Schedule Rating Method based on a Fragmentation Criterion

Schedule Optimization in Corporate Carsharing of Electric Vehicles

Falko Koetter, Julien Ostermann, Daniel Valerian Jecan
Fraunhofer IAO, Germany
email: firstname.lastname@iao.fraunhofer.de

***Abstract*— Sharing electric vehicles among companies and users can increase efficiency of corporate car fleets. Due to charging times and limited ranges, high quality scheduling is necessary to achieve a high degree of utilization and in turn economic efficiency. Schedule optimization can help improving a schedule regarding utilization and cost, but does not take into account possible future bookings. If a schedule is fragmented, no future bookings can fit in. In this work, we introduce a measurement for the fragmentation of schedules and use it to minimize fragmentation and in turn maximize future booking potential. We evaluate the approach using synthetic and real-life trials, showing that fragmentation reduction can lead to increased utilization in electric vehicle fleets.**

Keywords-corporate carsharing; fragmentation; schedule optimization; schedule management; electric vehicle fleets.

I. INTRODUCTION

Even though electric vehicles (EVs) have been available for some time, recent developments like the rising costs of fossil fuels, technological progress in vehicle technology and availability of regenerative energy make the economic use of EVs in company car fleets more and more feasible. The market potential is noticeable, estimated at about 1 million EVs until 2020 just for Germany and in particular for company car fleets at 30% of newly bought cars [5]. The drawbacks of EVs, such as the high fixed cost for vehicles and charging infrastructures, have to be overcome by employing high utilization. The Shared E-Fleet project [15] researches the economic operation of shared car fleets, making it possible even for small and medium enterprises, which could not economically operate a fleet on their own.

Charging times and limited range are special challenges in the context of using EVs for corporate car sharing, which need to be taken into consideration when scheduling vehicles and business trips. When operating at capacity, the consequences of minor disruptions like delays or lost battery level can affect the future schedule, as trips might not be started on time or without sufficient charge to reach the destination.

To reach high utilization of car fleets while minimizing cost or ecological impact and compensating these disruptions, continuous optimization of the schedule is used.

While regular optimization techniques can be used to optimize a fleet schedule, e.g., regarding minimization of emissions, possible future states are not taken into account. These include the potential for future bookings, for which suitable timeslots in the schedule need to be available. If the schedule is fragmented, i.e., trips are distributed uniformly among vehicles, there may be no timeslot for a future booking, even though in aggregate, enough unused time on vehicles is available.

In this work, we describe a rating method for vehicle schedules based on a fragmentation criterion and use it to provide optimized schedules with minimum fragmentation, thus ensuring maximum opportunities for future bookings.

The contribution of this work is as follows: We introduce a fragmentation ratio, a measurement for the fragmentation of schedules and show how it is used as part of a closed loop optimization system.

The remainder of this work is structured as follows: Section II describes related work. Section III describes the schedule optimization problem. Section IV defines schedule fragmentation in relation to concepts from memory management. Section V presents the fragmentation rating and the algorithm for computing it. Section VI describes an evaluation scenario from the Shared E-Fleet project. Finally, Section VII gives the conclusion and outlines future work.

II. RELATED WORK

Optimization of vehicle fleets encompasses multiple domains like routing, charging and scheduling. Different schedule optimization algorithms exist, but do not match the corporate car sharing scenario [13].

In our case, trips have a fixed start and end time and they need to be distributed among a set of vehicles with the purpose of maximizing utilization and minimizing costs. The solution for the optimization of scheduling and charging presented in [3] proves that the routing problem is NP-complete. While this approach provides a solution for charging and schedule optimization, the reaction to disruptions and schedule fragmentation are not covered.

Several methods for a feasible solution to NP-hard schedule optimization problems can be found in [4], but they do not refer specifically to the shared fleet scenario.

