ \quad & \frac{4.4(d_i + R_{max} - \mathbb{E}[A_{n+i}])\sqrt{q} - (\mathbb{E}[A_{n+i}] - d_i - R_{max})^2 + (5 - 10\beta)q}{10q} \geq 0. \qquad (6.22)
\end{aligned}$$

Setting this equal to zero and using the abc-formula gives two zeros:

$$\mathbb{E}[A_i] \quad = \quad 2.2\sqrt{q} + \mathbb{E}[A_{n+i}] - d_i - R_{max} \pm \sqrt{9.84 - 10\beta}\sqrt{q}.$$

Since the derivative of (6.22) at the left zero is positive and the derivative at the right zero is negative, Equation (6.22) becomes:

$$\begin{aligned}
2.2\sqrt{q} + \mathbb{E}[A_{n+i}] - d_i - R_{max} - \sqrt{9.84 - 10\beta}\sqrt{q} \leq \mathbb{E}[A_i] \\
\leq 2.2\sqrt{q} + \mathbb{E}[A_{n+i}] - d_i - R_{max} + \sqrt{9.84 - 10\beta}\sqrt{q}.
\end{aligned}$$

Since in this case we have

$$\mathbb{E}[A_i] \leq \mathbb{E}[A_{n+i}] - d_i - R_{max} + 2.2\sqrt{q} \leq \mathbb{E}[A_{n+i}] - d_i - R_{max} + 2.2\sqrt{q} + \sqrt{9.84 - 10\beta}\sqrt{q},$$

we only need the left boundary for $\mathbb{E}[A_i]$:

$$2.2\sqrt{q} + \mathbb{E}[A_{n+i}] - d_i - R_{max} - \sqrt{9.84 - 10\beta}\sqrt{q} \leq \mathbb{E}[A_i].$$

■

Lemma 6.11 *Using the approximation of* (6.5) *we have in Case 2:*

$$P(A_{n+i} \leq A_i + d_i + R_{max}) \approx 1 \geq \beta. \tag{6.23}$$

Proof In Case 2 we have:

$$\mathbb{E}[A_i] \geq \mathbb{E}[A_{n+i}] - d_i - R_{max} + 2.2\sqrt{q},$$

which means that we are in the second case of (6.5). This means the Φ-function is approximately equal to one, so Equation (6.4) becomes:

$$1 \geq \beta.$$

Since β is a value between zero and one this is always true. This means that we have:

$$P(A_{n+i} \leq A_i + d_i + R_{max}) \approx 1 \geq \beta.$$

■

From these lemmas we formulate the following theorem:

Theorem 6.12 *Using the approximation of* (6.5) *we can rewrite Constraint* (6.4) *as:*

$$\mathbb{E}[A_{n+i}] - \mathbb{E}[A_i] \leq d_i + R_{max} + \left(\sqrt{9.84 - 10\beta} - 2.2\right)\sqrt{q}. \tag{6.24}$$

Proof From Lemma 6.10 we know that the maximum ride time constraint is satisfied if $d_i + R_{max} - 2.2\sqrt{q} \leq \mathbb{E}[A_{n+i}] - \mathbb{E}[A_i] \leq d_i + R_{max} + \left(\sqrt{9.84 - 10\beta} - 2.2\right)\sqrt{q}$ and from Lemma 6.11 we know that it is also satisfied if $\mathbb{E}[A_{n+i}] - \mathbb{E}[A_i] < d_i + R_{max} - 2.2\sqrt{q}$. Together, this results in $\mathbb{E}[A_{n+i}] - \mathbb{E}[A_i] \leq d_i + R_{max} + \left(\sqrt{9.84 - 10\beta} - 2.2\right)\sqrt{q}$. ■

Concluding, in the ILP of Section 5.2 we replace Constraint (5.15) by Constraint (6.24) to achieve a reliability of β.

7 Model for combination of static and dynamic requests

In this section we want to take the dynamic requests into account while making the static schedule. In order to do so we assume a certain arrival process of the dynamic requests. Using this information we try to create space such that the dynamic requests can be scheduled within the static schedule. In this section we assume that the number of vehicles needed to schedule both the static and dynamic requests at once (so when also the dynamic requests are known beforehand) is the number of vehicles available. If a dynamic request cannot be scheduled in one of these vehicles anymore, it is rejected. In reality another option is to use a new vehicle for a dynamic request, but since this is also not desirable we do not use this option. So our objective is to minimize the number of rejected requests.

We start with a simplified model where we have only two locations and each request needs to go from a pickup location P to a destination location D. For this model we define a Markov reward process for the number of static and dynamic requests present in a certain time slot. From this the expected number of rejected requests can be calculated analytically and used to determine in which time slot extra space needs to be reserved for the dynamic requests.

If we have more than two locations we cannot calculate the number of rejected requests analytically anymore. Another difficulty is that the probability that a dynamic request is from one specific location to another specific location at a specific time is really small. Taking into account all these small probabilities separately is not possible. However, when we cluster the requests whose locations are close and have approximately the same time, we can sum their probabilities. This is what we do in our complete model. Dummies are created from these clustered requests and these dummies are used to create space for the dynamic requests.

In some cases it might be better to reserve special vehicles for the dynamic requests and in some cases it might be better to schedule the dynamic requests in the same vehicles as the static requests. A similar problem occurs when emergency patients need to be scheduled in operating rooms. Here there is some switching point when it is better to reserve specific emergency operating rooms or schedule the emergency patients in the same operating rooms as the non-emergency patients. With a machine scheduling problem we try to find out if we have a similar switching point in this case.

7.1 Simplified model for the combination of static and dynamic requests

A simplified version of the model is introduced with only two locations. The model of Kortbeek et al. (2011) is followed to take the dynamic requests into account while making the static schedule.

There are two locations, each customer has to go from a pickup location P to a destination location D. A day consists of T time slots of length h. The time for a vehicle to drive from location P to location D and back is equal to p time slots. At the beginning of the day there are n customers that need to be scheduled at that day, these are the static requests. The number of static requests in each time slot is A_t^s. Each static request has a time window of x time slots after the time A_t^s in which they need to be picked up. During the day new customers arrive, these are the dynamic requests. These dynamic requests have to be served in g time slots and are served according to the FCFS discipline. If it is not possible to serve a dynamic customer within this time, the customer is rejected and not scheduled again. There are B vehicles, each vehicle b has capacity c_b. The dynamic customers arrive according to a Poisson process with arrival rate χ_t in each time slot $(t-1, t]$.

We assume we have a static schedule where all the n static customers are planned in the desired time window. We call this schedule $\Gamma = (k_1^b, ..., k_T^b)$, where k_t^b is the number of customers scheduled to leave at the beginning of time slot t in vehicle b. Since a vehicle takes p time slots to drive from A to B we require that if $k_t^b > 0$, then $k_{t+1}^b, ..., k_{t+p-1}^b$ are all equal to zero. A feasible schedule should also satisfy $k_t^b \leq c_b$. We define an extra variable I_t^b which is equal to one if it is possible for vehicle b according to the static schedule to leave location P at the beginning of time slot t and zero otherwise. So, if the bus is either busy driving at the beginning of time slot t or is scheduled to leave within the next $p-1$ time slots, the value of I_t^b is zero.

With a probability of q a static customer does not show, for example because of illness. If $I_t^b = 1$ and in the schedule Γ we have $k_t^b < c_b$, there is capacity left for the dynamic requests in time slot t. We call the number of anticipated time slots to be available for dynamic requests in the upcoming g time slots in the beginning of time slot t, e_t. This is without potential no-shows, since we assume they are not known beforehand. The value of e_t can be calculated as follows:

$$e_t = \sum_{b=1}^{B} \sum_{j=t}^{\min\{t+g-1,T\}} (c_b - k_j^b) I_j^b.$$

We now define a Markov reward process by defining the states of the system, the transition probabilities and the reward.

The state is defined by (t, s, d) which means that at the beginning of time slot t, s static and d dynamic requests are present. The number of static requests that are present are the number of static requests scheduled to leave at the beginning of time slot t minus the no-shows. The number of dynamic requests are the dynamic requests arrived before time slot t that were not rejected, but still need to be transported.

The number of static requests arriving at the beginning of time slot t is equal to the number of static requests scheduled minus the number of no-shows. Let $p_t^s(s)$ be the probability that s static requests arrive at the beginning of time slot t, which can be calculated as follows:

$$p_t^s(s) = \begin{cases} \binom{\sum_{b=1}^{B} k_t^b}{s} (1-q)^s q^{\sum_{b=1}^{B} k_t^b - s} & \text{if } 0 \leq s \leq \sum_{b=1}^{B} k_t^b, \\ 0 & \text{if } s > \sum_{b=1}^{B} k_t^b. \end{cases}$$

Let $p_t^d(d)$ be the probability that d dynamic requests arrive during the time slot t, which is Poisson distributed with time slot dependent parameter χ_t. Let $\mathbb{P}[(s,d)_{t+1}|(v,w)_t]$ denote the transition probability of going from state (t, v, w) to state $(t+1, s, d)$. To calculate these transition probabilities we need two sums, defined as follows:

$$C_t = \sum_{b=1}^{B} c_b \cdot \mathbb{1}_{(I_t^b = 1)},$$
$$K_t = \sum_{b=1}^{B} k_t^b.$$

The first sum, C_t, is the capacity at time t of all busses that are starting to drive at time t. The second sum, K_t, is the total number of scheduled static requests at time t. With these values we can derive the transition probabilities in five different cases:

Case 1 $v \leq K_t$ and $w \leq e_t$

$$\mathbb{P}[(s,d)_{t+1}|(v,w)_t] = p_{t+1}^s(s) p_{t+1}^d(d - w + \min\{C_t - v, w\}) \mathbb{1}_{(d \geq w - \min\{C_t - v, w\})},$$

Case 2 $v \leq K_t$ and $w > e_t$ and $C_t - v \leq e_t$

$$\mathbb{P}[(s,d)_{t+1}|(v,w)_t] = p^s_{t+1}(s)p^d_{t+1}(d - e_t + C_t - v)\mathbb{1}_{(d \geq e_t - C_t + v)},$$

Case 3 $v \leq K_t$ and $w > e_t$ and $C_t - v > e_t$

$$\mathbb{P}[(s,d)_{t+1}|(v,w)_t] = p^s_{t+1}(s)p^d_{t+1}(d)\mathbb{1}_{(d \geq 0)},$$

Case 4 $v > K_t$ and $w \leq e_t$

$$\mathbb{P}[(s,d)_{t+1}|(v,w)_t] = p^s_{t+1}(s)p^d_{t+1}(d - w + \min\{C_t - K_t, w\})\mathbb{1}_{(d \geq w - \min\{C_t - K_t, w\})},$$

Case 5 $v > K_t$ and $w > e_t$

$$\mathbb{P}[(s,d)_{t+1}|(v,w)_t] = p^s_{t+1}(s)p^d_{t+1}(d - e_t + C_t - K_t)\mathbb{1}_{(d \geq e_t - C_t + K_t)}.$$

In Case 1 the number of static customers that shows up at time t is less or equal to the number of scheduled static requests and the number of dynamic customers is less or equal to the number of anticipated available time slots. This means that all static and all dynamic customers are served, the static customers immediately and the dynamic customers within g time slots. The number of dynamic customers that is served immediately is equal to the minimum of $C_t - v$ and w.

In Case 2 and Case 3 some dynamic customers are rejected, since the number of dynamic customers is now higher than the number of anticipated available time slots. If there are so many no-shows among the static customers such that the available capacity for dynamic customers is higher than e_t we are in Case 3. In this case all e_t dynamic customers that are not rejected are served in time slot t. In Case 2 only $C_t - v$ dynamic customers can be served in time slot t.

If the number of static customers that shows up at time t is higher than the number of scheduled static requests we are in Case 4 or Case 5. This happens when part of the static requests that show up is scheduled in a later time slot. If the number of dynamic requests is less or equal than e_t we are in Case 4 in which K_t static customers are served and all w dynamic customers are served in the next g time slots. In Case 5 only e_t dynamic customers are not rejected and $C_t - K_t$ of them are served in time slot t.

The reward is the number of dynamic requests that are not rejected each time period. So we formulate the expected number of rejected requests. To do this, we define $Q_t(s,d)$ as the probability that at the start of time slot t there are s static and d dynamic requests present. This is calculated as follows:

$$\begin{aligned} Q_1(s,d) &= p^s_1(s) \cdot p^d_1(d), \\ Q_{t+1}(s,d) &= \sum_{v=0}^{c}\sum_{w=0}^{\infty} Q_t(v,w)\mathbb{P}[(s,d)_{t+1}|(v,w)_t] \text{ for } t = 1,...,T-1. \end{aligned}$$

We define R_t as the expected number of rejected jobs until the end of time slot t, which is:

$$\begin{aligned} R_1 &= \sum_{s=0}^{\infty}\sum_{d=e_1+1}^{\infty}(d - e_1)Q_1(s,d), \\ R_t &= R_{t-1} + \sum_{s=0}^{\infty}\sum_{d=e_t+1}^{\infty}(d - e_t)Q_t(s,d) \text{ for } t = 2,...,T. \end{aligned}$$

The last value, R_T, is het expected total number of rejected requests. The expected number of rejected requests in time slot t is equal to:

$$r_t = \sum_{s=0}^{\infty} \sum_{d=e_t+1}^{\infty} (d - e_t) Q_t(s, d).$$

We now want to reserve some space for the rejected requests to prevent them from being rejected again. The expected total number of rejected requests is R_T. This is the number of places we want to reserve during the whole day, rounded down. We want to reserve those places in the time slots where most requests are expected to be rejected. So we reserve places in the $\lfloor R_T \rfloor$ time slots with highest values of r_t. These 'reservations' are added to the arrival stream of static customers and are seen as extra static customers, which have to be served within g time slots. This means they might have a different time window length than the original static customers.

A new static schedule is created with the extra static requests implemented. Again the number of expected rejected requests are calculated.

