3rd International Conference on Connected Vehicles & Expo (ICCVE 2014), November 3-7, 2014, Vienna, Austria

Automated Planning of Charge Processes for Privately Owned Electric Vehicles

Tillmann Nett, Jörn Schneider
Trier University of Applied Sciences
Trier, Germany
{T.Nett, J.Schneider}@Hochschule-Trier.de

***Abstract*—Following the vision of decentrally generated and locally out-balanced renewable electric energy the research project econnect Germany investigates the end user acceptance of Vehicle2Grid and Grid2Vehicle applications. The algorithm used to compute the charging/discharging of individual batteries in electric vehicles (EV) will be an essential factor for end user acceptance. This is because it controls the achievable driving range and the cost savings for the individual user of an electric vehicle. Additionally, the algorithm has to allow for an effective control by the energy provider to compensate for the natural fluctuations of wind and solar energy in the region. As the final authority to decide about charging/discharging needs to be the battery management system of the EV, it is a natural choice to embed the algorithm in an electronic control unit of the car. This paper presents an algorithm designed to meet these requirements and demonstrates that it scales down to the low computing power of embedded automotive systems.**

I. Introduction

Two important trends are currently underway to reduce the negative impact our society has on the environment. First, electric vehicles are becoming more and more common, reducing fossil fuel consumption and carbon dioxide emissions compared to conventional vehicles [1]. Second, renewable energy sources are used increasingly, improving the sustainability of energy production [2]. These two trends come with complementary challenges and opportunities, and offer significant synergies [3]. The batteries of EVs can be utilized to compensate for fluctuations of renewable energy (e.g. wind and solar energy) availability in a region by Grid2Vehicle (G2V) and Vehicle2Grid (V2G) technology [3]. For these future scenarios to work, a high percentage of future EV owners has to support the usage of their batteries and accept the consequences. The algorithms controlling the G2V and V2G functionality will be an essential factor for end user acceptance. This is because their outcome controls the achievable driving range and the cost savings for the individual user of an EV. Additionally, the algorithms have to allow for an effective control by the energy provider to compensate for the fluctuations of wind and solar energy in the region. Furthermore, the final authority to decide about charging/discharging is the battery management system of the EV itself. Therefore, it is a natural choice to embed the algorithm in an electronic control unit of the car. For a field trial on the end user acceptance of G2V and V2G in the econnect Germany project five series EVs have been upgraded with an in-car computer that computes the charging/discharging plan for the individual EV based on a price signal. In this paper the algorithm computing the G2V and V2G plans is presented. The algorithm is tailored to allow for a high user acceptance and reflect the situation of municipal energy providers in a deregulated market.

This work is partly funded by the German Federal Ministry for Economic Affairs and Energy under grant number 01ME12043 within the econnect Germany project.

The rest of the paper is structured as follows. In Section II, we will give an outline of the problem statement that was used as a basis. Section III gives a short outline of related work. The main algorithm of the system is described in Section IV. Thereafter the results of an experimental evaluation regarding the scalability of the algorithms computation time to embedded systems with less computing power is presented in Section V. Concluding remarks and an outlook on future research are given in the last Section.

II. Problem Statement

This section provides an informal description of the use case and usage of the systems and algorithms designed for planning of charge processes. A formal specification of the main algorithm is given in Section IV-B.

To achieve the desired synergies between EVs and renewable energies, a high market penetration and usage of G2V/V2G technology is required. Hence, it is necessary to consider economic and technical factors, as well as user acceptance. The system should be kept as simple as possible at first. Therefore, the following basic decisions were taken for the field trial on end user acceptance: (1) Centralized charging puts the vehicle under external control. As drivers may be reluctant to give up control of their vehicle, this may reduce user acceptance. Thus, a decentralized method is investigated in which each vehicle produces an individual charge plan, which is carried out by the energy provider. (2) A final solution should be simple and explainable in simple terms like "The vehicle buys energy when it is cheap and sells energy when it is expensive". (3) The interface is to be kept simple and be developed in a user centered development process. Planned trips should not be predicted automatically, since mispredictions may render the car unusable when needed and hence reduce user acceptance. (4) Energy should be traded with a single energy provider to eliminate influence from different market reputations or by confusing users with complicated market constructs. (5) Dynamic price information set by the energy provider shall be used as lean and transparent interface to control the charging/discharging process of individually owned EVs. The energy provider can set the prices in accordance with the predicted and actual energy availability of renewable energy sources.

