FEASIBILITY CHECKS FOR THE OPTIMIZATION OF THE MATCHING IN DIAL-A-RIDE PROBLEM WITH NO PRE-BOOKING

Urban Transport Systems Laboratory

Bachelor project

Professor
Geroliminis Nikolaos

Assistant
Fayed Lynn

Students
Ballinari Lorenzo
Ribeiro de Carvalho Paulo

July 9, 2022

Contents

1 Introduction

In urban contexts, taxi is a really important type of transportation as it offers excellent services in terms of shorter waiting times and better passenger travel convenience. However, there are a few critical aspects about this service that need to be addressed and improved. The higher demand during peak hours is one of them and increasing the number of taxis seems the obvious solution to this problem, but it causes other negative effects, as an increase in accumulation (congestion) and consumption of energy. Employing the shared-ride concept in this context has surely a great potential to maximize its efficiency. Developing advanced shared-taxi-dispatch algorithms creates substantial benefits, reducing energy consumption and increasing incomes by creating a more efficient taxi network.

Then the objective of this bachelor project is to propose a functional algorithm that creates sharing trips allowing to have up to two passengers in one vehicle at the same time. Two models are given in this work, so their results and performances are compared to better understand which one is the more efficient. The performances of the algorithms are generated through a simulation that has been voluntarily made pseudo-random in order to repeat the same simulation for different values of each constraint (maximum waiting and detour time, percentage of passengers willing to share, ...) and see the evolution.

To better understand the main concepts and challenges beyond the shared-taxi-dispatch algorithms, we started by reading a few reference articles, which allowed us to start thinking where to begin the implementation of the algorithm [1] [3] [4] [5].

2 Algorithms

Over this semester we implemented two algorithms that are presented in this section. It has been chosen to do this in order to quantify the difference in results between the two. The first model does the sharing matching after the solo matching (sequential). The second model does the same matching, but both at the same time (simultaneous). The goal is to know if what we gain in terms of network optimization is drastically different. Our study focuses on sharing with a capacity of two passengers for every vehicle in the network. In this case, the first step to create the algorithms is to compose all the possible scenarios that exist between two passengers who want to share a trip. The figures below try to summarize them in a clear way.

The first group of scenarios is done in the following environment. A vehicle has been assigned to a passenger (1), but he is still on his way to pick him up. A new request (2) is then looking for a vehicle. In this environment, we must consider four possible scenarios displayed in *Figure 1* below.

Figure 1: First group of scenarios

The second group of scenarios happens in a different environment. The vehicle has already picked up its passenger (1) and is therefore heading towards its destination. A new request (2) is then looking for a vehicle. In this new environment, both passengers are subjected to the scenarios in Figure 2 below.

Figure 2: Second group of scenarios

Our algorithms have two objectives. The first one is to return for each scenario if it is feasible. Indeed, if a scenario proposes a waiting or detour time considered as unacceptable for one of the two passengers, then we want this scenario to be refuted. The second is to know the cost of each of these scenarios for the network. This is still to be defined and depends on which point of view we want to optimize. In this work, it is chosen to take the community and the passengers point of view by optimizing the cost in terms of waiting or detour time. However, it is possible to take the position of the platform that would propose its services and optimize according to the incomes or even the position of a state that would like to reduce CO_2 emissions by optimizing them.

2.1 First model : Sequential

The first model simply proposes to make a first solo matching (one vehicle with one request) and then takes the unassigned requests, that are ready to share a trip, and try to match them to the already assigned drivers where their passengers are ready to share too.

A superficial explanation through simple pseudo-code cases is proposed in the following sections. They allow to quickly understand the main ideas of our algorithm. As the code is long, it would not be appropriate to display it in this report.

2.1.1 Main code

The main code is where all the next functions are called to reach the objective of matching the requests with a vehicle in the most effective way.

Algorithm 1 Main Code (sequential method)

```
1: Discretize the simulation duration in time steps.
2: Initiate lists & variables that will contain performance metrics of the
3: current algorithm.
4:
5: for time = 1 · steps, 2 · steps, ... do
6:     Update the network, arrivals, abandonments, the number of private
7:     vehicles and some performances metrics.
8:
9:     if time is a multiple of the batch size then
10:        MatchSolo(requests, vehicles)
11:        Create a list with Unassigned Requests that are ready to share.
12:        Create a list with Feasible Vehicles for sharing.
13:        Initialize the cost matrix.
14:
15:        for each request in Unassigned Requests do
16:            for each vehicle in Feasible Vehicles do
17:                Initialize a list that will stock the cost & its feasibility.
18:
19:                for scenario = 1, 2, ..., 6 do
20:                    Check waiting time constraint.
21:                    Check detour time constraint. (if l.20 → True)
22:                    Stock costs & feasibility in their respective metrics list.
23:                end for
24:
25:                Update matrix with the lowest chosen cost (obj.
26:                function can be waiting or detour time) and stock which
27:                scenario gives this lowest cost.
28:            end for
29:
30:        end for
31:
32:        Match with sharing(CostMatrix)
33:    end if
34:
35:    Update the last performance metrics.
36: end for
```

Figure 3: Pseudo code for the main separate code

2.1.2 Check waiting & detour time constraint

These two functions are used in the algorithm to define which scenario is feasible and what is the cost of it. We are not yet looking to keep the one with the lowest cost. We only do an analysis of all the scenarios and store the results.

Algorithm 2 Check waiting time constraint

```
# First condition for scenarios
if vehicle is on its way to pick-up its request & occupancy=0 then
    if already assigned request allows sharing then
        Compute the current waiting time of both passengers.
        Compute the estimate waiting time depending on the scenario.
        Return the waiting time and if it is feasible.
    end if
end if

# Second condition for scenarios
if vehicle is on its way to drop-off its request & occupancy=1 then
    if already assigned request allows sharing then
        Compute the current waiting time of both passengers.
        Compute the estimate waiting time depending on the scenario.
        Return the waiting time and if it is feasible.
    end if
end if

Return an aberrant waiting time and a False for the check
```

Figure 4: Pseudo code for the check of the waiting time constraint

Algorithm 3 Check detour time constraint

```
if waiting time check is True then

    # First condition for scenarios
    if vehicle is on its way to pick-up its request & occupancy=0 then
        if already assigned request allows sharing then
            Compute the estimate detour time depending on the scenario.
            Return the detour time and if it is feasible.
        end if
    end if

    # Second condition for scenarios
    if vehicle is on its way to drop-off its request & occupancy=1 then
        if already assigned request allows sharing then
            Compute the already traveled time for request in the vehicle.
            Compute the estimate detour time depending on the scenario.
            Return the detour time and if it is feasible.
        end if
    end if
end if

Return an aberrant detour time and a False for the check
```

Figure 5: Pseudo code for the check of the detour time constraint

2.1.3 Update cost matrix

It is at this step that we take the results obtained by the two previous functions and choose the optimal scenario (if it exists, because it is very likely that for some pair of requests, no scenario satisfies the time constraints). We finish by updating the cost matrix with the optimal scenario.

Algorithm 7 Update matrix

```
Determine position to be update in cost matrix

for each cost stock in metrics do
    Determine the smallest cost and its associated scenario.
end for

Update the cost matrix with the smallest cost and stock its associated
scenario.

Return cost matrix & scenario matching
```

Figure 6: Pseudo code for updating the cost matrix

2.1.4 Matching with sharing

Once all possible pairs between requests have been analyzed, our cost matrix is complete. Now we have to associate the request in order to minimize (or maximize in some cases) the global cost of the network. For this, we use the Bipartite Method which returns the peers that globally have the lowest cost. Finally, we just have to update the different instances of the vehicles and requests.

Algorithm 6 Match with sharing

```
Use Bipartite method for creating the pairs of request & vehicle given
the lowest global cost.

if there is any sharing trips to create then
    Stop
end if

if there is sharing trips to create then
    Update all necessary instances
    Add the sharing trip in the history variable
end if

Return history of sharing
```

Figure 7: Pseudo code for the matching with sharing

2.1.5 Calculate waiting & detour time performance

These last two functions have no purpose in the matching. They have been implemented in order to measure the performance of our algorithms when changing parameters such as the maximum waiting or detour time or the rate of willingness to share.

Algorithm 4 Calculate waiting time performance

```
for every request that are traveling do
    if request in already analyze then
        Pass
    end if

    if request not in already analyze then
        Calculate the pick-up time.
        Calculate the waiting time.
        Stock the value in Waiting time.
        Stock the name of the request in the list. already analyze
    end if
end for

Return both list : Waiting time & already analyze
```

Figure 8: Pseudo code to calculate the waiting time performance

Algorithm 5 Calculate detour time performance

```
Initialize list to delete request's name at the end.

# First step : Knowing who is sharing → experiencing detour.
for every request that are travelling do
    if another request is in the same vehicle then
        Stock requests name in request are sharing if not already in it.
    end if
end for

# Second step : Knowing when requests arrive at their destination
for every request in request are sharing do
    if request not in traveling passenger list then
        Calculate approximate travel time without sharing.
        Stock the actual traveled time.
        Compute the experienced detour time.
        Stock request name (not sharing anymore).
    end if
end for

for every request that are not sharing anymore do
    Remove request of request are sharing
end for

Stock the actual traveling passenger in traveling passenger old
variable to always have the state at time − step

Return traveling passenger old
```

Figure 9: Pseudo code to calculate the detour time performance

2.2 Second model : Simultaneous

The second model has no major difference with the first one. In this model the solo and duo matching is done simultaneously, increasing therefore the number of scenarios to check. Then the computational time for this model can be relatively long (twice the time of the first model) but as the number of scenario of matching grows, it is more accurate by choosing the more efficient one. Each function has been updated according to these small changes, like the addition of a scenario (the 0, which is simply a scenario without sharing) for the waiting and detour time check.

2.2.1 Main code

Only the main code is presented because the changes in the functions are not major enough to be relevant in this presentation. It is possible to see the removal of the solo matching as well as the addition of the scenario 0 in the waiting and detour time checking loop.

Algorithm 1 Main Code (simultaneous method)

```
Discretize the simulation duration in time steps.
Initiate lists & variables that will contain performance metrics of the
current algorithm.

for time = 1 · steps, 2 · steps, ... do
    Update the network, arrivals, abandonments, the number of private
    vehicles and some performances metrics.

    if time is a multiple of the batch size then

        Create a list with Unassigned Requests.
        Create a list with Feasible Vehicles.
        Initialize the cost matrix.

        for each request in Unassigned Requests do
            for each vehicle in Feasible Vehicles do
                Initialize a list that will stock the cost & its feasibility.

                for scenario = 0, 1, 2, ..., 6 do
                    Check the waiting time constraint.
                    Check the detour time constraint. (if l.20 → True)
                    Stock costs & feasibility in their respective metrics list.
                end for

                Update the cost matrix with the lowest chosen cost (obj.
                function can be waiting or detour time) and stock which
                scenario gives this lowest cost.
            end for

        end for

        MatchSharing(CostMatrix)
    end if

    Update the last performance metrics.
end for
```

Figure 10: Pseudo code for the main simultaneous code

3 Results

3.1 Evaluation plan

After writing the code for the two models we have then proceeded with their evaluation. We were able to simulate a real application of our algorithm by importing data regarding the requests arrival in a spatial domain originated from the city of Shenzehn. Note that the speed in the network is dynamic while the available operating modes are private vehicles and ride-hailing services. Their fraction in the network can be changed, but for all our simulations the fraction of private vehicles is set to 80%.

For the evaluation, we had to generate the results by running a simulation for each combination of the problem variables. Each simulation being very long to compile, we had to properly chose an evaluation plan to investigate the influence of each parameter without wasting too much time generating useless data. In particular, it wasn't possible to create lists of values for all the studied variables and then iterate all over the possible combinations because it would have taken too much time and it would not have made sense for the purpose of our project. Actually, the goal was just to generally observe how our algorithm works and see if the obtained results are reasonable.

Therefore, we decided to take a baseline scenario, defining fixed parameter values, and then investigate for each case of study the influence of one parameter by taking a few different associated values and running the simulation for each of these. We then evaluated the results by plotting for each case of study the performance in terms of abandonments, sharing fraction, mean waiting time, mean detour time and vehicle occupancy. The results obtained with different simulations are comparable because we always ran the same pseudo-random simulation, by only changing the parameters of the problem.

The percentage of abandonments is computed with respect to the total number of requests and the sharing fraction represents the percentage of passengers who shared their trip with respect to the total number of assigned passengers, i.e. the passengers who did not abandon. To compute the mean detour time we consider only the passengers having shared a trip while the mean waiting time is found on average considering all the assigned passengers.

It is also important to underline that for each case study the objective function for the assignment corresponds to the optimization of one unique quantity. In particular, we tried three optimization functions, one aiming to maximize the number of assignment, another to minimize the waiting time and the last one to minimize the detour time. We then tried to select at each time the objective function more closely related to the analysed parameter of the concerned case study.

We therefore used a very simple optimization function and a very few problem variables and perspectives. We then made a few simplification, but if we were treating an algorithm for a real application we should obviously consider also other aspects.

The second model we implemented - the simultaneous matching model - has a greater time cost as the code is almost three time as long to compile. For the baseline scenario, using the first model - the sequential matching model - the code takes around 1 and 2 hours to compile, while for the second model and the same scenario, the time cost is between 4 and 5 hours. For this reason, since the time was limited, for the evaluation of our results we have focused more on the first model, while to evaluate the second model and be able to obtain results in a reasonable time we had to reduce the number of vehicles and the number of requests in the simulations.

In *Table 1* the baseline scenario considered is defined.

We established four cases study for the first model to examine, in order, the influence of the willingness to share of the passengers - which is the fraction of requests available to share a trip - the fleet size - that is the number of vehicles in the simulation - the maximum waiting time accepted by the users before they are picked up and the maximum detour time accepted by the passengers who are prepared to share their trip. We started by considering just three values for each one of these parameters but we were then forced to take a few more values for the cases study of the maximum waiting time and detour time to obtain clearer

Table 1: Baseline scenario

Regular baseline scenario	**Reduced baseline scenario**
Number of requests: 31500	Number of requests: 7850
Fleet size: 2000 veh	Fleet size: 500 veh
Willingness to share: 80% of the requests	
Maximum waiting time: 5 minutes	
Maximum detour time: 10 minutes	

tendencies on the graphs.