Pseudocode 1 describes the construction of the best static schedule Γ^* with the dynamic requests taken into account. As input we have the set of static requests S that need to be scheduled and the arrival process in every time interval λ_t of the dynamic requests. First the set of static requests S is scheduled optimally which gives the static schedule Γ. If there are requests that cannot be planned they are stored in the vector $Unassigned$. Next the values of R_T and r_t are calculated for the constructed schedule Γ and the arrival process λ_t. In the main body of the algorithm a new schedule is constructed as long as all requests can still be scheduled ($Unassigned$ = $\emptyset$) and an improvement is made somewhere in the last three loops. To check this last condition f counts the number of loops no improvement was found. In every loop a vector r_{max} is filled with the $\lfloor R_T \rfloor$ time slots with the highest values of r_t. In these time slots dummy requests are created and stored in the set D. A new static schedule Γ is created with requests from both S and D. Next the requests from D are removed again from Γ and new values of R_T and r_t are calculated. If there are less rejected requests than with the previous static schedule, the current schedule is stored as the best schedule in Γ^*.

Pseudocode 1 Simplified model

Input: S = static requests, λ_t = arrival process dynamic requests
Output: Γ^* = best static schedule

```
Make static schedule of S →(Γ, Unassigned)
Calculate R_T and r_t with Γ and λ_t
f = 0
while Unassigned = ∅ and f < 3 do
   R'_T = R_T
   Calculate r_max = ⌊R_T⌋ time slots with highest value of r_t
   Create dummy requests with times from r_max → D
   Make static schedule of S and D →(Γ, Unassigned)
   Remove D from Γ
   Calculate R_T and r_t with Γ and λ_t
   if R_T < R'_T then
      Γ* = Γ
      f = 0
   else
      f = f + 1
   end if
end while
```

7.2 Complete model for the combination of static and dynamic requests

If we have more than two locations we cannot use the model of the previous section. In this analytical model we use the fact that if a vehicle starts driving, it is only driving for the next p time slots and after that it is at location P again to pick up new customers. In the complete model there are several locations and the travel times are not all equal. In the simplified model we could determine whether a dynamic customer could be planned by counting the number of available space in the vehicles that are scheduled to drive to location D. In the complete model this is not this easy. For each vehicle it needs to be calculated whether if the vehicle picks up and delivers the dynamic request, all other requests in the vehicle are still arriving on time. Only in this way we can determine whether a dynamic request is rejected. To do this we use our heuristic planning model described in Section 5.3 combined with a simulation of the dynamic requests. We first create a planning with the static requests and then dynamic requests are simulated according to the probability distribution of the dynamic requests, which is the probability that a request arrives from location a to location b on time t. These new requests are tried to add to the new planning using the heuristic and the number of requests that have to be rejected is counted.

The algorithm where we do take the dynamic requests into account is described in Pseudocode 2. In this case we first cluster both the locations as well as the dynamic requests. Locations that are close to each other are modelled as one clustered location and requests that have the same clustered pickup and arrival location and approximately the same time window are seen as the same request and their arrival rates are added. This is necessary because the specific arrival rates are usually really low. These clustered requests we add to the static requests as dummy requests and a planning is made with both kind of requests. In this planning as much space as possible is created for the dynamic requests. After this planning is made the dynamic dummy requests are removed from the planning again. The planning that remains is a static planning with space reserved for the expected dynamic requests. These dynamic requests are simulated according to their probability distribution and tried to fit in the static schedule.

Pseudocode 2 Complete model

Input: probability distribution of dynamic requests, static requests
Output: Unassigned requests

```
Cluster locations
Cluster dynamic requests
Plan clustered dynamic requests as dummy requests
Make planning with static requests and dummy requests (static requests leave as much space
as possible for dummy requests)
Remove dummy requests from planning
Simulate requests according to the probability distribution of the dynamic requests
Plan simulated requests in planning ⇒ Unassigned requests
```

In Pseudocode 3 the algorithm is described when the dynamic requests are not taken into account while making the static schedule. The aim of this algorithm is to compare it to the algorithm where the dynamic requests are taken into account. We would expect that in this case more dynamic requests are rejected, since no space is reserved for them.

Pseudocode 3 Compared model

Input: probability distribution of dynamic requests, static requests
Output: Unassigned requests

```
Make planning with static requests
Simulate requests according to the probability distribution of the dynamic requests
Plan simulated requests in planning ⇒ Unassigned requests
```

7.3 Combination of stochastic travel times and dynamic requests

One of our sub questions is whether we can combine stochastic travel times and taking into account the dynamic requests. The stochastic constraints are added as feasibility tests when a request is added and the dynamic requests are added as dummy requests. Because these are two different parts of the model, these two could easily be combined.

7.4 Machine scheduling problem

To determine when it may be better to reserve special vehicles for the dynamic requests and when not, we introduce a machine scheduling problem. The vehicles are represented by machines and the requests by jobs. We have both static and dynamic jobs and want to calculate in which cases we should leave a machine idle for the dynamic jobs. To do so, we first calculate the probability that the dynamic jobs can be scheduled.

We have m machines on which n jobs need to be scheduled. A machine can only serve one job at a time. The first q jobs are static, which means they are already planned on the machines. The other $r = n - q$ jobs are dynamic, they arrive later. We assume that all r jobs arrive at the same time. The times the first q jobs need to be scheduled are homogeneously distributed over an interval from 0 to N. The last r jobs arrive somewhere uniformly distributed on the interval. We call the desired time of job k to be scheduled A_k. Each job has a time interval in which the service has to start with length 1, $[A_k - \frac{1}{2}, A_k + \frac{1}{2}]$. Each job has the same service time s. We want to determine the probability p that the last r jobs can be served by the machine if the machine also serves the first q jobs, where all jobs start within the desired time window. Since the first q jobs are homogeneously distributed on the interval $[0, N]$, each $\frac{N}{q+1}$ time units a job is planned, so $A_k = \frac{kN}{q+1}$. In this way the time before the first job, the time between all jobs and the time after the last job are all equal. The first q jobs are scheduled on times $(S_1, ..., S_q)$, which we assume is a feasible planning. We can still shift these jobs within their time windows.

Two examples of how the planning of the first q jobs on one machine looks are given in Figure 3. In the first figure $\frac{N}{q+1} \geq 1$, which means the time windows do not overlap. In the second figure the time windows do overlap since $\frac{N}{q+1} < 1$. In these examples each job is scheduled in the middle of its time window, so $S_k = \frac{kN}{q+1}$ for each job k. Whether the r dynamic jobs can still be scheduled depends on the ratio between the service time, length of the time window, time frame and number of jobs. We first look at the case where there is one machine and one dynamic job.

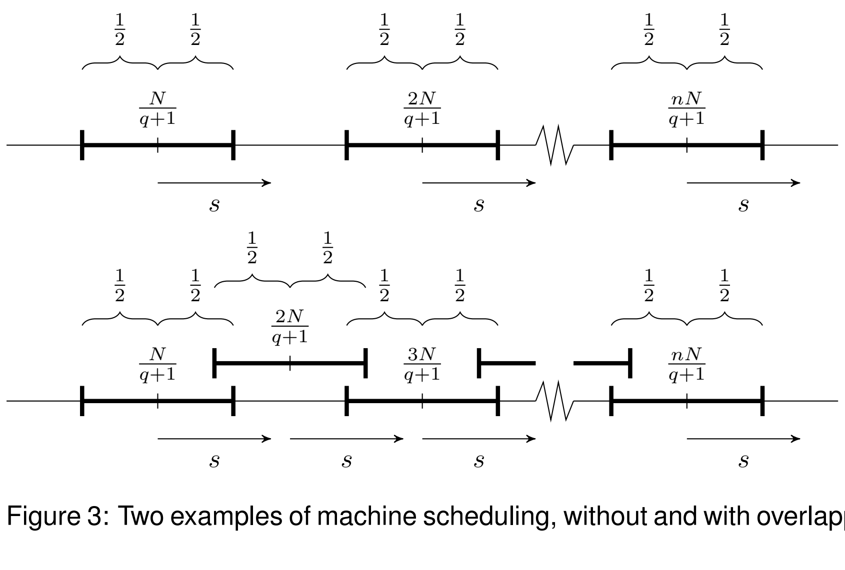

Figure 3: Two examples of machine scheduling, without and with overlapping of the time windows.

7.4.1 One machine and one dynamic job

We start with the case where $m = 1$ and $r = 1$. We call the dynamic job we have to add to the q static jobs already scheduled job R. We formulate five theorems. In each theorem p, the probability that job R can be scheduled, is given for a range of the service time s.

Theorem 7.1 *If $s \leq \min\{1, \frac{N}{q+1}\}$ then $p = 1$.*

Proof We prove that if $s \leq \min\{1, \frac{N}{q+1}\}$ we can always schedule job R. We first schedule the first q jobs in the middle of their time windows, so $S_k = \frac{kN}{q+1}$ for each job $1 \leq k \leq q$.

Suppose the time job R needs to be scheduled is during a time where a job $1 \leq k \leq q$ is already served on the machine, so $\frac{kN}{q+1} \leq A_R < \frac{kN}{q+1} + s$. In this case we shift jobs $(1, ..., k)$ $\frac{1}{2}$ time units to the left, so $S_k = \frac{kN}{q+1} - \frac{1}{2}$. All jobs $(k+1, ..., q)$ we shift to the right, so $S_{k+1} = \frac{(k+1)N}{q+1} + \frac{1}{2}$. Now we can prove that we can always schedule job R at time $S_R = \frac{kN}{q+1} - \frac{1}{2} + s$. First we need to prove that this time is always within the time window of job R. We have $A_R < \frac{kN}{q+1} + s$, so $A_R - \frac{1}{2} < \frac{kN}{q+1} - \frac{1}{2} + s$, which means the left boundary of the time window is satisfied. Since $s \leq 1$ and $\frac{kN}{q+1} \leq A_R$ we have $\frac{kN}{q+1} + s \leq A_R + 1$, which results in $\frac{kN}{q+1} - \frac{1}{2} + s \leq A_R + \frac{1}{2}$, satisfying the right boundary of the time window. Second we have to prove that job k will be finished before job R is scheduled to start and job R will be finished before job $k+1$ is scheduled to start. Since job k is scheduled at time $\frac{kN}{q+1} - \frac{1}{2}$ it is finished at time $\frac{kN}{q+1} - \frac{1}{2} + s$. Job R will end at time $\frac{kN}{q+1} - \frac{1}{2} + 2s$. Since $s \leq \frac{N}{q+1}$ and $s \leq 1$ we have $\frac{kN}{q+1} + 2s \leq \frac{kN}{q+1} + \frac{N}{q+1} + 1$, which results in $\frac{kN}{q+1} - \frac{1}{2} + 2s \leq \frac{(k+1)N}{q+1} + \frac{1}{2} = S_{k+1}$ so job R will be finished before the start of job $k+1$.

Suppose now the time job R needs to be scheduled is during an idle time on the machine, so for some job $1 \leq k \leq q$ we have $\frac{kN}{q+1} + s \leq A_R < \frac{(k+1)N}{q+1}$. Again we shift jobs $(1, ..., k)$ $\frac{1}{2}$ time units to the left and jobs $(k+1, ..., q)$ $\frac{1}{2}$ to the right, so $S_k = \frac{kN}{q+1} - \frac{1}{2}$ and $S_{k+1} = \frac{(k+1)N}{q+1} + \frac{1}{2}$. We prove that we can schedule job R now at time $A_R - \frac{1}{2}$, which is the left boundary of its time window. We only have to prove that job k will be finished before job R is scheduled to start and job R will be finished before job $k+1$ is scheduled to start. Since $\frac{kN}{q+1} + s \leq A_R$ we have $S_k + s = \frac{kN}{q+1} - \frac{1}{2} + s \leq A_R - \frac{1}{2} = S_R$, which means job k is finished before the start of job R. Since $A_R < \frac{(k+1)N}{q+1}$ and $s \leq 1$ we have $A_R + s < \frac{(k+1)N}{q+1} + 1$, which results in $S_R + s = A_R - \frac{1}{2} + s < \frac{(k+1)N}{q+1} + \frac{1}{2} = S_{k+1}$ so job R ends before the start of job $k+1$. ■

Theorem 7.2 *If $1 < s \leq \frac{N}{q+1}$ and $s > \frac{1}{2}\left(\frac{N}{q+1} + 1\right)$ then $p = \frac{2}{q+1} + \frac{2(1-s)}{N}$.*

Proof If $s \leq \frac{N}{q+1}$ we can schedule all jobs $1 \leq k \leq q$ on time $S_k = \frac{kN}{q+1}$. If we need to schedule job R in between job k and $k+1$ the maximum space we can create for this job is $\frac{N}{q+1} + 1 - s$, which is the case if we schedule the jobs at times $S_k = \frac{kN}{q+1} - \frac{1}{2}$ and $S_{k+1} = \frac{(k+1)N}{q+1} + \frac{1}{2}$. If this space is smaller than s, the machine cannot serve job R in this interval. So if $s > \frac{1}{2}\left(\frac{N}{q+1} + 1\right)$, we cannot schedule job R in between two jobs.

If we shift all jobs to their latest start time, so $S_k = \frac{kN}{q+1} + \frac{1}{2}$, the latest time job R can start before the first static job is $S_R = \frac{N}{q+1} + \frac{1}{2} - s$. Because of the time window of job R it can arrive between 0 and $S_R = \frac{N}{q+1} + 1 - s$. Since $s \leq \frac{N}{q+1}$ the length of this interval is positive. If we shift all jobs to their earliest start time, so $S_k = \frac{kN}{q+1} - \frac{1}{2}$, the earliest time job R can start after the last static job is $S_R = \frac{qN}{q+1} - \frac{1}{2} + s$. Because of the time window of job R it can arrive between $S_R = \frac{qN}{q+1} - 1 + s$ and N. The length of this interval is again $\frac{N}{q+1} + 1 - s$ since $N - \frac{qN}{q+1} = \frac{N}{q+1}$.