Schedule defragmentation is a problem in other fields of application as well. For example, [8] introduces an algorithm for scheduling lectures to classrooms by moving chunks from the least to the most occupied rooms. The implementation is described in [11], defragmenting a classroom schedule. However, compared to our work, defragmentation is the primary and not a secondary optimization goal. The schedule of healthcare professionals

 ISBN: 978-1-61208-434-3

can be defragmented to provide maximum potential for additional appointments by preserving large chunks of free time, similar to how this work aims at maximizing potential for additional bookings [9]. In a similar manner, [10] provides a defragmentation algorithm to minimize patient waiting times. In comparison, our scenario has additional constraints such as vehicle range.

While the general concept can be applied to schedule defragmentation, different consistency conditions specific to EV use are not covered. Additionally, these static scheduling problems do not cover frequent re-optimization, as needs to be applied in a car fleet schedule.

III. Schedule Optimization

In a shared car fleet, users book trips for a predefined time and destination, starting and ending at a car fleet station, where the vehicle can be charged. A schedule decides which trips are performed by which vehicles. As vehicles differ in range, cost per kilometer and emissions, there is potential for optimization. Another goal of optimization is enabling a high degree of utilization, which includes leaving a maximum potential for future trips. A user books a trip in the fleet, a specific vehicle is only assigned shortly before the trip starts, enabling trips to be moved by optimization beforehand.

Two different algorithms are used for schedule optimization. An alternative search algorithm searches a feasible fit for a trip in the schedule. This algorithm is used to check availability during booking and needs to provide immediate answers. The other algorithm, periodic optimization, performs a full optimization of the schedule, potentially redistributing any future trips [12].

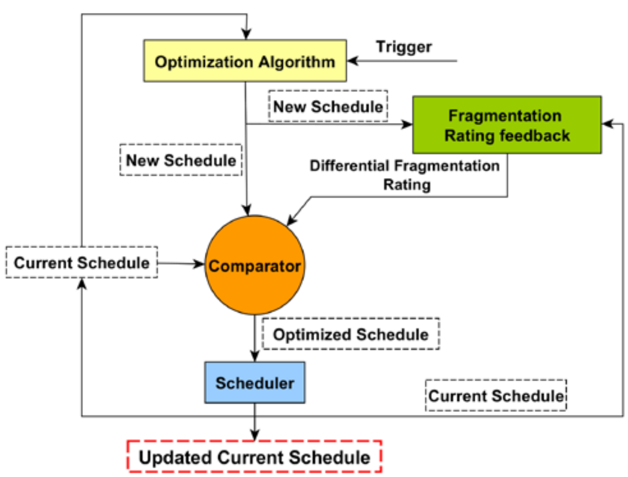

Figure 1 Closed loop schedule optimization scheme.

For the periodic optimization, we take into account two main components of schedule optimization, namely the algorithm used for optimizing the scheduling procedure and the feedback mechanism that has the purpose to indicate the degree of optimization or in other words the optimization rating of a schedule. As in the case of a typical closed loop control system [6], the optimization algorithm would be used as an actuator, changing the state of the schedule and the optimization rating as the feedback responsible for providing the necessary insight for improving the scheduling algorithm. This article focuses on one proposed feedback method, namely the fragmentation of a schedule. Fragmentation reduces the possible utilization of vehicles, as fragmented schedules offer shorter slots for future trips during alternative search. Therefore low fragmentation is a prerequisite for high utilization, so it is an important aspect to be evaluated and optimized.

Figure 1 shows the closed loop system used in the optimization process. In a closed loop configuration, the components involved are connected in a cyclic manner, influencing each other. Therefore, a holistic approach towards the analysis of the system is necessary [2].