For the second model we focused only on the influence of the maximum detour time because of the greater time cost of the algorithm, as explained before. We recall that in this case we had to use the reduced scenario. We finally established also another case study for the first model, considering once again the influence of the detour time, but this time for the reduced scenario, as for the second model. This was necessary to have comparable results between the first and the second model. This last case study is only considered in the comparison section, while the detailed performances graphs are given in *Appendix A*

The four case study of the first model are illustrated in *Table 2* below, while the case study of the second model is defined in *Table 3*.

Table 2: Cases study for the first model

Case study	Considered values	Other parameters	Objective function
1. Willingness to share	{0, 50, 100} (%)	As for the **regular** baseline scenario	Maximize the assignment
2. Fleet size	{1800, 2000, 2300} (veh)	As for the **regular** baseline scenario	Maximize the assignment
3. Max. waiting time	{3, 5, 7, 9, 12, 15} (min)	As for the **regular** baseline scenario	Minimize the witing time
4. Max. detour time	{3, 5, 7, 10, 15, 20} (min)	As for the **regular** baseline scenario	Minimize the detour time
4.1 Max. detour time	{3, 5, 7, 10, 15, 20} (min)	As for the **reduced** baseline scenario	Minimize the detour time

Table 3: Case study for the second model

Case study	Considered values	Other parameters	Objective function
5. Max. detour time	{3, 5, 7, 10, 15, 20} (min)	As for the **reduced** baseline scenario	Minimize the detour time

Note that the performances graphs are showed in *Tables 5-10* and they are discussed in the following section, while the complete visualization of simulations data are reported in *Appendix B*

3.2 First model - Sequential matching

3.2.1 Case study 1: Willingness to share influence

The results for the case study of the willingness to share are visible in *Table 5*.

As shown by the graphs in the table, if the willingness to share increases, there are more passengers who accept to share a trip and the fraction of vehicles carrying two passengers increases - as it can be seen on the occupancy graph - while the fraction of vehicles carrying only one or zero passengers decreases. The assignment is actually optimized for greater values of the willingness to share. The sharing fraction goes from almost 23% for a willingness to share of 50% - i.e. the half of the requests - to a percentage around 45% when the willingness to share is set to 100% and every user is available to share.

For this case study the objective function aims to maximize the assignment, thus with the increase of the willingness to share the percentage of abandonment decreases a lot because of the more flexible request of the passengers and the resulting greater sharing fraction. With more passengers available to share their trip and a greater vehicle occupancy on average, it is purely logical to imagine that there are more free vehicles ready to pick up new users. The abandonments go from an initial percentage of 7% to the final value close to 0%.

What is interesting to see is that the mean waiting time slightly increases. This may happen, first of all because the objective function aims to maximize the assignment and it is not directly related to the waiting time, but in particular because more passengers are sharing a trip and this may cause a longer waiting time for the second passenger who is picked up because the vehicle has to go picking up the first passenger before. Therefore, to maximize the assignment and increasing the fraction of passengers sharing their trip the algorithm sacrifices other aspects. However the value of the mean waiting time is kept in a range of values between 1min 52s and 2min 21s, which is actually a very short time.

We can finally see that with the increase of the willingness to share and the fraction of vehicle carrying 2 passengers the mean detour time also increases. To maximize the assignment the algorithm forces more passengers to share even if their routes are not very similar, causing their travel time to increase. However the increase of detour time is very small: 13s for a willingness to share going from 50% up to 100%.

3.2.2 Case study 2: Fleet size influence

Regarding the fleet size influence, the results of this second case study can be seen in *Table 6*.

If the fleet size increase, we have a greater density of vehicles in the simulation. Hence, there are more available vehicles on a shorter range of distance and we can imagine intuitively that each passenger should be assigned quicker to a free vehicle in his proximity. Therefore, the mean waiting time slightly decreases - as you can see on the graphs - while the percentage of abandonments decreases exponentially, from 5.71% for a fleet size of 1800 vehicles to almost 0% for a fleet size of 2300. This may happens because the goal of the objective function is actually to minimize the abandonments and the greater number of vehicles is immediately exploited for this purpose.

It can then be observed a decrease in the fraction of vehicles carrying 2 passengers, as showed by the occupancy graph. The sharing fraction changes a lot and goes from 55.4% to 12.1%. This is also followed by a decrease of the mean detour time. This means that for a greater number of vehicles available to serve the users, the algorithm is actually able to globally increase the level of service of the passengers by decreasing the sharing fraction - which decreases a lot the travel time of the passengers because they have no detour - but also slightly decreasing the mean waiting time and the mean detour time of the passengers who still share their travel. The level of service could be even greater with a different objective function, directly considering also the minimization of the waiting time or the detour time for example.

Globally we can deduce from our results that the theoretical optimal situation from passengers point of view would be here to have a greater number of vehicles. This is obvious but also simplistic, because it does not consider the point of view of the drivers. With a bigger number of vehicles, the income of every single driver would be smaller. Moreover, the goal of the share-ride concept is to maximize the sharing trips to minimize the number of vehicles and therefore also the congestion related problems and the energy consumption. Hence, an equilibrium should be found by a ride-hailing platform.

3.2.3 Case study 3: Waiting time influence

The results in this case are exposed in *Table 7*.

As it can be seen from the graphs, the distribution of the 'vehicle occupancy' is quite constant for every value of the maximum waiting time, except for a waiting time of 3 minutes. The sharing fraction initially decreases for a greater maximum waiting time but then slightly restart increasing, always remaining between a value of 48.5% and 44.4%. The graphs of the detour time has the same tendency and varies by only 7 seconds from the maximum to the minimum values, which means that the inconvenience in terms of time for the passengers who share does not change depending on the maximum waiting time.

Then, as reported on the graph, the mean waiting time increases and progressively tends to stabilize because the aim of the simulation is to minimize it. This is combined with an important initial decrease in the number of abandonments, which comes from the fact that the constraint concerning the waiting time is less restrictive and it is therefore easier to find a free vehicle at a sufficiently close distance to pick up the passenger before he renounces. However, it is interesting to see that the number of abandonments slightly start increasing again for progressively greater values of the maximum waiting time. Actually, this could be related to a phenomenon referred as wild goose chase.

The wild goose chase is an expression used to talk about a long search or pursuit that was unsuccessful or impossible, which is probably what happens here. By increasing the maximum waiting time, we accept some request that are unaffordable and unjustified, and this led to longer waiting times and less vehicles available, and therefore more abandonments. In ride-hailing platforms this phenomenon is considered as a common failure state which occurs during high demand times, and which must be considered to prevent inefficiencies. One solution is often to raise the prices during high demand times, which brings demand back under control. However, in our case the raise in the number of abandonments is very low. [2]

3.2.4 Case study 4: Detour time influence

The performances graphs for this fourth case study are reported in *Table 8*.

We can see that the mean waiting time is not affected, while the number of abandonments initially decreases - probably because the detour constraint is less restrictive - and then slightly increases again without a precise explainable tendency.

Once again, the distribution of the vehicle occupancy does not vary much. The sharing fraction of the passengers is fairly constant for every value of the maximum detour time. The mean detour time slightly increases at the beginning and stabilizes quickly as the objective function is to minimize it, but the difference between his maximum and minimum value is 3 seconds. Hence, we can say that the mean detour time in not affected by the change of the maximum detour time and remains very low. The maximum obtained value corresponds to 21 seconds for a maximum accepted detour time of 20 minutes, which is clearly a surprising result. Moreover, if we look on the detour times distribution in *Table 16* at *Appendix B.4*, we can see that the results remain unchanged for a maximum detour time bigger than 10 minutes and any detour time changes from one configuration to another. This seems to indicate that in these cases the critical constraint is not the one regarding the detour time.

We therefore supposed this critical constraint could be the maximum waiting time and we established a new similar case study, this time with a maximum waiting time of 10 minutes instead of the previous 5 minutes, as reported below in *Table 4*. The performances graphs are reported in *Table 9*

Table 4: Case study 4.2: Detour time influence with different waiting time, first model

Case study	Considered values	Other parameters	Objective function
4.2 Max. detour time	{3, 5, 7, 10, 15, 20} (min)	**Regular** baseline scenario **Waiting time: 10 min**	Minimize detour time

We can see that for a maximum waiting time of 10 minutes the tendencies of the graphs are the same as before, but the values change, except for the fraction of sharing which is almost the same as before. The abandonments decreases, as the maximum waiting time is greater and the constraint is easier to satisfy. The mean waiting times increase and the detour times decreases. Actually the objective function is to minimize the detour time and the algorithm takes advantage of the bigger accepted waiting time to make better assignments corresponding to shorter detours.

The results show that our supposition could be at least partially right, as the change of the maximum detour time actually changed the results, but not the tendencies. This because the maximum waiting time plays a role in the assignment, and could still be a critical constraint.

It is globally deduced that for the given model and the given parameters the increase of the maximum detour time has not a relevant influence on the model performances. Especially for greater value of the maximum detour time, this may be related to the objective function aiming to minimize the detour time, which causes the algorithm not to increase the number of sharing trips nor the detour time of the passengers who shared.

3.3 Second model - Simultaneous matching

3.3.1 Case study 5: Detour time influence

Once again, we considered the influence of the maximum detour time in a simulation where the objective is to minimize the detour time, but this time we used the second model with a reduced fleet size and a reduced number of requests, because of the greater time cost. We kept almost the same proportion between the number of vehicles and the number of requests, but their density in the simulation is decreased by a factor of four. The results are reported in *Table 10*.

Some tendencies are similar to the ones observed before for the first model. The mean waiting time is not affected by the change of the maximum detour time. However, we see that in this case the mean detour time increases and stabilizes progressively, doubling his value from 45s for a maximum detour time of 3 minutes up to 1min 33s for a maximum detour time of 15 minutes. At the same time the sharing fraction increases from 50.8% to 58.1% and the abandonments decrease from 4.51% to 2.59%. Therefore, the algorithm better reflect the effects of the change of the maximum detour time for the reduced scenario. Note that the same case study for a reduced scenario using the first model (graphs used for the comparison in next section and reported in detail in *Table 12* in *Appendix A*) gives the same tendencies.

Table 5: Performances graphs
Case study 1: Willingness to share influence, First model

Objective function: Maximise the assignment
Fleet size: 2000
Number of requests: 31500
Maximum waiting time: 5 min
Maximum detour time: 10 min

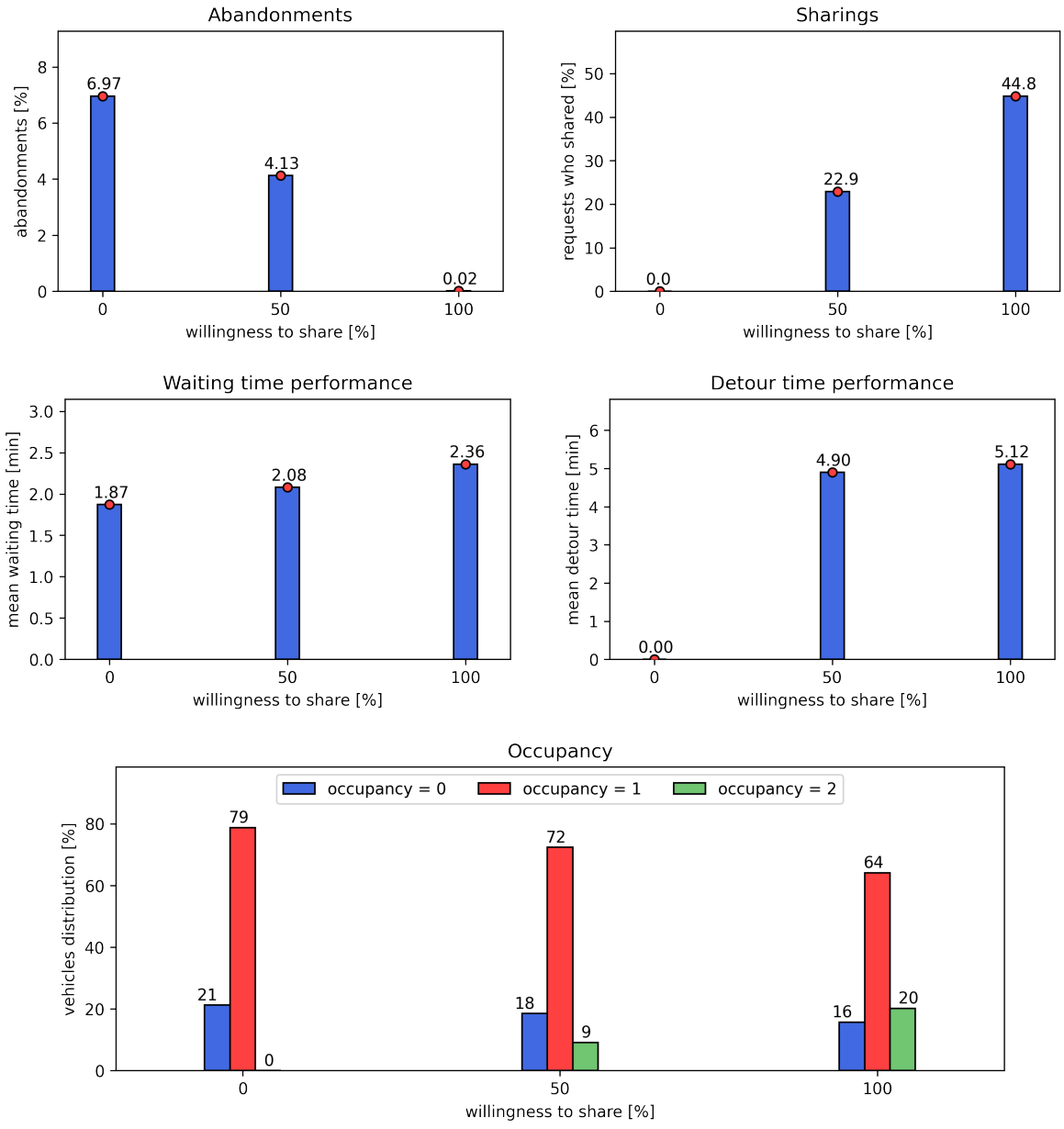


Table 6: Performances graphs
Case study 2: Fleet size influence, First model

Objective function: Maximise the assignment
Willingness to share: 80%
Number of requests: 31500