Together, before and after the static jobs, we have an interval of length $\frac{2N}{q+1} + 2(1-s)$, which gives $p = \frac{2}{q+1} + \frac{2(1-s)}{N}$. ∎

Theorem 7.3 *If* $1 < s \le \frac{N}{q+1}$ *and* $s \le \frac{1}{2}\left(\frac{N}{q+1} + 1\right)$ *then* $p = 1 - \frac{2q(s-1)}{N}$.

Proof Since $\frac{N}{q+1} \ge 1$ the time windows do not overlap. However, since $s > 1$ for each job we have a time in which job R cannot be scheduled. For some job $1 \le k \le q$ the time in which job R cannot be scheduled is from $\frac{kN}{q+1} + 1 - s$ to $\frac{kN}{q+1} - 1 + s$. The latest time job R can start before job k is the latest time job k can start minus s, $\frac{kN}{q+1} + \frac{1}{2} - s$. We add an extra $\frac{1}{2}$ because of the time window of job R. For the first job this is $\frac{N}{q+1} + 1 - s$, which is greater than zero since $s \le \frac{N}{q+1}$. The earliest time job R can start after job k is the earliest time job k can start plus s, $\frac{kN}{q+1} - \frac{1}{2} + s$. We subtract an extra $\frac{1}{2}$ because of the time window of job R. For the last job this is $\frac{qN}{q+1} - 1 + s$, which is smaller than N since $s \le \frac{N}{q+1}$. The length of this interval in which job R cannot be scheduled is $2(s-1)$, which is greater than zero since $s > 1$. Since for each job we have an interval of this length, the total length in which job R cannot be scheduled is equal to $2q(s-1)$. This results in a probability of $p = 1 - \frac{2q(s-1)}{N}$ that job R can be scheduled. ∎

Lemma 7.4 *If* $s > \frac{N}{q+1}$, $s > \frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right)$ *and* $\frac{qN}{q+1} + 1 - qs < A_R < \frac{N}{q+1} - 1 + qs$ *job* R *cannot be scheduled.*

Proof If $s > \frac{N}{q+1}$ we cannot schedule the first q jobs in the middle of their time window, since the service time of each job is longer than the time between each job. If we schedule job R in between job k and $k+1$ we create the largest space if we schedule job k to end at $s_k + s = \frac{N}{q+1} - \frac{1}{2} + ks$ and job $k+1$ to start at $s_{k+1} = \frac{qN}{q+1} + \frac{1}{2} - (q-k-1)s$. The service time of job R needs to be smaller than this space in order to be able to plan job R, so we need $s \le \frac{(q-1)N}{q+1} + 1 - (q-1)s$. This means that since $s > \frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right)$ we cannot schedule job R in between two jobs, but we might be able to schedule job R before or after all jobs.

If we shift all jobs to their latest start times, the first job starts at $S_1 = \frac{qN}{q+1} + \frac{1}{2} - (q-1)s$, so the latest time job R can start before the first job is $S_R = \frac{qN}{q+1} + \frac{1}{2} - qs$. Because of the time window of job R it can arrive $\frac{1}{2}$ time unit later, so the latest time job R can arrive before the first job is $A_R = \frac{qN}{q+1} + 1 - qs$.

The earliest time the last static job can be finished is $S_q + s = \frac{N}{q+1} - \frac{1}{2} + qs$. So the earliest time job R can arrive after the last static job is $A_R = \frac{N}{q+1} - 1 + qs$. ∎

Theorem 7.5 *If* $s > \frac{N}{q+1}$ *and* $s > \frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right)$ *we have the following cases and their probabilities that job* R *can be scheduled:*
if $s > \frac{N}{q+1} + \frac{1}{q}$ *then* $p = 0$,
if $s \le \frac{N}{q+1} + \frac{1}{q}$ *then* $p = \frac{2q}{q+1} - \frac{2(qs-1)}{N}$.

Proof From Lemma 7.4 we know that if job R arrives between $\frac{qN}{q+1} + 1 - qs$ and $\frac{N}{q+1} - 1 + qs$ it cannot be scheduled. Whether job R can arrive before or after all static jobs depends on whether $\frac{qN}{q+1} + 1 - qs$ is smaller or larger than zero and whether $\frac{N}{q+1} - 1 + qs$ is smaller or larger than N. Note that if $\frac{qN}{q+1} + 1 - qs < 0$ we also have $\frac{N}{q+1} - 1 + qs > N$ since $N - \frac{N}{q+1} = \frac{qN}{q+1}$.

If $s > \frac{N}{q+1} + \frac{1}{q}$ we have both $\frac{qN}{q+1} + 1 - qs < 0$ and $\frac{N}{q+1} - 1 + qs > N$, so there is no space for job R to be scheduled before the first static job or after the last static job. This means that in this case $p = 0$.

If $s \leq \frac{N}{q+1} + \frac{1}{q}$ there is space for job R to be scheduled both before the first static job and after the last static job. The length of the interval in which job R can arrive is on both sides $\frac{qN}{q+1} + 1 - qs$, which results in a probability of $p = \frac{2q}{q+1} - \frac{2(qs-1)}{N}$ that job R can be scheduled. ■

Theorem 7.6 *If* $\frac{N}{q+1} < s \leq \frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right)$ *then* $p = 1$.

Proof We prove that if $\frac{N}{q+1} < s \leq \frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right)$ we can always schedule job R. Since in this case $\frac{N}{q+1} < s$, each job needs to start later in its time window than the previous one. We schedule all jobs $1 \leq k \leq q$ at time $S_k = \frac{N}{q+1} - \frac{1}{2} + (k-1)s$, so at the earliest possible time. Since in this way there is no idle time on the machine between jobs, we are sure that A_R is during a time when a job is processed on the machine. We assume that for some job $1 \leq k \leq q$ we have $S_k - \frac{1}{2} = \frac{N}{q+1} - 1 + (k-1)s \leq A_R < \frac{N}{q+1} - 1 + ks = S_k - \frac{1}{2} + s = S_{k+1} - \frac{1}{2}$. In this case we shift jobs $(k, ..., q)$ to the right as much as possible, so $S_k = \frac{qN}{q+1} + \frac{1}{2} - (q-k)s$. Now we can prove that we can always schedule job R at time $S_R = S_{k-1} + s = \frac{N}{q+1} - \frac{1}{2} + (k-1)s$.

First we need to prove that this time is always within the time window of job R. We have $A_R < \frac{N}{q+1} - 1 + ks$, so $A_R - \frac{1}{2} < \frac{N}{q+1} - \frac{3}{2} + ks < S_R$, since $s \leq 1$. Because $\frac{N}{q+1} \leq 1$, we have $s \leq \frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right) \leq 1$. This means the left boundary of the time window is satisfied. Since $\frac{N}{q+1} - 1 + (k-1)s \leq A_R$ we have $S_R = \frac{N}{q+1} - \frac{1}{2} + (k-1)s \leq A_R + \frac{1}{2}$, satisfying the right boundary of the time window. Second we have to prove that job $k-1$ will be finished before job R is scheduled to start and job R will be finished before job k is scheduled to start. Since $S_R = S_{k-1} + s$ job R will start when job $k-1$ is finished. Since $s \leq \frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right)$ we have $S_k = \frac{qN}{q+1} + \frac{1}{2} - (q-k)s \geq \frac{N}{q+1} - \frac{1}{2} + ks = S_R + s$, which means job R will be finished before the start of job k. ■

In Table 1 a summary of the conclusions is shown. For each set of conditions for s the probability that job R can be scheduled is given.

Theorem	Conditions					Probability
7.1	$s \leq \min\{1, \frac{N}{q+1}\}$					$p = 1$
7.2	$1 < s \leq \frac{N}{q+1}$	and	$s > \frac{1}{2}\left(\frac{N}{q+1} + 1\right)$			$p = \frac{2}{q+1} + \frac{2(1-s)}{N}$
7.3	$1 < s \leq \frac{N}{q+1}$	and	$s \leq \frac{1}{2}\left(\frac{N}{q+1} + 1\right)$			$p = 1 - \frac{2q(s-1)}{N}$
7.5	$s > \frac{N}{q+1}$	and	$s > \frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right)$	and	$s > \frac{N}{q+1} + \frac{1}{q}$	$p = 0$
7.5	$s > \frac{N}{q+1}$	and	$s > \frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right)$	and	$s \leq \frac{N}{q+1} + \frac{1}{q}$	$p = \frac{2q}{q+1} - \frac{2(qs-1)}{N}$
7.6	$s > \frac{N}{q+1}$	and	$s \leq \frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right)$			$p = 1$

Table 1: Probability that job R can be scheduled in different situations when we have one machine

7.4.2 Multiple machines and one dynamic job

Next we examine the case where $m > 1$ and $r = 1$. Again we call the dynamic job we have to add to the q static jobs already scheduled job R. We divide all jobs q over all m machines as follows: the first job is served by the first machine, the second job by the second machine, the m^{th} job by the m^{th} machine, the $m + 1^{th}$ job by the first machine etc. In this way machine v has to serve $q_v = \lfloor \frac{q}{m} \rfloor + \lfloor \min \left\{ \frac{q - m \cdot \lfloor \frac{q}{m} \rfloor}{v}, 1 \right\} \rfloor$ jobs. The times at which these jobs need to be scheduled on machine v are $\left(\frac{vN}{q+1}, \frac{(v+m)N}{q+1}, ..., \frac{(v+(q_v-1)m)N}{q+1} \right)$, so $A_k = \frac{(v+(k-1)m)N}{q+1}$. We assume that the number of jobs is larger or equal than the number of machines, so $q \geq m$, such that every machine serves at least one job. If this is not the case we have at least one empty machine where the dynamic job can always be scheduled. We now formulate five theorems about the probability p that job R can be scheduled.

Theorem 7.7 *If $s \leq \min\{1, \frac{mN}{q+1}\}$ then $p = 1$.*

Proof We prove that if $s \leq \min\{1, \frac{mN}{q+1}\}$ we can always schedule job R on each machine v. We first schedule the first q_v jobs in the middle of their time windows, so $S_k = \frac{(v+(k-1)m)N}{q+1}$ for each job $1 \leq k \leq q_v$.

Suppose the time job R needs to be scheduled is during a time where a job $1 \leq k \leq q_v$ is already served on machine v, so $\frac{(v+(k-1)m)N}{q+1} \leq A_R < \frac{(v+(k-1)m)N}{q+1} + s$. In this case we shift jobs $(1, ..., k)$ $\frac{1}{2}$ time units to the left, so $S_k = \frac{(v+(k-1)m)N}{q+1} - \frac{1}{2}$. All jobs $(k+1, ..., q_v)$ we shift to the right, so $S_{k+1} = \frac{(v+km)N}{q+1} + \frac{1}{2}$. Now we can prove that we can always schedule job R at time $S_R = \frac{(v+(k-1)m)N}{q+1} - \frac{1}{2} + s$. First we need to prove that this time is always within the time window of job R. We have $A_R < \frac{(v+(k-1)m)N}{q+1} + s$, so $A_R - \frac{1}{2} < \frac{(v+(k-1)m)N}{q+1} - \frac{1}{2} + s$, which means the left boundary of the time window is satisfied. Since $s \leq 1$ and $\frac{(v+(k-1)m)N}{q+1} \leq A_R$ we have $\frac{(v+(k-1)m)N}{q+1} + s \leq A_R + 1$, which results in $\frac{(v+(k-1)m)N}{q+1} - \frac{1}{2} + s \leq A_R + \frac{1}{2}$, satisfying the right boundary of the time window. Second we have to prove that job k will be finished before job R is scheduled to start and job R will be finished before job $k+1$ is scheduled to start. Since job k is scheduled at time $\frac{(v+(k-1)m)N}{q+1} - \frac{1}{2}$ it is finished at time $\frac{(v+(k-1)m)N}{q+1} - \frac{1}{2} + s$. Job R will end at time $\frac{(v+(k-1)m)N}{q+1} - \frac{1}{2} + 2s$. Since $s \leq \frac{mN}{q+1}$ and $s \leq 1$ we have $\frac{(v+(k-1)m)N}{q+1} + 2s \leq \frac{(v+(k-1)m)N}{q+1} + \frac{mN}{q+1} + 1$, which results in $\frac{(v+(k-1)m)N}{q+1} - \frac{1}{2} + 2s \leq \frac{(v+km)N}{q+1} + \frac{1}{2} = S_{k+1}$ so job R will be finished before the start of job $k+1$.

Suppose now the time job R needs to be scheduled is during an idle time on the machine, so for some job $1 \leq k \leq q_v$ we have $\frac{(v+(k-1)m)N}{q+1} + s \leq A_R < \frac{(v+km)N}{q+1}$. Again we shift jobs $(1, ..., k)$ $\frac{1}{2}$ time units to the left and jobs $(k+1, ..., q_v)$ $\frac{1}{2}$ to the right, so $S_k = \frac{(v+(k-1)m)N}{q+1} - \frac{1}{2}$ and $S_{k+1} = \frac{(v+km)N}{q+1} + \frac{1}{2}$. We prove that we can schedule job R now at time $A_R - \frac{1}{2}$, which is the left boundary of its time window. We only have to prove that job k will be finished before job R is scheduled to start and job R will be finished before job $k+1$ is scheduled to start. Since $\frac{(v+(k-1)m)N}{q+1} + s \leq A_R$ we have $S_k + s = \frac{(v+(k-1)m)N}{q+1} - \frac{1}{2} + s \leq A_R - \frac{1}{2} = S_R$, which means job k is finished before the start of job R. Since $A_R < \frac{(v+km)N}{q+1}$ and $s \leq 1$ we have $A_R + s < \frac{(v+km)N}{q+1} + 1$, which results in $S_R + s = A_R - \frac{1}{2} + s < \frac{(v+km)N}{q+1} + \frac{1}{2} = S_{k+1}$ so job R ends before the start of job $k+1$. ■

Lemma 7.8 *If $1 < s \leq \frac{mN}{q+1}$, $s > \frac{1}{2}\left(\frac{mN}{q+1} + 1\right)$ and $\frac{vN}{q+1} + 1 - s < A_R < \frac{(v+(q_v-1)m)N}{q+1} - 1 + s$ job R cannot be scheduled on machine v.*

Proof If $s \leq \frac{mN}{q+1}$ we can schedule all jobs $1 \leq k \leq q_v$ on time $S_k = A_k = \frac{(v+(k-1)m)N}{q+1}$. If we need to schedule job R in between job k and $k+1$ the maximum space we can create for this job is $\frac{mN}{q+1}+1-s$, which is the case if we schedule the jobs at times $S_k = \frac{(v+(k-1)m)N}{q+1} - \frac{1}{2}$ and $S_{k+1} = \frac{(v+km)N}{q+1} + \frac{1}{2}$. If this space is smaller than s, the machine cannot serve job R in this interval, this is the case since $s > \frac{1}{2}\left(\frac{mN}{q+1}+1\right)$. So we cannot schedule job R in between two jobs, but we might be able to schedule job R before or after all jobs.