The closed loop has four main components:

Optimization algorithm – used for scheduling based on an algorithm that is meant to optimize specific goals

Scheduler – deploys the current schedule

Fragmentation Rating – the feedback mechanism used for comparing the old and the new schedule

Comparator – based on the feedback received from the Fragmentation Rating component, delivers the optimum schedule

The optimization process starts with the current schedule, containing all current and future trips assigned temporally or permanently to vehicles. This schedule is given to the optimization algorithm, which after being triggered (periodically, by an event, e.g., a delay, or by an administrator), delivers a new schedule, using a greedy optimization algorithm, which is described in [12]. The new schedule and the current schedule are next given to the fragmentation rating component, where the computation of the rating takes place. The fragmentation rating is implemented as a maximization function (the higher the rating, the lower the fragmentation ratio) and the output is given in the form of a differential rating between the new and the current schedule. Using the provided rating, the comparator decides between the current and the new schedule and passes on the optimal schedule to the scheduler. At this point, the current schedule is replaced with the new optimized schedule (which might be the same one, if the new schedule does not receive a better rating, as the optimization algorithm is not fragmentation aware at the stage covered by the article). The optimization process is repeated every time the optimization algorithm is triggered.

IV. Fragmentation of a Schedule

While the initial schedule optimization only took into account a goal function (for either cost or emission minimization), during model trials we noticed that high utilization was hard to be achieved, as end-users tended to book trips which could not be scheduled to any vehicle, though globally enough free capacity was available. We noticed the problem occurred when trips were evenly distributed between vehicles, providing many opportunities for a future trip to overlap existing trips, therefore becoming impossible to book without re-optimization.

Thus, we added a soft goal to our approach in the form of the fragmentation of a schedule. Borrowing from operating systems memory fragmentation, we applied this concept to the area of scheduling trips among cars. In the context of computer memory, fragmentation usually refers to storage space that is not used efficiently, meaning we are dealing with reduced capacity and/or performance [7]. This can lead to different undesirable situations, one of which is not being able to allocate memory in certain areas of the storage space.

We will now introduce the basic principles related to memory management and fragmentation in the area of computer systems and we will transfer them to our proposed concept of the fragmentation of a schedule.

The dynamic memory is designed as a buffer between the physical big storage devices (e.g., hard-disks) and the small size but high speed memory of the processor, i.e., the cache. Basically any application needs to allocate memory from dynamic memory in order to run. Blocks of memory are allocated in chunks and whenever the application does not need such a chunk anymore, that particular space can be freed. However, because the size of these chunks is variable, after a while, depending on the actual memory usage of the application, the number and the size of long continuous regions of memory space could reduce significantly [1].

As an analogy, in order for a trip to take place, it needs to be booked in the schedule, which means we need to allocate that trip a time slot on a single vehicle inside the schedule (the schedule corresponds to the dynamic memory).

In the case of memory management, we work with allocating space chunks, while in the case of schedule management we are dealing with allocating time chunks. However, there is an important difference to be mentioned. The time to be allocated in the schedule is replicated among the vehicles available in the fleet, i.e., every vehicle has its own timeline, parallel to the others, so a specific time interval can be booked to any available vehicle. This is not true in the case of memory management, where every chunk of storage space is unique and it cannot be allocated to two or more processes at the same time. When a trip is cancelled for whatever reason, that specific time interval can be freed, as in the case of memory allocation. To continue the analogy, the time chunks allocated for trips are variable in size and when trips get cancelled, some variably sized slots are left unused, which can make booking longer trips harder. On the other hand, memory management does not depend directly on time, so the state of the memory could stay the same even if time passes. That is not the case with schedule management, because we are actually dealing with allocating time and the mere passing of it determines the state of the schedule to change along with it. For example, currently running trips cannot be reallocated and late trips may extend their allocated chunks, necessitating future changes.

The two main types of fragmentation related to memory are internal and external. Internal fragmentation usually occurs when the allocated memory (addressed in fixed size partitions) does not match the requested memory and the remaining unused part is wasted, as it cannot be allocated to other processes. External fragmentation however is generated when variably sized partitions are used, but as soon as some segments of memory are freed, some unusable small gaps can appear between the occupied blocks of memory. If we analyze the schedule management situation, we can only have external fragmentation, because when we allocate time for a trip, we can allocate the exact interval needed, if the vehicle is available for booking, so we are not bound to some fixed sized intervals, as in the case of memory fragmentation.