Maximum waiting time: 5 min
Maximum detour time: 10 min

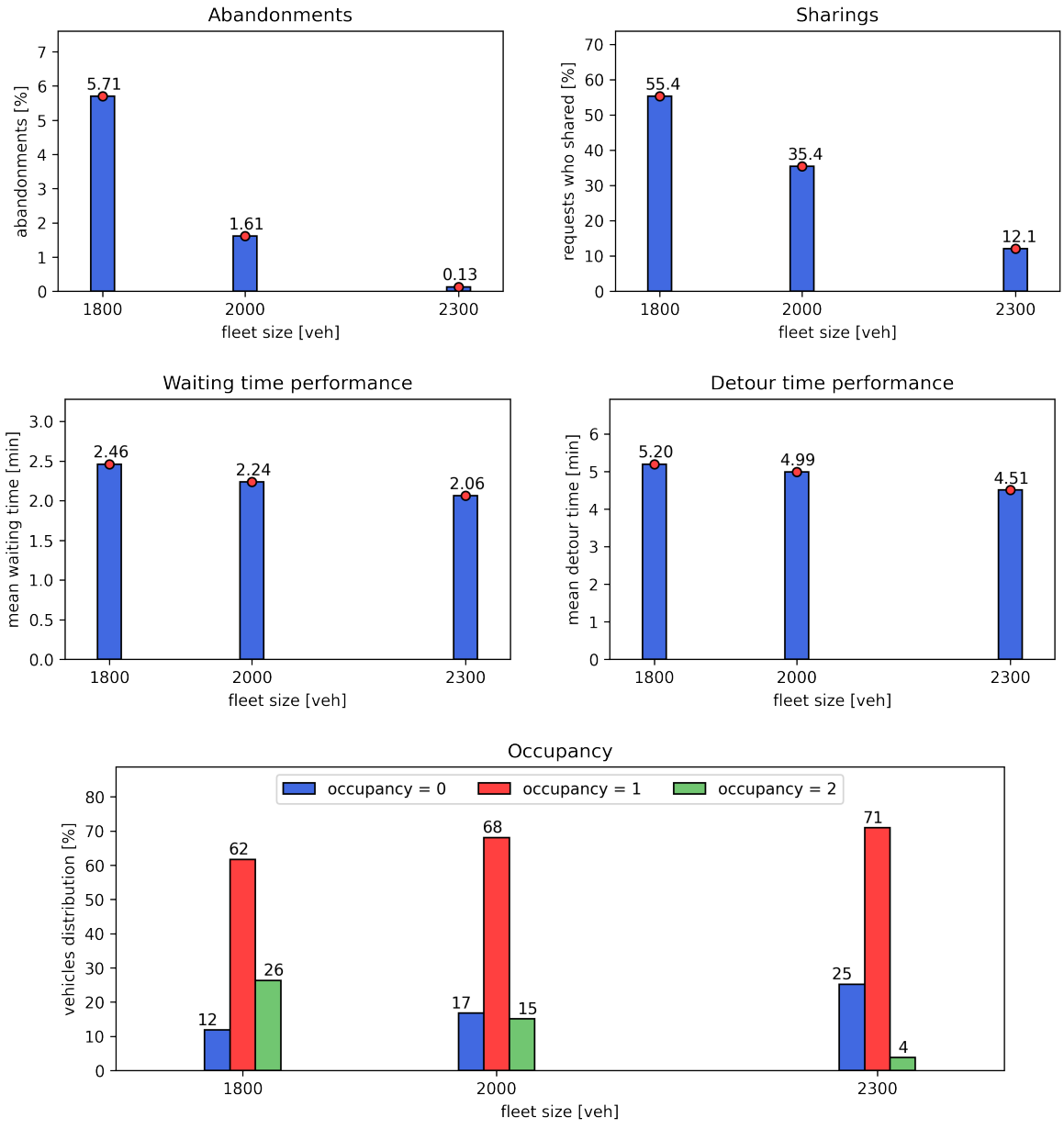


Table 7: Performances graphs
Case study 3: Waiting time influence, First model

Objective function: Minimize the waiting time
Fleet size: 2000
Number of requests: 31500
Willingness to share: 80%
Maximum detour time: 10 min

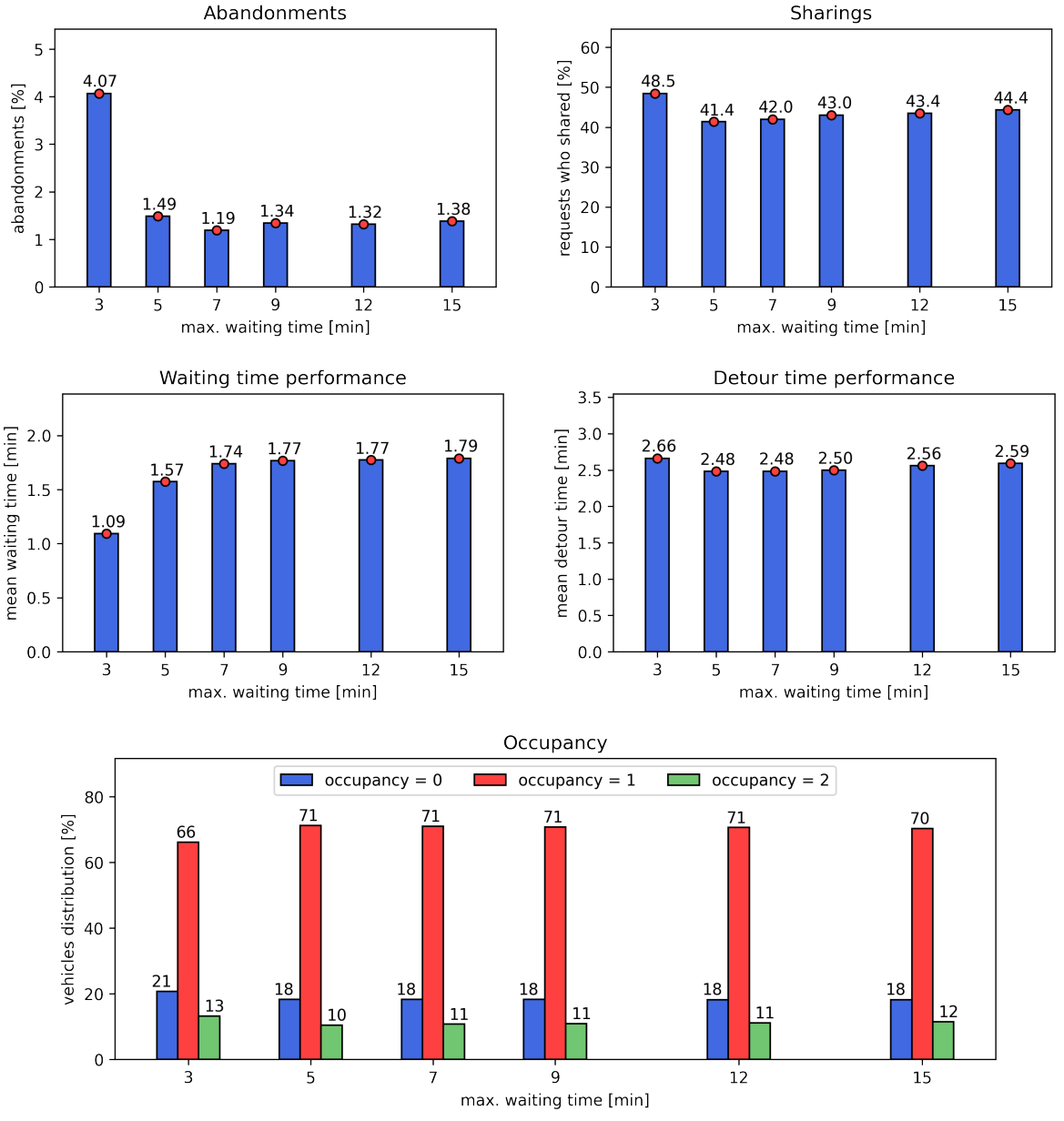


Table 8: Performances graphs
Case study 4: Detour time influence, First model

Objective function: Minimize the detour time

Fleet size: 2000

Number of requests: 31500

Willingness to share: 80%

Maximum waiting time: 5 min

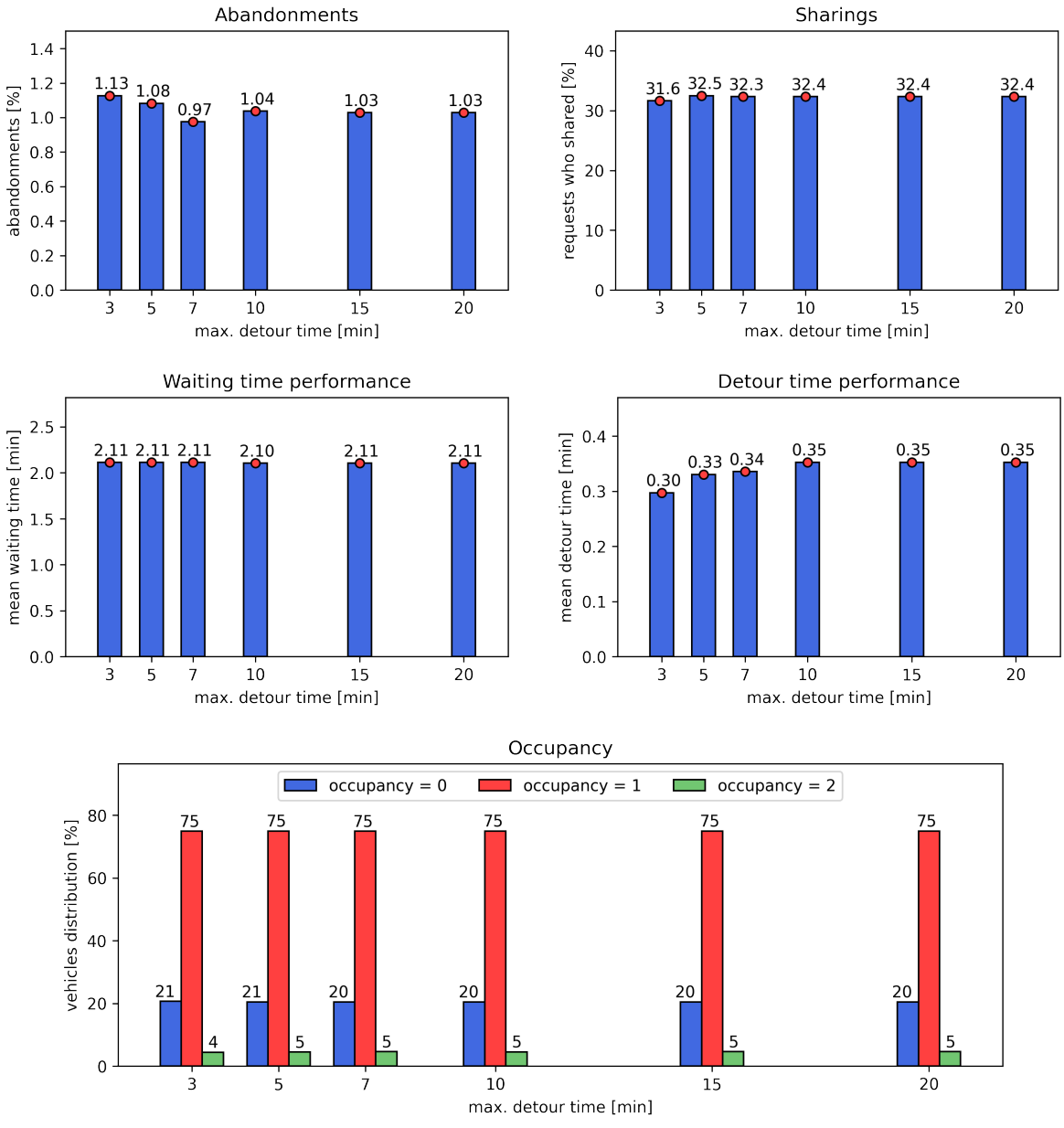


Table 9: Performances graphs

Case study 4.2: Detour time influence considering a greater waiting time, First model