If we shift all jobs to their latest start times, the first job starts at $S_1 = \frac{vN}{q+1} + \frac{1}{2}$, so the latest time job R can start before the first job is $S_R = \frac{vN}{q+1} + \frac{1}{2} - s$. Because of the time window of job R it can arrive $\frac{1}{2}$ time unit later, so the latest time job R can arrive before the first job is $A_R = \frac{vN}{q+1} + 1 - s$.

The earliest time the last static job can be finished is $S_{q_v} + s = \frac{(v+(q_v-1)m)N}{q+1} - \frac{1}{2} + s$. So the earliest time job R can arrive after the last static job is $A_R = \frac{(v+(q_v-1)m)N}{q+1} - 1 + s$. ■

Theorem 7.9 *If $1 < s \leq \frac{mN}{q+1}$, $s > \frac{1}{2}\left(\frac{mN}{q+1}+1\right)$ we have the following cases and their probabilities that job R can be scheduled:*
if $s \leq \frac{mN}{q+1} + 1 - \frac{1}{2}N$ then $p = 1$,
if $s > \frac{mN}{q+1} + 1 - \frac{1}{2}N$ then $p = 2\left(\frac{m}{q+1} - \frac{s-1}{N}\right)$.

Proof From Lemma 7.8 we know that if job R arrives between $\frac{vN}{q+1}+1-s$ and $\frac{(v+(q_v-1)m)N}{q+1}-1+s$ it cannot be scheduled on machine v. It only has to be able to be scheduled on one machine, so we need to calculate the length of the interval in which job R cannot be scheduled on any of the machines. In order to calculate this we need the machine for which the lower bound $\frac{vN}{q+1}+1-s$ is maximal and the machine for which the upper bound $\frac{(v+(q_v-1)m)N}{q+1}-1+s$ is minimal. The maximal lower bound is achieved when $v = m$ and the minimal upper bound is achieved when $v = q - m\lfloor\frac{q}{m}\rfloor + 1$. This means that job R cannot be scheduled if $\frac{mN}{q+1}+1-s < A_R < \frac{(q-m\lfloor\frac{q}{m}\rfloor+1+(\lfloor\frac{q}{m}\rfloor-1)m)N}{q+1} - 1 + s = N - \frac{mN}{q+1} - 1 + s$. Since $s \leq \frac{mN}{q+1}$ we know that $\frac{mN}{q+1}+1-s > 0$ and $N - \frac{mN}{q+1} - 1 + s < N$.

If $N - \frac{mN}{q+1} - 1 + s \leq \frac{mN}{q+1} + 1 - s$ then there is no interval in which job R cannot be scheduled resulting in $p = 1$, which is the case if $s \leq \frac{mN}{q+1} + 1 - \frac{1}{2}N$.

If $N - \frac{mN}{q+1} - 1 + s > \frac{mN}{q+1} + 1 - s$ there is an interval of length $N - \frac{mN}{q+1} - 1 + s - \frac{mN}{q+1} - 1 + s = N - 2\left(\frac{mN}{q+1}+1-s\right)$ in which job R cannot be scheduled on any machine. So the probability that job R can be scheduled is $p = 2\left(\frac{m}{q+1} - \frac{s-1}{N}\right)$. ■

Lemma 7.10 *If $1 < s \leq \frac{mN}{q+1}$, $s \leq \frac{1}{2}\left(\frac{mN}{q+1}+1\right)$ then for each job $1 \leq k \leq q_v$ we have an interval in which job R cannot arrive. If $\frac{(v+(k-1)m)N}{q+1} + 1 - s < A_R < \frac{(v+(k-1)m)N}{q+1} - 1 + s$ for some job k, job R cannot be scheduled on machine v.*

Proof Since $1 < \frac{mN}{q+1}$ the time windows do not overlap. Since $s \leq \frac{mN}{q+1}$, we can schedule each job at the same place in their time window. Since $s \leq \frac{1}{2}\left(\frac{mN}{q+1}+1\right)$ we can create some space to schedule job R between each two jobs, but since $s > 1$ for each job we also have a time in which job R cannot be scheduled. If we want to schedule job R after job k we can shift job k to the beginning of its time window. The earliest time job R can start after job k is then $\frac{(v+(k-1)m)N}{q+1} - \frac{1}{2} + s$. The latest time job R can start before job k is $\frac{(v+(k-1)m)N}{q+1} + \frac{1}{2} - s$. Because of the time window of job R it can arrive either before $\frac{(v+(k-1)m)N}{q+1} + 1 - s$ or after $\frac{(v+(k-1)m)N}{q+1} - 1 + s$, if it arrives in between those values it cannot be scheduled. ■

Theorem 7.11 *If $1 < s \leq \frac{mN}{q+1}$, $s \leq \frac{1}{2}\left(\frac{mN}{q+1}+1\right)$ we have the following cases and their probabilities that job R can be scheduled:*
if $s \leq \frac{1}{2}\frac{(m-1)N}{q+1}+1$ then $p = 1$,
if $s > \frac{1}{2}\frac{(m-1)N}{q+1}+1$ then $p = 1 - m\left(\lfloor\frac{q}{m}\rfloor - 1\right)\left(\frac{1-m}{q+1}+\frac{2(s-1)}{N}\right)$.

Proof From Lemma 7.10 we know that if $\frac{(v+(k-1)m)N}{q+1} + 1 - s < A_R < \frac{(v+(k-1)m)N}{q+1} - 1 + s$ for some job $1 \leq k \leq q_v$, job R cannot be scheduled on machine v. On the first machine this interval in which job R cannot arrive starts earliest for each job $1 \leq k \leq \lfloor\frac{q}{m}\rfloor$ and on the last machine this interval starts latest. If the interval on the first machine ends before the interval on the last machine starts, job R can always be scheduled on one of the machines. So if $\frac{(1+(k-1)m)N}{q+1} - 1 + s \leq \frac{(m+(k-1)m)N}{q+1} + 1 - s$ we have $p = 1$, which is the case if $s \leq \frac{1}{2}\frac{(m-1)N}{q+1} + 1$.

If $s > \frac{1}{2}\frac{(m-1)N}{q+1} + 1$ we have an interval of length $\frac{(1+(k-1)m)N}{q+1} - 1 + s - \frac{(m+(k-1)m)N}{q+1} - 1 + s = \frac{(1-m)N}{q+1} - 2 + 2s$ for each $1 \leq k \leq \lfloor\frac{q}{m}\rfloor$ in which job R cannot arrive. Since $s \leq \frac{mN}{q+1}$ the interval for $k = 1$ starts after 0 and the interval for $k = \lfloor\frac{q}{m}\rfloor$ ends before N. So the total length of the interval in which job R cannot arrive is $\lfloor\frac{q}{m}\rfloor\left(\frac{(1-m)N}{q+1} - 2 + 2s\right)$.

Furthermore, if job $k+1$ on machine v starts earlier than job k on machine $v+1$ ends, there is also an interval in which job R cannot arrive. If $\frac{(v+km)N}{q+1} + 1 - s \geq \frac{(v+1+(k-1)m)N}{q+1} - 1 + s$, so if $s \leq \frac{1}{2}\frac{(m-1)N}{q+1} + 1$, this is not the case and we still have $p = 1$. If $s > \frac{1}{2}\frac{(m-1)N}{q+1} + 1$ we have an interval of length $\lfloor\frac{q}{m}\rfloor\left(\frac{(1-m)N}{q+1} - 2 + 2s\right) + \left(\lfloor\frac{q}{m}\rfloor - 1\right)(m-1)\left(\frac{(1-m)N}{q+1} - 2 + 2s\right) = m\left(\lfloor\frac{q}{m}\rfloor - 1\right)\left(\frac{(1-m)N}{q+1} - 2 + 2s\right)$, resulting in a probability of
$p = 1 - m\left(\lfloor\frac{q}{m}\rfloor - 1\right)\left(\frac{1-m}{q+1} + \frac{2(s-1)}{N}\right)$. ∎

Lemma 7.12 *If $s > \frac{mN}{q+1}$ and $s > \frac{1}{q_v}\left(\frac{(q_v-1)mN}{q+1}+1\right)$ and $\frac{(v+(q_v-1)m)N}{q+1} + 1 - q_v s < A_R < \frac{vN}{q+1} - 1 + q_v s$ job R cannot be scheduled on machine v.*

Proof If $s > \frac{mN}{q+1}$ we cannot schedule the first q_v jobs in the middle of their time window, since the service time of each job is longer than the time between each job. If we schedule job R in between job k and $k+1$ we create the largest space if we schedule job k to end at $S_k + s = \frac{vN}{q+1} - \frac{1}{2} + ks$ and job $k+1$ to start at $S_{k+1} = \frac{(v+(q_v-1)m)N}{q+1} + \frac{1}{2} - (q_v - k - 1)s$. If the service time of job R is larger than this maximum space, so if $s > \frac{(q_v-1)mN}{q+1} + 1 - (q_v - 1)s$ we cannot schedule job R in between two jobs. This is the case since $s > \frac{1}{q_v}\left(\frac{(q_v-1)mN}{q+1}+1\right)$. So we cannot schedule job R in between two jobs, but we might be able to schedule job R before or after all jobs.

If we shift all jobs to their latest start times, the first job starts at $S_1 = \frac{(v+(q_v-1)m)N}{q+1} + \frac{1}{2} - (q_v - 1)s$, so the latest time job R can start before the first job is $S_R = \frac{(v+(q_v-1)m)N}{q+1} + \frac{1}{2} - q_v s$. Because of the time window of job R it can arrive $\frac{1}{2}$ time unit later, so the latest time job R can arrive before the first job is $A_R = \frac{(v+(q_v-1)m)N}{q+1} + 1 - q_v s$.

The earliest time the last static job can be finished is $S_{q_v} + s = \frac{vN}{q+1} - \frac{1}{2} + q_v s$. So the earliest time job R can arrive after the last static job is $A_R = \frac{vN}{q+1} - 1 + q_v s$. ∎

Theorem 7.13 *If $s > \frac{mN}{q+1}$ and $s > \frac{1}{q_v}\left(\frac{(q_v-1)mN}{q+1}+1\right)$ $\forall v$ we have the following cases and their probabilities that job R can be scheduled:*
if $s \geq \frac{mN}{q+1} + \frac{1}{\lfloor \frac{q}{m} \rfloor}$ then $p = 0$,
if $s \leq \frac{mN}{q+1} + \frac{1-\frac{1}{2}N}{\lfloor \frac{q}{m} \rfloor}$ then $p = 1$,
if $\frac{mN}{q+1} + \frac{1-\frac{1}{2}N}{\lfloor \frac{q}{m} \rfloor} < s < \frac{mN}{q+1} + \frac{1}{\lfloor \frac{q}{m} \rfloor}$ then $p = 2\left(\frac{\lfloor \frac{q}{m} \rfloor s+1}{N} - \frac{m\lfloor \frac{q}{m} \rfloor}{q+1}\right)$.

Proof From Lemma 7.12 we know that if job R arrives between $\frac{(v+(q_v-1)m)N}{q+1} + 1 - q_v s$ and $\frac{vN}{q+1} - 1 + q_v s$ it cannot be scheduled on machine v. It only has to be able to be scheduled on one machine, so we need to calculate the length of the interval in which job R cannot be scheduled on any of the machines. In order to calculate this we need the machine for which the lower bound $\frac{(v+(q_v-1)m)N}{q+1} + 1 - q_v s$ is maximal and the machine for which the upper bound $\frac{vN}{q+1} - 1 + q_v s$ is minimal. The maximal lower bound is achieved when $v = m$, since in this case v is maximal and q_v is minimal, which we want since $s > \frac{mN}{q+1}$. The minimal upper bound is achieved when $v = 1$ if $s \leq \frac{(q-m\lfloor \frac{q}{m} \rfloor)N}{q+1}$ and $v = q - m\lfloor \frac{q}{m} \rfloor + 1$ when $s > \frac{(q-m\lfloor \frac{q}{m} \rfloor)N}{q+1}$. Since $\lfloor \frac{q}{m} \rfloor \geq \frac{q}{m} - 1$ we can prove that $\frac{(q-m\lfloor \frac{q}{m} \rfloor)N}{q+1} \leq \frac{mN}{q+1}$. Since $s > \frac{mN}{q+1}$ we also have $s > \frac{(q-m\lfloor \frac{q}{m} \rfloor)N}{q+1}$, so the minimal upper bound is achieved when $v = q - m\lfloor \frac{q}{m} \rfloor + 1$.