Therefore, the fragmentation for a schedule is computed using the chunks of time which are not used and their associated properties (duration and time interval of the day).

V. FRAGMENTATION COMPUTATION

The initial configuration is that every vehicle is fully charged and ready to be booked.

There are a couple of terms to be defined:

Window – The time interval (in the future) the fragmentation is calculated for (a default one day window starts at 7 AM and ends the next day at 6:59AM)

Fragment – a chunk of time between two bookings which is not used (as seen in Figure 2)

Maximum Fragmentation Ratio – the maximum rating you could get within a given window when the vehicles are fully charged without any booking.

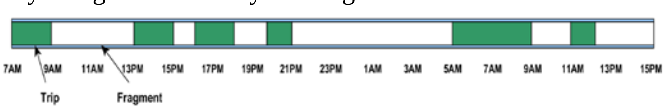

Figure 2 Example of a schedule.

The computation of fragmentation is meant to be used for optimizing the fragmentation goal, which means reducing the actual fragmentation of the schedule. By taking a snapshot of the schedule at different moments in time, we can analyze its structure and extract a list of fragments for each vehicle. The span of a fragment is between the end of the previous trip and the start of the next one, all within the given window for analysis. After getting the list of all fragments in the schedule, we compute the fragmentation ratio using two general, relevant properties of the fragments: duration and the time interval in which the trip is located in.

We incorporated these properties in the fragmentation ratio formula (1), represented by weights, as follows:

$$\mathrm{FR} = \sum_{1}^{n}(\mathrm{duration}(\mathrm{frag_i}) * \mathrm{w1}(\mathrm{frag_i}) * \mathrm{w2}(\mathrm{frag_i})) \quad (1)$$

where n is the number of fragments, w1 is the duration weight (computed in (2)) and w2 is the time interval weight (computed using (3)).

The fragments which have a longer duration have a higher weight w1, because longer and shorter trips can all fit in an extended time interval (see Figure 4). However, shorter

fragments have a lower usability (trips of less than 20 min are less likely). Regarding w2, the weight for the time interval, trips during regular business hours (7AM till 17PM), have a higher weight than the trips outside of this interval (see Figure 5), taking into consideration that the main target of the Shared-E-Fleet project is the business sector [3].
The functions used for the weight computation are synthetic functions, but using the real data coming from statistics regarding usage of the electronic fleet, new and more relevant weighting functions can be computed. The initial formula (with x as the duration of the trip) used is:

$$\mathrm{DurationWeight} = \frac{0.79}{\mathrm{fw}*60}\mathrm{x} + 0.01 - \frac{0.79}{\mathrm{fw}*60} \quad (2)$$

The *fw* parameter stands for fragmentation window, which is always a factor of 1440, i.e., how many minutes there are in one day. A window starts at 7AM in the morning (0) and it ends at 6.59AM the next day (1440). If there is one trip today at 1PM and 1 trip tomorrow at 10AM, the fragmentation window is 2880 minutes (48H) (see Figure 3).

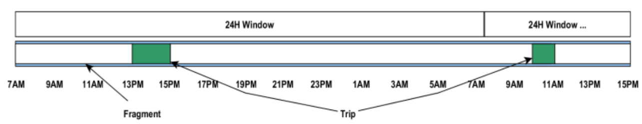

Figure 3 Fragmentation window.

An arbitrary maximum weight of 0.79 is given to a fragment containing the whole analyzed window. We intend to adjust the weight functions according to the real life usage profile obtained from the model trials. The subtraction in the formula is used for keeping the weight normalized.