Objective function: Minimize the detour time

Fleet size: 2000

Number of requests: 31500

Willingness to share: 80%

Maximum waiting time: 10 min

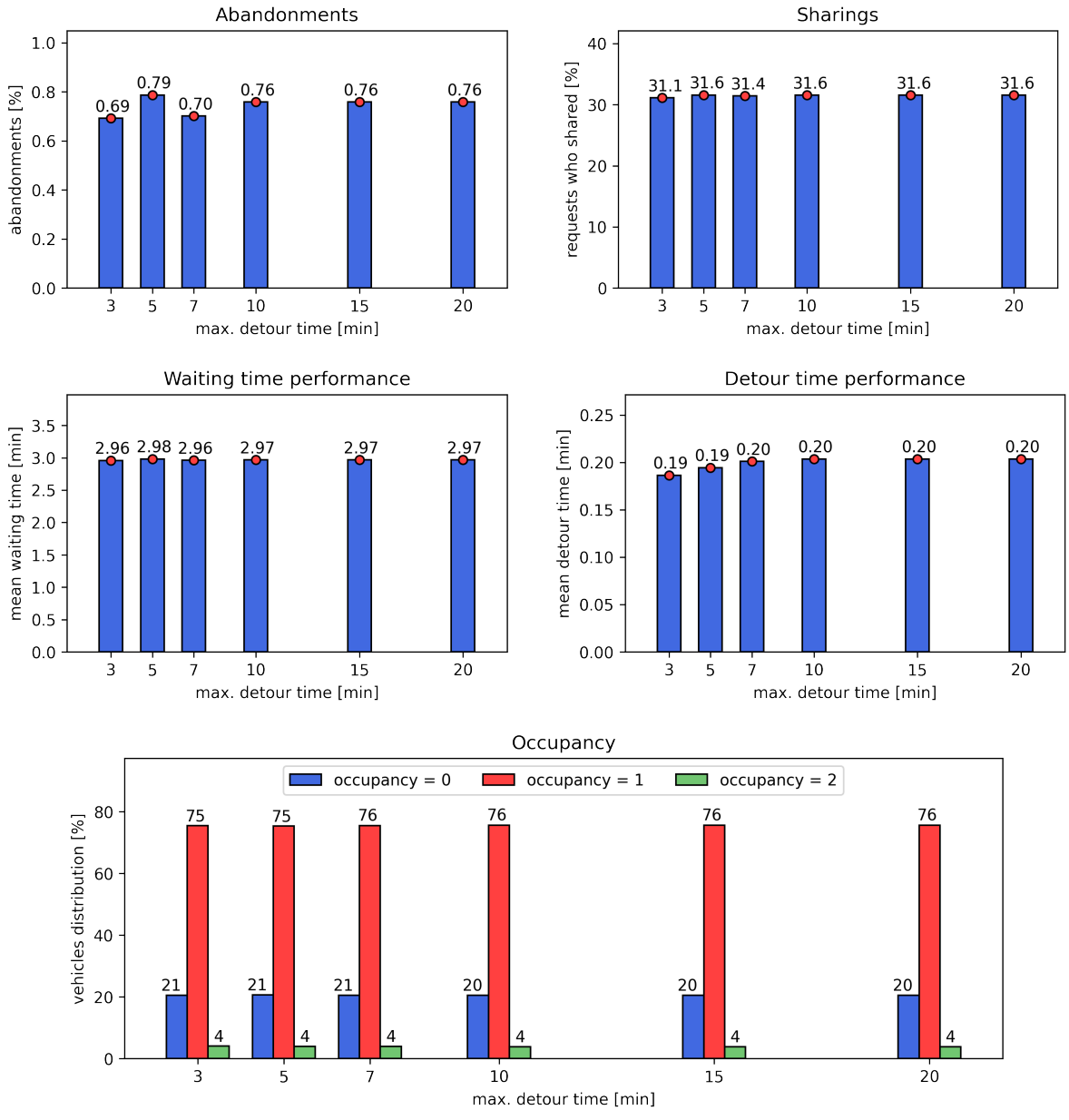


Table 10: Performances graphs

Case study 5: Detour time influence, Second model (reduced fleet size and requests number)

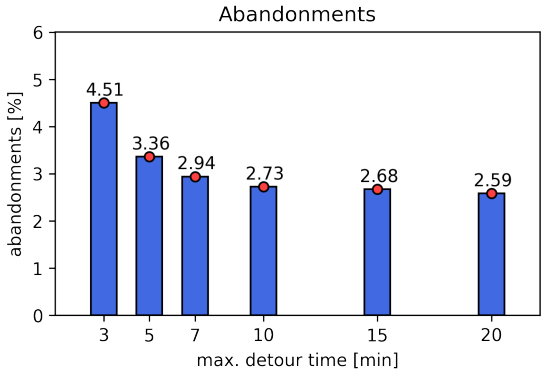
Objective function: Minimize the detour time

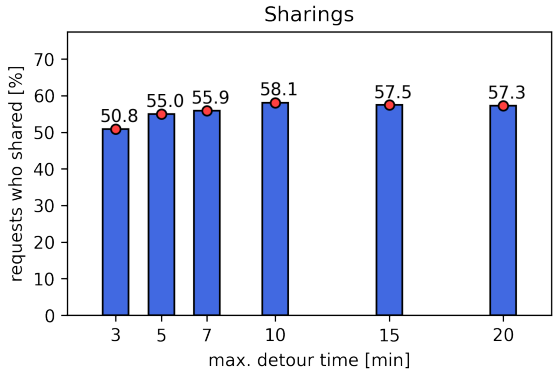
Fleet size: 500

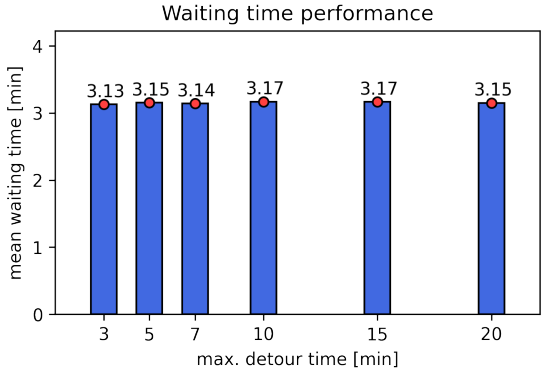
Number of requests: 7850

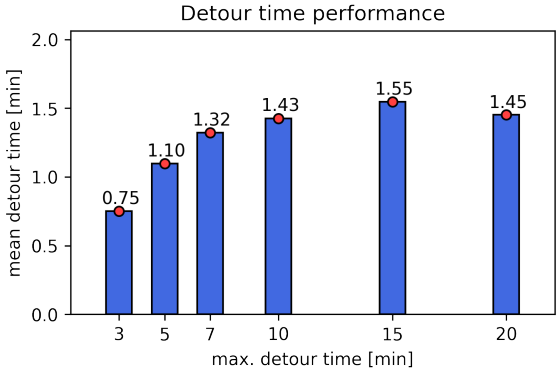
Willingness to share: 80%

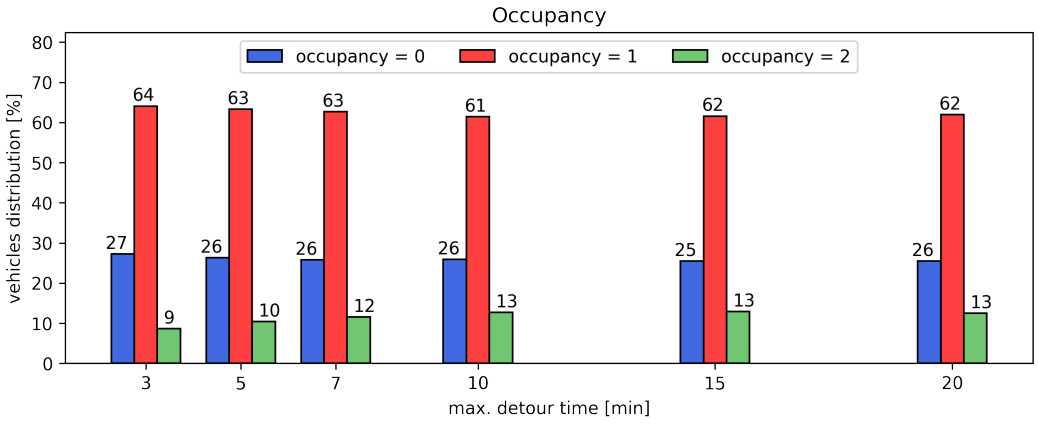
Maximum waiting time: 5 min











3.4 Comparison between the models

To compare the two models, we considered again the case study of the maximum detour time. To have comparable results for the two models we repeated the simulations of the first model with the reduced fleet size and number of requests as for the second model. As already said, the graphs for the reduced scenario considering the first model are reported in *Table 12* in *Appendix A* for the sake of completeness, while the graphs comparing the two model are in *Table 11*, on the next page.

The tendencies in each graph are very similar for the two models. However, we can see that with the simultaneous matching the mean waiting time and the mean detour time increases, which is less efficient from the point of view of the passengers. At the same time, we can see that the percentage of sharing increases significantly and the number of abandonments decreases a lot for the second model, which is positive from the perspective of the ride-hailing platform. We therefore conclude that the second algorithm is able to optimize the assignment of the shared trips by taking better advantage of the accepted waiting time and detour time of the passengers, which allows to globally reduce the abandonments.

As said, the efficiency of one model with respect to the other one depends on the perspective, as the passengers would probably consider primarily the waiting and the travel time, while the platform wants to maximize the incomes and his priority is therefore to minimize the abandonments. We can imagine that the perspective which counts more is the one of the platform, because it is the platform which make the strategic choices, but even if we were working for a ride-hailing platform and the main thing we would care about were the incomes, we should also consider the passengers satisfaction. Besides the fraction of abandonments, a larger detour and waiting time could also leave the client more unsatisfied, and this may lead them to rather consider other modes of transportation in the future. This just to point out how important it is to always consider all the perspectives, paying attention not to superficially promoting one algorithm as the best one without considering also his weaknesses.

We finally recall also that the cost in terms of time of the algorithm is much higher for the simultaneous matching and this may cause it to be less efficient.

Table 11: Comparison between the two models.

Cases study 4.1 and 5: Detour time influence (reduced fleet size and requests number)

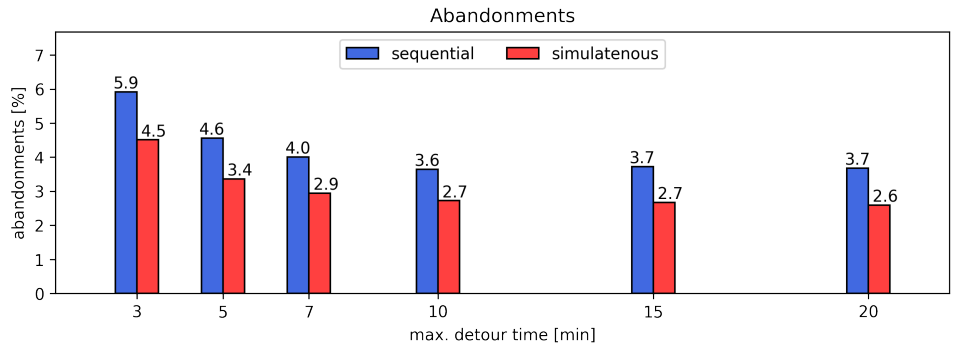
Objective function: Minimize the detour time

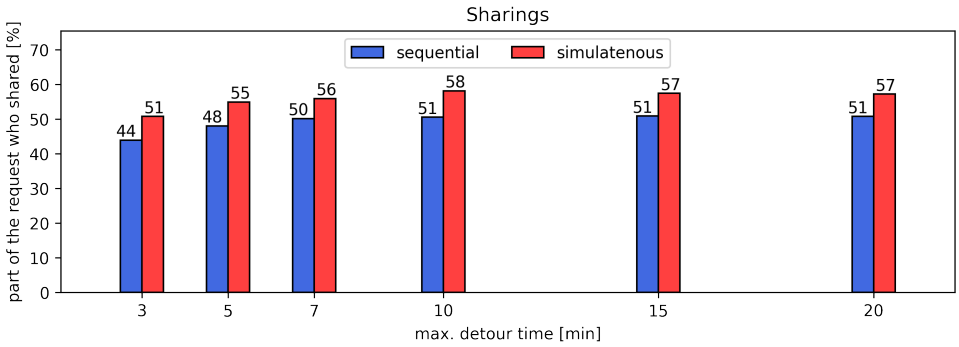
Fleet size: 500

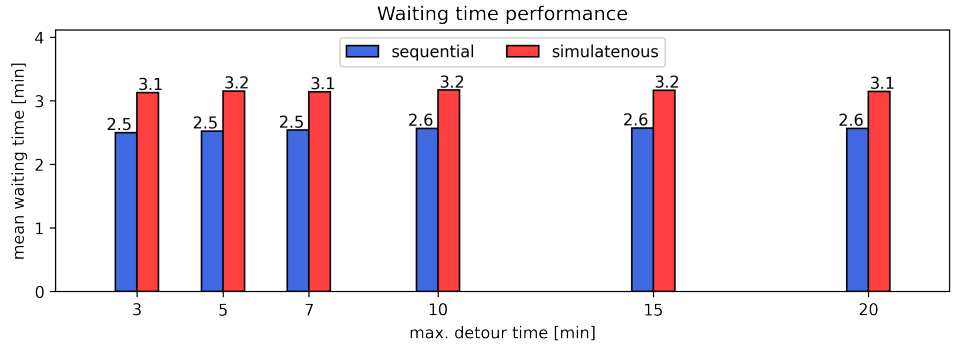
Number of requests: 7850

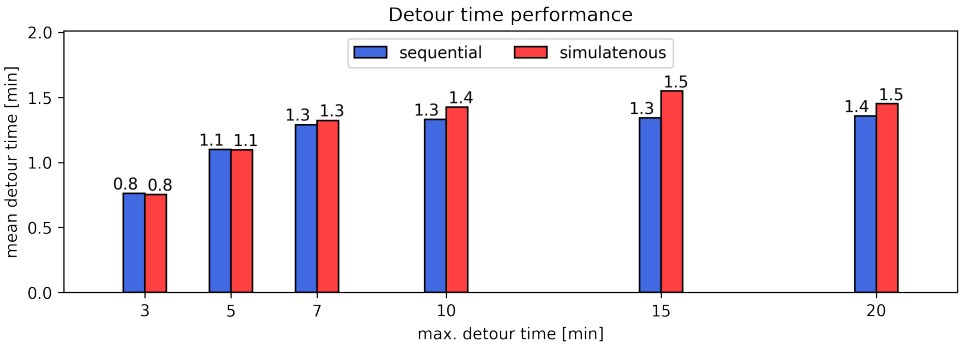
Willingness to share: 80%

Maximum waiting time: 5 min









4 Conclusion

Globally we can say that our results are consistent and we were able to justify them. The algorithms work well and especially the first model has an acceptable time cost.

However, our work has multiple limitations and is far from being considered for a real application. For example we considered a sharing of maximum two passengers, while if we had more time we could also consider a sharing of more passengers. Nevertheless, this would clearly lead to an important increase of the number of scenarios that should be considered by the algorithm during the assignment of shared trips, causing the assignments process to be longer to implement and the code longer to compile.