This means that job R cannot be scheduled if $\frac{(m+(\lfloor \frac{q}{m} \rfloor-1)m)N}{q+1} + 1 - \lfloor \frac{q}{m} \rfloor s < A_R < \frac{(q-m\lfloor \frac{q}{m} \rfloor+1)N}{q+1} - 1 + \lfloor \frac{q}{m} \rfloor s$, which is the case if $\frac{m\lfloor \frac{q}{m} \rfloor N}{q+1} + 1 - \lfloor \frac{q}{m} \rfloor s < A_R < N - \frac{m\lfloor \frac{q}{m} \rfloor N}{q+1} - 1 + \lfloor \frac{q}{m} \rfloor s$. In this case we cannot prove that this interval lies between 0 and N. If $\frac{m\lfloor \frac{q}{m} \rfloor N}{q+1} + 1 - \lfloor \frac{q}{m} \rfloor s \leq 0$ we also have $N - \frac{m\lfloor \frac{q}{m} \rfloor N}{q+1} - 1 + \lfloor \frac{q}{m} \rfloor s \geq N$, resulting in $p = 0$ if $s \geq \frac{mN}{q+1} + \frac{1}{\lfloor \frac{q}{m} \rfloor}$. If $s < \frac{mN}{q+1} + \frac{1}{\lfloor \frac{q}{m} \rfloor}$ there is a positive probability that job R can be scheduled. If $N - \frac{m\lfloor \frac{q}{m} \rfloor N}{q+1} - 1 + \lfloor \frac{q}{m} \rfloor s \leq \frac{m\lfloor \frac{q}{m} \rfloor N}{q+1} + 1 - \lfloor \frac{q}{m} \rfloor s$ then there is no interval in which job R cannot be scheduled resulting in $p = 1$, which is the case if $s \leq \frac{mN}{q+1} + \frac{1-\frac{1}{2}N}{\lfloor \frac{q}{m} \rfloor}$. If $s > \frac{mN}{q+1} + \frac{1-\frac{1}{2}N}{\lfloor \frac{q}{m} \rfloor}$ there is an interval of length $N - 2\left(\frac{m\lfloor \frac{q}{m} \rfloor N}{q+1} - \lfloor \frac{q}{m} \rfloor s + 1\right)$ in which job R cannot be scheduled on any machine. So the probability that job R can be scheduled is $p = 2\left(\frac{m\lfloor \frac{q}{m} \rfloor}{q+1} - \frac{\lfloor \frac{q}{m} \rfloor s-1}{N}\right)$. ■

Theorem 7.14 *If $\frac{mN}{q+1} < s \leq \frac{1}{q_v}\left(\frac{(q_v-1)mN}{q+1}+1\right)$ $\exists v$ then $p = 1$.*

Proof We prove that if $\frac{mN}{q+1} < s \leq \frac{1}{q_v}\left(\frac{(q_v-1)mN}{q+1}+1\right)$ we can always schedule job R on each machine v. Since in this case $\frac{mN}{q+1} < s$, each job needs to start later in its time window than the previous one. We schedule all jobs $1 \leq k \leq q_v$ at time $S_k = \frac{vN}{q+1} - \frac{1}{2} + (k-1)s$, so at the earliest possible time. Since in this way there is no idle time on the machine between jobs, we are sure that A_R is during a time when a job is processed on the machine. We assume that for some job $1 \leq k \leq q_v$ we have $S_k - \frac{1}{2} = \frac{vN}{q+1} - 1 + (k-1)s \leq A_R < \frac{vN}{q+1} - 1 + ks = S_k - \frac{1}{2} + s = S_{k+1} - \frac{1}{2}$. In this case we shift jobs $(k, ..., q_v)$ to the right as much as possible, so $S_k = \frac{(v+(q_v-1)m)N}{q+1} + \frac{1}{2} - (q_v - k)s$. Now we can prove that we can always schedule job R at time $S_R = S_{k-1} + s = \frac{vN}{q+1} - \frac{1}{2} + (k-1)s$.

First we need to prove that this time is always within the time window of job R. We have $A_R < \frac{vN}{q+1} - 1 + ks$, so $A_R - \frac{1}{2} < \frac{vN}{q+1} - \frac{3}{2} + ks < S_R$, since $s \leq 1$. Because $\frac{mN}{q+1} \leq 1$, we have $s \leq \frac{1}{q_v}\left(\frac{(q_v-1)mN}{q+1}+1\right) \leq 1$. This means the left boundary of the time window is satisfied. Since $\frac{vN}{q+1} - 1 + (k-1)s \leq A_R$ we have $S_R = \frac{vN}{q+1} - \frac{1}{2} + (k-1)s \leq A_R + \frac{1}{2}$, satisfying the right boundary of the time window. Second we have to prove that job $k-1$ will be finished before job R is scheduled to start and job R will be finished before job k is scheduled to start. Since $S_R = S_{k-1} + s$ job R will start when job $k-1$ is finished. Since $s \leq \frac{1}{q_v}\left(\frac{(q_v-1)mN}{q+1}+1\right)$ we have

$S_k = \frac{(v+(q_v-1)m)N}{q+1} + \frac{1}{2} - (q_v - k)s \geq \frac{vN}{q+1} - \frac{1}{2} + ks = S_R + s$, which means job R will be finished before the start of job k. ■

In Table 2 a summary of the conclusions is shown. For each set of conditions for s the probability that job R can be scheduled is given. For each combination of m, N and q only a selection of the theorems are applicable since never all of the cases occur.

In Figure 4 a flow chart shows in which situations which theorems can be used. For example, if $\frac{mN}{q+1} > 1$ and $\frac{N}{q+1} > 1$ we follow the corresponding path and see that we can use Theorems 7.7, 7.11(1), 7.11(2), 7.9(2), 7.13(3) and 7.13(1). Remember that the number in brackets indicates the case of the theorem. Which of these theorems we can use depends on the value of s. The theorems that result in $p = 1$ are surrounded by a green rectangle and the one resulting in $p = 0$ by a red one.

Thm	Conditions		Probability
7.7	$s \leq \min\{1, \frac{mN}{q+1}\}$		$p = 1$
7.9	$1 < s \leq \frac{mN}{q+1}$, $s \leq \frac{mN}{q+1} + 1 - \frac{1}{2}N$	$s > \frac{1}{2}\left(\frac{mN}{q+1} + 1\right)$,	$p = 1$
7.9	$1 < s \leq \frac{mN}{q+1}$, $s > \frac{mN}{q+1} + 1 - \frac{1}{2}N$	$s > \frac{1}{2}\left(\frac{mN}{q+1} + 1\right)$,	$p = 2\left(\frac{m}{q+1} - \frac{s-1}{N}\right)$
7.11	$1 < s \leq \frac{mN}{q+1}$, $s \leq \frac{1}{2}\frac{(m-1)N}{q+1} + 1$	$s \leq \frac{1}{2}\left(\frac{mN}{q+1} + 1\right)$,	$p = 1$
7.11	$1 < s \leq \frac{mN}{q+1}$, $s > \frac{1}{2}\frac{(m-1)N}{q+1} + 1$	$s \leq \frac{1}{2}\left(\frac{mN}{q+1} + 1\right)$,	$p = 1 - m\left(\lfloor\frac{q}{m}\rfloor - 1\right)\left(\frac{1-m}{q+1} + \frac{2(s-1)}{N}\right)$
7.13	$s > \frac{mN}{q+1}$, $s \geq \frac{mN}{q+1} + \frac{1}{\lfloor\frac{q}{m}\rfloor}$	$s > \frac{1}{q_v}\left(\frac{(q_v-1)mN}{q+1} + 1\right)$ $\forall v$,	$p = 0$
7.13	$s > \frac{mN}{q+1}$, $s \leq \frac{mN}{q+1} + \frac{1-\frac{1}{2}N}{\lfloor\frac{q}{m}\rfloor}$	$s > \frac{1}{q_v}\left(\frac{(q_v-1)mN}{q+1} + 1\right)$ $\forall v$,	$p = 1$
7.13	$s > \frac{mN}{q+1}$, $s > \frac{mN}{q+1} + \frac{1-\frac{1}{2}N}{\lfloor\frac{q}{m}\rfloor}$	$s > \frac{1}{q_v}\left(\frac{(q_v-1)mN}{q+1} + 1\right)$ $\forall v$, $s < \frac{mN}{q+1} + \frac{1}{\lfloor\frac{q}{m}\rfloor}$	$p = 2\left(\frac{m\lfloor\frac{q}{m}\rfloor}{q+1} - \frac{\lfloor\frac{q}{m}\rfloor s-1}{N}\right)$
7.14	$s > \frac{mN}{q+1}$,	$s \leq \frac{1}{q_v}\left(\frac{(q_v-1)mN}{q+1} + 1\right)$ $\exists v$	$p = 1$

Table 2: Probability that job R can be scheduled in different situations when we have multiple machines

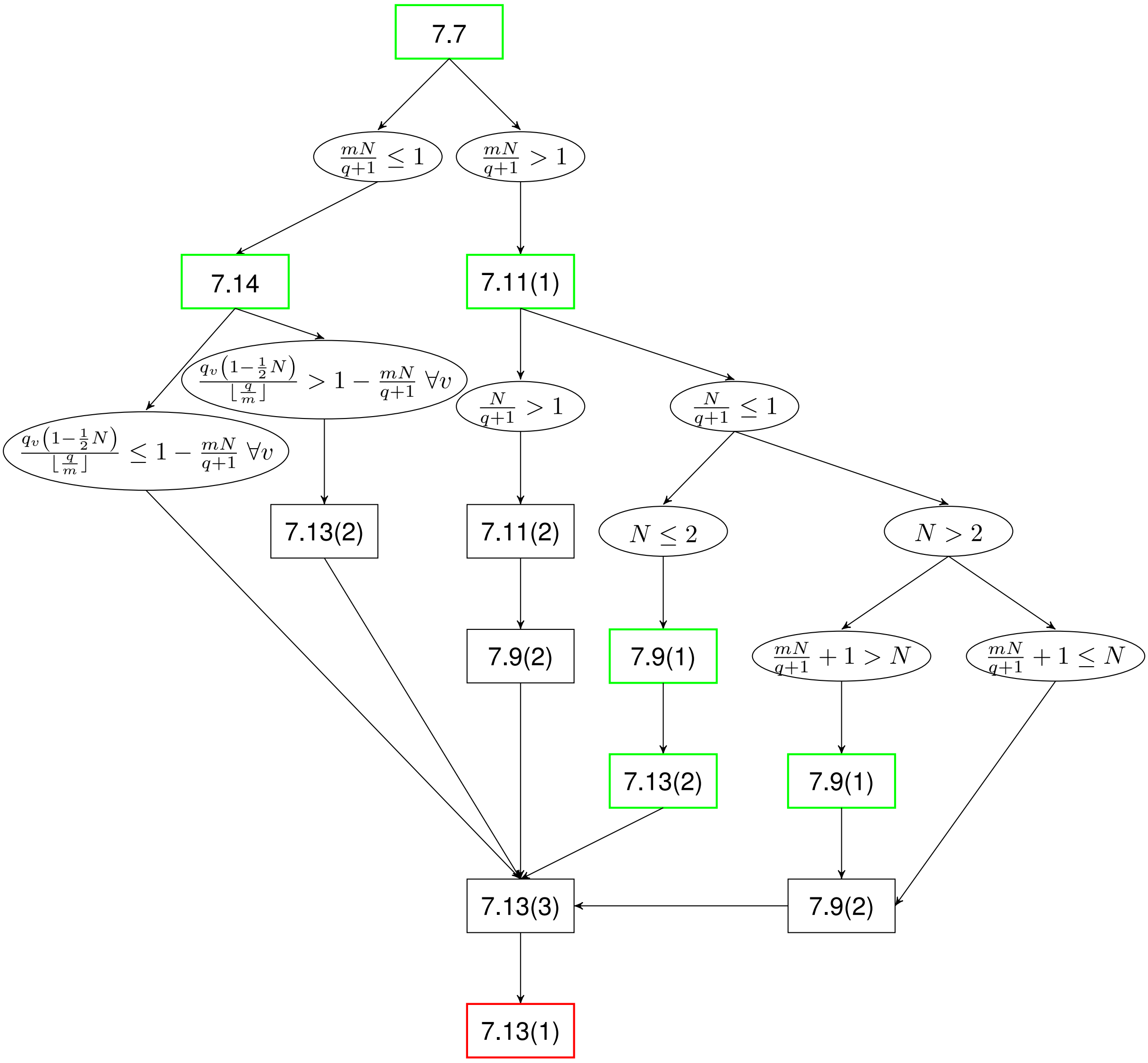

Figure 4: Flow chart of which theorems are applicable (green: $p = 1$, red: $p = 0$)

7.4.3 Multiple dynamic jobs

The case where multiple dynamic jobs are arriving is more difficult to examine analytically, since whether a dynamic job can be scheduled depends on where the previous dynamic jobs arrived and are scheduled. This is why we use simulation to examine the case where $r > 1$. For each set of values of N, q, m and r we simulate i times the r dynamic jobs, which are arriving uniformly on the interval $[0, N]$. The proportion of the i simulations in which all r jobs could be scheduled, is the probability that the r jobs can be scheduled.

7.4.4 Switching point

In the previous part we calculated the probability that the r jobs can be scheduled on m machines. For some values of N, q, s and r it might only be possible to schedule the q static jobs on all m machines. In this case $m - 1$ machines are not enough to schedule all jobs, so in this case we have no choice. In other cases it might be possible to schedule the jobs on $m - 1$ machines and leave one machine idle for the dynamic jobs. We want to determine the switching point from which service time or from which number of static jobs it is better to leave one machine idle. The switching point for the service time can be determined analytically when $r = 1$. When $r > 1$ we have to use simulation. The switching point for q is determined with simulation in both cases.

We start with only one dynamic job, so $r = 1$. If $p = 1$ this dynamic job can always be scheduled. In this case it is not useful to schedule the q other jobs on $m - 1$ machines. If $p < 1$ we can increase p by leaving one machine empty. This is only possible if it is possible to schedule all q static jobs on $m - 1$ machines. The values of s for which $p = 1$ can be found using Figure 4 in combination with Table 2. On the path corresponding to the chosen values of m, N and q in Figure 4 the green rectangles denote the theorems resulting in $p = 1$. The last theorem of this path can be looked up in Table 2 to find out for which values of s we have $p = 1$.