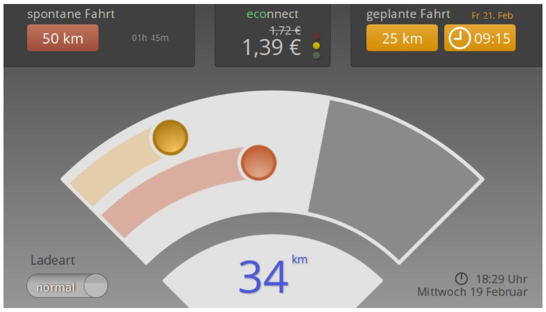

Figure 1. The user interface for planning of the charge process

The process of developing the human-machine interface in a user centered approach is described in [4]. A preliminary survey of possible users uncovered two main usage patterns: (1) Short unexpected rides and (2) longer rides at a planned point of time in the future. Examples for (1) are short spontaneous trips to a bar or the movies, but also unexpected emergencies such as a quick ride to the hospital. Examples for (2) may be daily commutes or longer rides on weekends. Hence, an interface is needed which allows configuration for both use cases. We decided on a simple interface in which the user can both set the distance that should always be provided for spontaneous trips as well as the desired distance and departure time for the next planned trip [4]. The owner of an electric vehicle plans his next charging period and leaves it to the in-car computer to calculate a cost-optimized charging/discharging cycle.

The algorithm to calculate the charge plans was also designed to handle both use cases. The charge process was divided in three phases. In the first phase the distance needed for spontaneous trips is charged as fast as possible without regards to the current price. This phase is referred to as the *spontaneous plan*. In the second phase the distance configured for the next planned trip is provided by the time of departure, with the goal of keeping the total cost as low as possible. This is called the *planned plan*. After the departure time has passed and before the owner has picked up his car, only optimization of the costs may be performed, but the available distance must not be discharged below the configured limit. This part of the plan is called the *refining plan*.

Furthermore, the algorithm is designed to take the realistic behavior of the battery into account. As the new state of charge after a period of charging or discharging depends on the applied voltage as well as the current state of charge of the battery (see e. g. [5]), it must be possible to model such a behavior. The system itself should be independent from the model. For now the model does not include other factors which may influence the change of available distance – such as temperature or humidity – because the car that was used does not provide sensor readings for these values.

As a last constraint the algorithm has to be efficient enough to run on an electronic control unit in a car. For the field trial the program runs on one partition of a mixed criticality platform within the car [6]. The main processor was a TI OMAP4460 running at 1GHz. Because of the partitioning only one core was available so no true parallelisation of the algorithm could be used for speedup.

III. Related Work

Research on V2G and optimal charging of electric vehicles is mainly focused on centralized methods, as these represent the traditional view point of large energy providers and grid operators [7]. Charge plans for centralized charging can be computed using common optimization techniques, e. g. convex optimization [8]. Research on decentralized planning focuses mainly on coordination of charge processes of multiple vehicles, e. g. using game theoretic approaches [9] or by considering vehicles as part of a multi-agent system [10].

In [10] vehicles are part of a multi-agent system which also includes aggregators and grid operators. Planning of the charge process is done by trading energy on different markets depending on the time until departure. Contracts and coalitions necessary for trading on these markets are formed directly between the agents. In [9] a game theoretic analysis of charging strategies based on real time price information is performed. To achieve a valley filling effect, a penalty for deviating from the population average is added to the prices. This eliminates oscillations of charge plans, which are caused when all vehicles shift charging to the cheapest moment and simultaneously increase the predicted load at that time. Discharging of vehicles is not considered.

In [11] a dynamic programming algorithm is presented, which can be used to calculate control strategies for plug-in hybrid vehicles. These control strategies include trading energy on different energy markets while plugged in as well as switching between battery or internal generator.

IV. Planning of the Charge Process

A. Definitions

Prices and charge operations are planned in 15 minute intervals by the energy provider. We will take the number t to indicate the t-th 15 minute interval based on some arbitrary reference point. The whole set of 15 minute intervals will be denoted by T. Furthermore let $T_{a,b} := \{a, a+1, \ldots, b\}$ for $a < b$ denote a continuous subset of T.

A charge plan u for the time $T_{t_{start},t_{end}}$ is defined as a function from the continuous subset $T_{t_{start},t_{end}}$ to the set of operations $O := \{-1, 0, +1\}$. The function $u_{k,l}$ will denote the partial plan defined over the continuous subset $T_{k,l}$. The set $U_{t_{start},t_{end}}$ will denote the set of all possible plans for the Time $T_{t_{start},t_{end}}$.

A price list p is defined as a function from the set T and an operation in O to a price given in $cent/kWh$ in steps of one-hundreth of a cent. $p(+1, t)$ denotes the price to be paid by the end user for charging at time interval t. $p(-1, t)$ indicates the price paid by the energy provider for discharging at time interval t. All prices must be positive, $p(0, t)$ must always be 0.