Another limitation concern the selected optimization function, which is very basic in our cases study. Actually we optimized only one parameter per time because the goal of our analysis was just to generally evaluate the functioning of the algorithm. If we had more time, the first thing we would probably do would be the implementation a more sophisticated optimization function and then study his influence on the results.

We have then to recall that all the analysed data come from one unique pseudo-random simulation. This choice was made with the aim to have comparable results between the different simulations during the evaluation avoiding noise related problems, but it cause the results to be restricted to this single simulation. It could therefore be useful to consider also other random simulations for the generalization of the results.

Moreover, the graphs are obtained with a limited number of simulations. To study the influence of each parameter we selected just a few associated values for the evaluation. With more time available we could therefore realize more simulations to get more data and clearer trends in the evaluation graphs, that could allow us to make better conclusions and analysis.

Finally, we are conscious that we considered just a few performance metrics, that is the waiting time, the detour time, the abandonments and the sharing fraction. It is a very limited set of performance metrics that mainly regard the point of view of the passengers. The essential parameter that we did not consider are the incomes, that is the primary aspect considered by the ride-hailing platform. Actually, by considering the perspective of the ride-hailing company we would also consider implicitly the perspective of passengers and drivers, as the incomes are related to their use of the platform and therefore their level of satisfaction for the offered service.

5 Thanks

We would like to express our sincere thanks to all the people who helped us through this bachelor project. In particular, we would like to thank our project assistant Fayed Lynn who was always available and ready to help us. Her great knowledge in the field has brought us a lot and enriched our knowledge in this sector. It was a great pleasure to work with her. Thank you.

References

[1] Alex Wallara Emilio Frazzolic Javier Alonso-Moraa, Samitha Samaranayakeb and Daniela Rusa. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *PNAS*, 2016.

[2] E. Glen Weyl Juan Camilo Castillo, Dan Knoepfle. Surge pricing solves the wild goose chase. *Stanford University*, July 2017.

[3] Jaeyoung Jung R. Jayakrishnan Ji Young Park. Dynamic shared-taxi dispatch algorithm with hybrid-simulated annealing. *Computer-Aided Civil and Infrastructure Engineering 31, 275–291*, 2016.

[4] David Røpke, Stefan; Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 2006.

[5] Senior Member IEEE Shuo Ma, Yu Zheng and IEEE Ouri Wolfson, Fellow. Real-time city-scale taxi ridesharing. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 7*, July 2015.

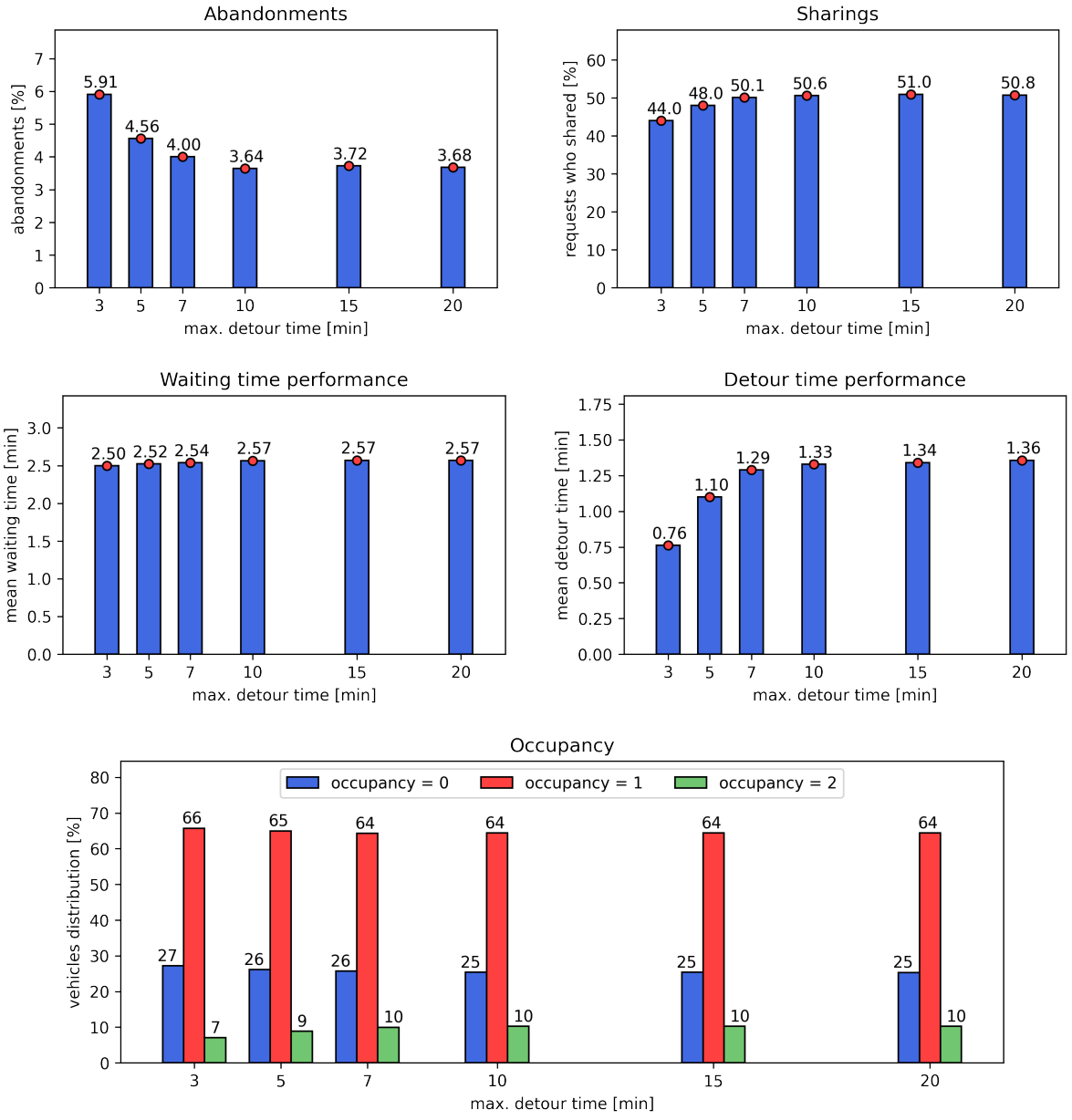
List of Figures

List of Tables

A Appendix: Performances graphs, Case study 4.1

Table 12: Performances graphs
Case study 4.1: Detour time influence, First model (reduced fleet size and requests number)

Objective function: Minimize the detour time
Fleet size: 500
Number of requests: 7850
Willingness to share: 80%
Maximum waiting time: 5 min



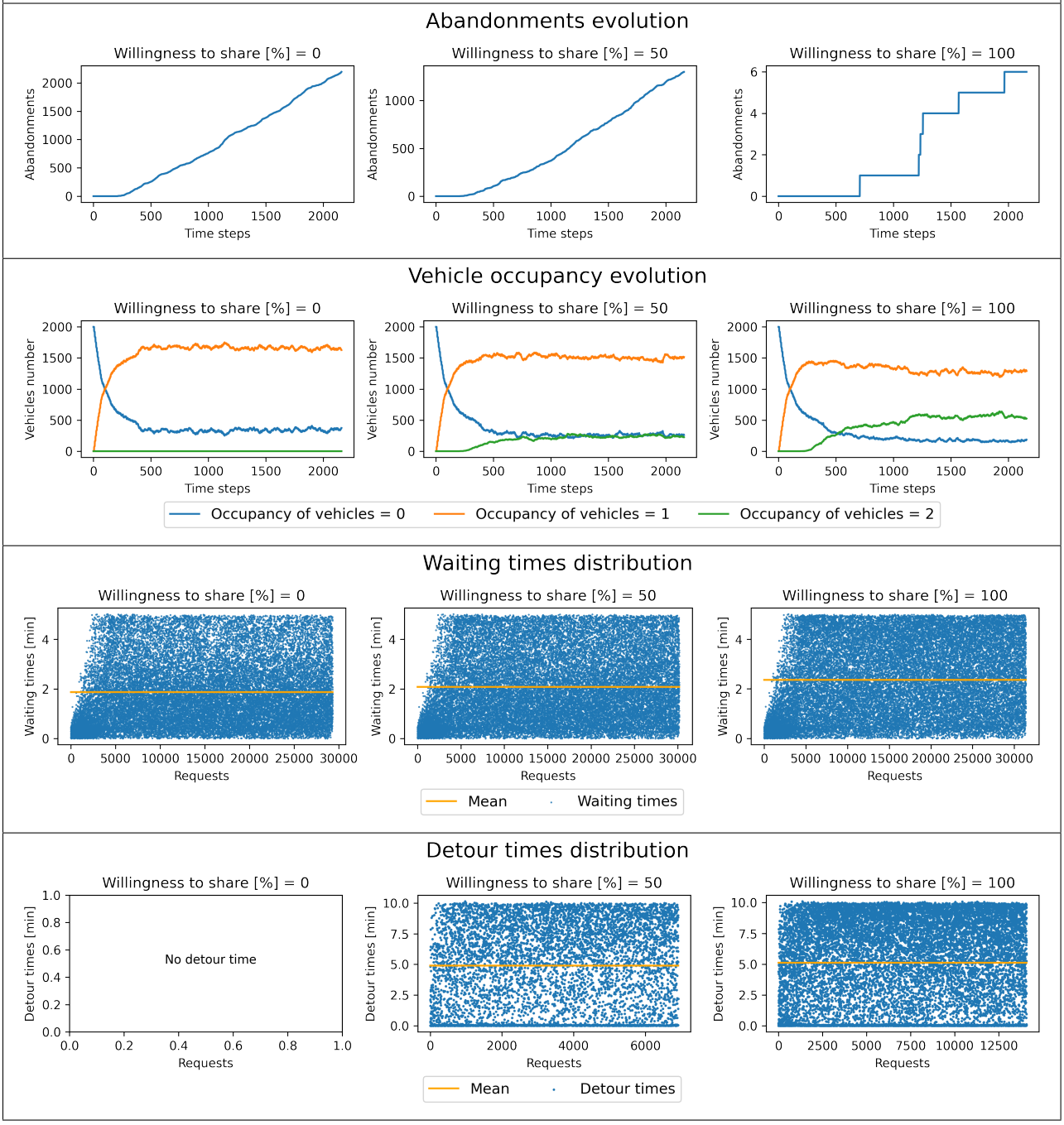
B Appendix: Complete visualization of simulations data

B.1 Case study 1: Willingness to share influence

Table 13: Complete visualization of simulations data
Case study 1: Willingness to share influence

Objective function: Maximise the assignment
Fleet size: 2000
Number of requests: 31500

Maximum waiting time: 5 min
Maximum detour time: 10 min

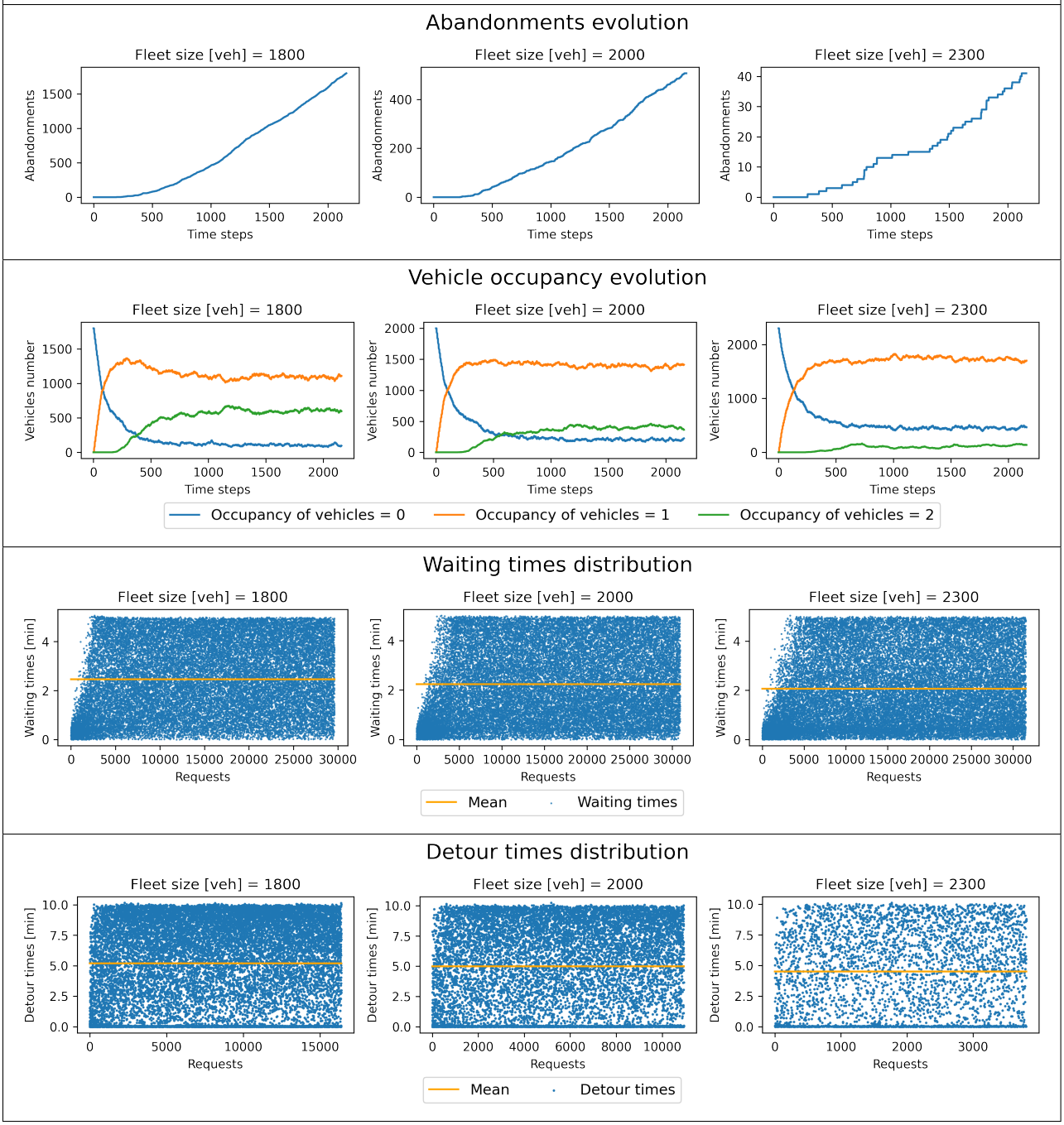


B.2 Case study 2: Fleet size influence

Table 14: Complete visualization of simulations data
Case study 2: Fleet size influence