In Table 3 a summary of these switching points is given.

Conditions	Range of s for which $p = 1$
$\frac{mN}{q+1} > 1, \ \frac{N}{q+1} > 1$	$s \leq \frac{1}{2}\frac{(m-1)N}{q+1} + 1$
$\frac{mN}{q+1} > 1, \ \frac{N}{q+1} \leq 1, \ N \leq 2$	$s \leq \frac{mN}{q+1}$
$\frac{mN}{q+1} > 1, \ \frac{N}{q+1} \leq 1, \ N > 2, \ \frac{mN}{q+1} + 1 > N$	$s \leq \frac{mN}{q+1} + 1 - \frac{1}{2}N$
$\frac{mN}{q+1} > 1, \ \frac{N}{q+1} \leq 1, \ N > 2, \ \frac{mN}{q+1} + 1 \leq N$	$s \leq \frac{1}{2}\left(\frac{mN}{q+1} + 1\right)$
$\frac{mN}{q+1} \leq 1, \ \frac{q_v\left(1-\frac{1}{2}N\right)}{\lfloor\frac{q}{m}\rfloor} > 1 - \frac{mN}{q+1} \ \forall v$	$s \leq \frac{mN}{q+1} + \frac{1-\frac{1}{2}N}{\lfloor\frac{q}{m}\rfloor}$
$\frac{mN}{q+1} \leq 1, \ \frac{q_v\left(1-\frac{1}{2}N\right)}{\lfloor\frac{q}{m}\rfloor} \leq 1 - \frac{mN}{q+1} \ \forall v$	$s \leq \frac{1}{q_v}\left(\frac{(q_v-1)mN}{q+1} + 1\right) \exists v$

Table 3: Ranges of s for which $p = 1$ in different cases for $r = 1$

To determine for which values of s it is still possible to schedule the q static jobs on $m-1$ machines we derive the following theorem:

Theorem 7.15 *If $s \leq \frac{mN}{q+1} + \frac{1}{\lceil\frac{q}{m}\rceil-1}$ then we can schedule q jobs on m machines.*

Proof In order to determine for which value of s we can schedule all q jobs on m machines we have to take a machine with a maximum number of jobs scheduled on it. The maximum number of jobs scheduled on a machine is $\lceil\frac{q}{m}\rceil$. If these jobs can be scheduled, also the jobs on the machines with less jobs can be scheduled. In order to schedule all jobs, we shift all jobs as much as possible to the right. If the last job still starts in its time window, then all jobs can be scheduled. This is possible when $\frac{vN}{q+1} - \frac{1}{2} + (\lceil\frac{q}{m}\rceil - 1)s \leq \frac{(v+(\lceil\frac{q}{m}\rceil-1)m)N}{q+1} + \frac{1}{2}$, which is the case when $s \leq \frac{mN}{q+1} + \frac{1}{\lceil\frac{q}{m}\rceil-1}$. ■

With this theorem we know that if $s > \frac{(m-1)N}{q+1} + \frac{1}{\lceil\frac{q}{m-1}\rceil-1}$ it is not possible to schedule the q static jobs on $m-1$ machines, so we do not have the choice to do so. If $s > \frac{mN}{q+1} + \frac{1}{\lceil\frac{q}{m}\rceil-1}$ it is not even possible to schedule the jobs on m machines.

We define four ranges for the value of s. The first range, $s_{p=1}$, is the range for which the r dynamic jobs can be scheduled for sure, so in this case leaving one machine empty does not make a difference for p. The second range, s^h_{m-1} ,is the range for which $0.9 < p < 1$ and it is possible to schedule all q jobs on $m-1$ machines. In this case p could be increased by leaving a machine empty, but since p is already quite high it could also be better to divide the jobs over all m machines. The values of s^l_{m-1} are the values of s satisfying $0 \leq p \leq 0.9$ for which it is possible to schedule the q jobs on $m-1$ machines. In this case it is better to leave one machine empty. The last range we call s_m and is the range for which $p < 1$ but it is not possible to schedule the q jobs on $m-1$ machines. This range ends when it is not possible anymore to schedule all jobs on

m machines.

For multiple dynamic jobs we use simulation to determine the range of s for which $p = 1$. The ranges for which it is possible to schedule the q static jobs on $m - 1$ and on m machines can still be calculated analytically with Theorem 7.15. These values do not depend on the number of dynamic jobs. In a similar way we also determine the switching points for q.

8 Results

In this section we present the results following from the previous section.

8.1 Stochastic travel times

We first define the input used for the results. For three different values of the reliabilities α and β and three different values of the variance of the travel times the number of times a request arrives outside a time window or exceeds the maximum ride time is calculated. From this we can test whether more constraints are satisfied when we take the variance into account in our model.

Input
The data we use is an anonymized data set from a municipality in the Netherlands for school transport. It contains 133 requests, which are all walking persons, so no wheelchairs or rollators. The service time we assume for getting in or out of the vehicle is one minute for each person. The requests are all going from a home location to a school, so each request has a desired arrival time. The time window for arrival starts 15 minutes before this arrival time. For departure there is no time window, although a maximum ride time of 90 minutes has to be satisfied.

As objective function we take $c = 50ab + cr$, where ab is the number of buses used and cr is the travel time costs, which consists of the constant costs for each bus and the variable costs per kilometre.

Results
We want to test whether our model can satisfy the desired percentage α arriving on time and the desired percentage β travelling shorter than the maximum ride time. We test this for different variances of the travel time. For each variance Var_r we simulate three different situations. In the first situation we do not take the variance into account at all, so the variance we use, Var_u is equal to zero. In the second situation we make a planning with a smaller variance than the real variance and in the last situation we take into account the real variance, so $Var_u = Var_r$. For each pair of variances we output the number of buses needed and the travel time costs. Then we run the simulation for 10000 iterations and determine for each iterations whether each request arrives outside its time window or exceeds the maximum ride time. The results for α and β equal to 0.9, 0.925 and 0.95 are given in Table 4, Table 5 and Table 6. The information that is given in each column of these tables is:

$$
\begin{aligned}
Var_r &= \text{Real variance,} \\
Var_u &= \text{Variance used for the planning,} \\
o_1 &= \text{Number of buses needed,} \\
o_2 &= \text{Travel time costs,} \\
tw_1 &= \text{Number of requests of which at least one iteration is outside the time window,} \\
tw_2 &= \text{Number of times outside the time window over all iterations and all requests,} \\
tw_3 &= \text{Total number of requests with a confidence smaller than } \alpha \text{ for the time window,} \\
tw_4 &= \text{Average percentage outside the time window over all requests,} \\
mr_1 &= \text{Number of requests of which at least one iteration exceeds the maximum ride time,} \\
mr_2 &= \text{Number of times the maximum ride time is exceeded over all iterations and all requests,} \\
mr_3 &= \text{Total number of requests with a confidence smaller than } \beta \text{ for the maximum ride time,} \\
mr_4 &= \text{Average percentage exceeding the maximum ride time over all requests.}
\end{aligned}
$$

$\alpha = \beta = 0.9$

Var_r	Var_u	o_1	o_2	tw_1	tw_2	tw_3	tw_4	mr_1	mr_2	mr_3	mr_4
3	0	21	3197	84	96351	42	92.8%	7	8905	3	99,3%
3	1.5	21	3205	65	18530	0	98.6%	3	1093	1	99,9%
3	3	22	3349	65	8667	0	99.3%	2	800	1	99,9%
6	0	21	3197	106	141831	57	89.3%	10	11535	4	99,1%
6	3	22	3349	106	32269	23	97.6%	4	1912	1	99,9%
6	6	21	3236	97	14343	0	98.9%	4	1022	0	99,9%
9	0	21	3197	115	172823	66	87.0%	12	13413	4	99,0%
9	4.5	21	3225	125	40366	21	97.0%	7	1956	0	99,9%
9	9	23	3472	132	20241	0	98.5%	9	1202	0	99,9%

Table 4: Results for stochastic travel times, $\alpha = \beta = 0.9$

$\alpha = \beta = 0.925$

Var_r	Var_u	o_1	o_2	tw_1	tw_2	tw_3	tw_4	mr_1	mr_2	mr_3	mr_4
3	0	21	3197	85	96363	42	92.8%	7	8979	3	99.3%
3	1.5	21	3221	69	15473	0	98.8%	2	1907	2	99.9%
3	3	21	3247	66	6192	0	99.5%	2	78	0	100.0%
6	0	21	3197	108	142342	57	89.3%	8	11448	4	99.1%
6	3	21	3247	102	26870	20	98.0%	5	588	0	100.0%
6	6	22	3378	97	10700	0	99.2%	3	970	0	99.9%
9	0	21	3197	117	172339	66	87.0%	11	13400	4	99.0%
9	4.5	21	3236	126	33738	21	97.5%	10	1808	1	99.9%
9	9	28	4053	131	17804	0	98.7%	6	1168	1	99.9%

Table 5: Results for stochastic travel times, $\alpha = \beta = 0.925$

$\alpha = \beta = 0.95$

Var_r	Var_u	o_1	o_2	tw_1	tw_2	tw_3	tw_4	mr_1	mr_2	mr_3	mr_4
3	0	21	3197	87	96094	43	92.8%	7	8902	3	99.3%
3	1.5	21	3226	72	11552	0	99.1%	1	971	1	99.9%
3	3	22	3344	57	3685	0	99.7%	7	630	0	100.0%
6	0	21	3197	106	141500	65	89.4%	9	11455	4	99.1%
6	3	22	3344	97	19052	22	98.6%	7	2590	2	99.8%
6	6	25	3703	102	7391	0	99.4%	4	711	0	99.9%
9	0	21	3197	115	172523	66	87.0%	13	13298	4	99.0%
9	4.5	21	3269	122	25142	21	98.1%	12	2698	3	99.8%
9	9	31	4372	133	13613	0	99.0%	4	516	0	100.0%

Table 6: Results for stochastic travel times, $\alpha = \beta = 0.95$

For each value of α and β we see that in general a higher variance results in more buses needed and higher travel time costs, like we would expect. Only for $\alpha = \beta = 0.9$ and $Var_r = 6$ we see that the number of buses is lower for $Var_u = 6$ than for $Var_u = 3$. The reason for this could be that by coincidence a better optimization step is found in the first case than in the second case.

The number of requests of which at least one iteration is outside the time window is increasing when the real variance is increasing. This is an effect we would expect, since increasing the travel time variance increases the possibility that a request arrives outside its time window.

One might expect that increasing the variance taken into account for the planning would decrease the number of requests of which at least one iteration is outside the time window, but this is not always the case. The reason for this is that for each different variance used a different planning is made. Even if a higher variance is used to make the planning, the arrival times could still be such that more requests arrive outside the time window in at least one of the 10000 iterations. For $\alpha = \beta = 0.95$ and $Var_r = Var_u = 9$ even all requests arrive outside the time window at least one of the iterations. Still, both the total number of times a request is arriving outside the time window as well as the number of requests arriving more than α% of the times outside the time window, are decreasing when the variance used is decreasing.

We see that in all cases when $Var_r = Var_u$ the number of requests for which the confidence of α is not satisfied is zero. For a small variance of $Var_r = 3$ a variance used of $Var_u = 1.5$ is also enough to satisfy this condition. So we can conclude that the addition of stochastic travel times leads to more constraints satisfied in practice.

8.2 Simplified model for the combination of static and dynamic requests

In this subsection we give the results of the simplified model for the combination of static and dynamic requests. In this simplified version of the model we have only two locations and each request needs to be transported from location P to location D. We want our model to decrease the number of rejected dynamic requests. We first give the input used to test this, followed by the results for several arrival rates for the dynamic customers.

Input
The input parameters we use are:

$$\begin{aligned}
T &= 22, \\
h &= 10, \\
p &= 2, \\
x &= 3, \\
g &= 3, \\
B &= 3, \\
c &= [2,4,6], \\
n &= 39, \\
A^s &= [4,0,1,0,3,3,0,11,1,3,2,6,0,0,0,2,0,0,3,0,0,0], \\
q &= 0.
\end{aligned}$$

We have 22 time slots of length 10. It takes 2 time slots to drive from location P to location D and back. We have 39 static customers that need to be transported from location P to location D. If customer i arrives in time slot t it needs to be picked up in the time window $[t, t+3]$. Also the dynamic requests have to be picked up in 3 time slots after they arrive. We have 3 vehicles to transport the customers, with capacities 2, 4 and 6. The probability that a customer does not show is taken as zero.

Results
We use different arrival rates for the dynamic customers (χ_t), which results in a different number of expected rejected requests (R_T). We first use a different arrival rate in each time slot and then a homogeneous arrival rate. The results we get are:

$$\begin{aligned}
\chi &= [0.1, 0.1, 0.2, 0.2, 0.3, 0.3, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.01, 0.01, 0.01, 0.02, 0.02, 0.02, 0.02, 0, 0] \\
&\Rightarrow R_T = [5.3659, 3.7084, 0.8999, 0.8999, 0.8999, 0.8999], \\
\chi &= [0.1, 0.3, 0.1, 0, 0.1, 0.3, 0.1, 0, 0.1, 0.3, 0.1, 0, 0.1, 0.3, 0.1, 0, 0.1, 0.3, 0.1, 0.1, 0, 0] \\
&\Rightarrow R_T = [6.0860, 4.0120, 4.8916, 4.3663, 3.8901, 2.2349, 2.2562, -], \\
\chi_t &= 0.05\ \forall t \\
&\Rightarrow R_T = [1.8698, 1.8698, 1.8698, 1.8698], \\
\chi_t &= 0.1\ \forall t \\
&\Rightarrow R_T = [4.1592, 3.6390, -], \\
\chi_t &= 0.15\ \forall t \\
&\Rightarrow R_T = [6.6481, 4.5798, 2.4431, 2.5939, 4.4811, -], \\
\chi_t &= 0.2\ \forall t \\
&\Rightarrow R_T = [9.2927, 4.8880, -].
\end{aligned}$$

The first value of R_T is the number of rejected requests when the dynamic requests are not taken into account. We see that if the homogeneous arrival rate increases this value also increases since more dynamic requests are arriving and thus also rejected. In each following value of R_T space is reserved for the previously rejected requests. The runs stop when either three times in a row no improvement in the number of rejected requests could be found or when the space created for the dynamic requests is too large such that not all static requests could be planned anymore.