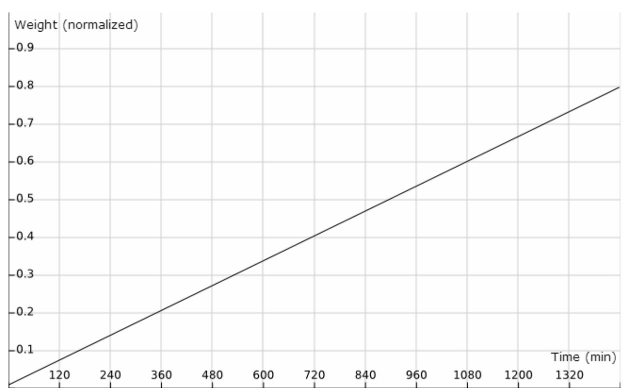

Figure 4 Weight function for the duration of the fragment.

$$\mathrm{TimeIntervalWeight} = \left|\frac{\mathrm{w_{end}} - \mathrm{w_{start}}}{2}\right| \quad (3)$$

$$\mathrm{w_{start}} = \frac{-0.8}{\mathrm{fw}*60}\mathrm{startInterval} + 0.9 \quad (4)$$

$$\mathrm{w_{end}} = \frac{-0.8}{\mathrm{fw}*60}\mathrm{endInterval} + 0.9 \quad (5)$$

The startInterval and endInterval variables are normalized for a one day duration, so values are between 0 and 1440 by using the fragmentation window defined previously. The parameters w_{start} and w_{end} are computed using formulae (4) and (5) and used for determining the time interval weight.

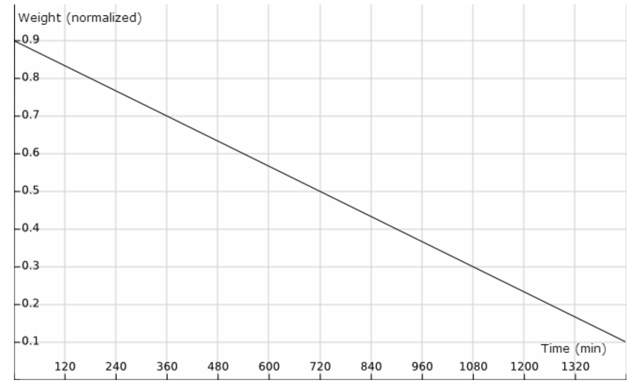

Figure 5 Weight function for the time interval of the fragment.

We plan to further adjust these synthetic functions by using statistical data from model trials.

Using those weights, the fragmentation ratio function is computed as a value between zero (meaning there is no fragment in the schedule, so no free slots) and maximum fragmentation ratio (meaning all vehicle schedules are empty and ready to be booked). Note that due to the duration weight, longer continuous fragments provide a higher fragmentation rating, and due to the time interval weight, fragments during business hours provide a higher fragmentation rating. Thus, schedules which provide large fragments of free time slots during business hours for future bookings are selected after optimization.

VI. EVALUATION

Within the Shared E-Fleet project, this method is integrated into an optimization system, containing algorithms for booking, alternative search, partial and full optimization implemented as a Java prototype [12]. This optimization system is part of the larger Shared E-Fleet architecture, providing a EV fleet management solution[14].

We evaluated the fragmentation ratio approach, both using synthetic tests and by application during three long-term model trials in German industrial parks.

The partial and full optimizations are implemented in the optimizer component. The state of an EV is updated if real-time notifications such as delays, malfunctions or returns are received. The optimization algorithm implemented is a greedy algorithm using backtracking [12], minimizing total emissions and compensating disruptions like delays, which was run as is during the initial phase model trials. As a whole, the optimization scales linearly with the number of trips, allowing use of large schedules.

Before deployment in the model trials, we used synthetic test data with randomized bookings to determine the suitability of the approach.

In the following, we show a simplified example using synthetic test data.

We used the fragmentation ratio computation in order to check the improvement of the fragmentation goal between two snapshots of a schedule, one with three trips, spanning over a one day window and the second one being the optimized version of the first one, using the optimization algorithm.