Objective function: Maximise the assignment
Willingness to share: 80%
Number of requests: 31500

Maximum waiting time: 5 min
Maximum detour time: 10 min

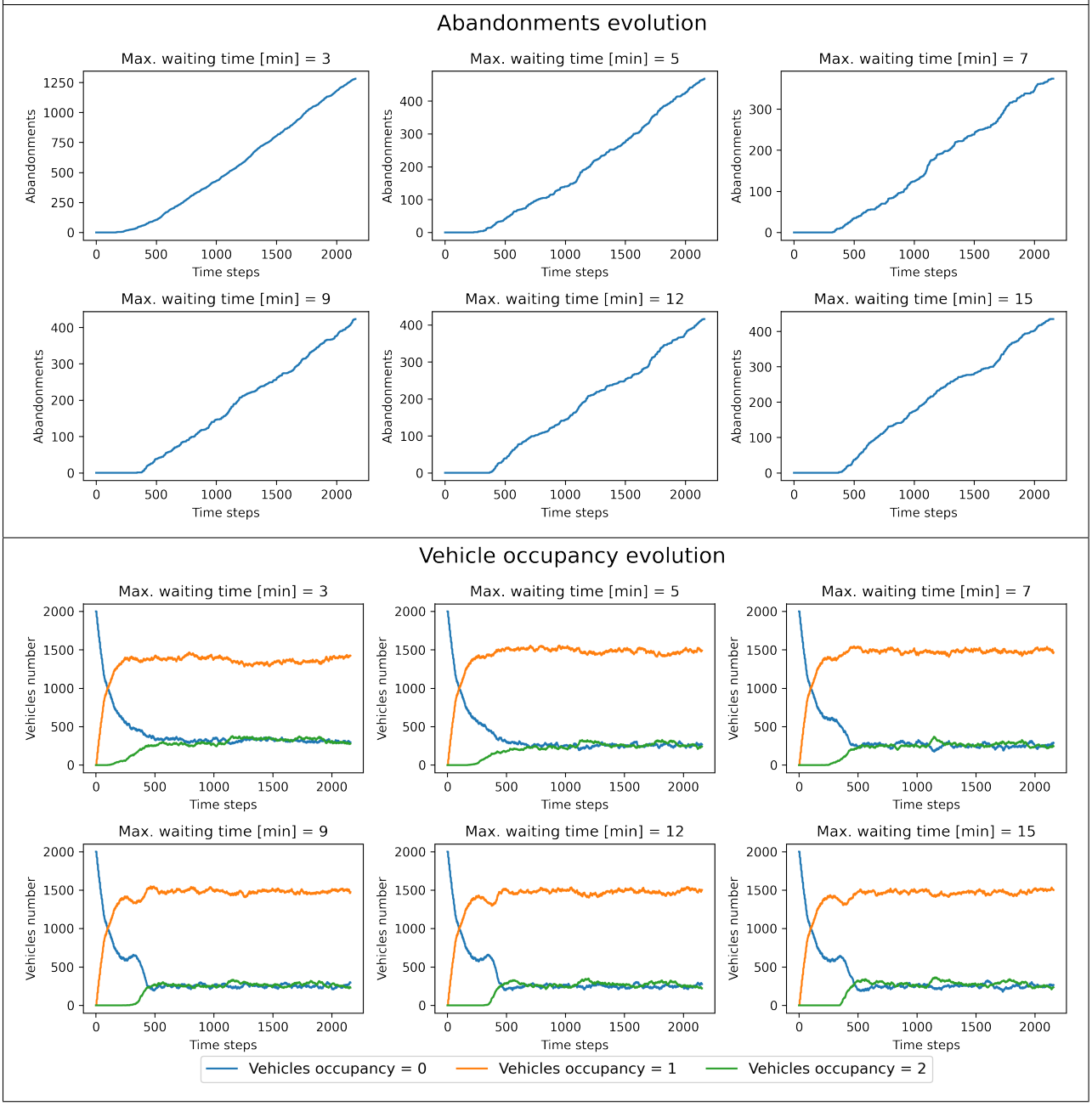


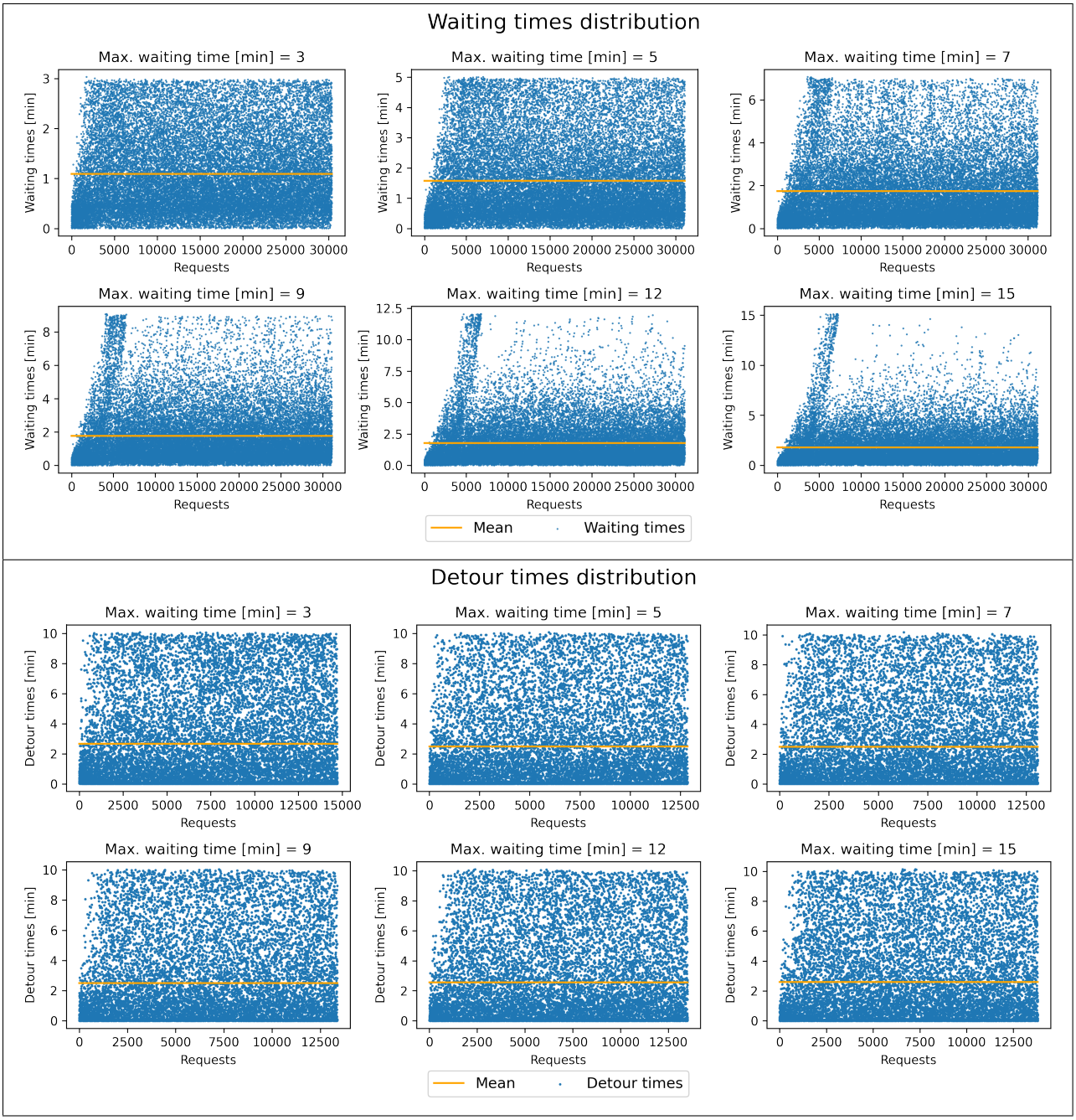
B.3 Case study 3: Waiting time influence

Table 15: Complete visualization of simulations data
Case study 3: Waiting time influence

Objective function: Minimize the waiting time
Fleet size: 2000
Number of requests: 31500

Willingness to share: 80%
Maximum detour time: 10 min



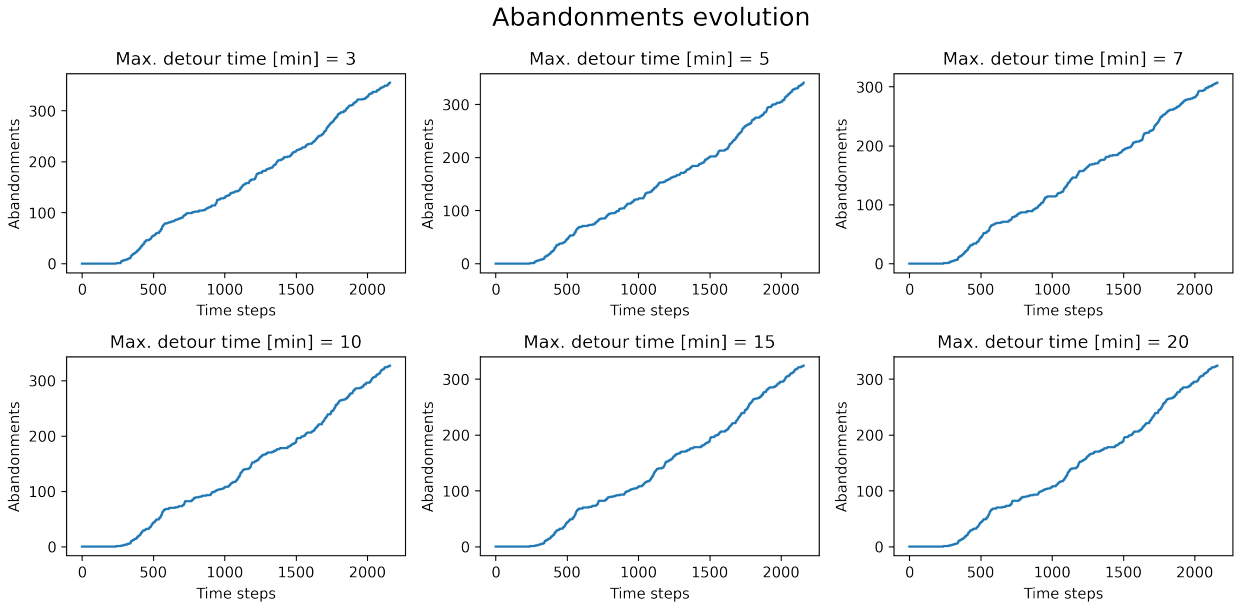

Waiting times distribution
Max. waiting time [min] = 3
Max. waiting time [min] = 5
Max. waiting time [min] = 7
Max. waiting time [min] = 9
Max. waiting time [min] = 12
Max. waiting time [min] = 15
Waiting times [min]
Requests
Mean
Waiting times
Detour times distribution
Detour times [min]
Detour times