In the first case where the value of χ_t differs each time interval, the expected number of dynamic requests decreases in two runs to a low value of 0.8999, which cannot be improved anymore in the subsequent rounds. If $\chi_t = 0.05$ for all values of t we get a value of $R_T = 1.8698$ when we do not take the dynamic requests into account and we are not able to improve this value. For $\chi_t = 0.1\ \forall t$ or $\chi_t = 0.2\ \forall t$ we see a decrease of the number of rejected requests in two runs, but after that we create too much space such that not all requests can be planned anymore. For $\chi_t = 0.15$ we see something quite unexpected, first the value of R_T decreases and then it increases again. Apparently the space created for the dynamic requests is such that the static requests have to be planned at times at which there are even more dynamic requests arriving. If in a previous run there were not so many static requests planned at a time where there are many dynamic requests arriving, the dynamic requests were not rejected. If the static requests are moved to this time slot, it could result in a high value of the number of rejected requests. But even though our model does not seem to do the exact job here, it still is able to reduce the number of rejected requests from 6.6481 in the first run to 2.4431 in the third run. In the second non-homogeneous case we even see a fluctuation in the number of rejected requests.

8.3 Complete model for the combination of static and dynamic requests

We use the same input as for the stochastic travel times, so we have 133 static requests. We simulate 50, 60 and 70 dynamic requests, who are clustered if their locations are less than 10 minutes travelling from each other. For each number of dynamic requests we use two different data sets. The number of rejected dynamic requests is calculated when using both algorithms presented in Section 7.2 for different number of vehicles. The value of R_1 is the number of rejected requests when we reserve space for the dynamic requests and R_2 is the number of rejected requests when we do not take the dynamic requests into account while making the static schedule.

The results can be found in Table 7. In most cases the number of rejected requests is higher in the first model than in the second one. This is not what we would expect, since we reserve space for the dynamic requests in the first model. Only the data sets with 60 dynamic requests show cases for which the number of rejected requests is lower or equal in the first model; for 29 and 30 vehicles. An explanation for this is that for some ratios of the number of static requests, dynamic requests and vehicles, it may be better to reserve separate vehicles for the dynamic requests. This is what is basically done when the dynamic requests are not taken into account, so in the second model. The machine scheduling problem is used to analyse this switching point.

Data set	Number of dynamic requests	Number of vehicles	R_1	R_2
1.	50	28	11	7
	50	29	11	1
	50	30	6	1
2.	50	28	10	7
	50	29	7	5
	50	30	8	3
3.	60	28	10	9
	60	29	8	8
	60	30	6	8
4.	60	28	14	9
	60	29	3	5
	60	30	1	3
5.	70	28	33	15
	70	29	18	11
	70	30	24	9
6.	70	28	30	14
	70	29	25	7
	70	30	19	8

Table 7: Number of rejected requests in both models

8.4 Machine scheduling problem

In this section we give the results of the machine scheduling problem. First the probability p that the r jobs can be scheduled is calculated in a few examples, depending on the service time s. Next, some results on the switching points are presented.

One machine and one dynamic job

For the case where $m = 1$ and $r = 1$ we can calculate the probability p analytically using Table 1. We analyse the value of p for three different examples.

First we take $N = 100$ and $q = 50$, resulting in $\frac{N}{q+1} \approx 1.96$. If $s \leq 1$ we can use Theorem 7.1 so in this case $p = 1$. If $1 < s \leq 1.96$ we can either use Theorem 7.2 or Theorem 7.3. If $1 < s \leq 1.48$ then $p = 2 - s$ and if $1.48 < s \leq 1.96$ then $p = 0.059 - 0.020s$. If $s > 1.96$ we could either use Theorem 7.5 or Theorem 7.6. However, since $\frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right) \approx 1.94 < 1.96 < s$, we do not need Theorem 7.6. If $s > \frac{N}{q+1} + \frac{1}{q} \approx 1.98$ then $p = 0$ and if $1.96 < s \leq 1.98$ then $p = 1.98 - s$.

Next we take $N = 100$ and $q = 99$, such that $\frac{N}{q+1} = 1$. If $s \leq 1$ then $p = 1$. If $s > 1$ we can either use Theorem 7.5 or Theorem 7.6. Since $\frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right) = 1$ we only have to use Theorem 7.5. If $s > \frac{N}{q+1} + \frac{1}{q} \approx 1.01$ then $p = 0$ and if $1 < s \leq 1.01$ then $p = 2 - 1.98s$.

In the last example we take $N = 100$ and $q = 150$. In this case we have $\frac{N}{q+1} \approx 0.662 < 1$ so depending on the value of s we can use Theorem 7.1, Theorem 7.5 or Theorem 7.6. If $s \leq 0.662$ then $p = 1$. If $s > 0.662$ we can use Theorem 7.5 if $s > \frac{1}{q}\left(\frac{(q-1)N}{q+1} + 1\right) \approx 0.665$ and Theorem 7.6 if $s \leq 0.665$. This results in a value of $p = 0$ if $s > \frac{N}{q+1} + \frac{1}{q} \approx 0.669$, $p = 2.01 - 3s$ if $0.665 < s \leq 0.669$ and $p = 1$ if $0.662 < s \leq 0.665$.

An overview of the probabilities that job R can be scheduled in these three examples is given in Table 8. In the second and third example p is always approximately equal to zero or one. We see that if the number of static requests, q, increases, p decreases for the same value of s.

Example	Value of s	Probability	Probability range
1. $N = 100$, $q = 50$	$s \leq 1$	$p = 1$	$p = 1$
	$1 < s \leq 1.48$	$p = 2 - s$	$0.52 \leq p < 1$
	$1.48 < s \leq 1.96$	$p = 0.059 - 0.020s$	$0.020 \leq p < 0.030$
	$1.96 < s \leq 1.98$	$p = 1.98 - s$	$0 \leq p < 0.020$
	$s > 1.98$	$p = 0$	$p = 0$
2. $N = 100$, $q = 99$	$s \leq 1$	$p = 1$	$p = 1$
	$1 < s \leq 1.01$	$p = 2 - 1.98s$	$0.00020 \leq p < 0.020$
	$s > 1.01$	$p = 0$	$p = 0$
3. $N = 100$, $q = 150$	$s \leq 0.665$	$p = 1$	$p = 1$
	$0.665 < s \leq 0.669$	$p = 2.01 - 3s$	$0 \leq p < 0.012$
	$s > 0.669$	$p = 0$	$p = 0$

Table 8: Probability that job R can be scheduled in different examples for different values of s for one machine

Multiple machines and one dynamic job

For the case where $m > 1$ and $r = 1$ we can calculate the probability p analytically using Table 2 and the flow chart of Figure 4. For different values of N, q and m the probability that job R can be scheduled is given in Table 9 for different ranges of s. Also the theorem used is given, where the number in brackets indicates which case it is of the theorem.

In the first four cases we see that if q decreases, p increases for the same value of s. The last three cases are special cases where we can use different theorems.

Example	Value of s	Theorem	Probability	Probability range
1. $N = 100$	$s \leq 1.50$	7.7 or 7.11(1)	$p = 1$	$p = 1$
$q = 250$	$1.50 < s \leq 1.99$	7.9(2)	$p = \frac{10}{251} + \frac{1}{50} - \frac{1}{50}s$	$0.0050 \leq p < 0.0075$
$m = 5$	$1.99 < s \leq 2.01$	7.13(3)	$p = \frac{500}{251} + \frac{1}{50} - s$	$0 \leq p < 0.0050$
	$s > 2.01$	7.13(1)	$p = 0$	$p = 0$
2. $N = 100$	$s \leq 2.98$	7.7 or 7.11(1)	$p = 1$	$p = 1$
$q = 100$	$2.98 < s \leq 4.95$	7.9(2)	$p = \frac{10}{101} + \frac{1}{50} - \frac{1}{50}s$	$0.020 \leq p < 0.060$
$m = 5$	$4.95 < s \leq 5.00$	7.13(3)	$p = \frac{200}{101} + \frac{1}{50} - \frac{2}{5}s$	$0 \leq p < 0.010$
	$s > 5.00$	7.13(1)	$p = 0$	$p = 0$
3. $N = 100$	$s \leq 4.92$	7.7 or 7.11(1)	$p = 1$	$p = 1$
$q = 50$	$4.92 < s \leq 5.40$	7.11(2)	$p = \frac{180}{51} + \frac{19}{10} - \frac{9}{10}s$	$0.57 \leq p < 1$
$m = 5$	$5.40 < s \leq 9.80$	7.9(2)	$p = \frac{10}{51} + \frac{1}{50} - \frac{1}{50}s$	$0.020 \leq p < 0.11$
	$9.80 < s \leq 9.90$	7.13(3)	$p = \frac{100}{51} + \frac{1}{50} - \frac{1}{5}s$	$0 \leq p < 0.020$
	$s > 9.90$	7.13(1)	$p = 0$	$p = 0$
4. $N = 100$	$s \leq 8.69$	7.7 or 7.11(1)	$p = 1$	$p = 1$
$q = 25$	$8.69 < s \leq 10.12$	7.11(2)	$p = \frac{80}{26} + \frac{7}{5} - \frac{2}{5}s$	$0.43 \leq p < 1$
$m = 5$	$10.12 < s \leq 19.23$	7.9(2)	$p = \frac{10}{26} + \frac{1}{50} - \frac{1}{50}s$	$0.020 \leq p < 0.20$
	$19.23 < s \leq 19.43$	7.13(3)	$p = \frac{50}{26} + \frac{1}{50} - \frac{1}{10}s$	$0 \leq p < 0.020$
	$s > 19.43$	7.13(1)	$p = 0$	$p = 0$
5. $N = 100$	$s \leq 50.50$	7.7, 7.11(1) or 7.9(1)	$p = 1$	$p = 1$
$q = 200$	$50.50 < s \leq 99.50$	7.9(2)	$p = \frac{400}{201} + \frac{1}{50} - \frac{1}{50}s$	$0.020 \leq p < 1$
$m = 200$	$99.50 < s \leq 100.50$	7.13(3)	$p = \frac{400}{201} + \frac{1}{50} - \frac{1}{50}s$	$0 \leq p < 0.020$
	$s > 100.50$	7.13(1)	$p = 0$	$p = 0$
6. $N = 1$	$s \leq 0.73$	7.7 or 7.14	$p = 1$	$p = 1$
$q = 10$	$0.73 < s \leq 0.95$	7.13(3)	$p = \frac{20}{11} + 2 - 4s$	$0 \leq p < 0.91$
$m = 5$	$s > 0.95$	7.13(1)	$p = 0$	$p = 0$
7. $N = 1$	$s \leq 1.33$	7.7, 7.14 or 7.13(2)	$p = 1$	$p = 1$
$q = 5$	$1.33 < s \leq 1.83$	7.13(3)	$p = \frac{5}{3} + 2 - 2s$	$0 \leq p < 1$
$m = 5$	$s > 1.83$	7.13(1)	$p = 0$	$p = 0$

Table 9: Probability that job R can be scheduled in different examples for different values of s

Multiple dynamic jobs

The case where we have multiple dynamic jobs we analyse by using simulation. We simulate the dynamic jobs 10,000 times. To check the simulation we first simulate the examples for one dynamic job. In Table 10 the maximum difference between the analytical calculation of p and the simulation is given for each example. By maximum difference we mean the maximum value of the differences for all values of s. The total maximum difference is equal to 0.035, so our simulation seems to work correctly.

We now move on to multiple dynamic jobs. In Table 11 the probability that all r dynamic jobs can be scheduled is given in different situations. One can see that if r increases, the probability that these r jobs can all be scheduled decreases. If the number of machines increases, the probability also increases. Both these results are logical.

m	N	q	Maximum difference of p
1	100	50	0.0045
1	100	99	0
1	100	150	0
5	100	250	0.0029
5	100	100	0.0040
5	100	50	0.011
5	100	25	0.035
200	100	200	0.0084
5	1	10	0.011
5	1	5	0.010

Table 10: Maximum difference between the analytical calculation and the simulation

m	N	q	s	r	p
1	100	50	1	1	1
1	100	50	1	5	0.78
1	100	50	1	10	0.30
1	100	50	1	15	0.05
1	100	50	1	20	0.01
1	100	150	0.5	1	1
1	100	150	0.5	5	0.99
1	100	150	0.5	10	0.90
1	100	150	0.5	15	0.68
1	100	150	0.5	20	0.39
5	100	250	1.4	1	1
5	100	250	1.4	5	0.999
5	100	250	1.4	10	0.98
5	100	250	1.4	15	0.91
5	100	250	1.4	20	0.74
5	100	25	5	1	1
5	100	25	5	5	0.998
5	100	25	5	10	0.97
5	100	25	5	15	0.83
5	100	25	5	20	0.58

Table 11: Probability that the r dynamic jobs can be scheduled in different situations

Switching point

For $r = 1$ the switching point for s can be calculated analytically. Remember that we defined four ranges for the value of s. The first range, $s_{p=1}$, is the range for which the r dynamic jobs can be scheduled for sure, so in this case leaving one machine empty does not make a difference for p. The second range, s^h_{m-1} ,is the range for which $0.9 < p < 1$ and it is possible to schedule all q jobs on $m-1$ machines. In this case p could be increased by leaving a machine empty, but since p is already quite high it could also be better to divide the jobs over all m machines. The values of s^l_{m-1} are the values of s satisfying $0 \leq p \leq 0.9$ for which it is possible to schedule the q jobs on $m-1$ machines. In this case it is better to leave one machine empty. The last range we call s_m and is the range for which $p < 1$ but it is not possible to schedule the q jobs on $m-1$ machines. This range ends when it is not possible anymore to schedule all jobs on m machines.