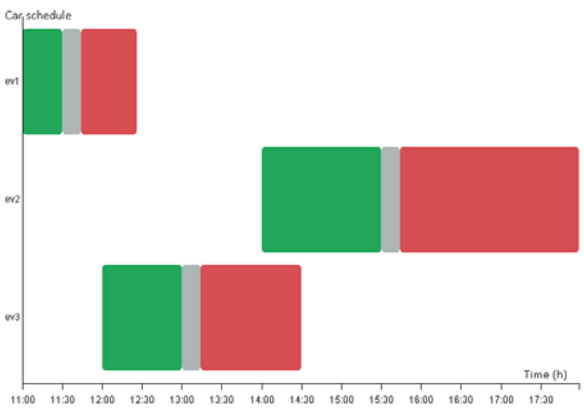

Figure 6 – Original version of the schedule.

A fleet of three vehicles is used in the example. As seen in Figure 6, in the original schedule we have a 30 min trip using ev1, starting at 11.00, a 1h 30min trip using ev2, starting at 14.00 and a 1h trip using ev3, starting at 12.00.

Figure 7 shows the optimized schedule (optimization was carried out at 10.38am). It can be observed that instead of sparsely using three vehicles, two of them were completely freed and all three trips are booked on one vehicle, therefore increasing utilization and decreasing fragmentation. Note that charging can be deferred as long as the remaining charge is sufficient for the next trip.

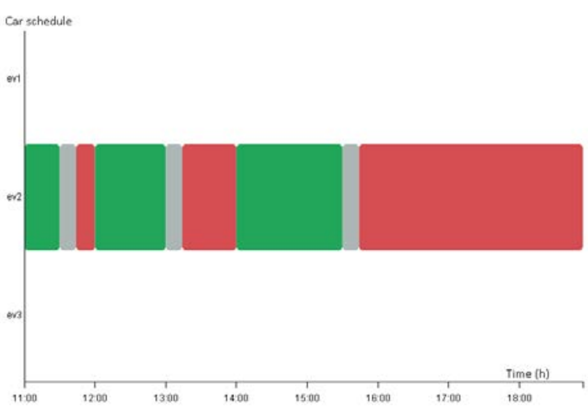

Figure 7 – Optimized version of the schedule.

The gray area after each trip represents a buffer which is intended to account for user behavior (minor delays, unloading the vehicle, connect to charge station, etc.). The red areas indicate charging times.

After running the analysis, we found 6 fragments for the old schedule (Figure 8) and 4 fragments for the new schedule (Figure 9) after optimization. The value of the fragmentation rating for the old schedule was approximately 794 and for the new one is around 1051, so the fragmentation was successfully reduced. The new schedule has two fragments of 1440 minutes, which means that two vehicles are completely free, as seen in Figure 7.

```
fragment start: Wed Jul 15 10:38:46 CEST 2015
fragment end: Wed Jul 15 11:00:00 CEST 2015
fragment duration: 21
---------------------------------------------
fragment start: Wed Jul 15 12:26:03 CEST 2015
fragment end: Thu Jul 16 10:38:46 CEST 2015
fragment duration: 1332
---------------------------------------------
fragment start: Wed Jul 15 10:38:46 CEST 2015
fragment end: Wed Jul 15 14:00:00 CEST 2015
fragment duration: 201
---------------------------------------------
fragment start: Wed Jul 15 17:58:36 CEST 2015
fragment end: Thu Jul 16 10:38:46 CEST 2015
fragment duration: 1000
---------------------------------------------
fragment start: Wed Jul 15 10:38:46 CEST 2015
fragment end: Wed Jul 15 12:00:00 CEST 2015
fragment duration: 81
---------------------------------------------
fragment start: Wed Jul 15 14:29:43 CEST 2015
fragment end: Thu Jul 16 10:38:46 CEST 2015
fragment duration: 1209
---------------------------------------------
Fragmentation of the schedule: 794.6474138932936
```