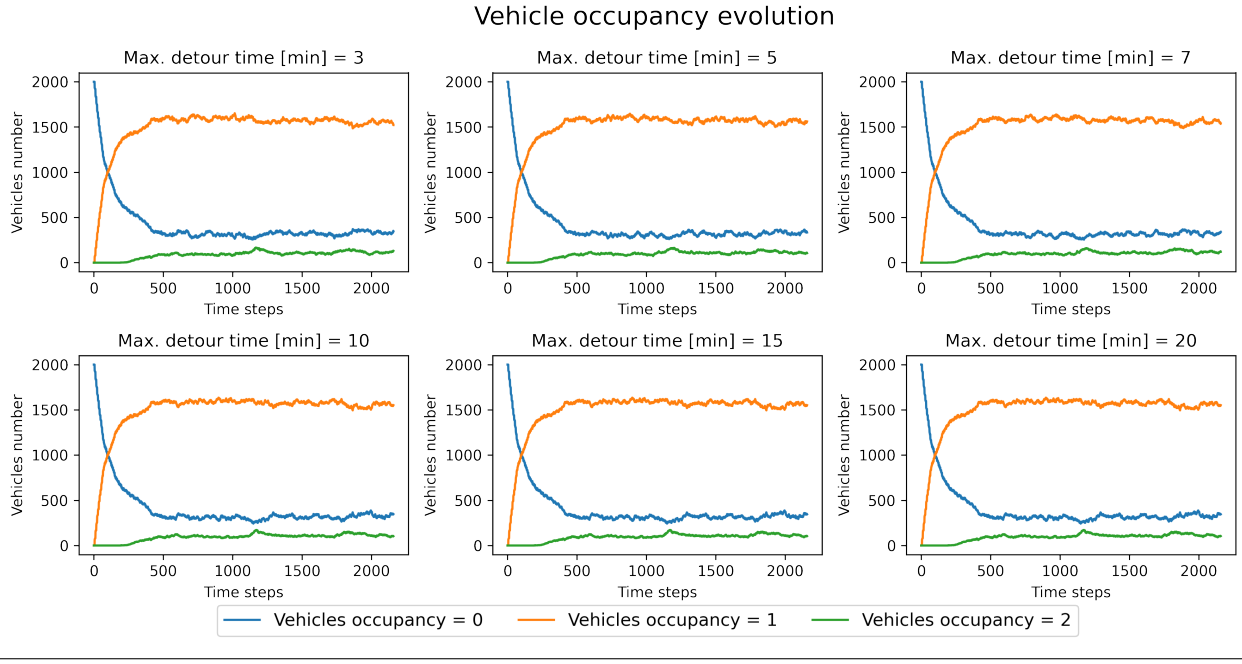
B.4 Case study 4: Detour time influence

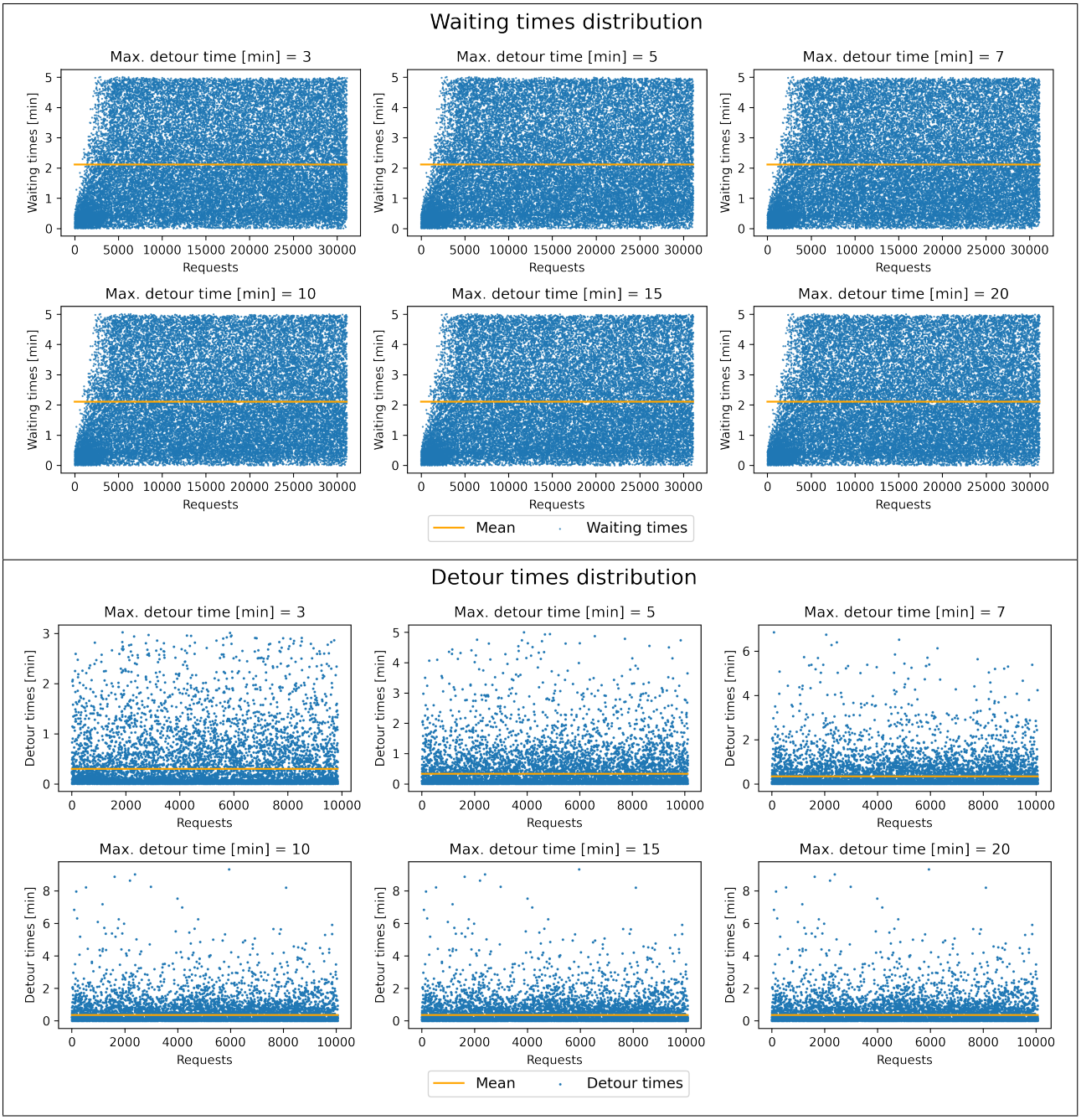
Table 16: Complete visualization of simulations data
Case study 4: Detour time influence

Objective function: Minimize the detour time
Fleet size: 2000
Number of requests: 31500

Willingness to share: 80%
Maximum waiting time: 5 min





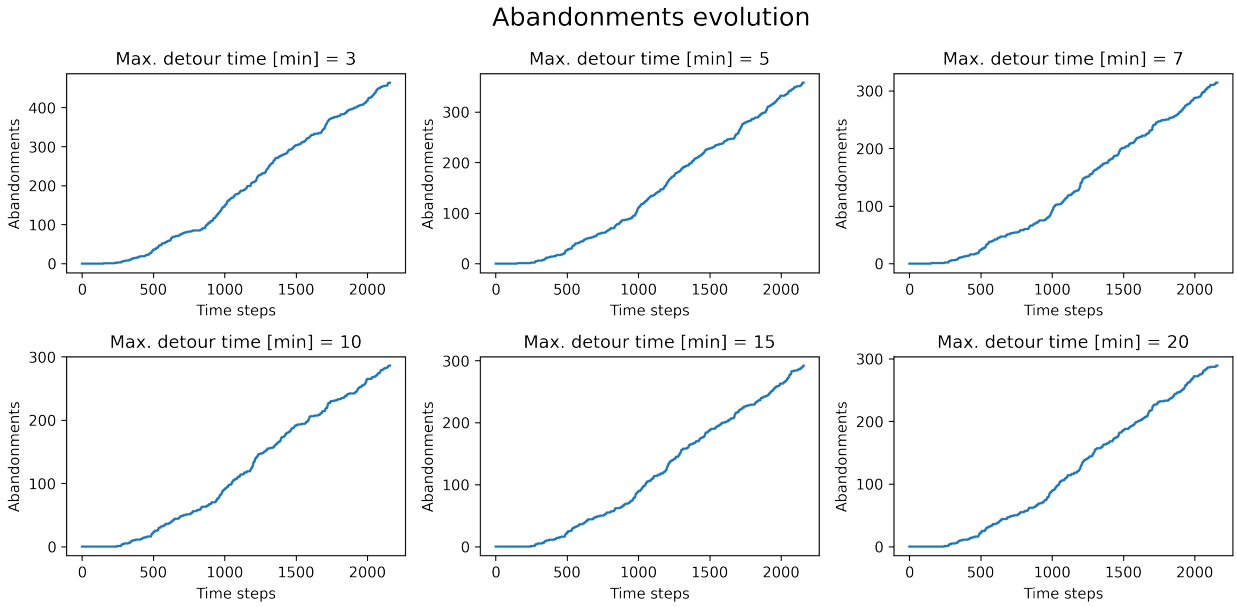

Waiting times distribution
Max. detour time [min] = 3
Max. detour time [min] = 5
Max. detour time [min] = 7
Max. detour time [min] = 10
Max. detour time [min] = 15
Max. detour time [min] = 20
Waiting times [min]
Requests
Mean
Waiting times
Detour times distribution
Detour times [min]
Detour times

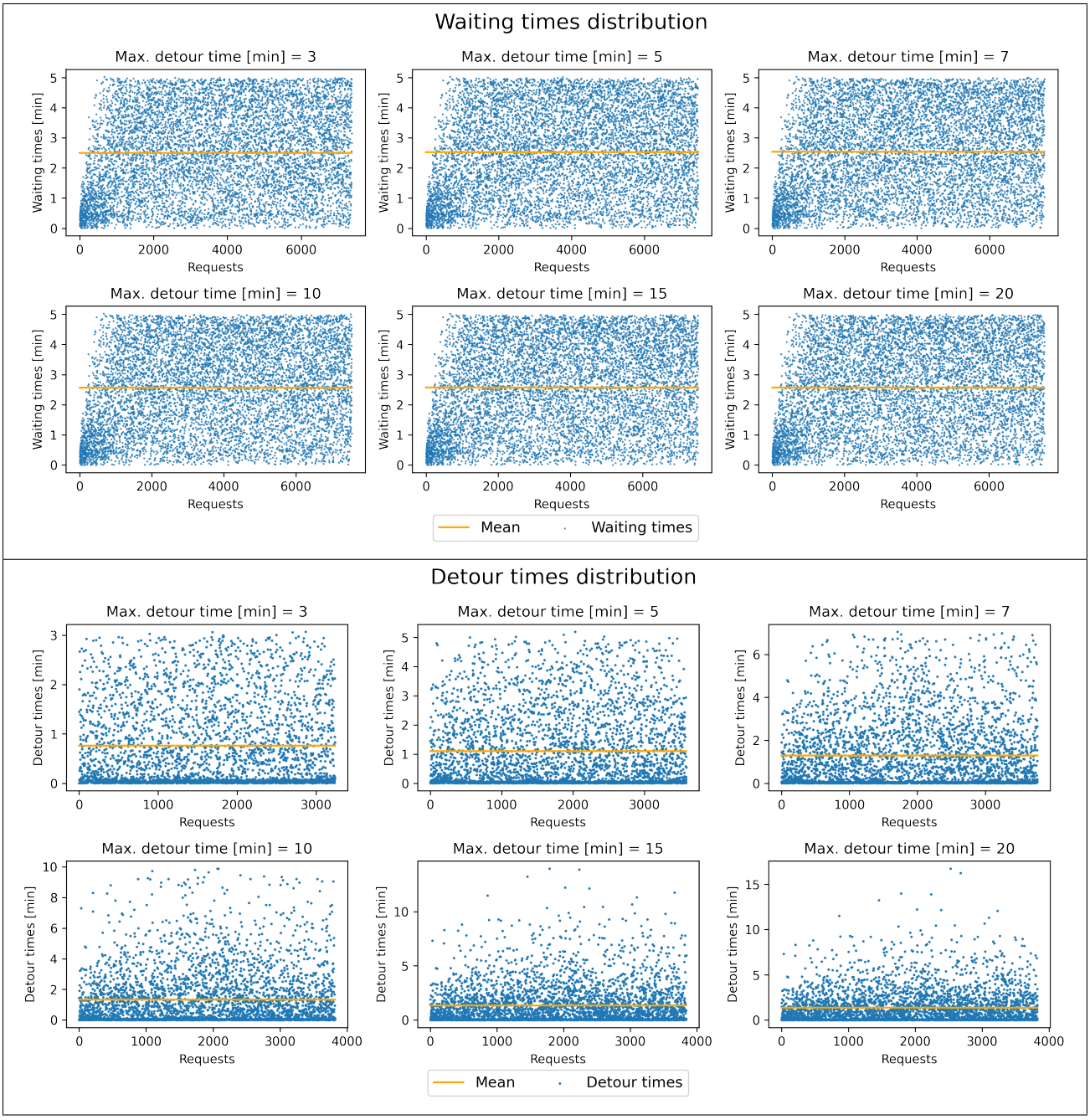
B.5 Case study 4.1: Detour time influence, First model, Reduced scenario

Table 17: Complete visualization of simulations data
Case study 4.1: Detour time influence, First model, Reduced scenario

Objective function: Minimize the detour time
Fleet size: 500
Number of requests: 7850

Willingness to share: 80%
Maximum waiting time: 5 min



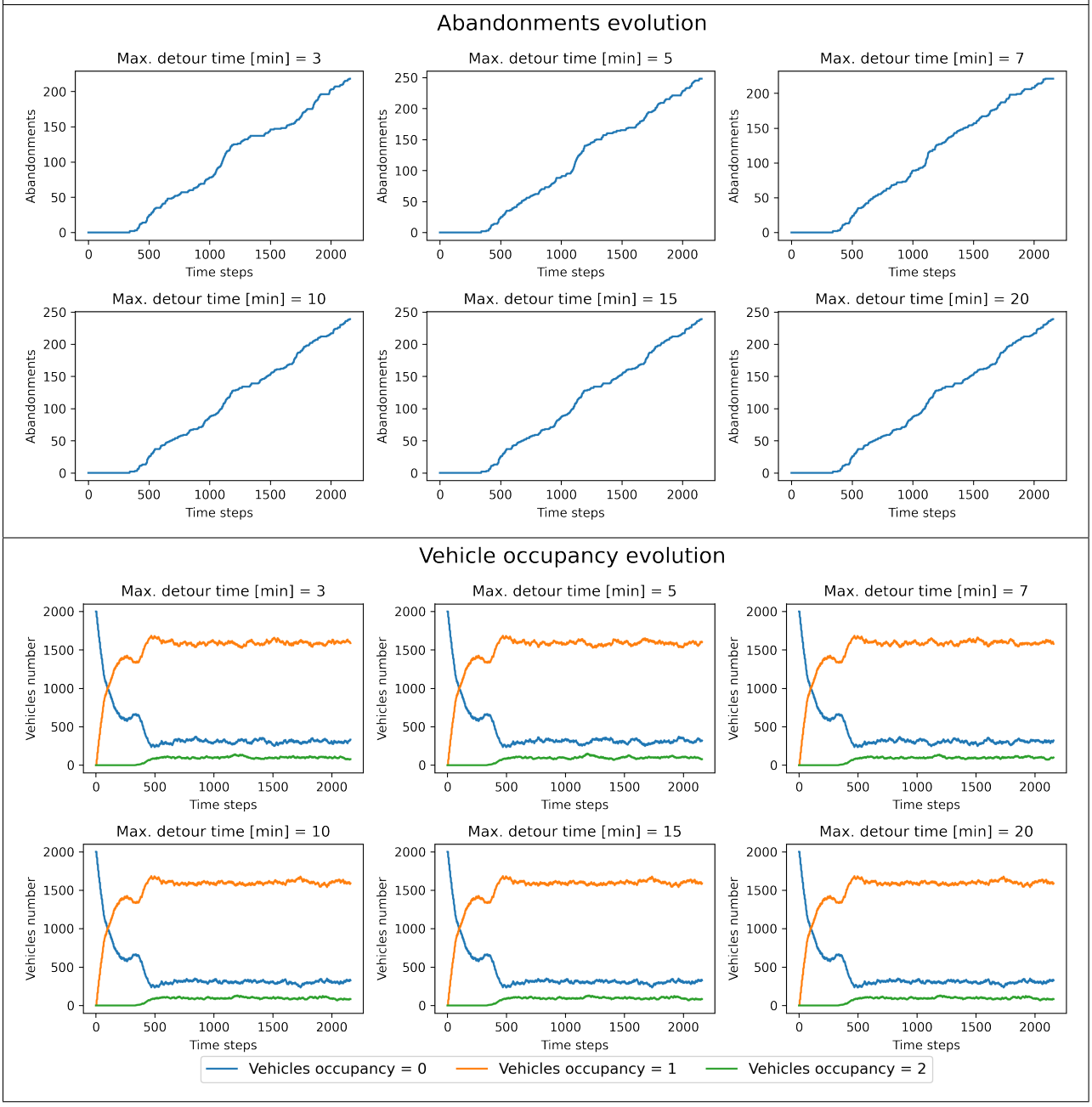

Waiting times distribution
Max. detour time [min] = 3
Max. detour time [min] = 5
Max. detour time [min] = 7
Max. detour time [min] = 10
Max. detour time [min] = 15
Max. detour time [min] = 20
Waiting times [min]
Requests
Mean
Waiting times
Detour times distribution
Detour times [min]
Detour times