We now determine these ranges for the examples of Table 9. The results can be found in Table 12. Note that if $q = m$ the q jobs can always be scheduled on m machines since each machine only serves one job, so we have no upper boundary for s.

m	N	q	r	$s_{p=1}$	s^h_{m-1}	s^l_{m-1}	s_m
5	100	250	1	$0 < s \leq 1.5$	-	$1.5 < s \leq 1.6$	$1.6 < s \leq 2$
5	100	100	1	$0 < s \leq 3.0$	-	$3.0 < s \leq 4.0$	$4.0 < s \leq 5.0$
5	100	50	1	$0 < s \leq 4.9$	$4.9 < s \leq 5$	$5 < s \leq 7.9$	$7.9 < s \leq 9.9$
5	100	25	1	$0 < s \leq 8.7$	$8.7 < s \leq 8.9$	$8.9 < s \leq 15.5$	$15.5 < s \leq 19.4$
200	100	200	1	$0 < s \leq 50.5$	$50.5 < s \leq 55.4$	$50.5 < s \leq 100.0$	$100.0 < s$
5	1	10	1	$0 < s \leq 0.73$	-	$0.73 < s \leq 0.80$	$0.80 < s \leq 1.4$
5	1	5	1	$0 < s \leq 1.3$	-	$1.3 < s \leq 1.6$	$1.6 < s$

Table 12: Different ranges for the values of s resulting in a different scheduling choice, $r = 1$

If there are multiple dynamic jobs, we use simulation. For the examples of Table 9 for $r = 1$ we get indeed the same results as given in Table 12. A few examples for $r > 1$ are given in Table 13.

m	N	q	r	$s_{p=1}$	s^h_{m-1}	s^l_{m-1}	s_m
5	100	50	5	$0 < s \leq 2.7$	$2.7 < s \leq 4.4$	$4.4 < s \leq 7.9$	$7.9 < s \leq 9.9$
10	100	50	5	$0 < s \leq 7.4$	$7.4 < s \leq 9.3$	$9.3 < s \leq 17.8$	$17.8 < s \leq 19.8$
5	100	50	20	$0 < s \leq 1.6$	$1.6 < s \leq 2.9$	$2.9 < s \leq 7.9$	$7.9 < s \leq 9.9$
10	100	50	20	$0 < s \leq 4.5$	$4.5 < s \leq 6.7$	$6.7 < s \leq 17.8$	$17.8 < s \leq 19.8$

Table 13: Different ranges for the values of s resulting in a different scheduling choice, $r > 1$

Results of a few examples for the ranges of q for which we have different optimal scheduling choices can be found in Table 14.

m	N	s	r	$q_{p=1}$	q^h_{m-1}	q^l_{m-1}	q_m
5	100	5	1	$1 \leq q \leq 49$	$q = 50$	$51 \leq q \leq 79$	$80 \leq q \leq 100$
10	100	5	1	$1 \leq q \leq 110$	-	$111 \leq q \leq 180$	$181 \leq q \leq 201$
5	100	5	5	$1 \leq q \leq 14$	$15 \leq q \leq 42$	$43 \leq q \leq 79$	$80 \leq q \leq 100$
10	100	5	5	$1 \leq q \leq 85$	$86 \leq q \leq 104$	$105 \leq q \leq 180$	$181 \leq q \leq 201$
5	100	5	20	$1 \leq q \leq 4$	$5 \leq q \leq 12$	$13 \leq q \leq 79$	$80 \leq q \leq 100$
10	100	5	20	$1 \leq q \leq 37$	$38 \leq q \leq 86$	$87 \leq q \leq 180$	$181 \leq q \leq 201$

Table 14: Different ranges for the values of q resulting in a different scheduling choice

9 Conclusion

In this report stochastics were added to the dial-a-ride problem in two different ways, by considering stochastic travel times and by taking into account the dynamic requests while making the static schedule. In this section the conclusions of this research are summarized.

By using stochastic travel times the time window constraints as well as the maximum ride time constraints are made stochastic. We started from the assumption that the travel times are normally distributed. In this way we can use an approximation of the normal distribution to make these constraints deterministic again. The results show that using these adapted constraints indeed makes sure the desired reliabilities for the time window and the maximum ride time constraints are satisfied, assuming the variance we use is a good approximation for the variance in practice.

To minimize the number of dynamic requests that are rejected because of the static schedule that was already created, we started analysing a simplified version of the problem with only two locations. By defining a Markov reward process the number of rejected requests in each time slot is calculated analytically. Space is then reserved in the time slots where most requests are rejected. In the results we see that this approach can indeed result in a reduction of the number of rejected requests. Sometimes the number of rejected requests also increases again after a few runs. In these cases too much space is created in certain time slots, pushing the static requests to time slots in which dynamic requests are arriving as well. But even in these cases the number of rejected requests diminished in the first few runs.

In the complete model of the dynamic requests, the requests with locations close to each other and similar time windows are clustered. These clustered requests are seen as dummy requests and in combination with the static requests a planning is made. The dummy requests are removed from the planning again, leaving space for the simulated dynamic requests. The results establish that this method often leads to a less optimal schedule than if we do not take the dynamic requests into account at all. An explanation for this is that for some ratios of the number of static requests, dynamic requests and vehicles, it may be better to reserve separate vehicles for the dynamic requests. This is what is basically done when the dynamic requests are not taken into account.

The stochastic travel times could also be combined with the model that takes the dynamic requests into account. However, since the model taking into account the dynamic requests does not decrease the number of dynamic requests in most of the cases, this is not really interesting.

To examine in which cases it is better to schedule the dynamic requests in the same vehicles as the static requests and when it is better to reserve one or more vehicles for the dynamic requests, we used a machine scheduling model. The probability that dynamic jobs can be added to machines where static jobs are already scheduled is calculated analytically in the case of one dynamic job and numerically in the case of more dynamic jobs. Based on this probability the switching points are determined, i.e. for which values of the service time or the number of static requests it is better to leave one machine idle and when it is better to divide all static jobs over all machines.

10 Discussion

In this section some of the assumptions made in this research are discussed, as well as some potential future work.

The first important assumption made in this report is that the travel times are normally distributed. In reality the distribution of the travel times can be different for each route. For example, if we have a route including a bridge that has to open once in a while to let boats through, the travel time roughly differs between two possible values; the travel time when the bridge is open and the travel time when the bridge is closed. However, the main fluctuations in travel times are caused by traffic, in which case a normal distribution seems more realistic. We also assumed the same variance of the travel times between each two locations. It is obvious that in reality this is rarely the case. In urban areas the travel times vary more than in rural areas and also long distances have a larger variance than short ones. These differences of variances are not included in our model yet, but it would not be complicated to add this to the model.

The travel times used in our model are not time-dependent. We could add time-dependency to the model by making the mean and variance of the travel times dependent on the time. Also stochastic service times could be added, but like said before, the variance of these services times can also be included in the variance of the travel times.

Another assumption that was made is that the variance of the travel time and the arrival process of the dynamic requests are known in advance. In reality of course we do not know these exactly. However, from historical data we could draw conclusions about the variance of the travel times and the dynamic arrival process. We did not do this in this research, but this would be necessary if the model is to be applied in practice, so this would be important for further research.

For the machine scheduling problem we analysed the switching point between when it is better to reserve one or more vehicles for the dynamic requests and when it is not. The main question for further research is whether a similar switching point can be determined for the vehicle planning problem. This will be more complicated than for the machine scheduling problem, since there are more factors involved that could influence this switching point. In the machine scheduling problem we assumed that each job has the same service time and the jobs are homogeneously arriving on the interval. In the vehicle planning problem the travel times differ and the jobs are not arriving homogeneously. In this problem the switching point also depends on the travel times and the arrival processes of the static as well as the dynamic requests. This would be an interesting subject for further research.

11 References

MuConsult. Krachten bundelen voor toekomstvast doelgroepenvervoer en OV, eindrapportage, 2013. URL http://www.rijksoverheid.nl/documenten-en-publicaties/rapporten/2013/06/28/krachten-bundelen-voor-toekomstvast-doelgroepenvervoer-en-ov.html. In collaboration with Significant and Alares on behalf of Ministerie VWS.

CROW-KpVV. Handreiking voor het organiseren van samenwerking in het doelgroepenvervoer en aanvullend OV, 2014. URL http://www.muconsult.nl/images/Downloads/CROW_Handreiking-organiseren-doelgroepenvervoer.pdf. In collaboration with MuConsult.

SEO-Economisch-Onderzoek. Contractvervoer op de rit, consequenties van marktwerking in het contractvervoer, 2011. URL http://www.seo.nl/uploads/media/2011-39_Contractvervoer_op_de_rit_01.pdf. On behalf of KNV taxi.

CROW-KpVV. De regiecentrale: wat is het en hoe werkt het?, 2013. URL http://beta.forseti.nl/dynamic/media/24/documents/KpVV-brochure-Regiecentrale-webpdf.pdf. In collaboration with Forseti and Mobycon.

N.H.M. Wilson, J.M. Sussman, H.K. Wong, and B.T. Higonnet. *Scheduling algorithms for a dial-a-ride system*. Massachusetts Institute of Technology, Cambridge, 1971.

J.-F. Cordeau and G. Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579–594, 2003a.

E. Melachrinoudis, A.B. Ilhan, and H. Min. A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research*, 34(3):742–759, 2007.

J.-J. Jaw, A.R. Odoni, H.N. Psaraftis, and N.H.M. Wilson. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B: Methodological*, 20(3):243–257, 1986.

M. Diana and M.M. Dessouky. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B: Methodological*, 38(6):539–557, 2004.

Y. Luo and P. Schonfeld. A rejected-reinsertion heuristic for the static dial-a-ride problem. *Transportation Research Part B: Methodological*, 41(7):736–755, 2007.

P. Healy and R. Moll. A new extension of local search applied to the dial-a-ride problem. *European Journal of Operational Research*, 83(1):83–104, 1995.

B. Rekiek, A. Delchambre, and H.A. Saleh. Handicapped person transportation: An application of the grouping genetic algorithm. *Engineering Applications of Artificial Intelligence*, 19(5):511–520, 2006.

D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & operations research*, 34(8):2403–2435, 2007.

Z. Xiang, C. Chu, and H. Chen. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European Journal of Operational Research*, 174(2):1117–1139, 2006.

S.N. Parragh, J.-F. Cordeau, K.F. Doerner, and R.F. Hartl. Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR spectrum*, 34(3):593–633, 2012.

G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte. Static pickup and delivery problems: a classification scheme and survey. *Top*, 15(1):1–31, 2007.

A. Attanasio, J.-F. Cordeau, G. Ghiani, and G. Laporte. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, 30(3):377–387, 2004.

O.B.G. Madsen, H.F. Ravn, and J.M. Rygaard. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of operations Research*, 60(1):193–208, 1995.

L. Coslovich, R. Pesenti, and W. Ukovich. A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. *European Journal of Operational Research*, 175(3): 1605–1615, 2006.

D. Teodorovic and G. Radivojevic. A fuzzy logic approach to dynamic dial-a-ride problem. *Fuzzy sets and systems*, 116(1):23–33, 2000.

G. Berbeglia, J.-F. Cordeau, and G. Laporte. Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202(1):8–15, 2010.

V. Pillac, M. Gendreau, C. Guéret, and A.L Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.

M.E.T. Horn. Fleet scheduling and dispatching for demand-responsive passenger services. *Transportation Research Part C: Emerging Technologies*, 10(1):35–63, 2002.

J.-F. Cordeau and G. Laporte. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153(1):29–46, 2007.

J.-F. Cordeau and G. Laporte. The dial-a-ride problem (darp): Variants, modeling issues and algorithms. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(2):89–101, 2003b.

S. Ichoua, M. Gendreau, and J.-Y. Potvin. Vehicle dispatching with time-dependent travel times. *European journal of operational research*, 144(2):379–396, 2003.

E. Taniguchi and H. Shimamoto. Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times. *Transportation Research Part C: Emerging Technologies*, 12(3):235–250, 2004.

A. Haghani and S. Jung. A dynamic vehicle routing problem with time-dependent travel times. *Computers & operations research*, 32(11):2959–2986, 2005.

A.L. Kok, E.W. Hans, J.M.J. Schutten, and W.H.M. Zijm. A dynamic programming heuristic for vehicle routing with time-dependent travel times and required breaks. *Flexible services and manufacturing journal*, 22(1-2):83–108, 2010.

X. Li, P. Tian, and S.C.H. Leung. Vehicle routing problems with time windows and stochastic travel and service times: models and algorithm. *International Journal of Production Economics*, 125(1):137–145, 2010.

L. Fu. Scheduling dial-a-ride paratransit under time-varying, stochastic congestion. *Transportation Research Part B: Methodological*, 36(6):485–506, 2002.

M. Schilde, K.F. Doerner, and R.F. Hartl. Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem. *European Journal of Operational Research*, 238(1):18–30, 2014.

Z. Xiang, C. Chu, and H. Chen. The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. *European Journal of Operational Research*, 185(2): 534–551, 2008.

M. Schilde, K.F. Doerner, and R.F. Hartl. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers & operations research*, 38(12):1719–1730, 2011.

J. Span, H. Boelema, J. de Cloet, L. van der Kaap, and S. Klootwijk. Optimalisatie van doelgroepenvervoer, 2013.

Z.A. Karian and E.J. Dudewicz. Modern statistical. *Systems, and GPSS Simulation-The First Course*, 1991.

N. Kortbeek, M.E. Zonderland, R.J. Boucherie, N. Litvak, and E.W. Hans. Designing cyclic appointment schedules for outpatient clinics with scheduled and unscheduled patient arrivals. 2011. Memorandum 1968.