Figure 8 – Fragmentation rating of the old schedule.

```
fragment start: Wed Jul 15 10:38:46 CEST 2015
fragment end: Thu Jul 16 10:38:46 CEST 2015
fragment duration: 1440
---------------------------------------------
fragment start: Wed Jul 15 10:38:46 CEST 2015
fragment end: Wed Jul 15 11:00:00 CEST 2015
fragment duration: 21
---------------------------------------------
fragment start: Wed Jul 15 18:54:00 CEST 2015
fragment end: Thu Jul 16 10:38:46 CEST 2015
fragment duration: 944
---------------------------------------------
fragment start: Wed Jul 15 10:38:46 CEST 2015
fragment end: Thu Jul 16 10:38:46 CEST 2015
fragment duration: 1440
---------------------------------------------
Fragmentation of the schedule: 1051.625424654092
```

Figure 9 – Fragmentation rating of the new schedule.

If the new schedule has a higher fragmentation rating, the new schedule actually has lower fragmentation, so it is better than the old one and the optimization was successful.

The model trials were implemented in three technology parks over a time period of a year, in the context of real small and medium sized enterprises. While the cars are providing a range of over 100 kilometers, the average booking contained a trip with less than 50 kilometers and 3 hours in length, facilitating optimization. Figure 10 shows the distribution of trips throughout the day, indicating predominant use during business hours, which we aimed to accommodate using the fragmentation ratio. Figure 11 shows the fleet utilization in one model trial, as well as the success percentage, indicating how many booking requests could be fulfilled. Due to novelty value, the demand reached its peaks in the first month. However, initial hardware problems in addition to suboptimal fragmentation created a perception of low availability. Bookings were rejected even though vehicles could be seen as available. To increase utilization, the fragmentation rating comparison was added to optimization at the end of August. The improvement in scheduling allowed on average 21.6% increase in utilization, with a higher success ratio. In December, utilization was lower due to weather conditions as well as holidays.

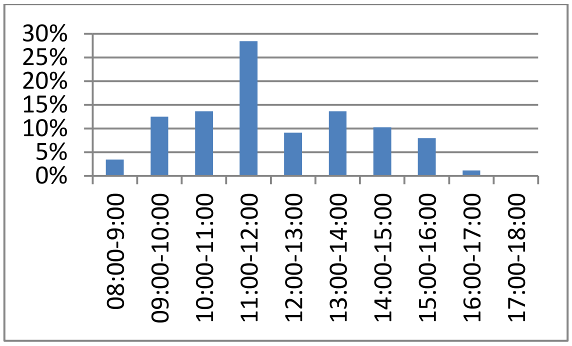

Figure 10 – Distribution of trips during the day.

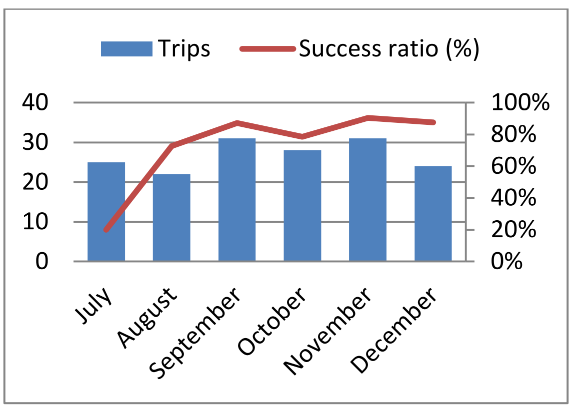

Figure 11 – Trips and success ratio.

Other model trials lack comparison values as they were started with fragmentation rating in place.

VII. CONCLUSION

Optimizing vehicle schedules is a necessity for economic operation of shared fleets. In this work, we have introduced the fragmentation ratio, a rating for fragmentation of vehicle schedules, based on concepts from memory management in operating systems. Fragmentation rating complements schedule optimization to further increase utilization by prioritizing large time slots for future booking.