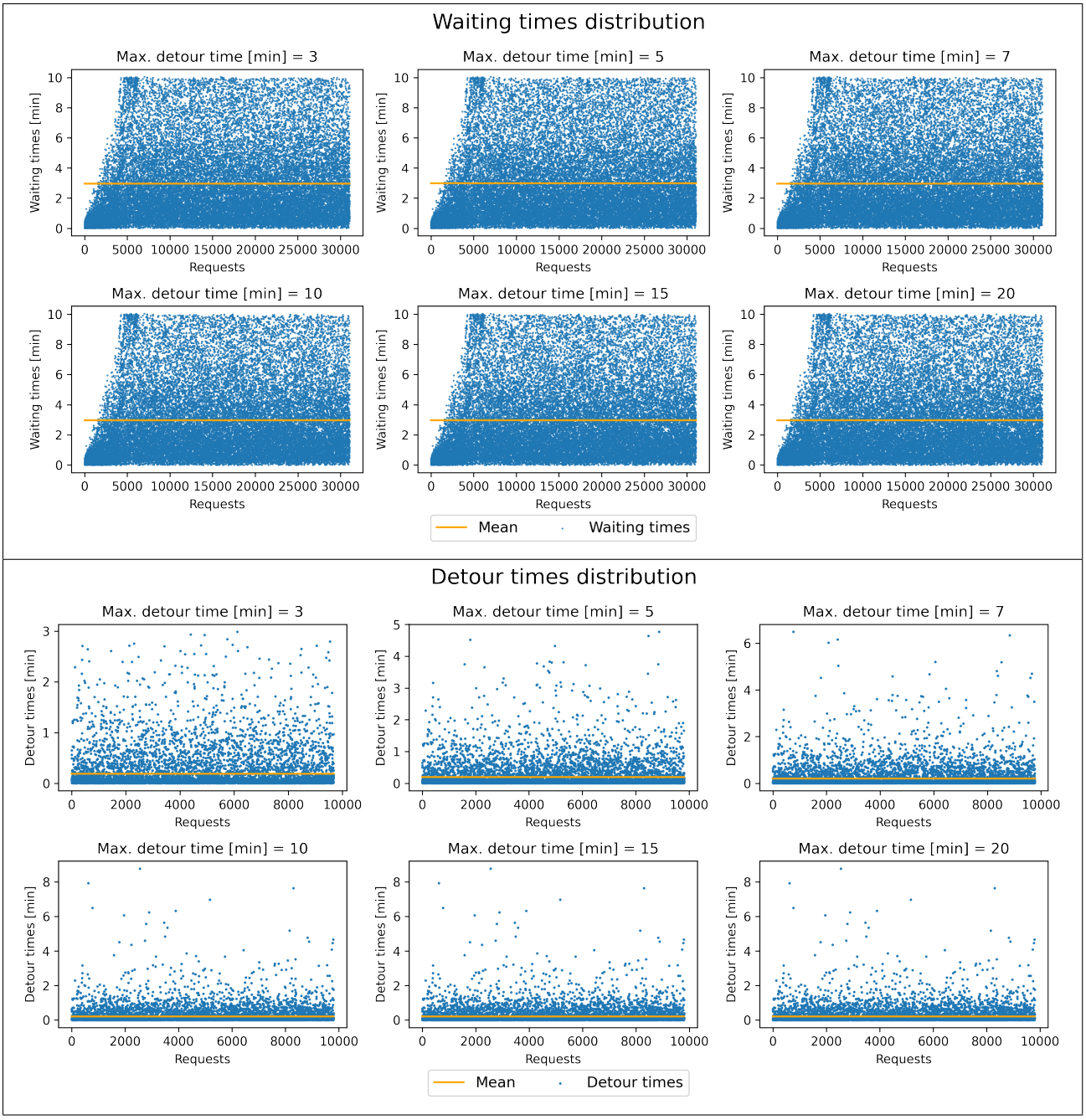
B.6 Case study 4.2: Detour time influence, First model, Greater waiting time

Table 18: Complete visualization of simulations data
Case study 4.2: Detour time influence, First model, Greater waiting time

Objective function: Minimize the detour time
Fleet size: 2000
Number of requests: 31500

Willingness to share: 80%
Maximum waiting time: 10 min



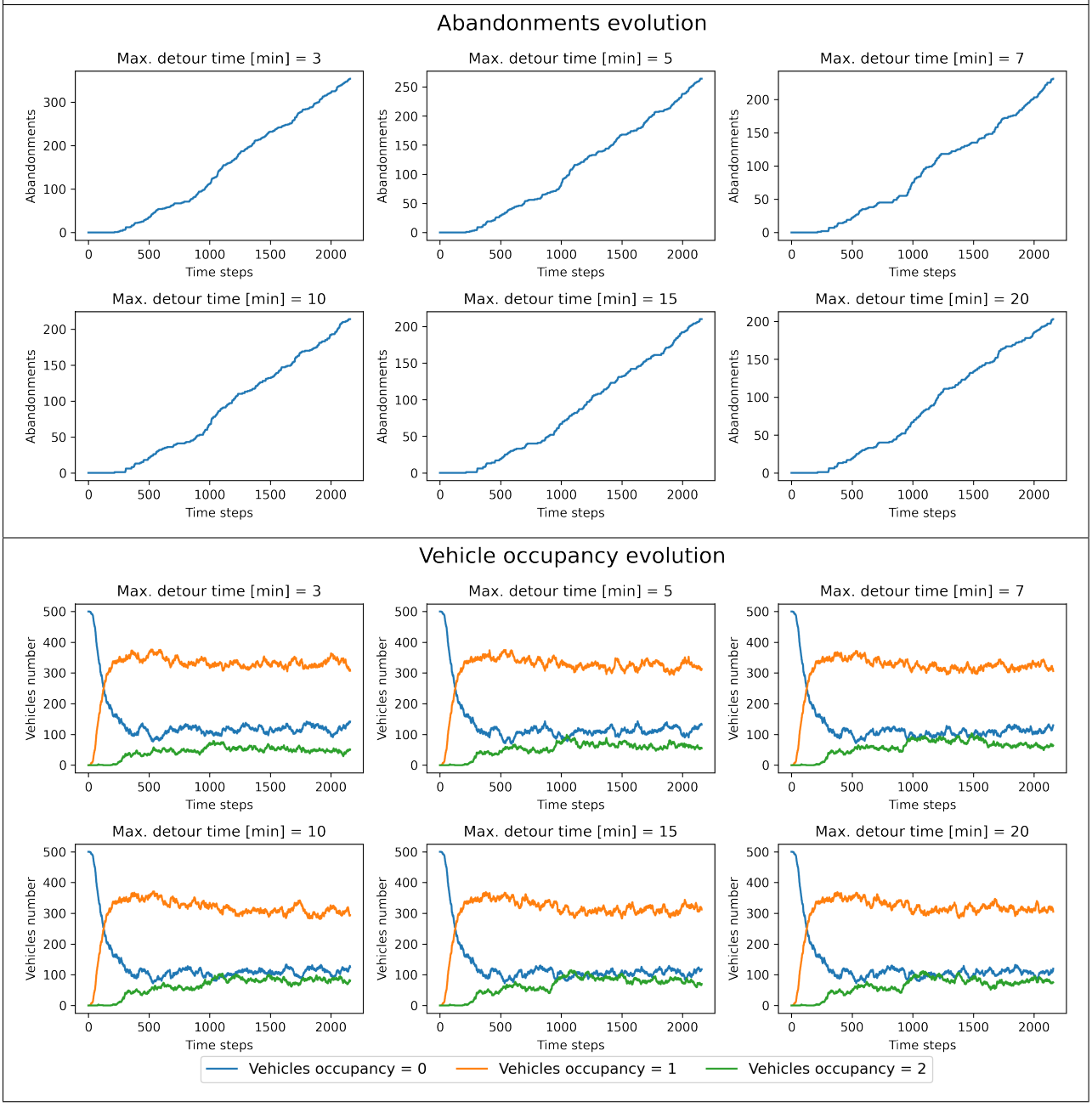

Waiting times distribution
Max. detour time [min] = 3
Max. detour time [min] = 5
Max. detour time [min] = 7
Max. detour time [min] = 10
Max. detour time [min] = 15
Max. detour time [min] = 20
Waiting times [min]
Requests
Mean
Waiting times
Detour times distribution
Detour times [min]
Detour times

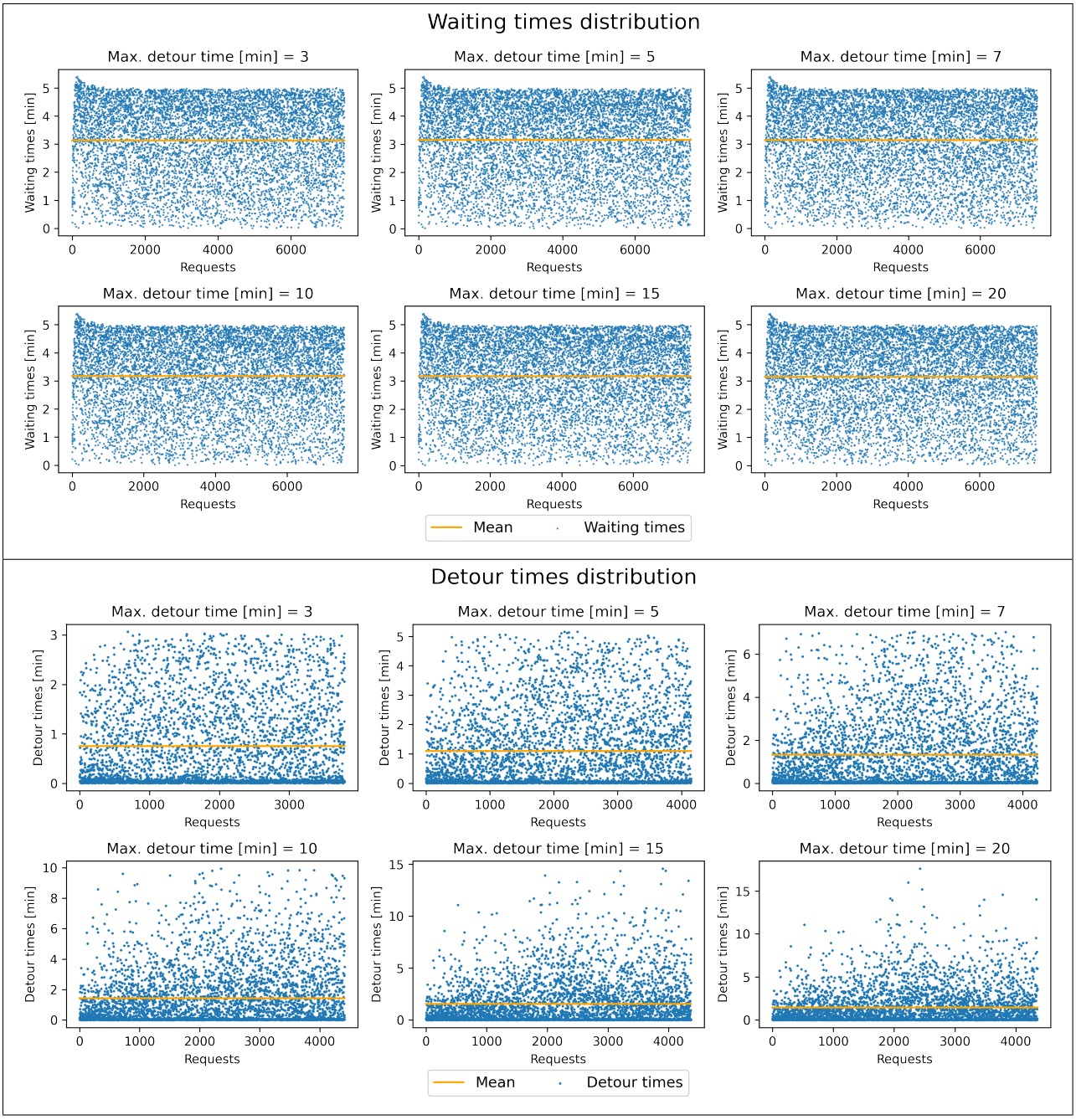
B.7 Case study 5: Detour time influence, Second model, Reduced scenario

Table 19: Complete visualization of simulations data
Case study 5: Detour time influence, Second model, Reduced scenario

Objective function: Minimize the detour time
Fleet size: 500
Number of requests: 7850

Willingness to share: 80%
Maximum waiting time: 5 min




Waiting times distribution
Max. detour time [min] = 3
Max. detour time [min] = 5
Max. detour time [min] = 7
Max. detour time [min] = 10
Max. detour time [min] = 15
Max. detour time [min] = 20
Waiting times [min]
Requests
Mean
Waiting times
Detour times distribution
Detour times [min]
Detour times