We evaluated the concepts by using synthetic data, as well as introducing them in a running model trial, showing notable improvements in utilization.

In future work, we will integrate fragmentation rating with other soft goals related to charging times and energy management (e.g., optimal utilization of photovoltaics for charging). Currently, the rating is applied after optimization, stopping fragmented schedules from replacing less fragmented schedules. In future work, we would like to adapt the optimization algorithm with fragmentation awareness in addition to other optimization goals. Additionally, we would like to evaluate weighing individual vehicle utilization using the fragmentation ratio, which could provide benefits in larger fleets.

ACKNOWLEDGEMENT

This research has been supported by the IKT II program in the Shared E-Fleet project. They are funded by the German Federal Ministry of Economics and Technology under the grant number 01ME12105. The responsibility for this publication lies with the authors.

REFERENCES

[1] A. S. Tanenbaum, “Modern Operating Systems (3rd Edition),” Prentice Hal, ISBN-13: 978-0136006633, 2007.

[2] K. J. Åström and R. M. Murray, “Feedback Systems: An Introduction for Scientists and Engineers,” Princeton University Press, ISBN 0-691-13576-2, 2008.

[3] O. Sassi and A. Oulamara, “Joint scheduling and optimal charging of electric vehicles problem,” Computation Science and its Applications-ICCSA 2014, Springer, 2014, pp. 76-91.

[4] V. Pillac, M. Gendreau, C. Gueret, and A. L. Medaglia, “A review of dynamic vehicle routing problems,” European Journal of Operational Research 225(1), 2013, pp. 1-11.

[5] P. Plötz, T. Gnann, A. Kühn, and M. Wietschel, “Market introduction scenarios for e-vehicles,“ Karlsruhe: Fraunhofer-Institut für System- und Innovationsforschung ISI, 2013, (in German).

[6] R. C. Dorf and R. H. Bishop, "Modern control systems," Pearson (Addison-Wesley), 1998.

[7] M. S. Johnstone and P. R. Wilson, "The memory fragmentation problem: solved?," ACM SIGPLAN Notices, Vol. 34, No. 3, ACM, 1998, pp. 26-36.

[8] E. Humo and Z. Vejzović, "A Heuristic Approach To Classroom-Period Schedule Optimization," Advanced Engineering, 1st year (Vol.2), 2007, pp. 165-172.

[9] J. Lian, et al., "Clinical appointment process: improvement through schedule defragmentation," Engineering in Medicine and Biology Magazine, IEEE 29.2 (2010), pp. 127-134.

[10] C. E. Nelson, et al., "Visual scheduling to improve workflow and throughput in the University of Virginia Health System Pediatric Specialty Clinic," Systems and Information Engineering Design Symposium (SIEDS), IEEE, 2011.

[11] Z. Vejzovic and E. Humo, "A software solution for a mathematical model of classroom-period schedule defragmentation," EUROCON, 2007, The International Conference on: Computer as a Tool, IEEE, 2007.

[12] F. Koetter, „Dynamic schedule optimization in shared EV fleets,“ http://www.shared-e-fleet.de/images/Dynamische_Einsatzoptimierung_von_gemeinsam_genutzten_ Elektrofahrzeugflotten.pdf, 2015, (In German).

[13] M. Bielli, A. Bielli, and R. Rossi, "Trends in models and algorithms for fleet management," Procedia-Social and Behavioral Sciences 20, 2011, pp. 4-18.

[14] J. Ostermann, T. Renner, F. Koetter, and S. Hudert, "Leveraging Electric Cross-Company Car Fleets through Cloud Service Chains: The Shared E-Fleet Architecture," Global Conference (SRII), 2014 Annual SRII, IEEE, 2014.

[15] Shared E-Fleet. [Online]. Available from: http://www.shared-e-fleet.de/ 2015.07.14