A battery model is defined as a triple $(\delta_{evse}, \delta_{bat}, s_{max})$. Where δ_{bat} and δ_{evse} are functions from a pair (o, s) where $o \in O$ denotes an operation and s the current state of charge of the battery. The function δ_{bat} returns the difference in charge of the battery if the operation o is applied on a battery with the current state of charge s while the function δ_{evse} returns the required energy taken in from the charge station (electric vehicle supply equipment, EVSE). The following constraints apply:

$$\delta_{bat}(-1, s) \leq \delta_{evse}(-1, s) \leq 0 \tag{1}$$
$$\delta_{bat}(0, s) = \delta_{evse}(0, s) = 0 \tag{2}$$
$$\delta_{evse}(+1, s) \geq \delta_{bat}(+1, s) \geq 0 \tag{3}$$

The state of the battery $s_{u,t}$ at the end of the 15 minute interval t for a charge plan u over $T_{t_{start},t_{end}}$ can be calculated as

$$s_{u,t} := \begin{cases} s_{init} & \text{for } t < t_{start} \\ s_{u,t-1} + \delta_{bat}(u(t), s_{u,t-1}) & \text{for } t_{start} \leq t \leq t_{end} \\ \widehat{s_u} & \text{for } t > t_{end} \end{cases} \tag{4}$$

where s_{init} indicates the state of the battery before the charge plan is executed and $\widehat{s_u} := s_{u,t_{end}}$ denotes the final charge after execution of u. A charge plan u is called s_1, s_2-bounded if there is no t for which $s_{u,t} < s_1$ or $s_{u,t} > s_2$.

The cost for executing a charge plan u over $T_{t_{start},t_{end}}$ up to the end of time interval t with $t_{start} \leq t \leq t_{end}$ can be calculated as

$$C_{u,t} := \sum_{k=t_{start}}^{t} \delta_{evse}(u(k), s_{u,k-1}) p(u(k), k). \tag{5}$$

The total cost $\widehat{C_u}$ of a charge plan u is $\widehat{C_u} := C_{u,t_{end}}$.

When comparing two charge plans, it is necessary not only to take into account that these two plans have different prices, but also that the battery may have a different state of charge after execution of these plans. Hence, the value $V_{u,t}$ for executing plan u until end of time interval t is defined as

$$V_{u,t} := (s_t - s_{init})\bar{p} - C_{u,t}. \tag{6}$$

Here $\bar{p}$ is the average price for the price list p. $\bar{p}$ computed from both prices for charging and discharging with equal weight. Other methods for the calculation of $\bar{p}$ were also considered, e. g. taking the minimum or maximum of average buy and sell prices. The final value of a charge plan is given as $\widehat{V_u} := V_{u,t_{end}}$. The charge plan with the highest value for all possible charge plans that reach the same final charge s at time t is denoted as

$$u^*_{s,t} := \underset{u \in U_{t_{start},t}}{\arg\max}\, \{\widehat{V_u}\}. \tag{7}$$

Furthermore, let $V^*_{s,t} = V_{u^*_{s,t}}$ denote the optimal value that can be achieved.

B. Formal Specification

The algorithm takes an initial charge of the battery s_{init}, a time interval t_{start} at which the charge plan must start and a time interval t_{end} at which it must end, a price list p, a battery model $b := (\delta_{evse}, \delta_{bat}, s_{max})$, and user settings. The user settings consist of the triple $(s_{sp}, s_{pl}, t_{dep})$ where s_{sp} indicates the required charge state for spontaneous trips, s_{pl} indicates the required charge for the next planned trip and t_{dep} indicates the planned departure time. It must be assured that $t_{start} \leq t_{dep} \leq t_{end}$.

The algorithm produces a $0, s_{max}$-bounded charge plan u over $T_{t_{start},t_{end}}$ with several hierarchical requirements.

1) Let t_{sp} denote the first time interval for which $s_{u,t_{sp}} \geq s_{sp}$. Then $t_{sp} - t_{start}$ must be minimal for all possible plans.
2) $u_{t_{sp},t_{dep}}$ must be s_{sp}, s_{max}-bounded.
3) $V_{u,t_{dep}}$ must be maximal for all plans satisfying the previous constraints.
4) $u_{t_{dep},t_{end}}$ must be s_{rf}, s_{max}-bounded, where $s_{rf} := max\{s_{sp}, s_{pl}\}$.
5) The values $V_{u,t}$ must be monotonous non-decreasing for $t > t_{dep}$.

All further analyses assume that at least one plan with $s_{u,t_{dep}} \geq s_{pl}$ and $s_{u,t_{dep}} \geq s_{sp}$ exists. If no such plan exists the algorithm fails. In the final system currently used in the field trial, failures are handled by indicating to the user that the given settings cannot be used to create a plan.

C. Algorithm for the spontaneous plan

The *spontaneous plan* refers to the part of the plan before the spontaneous distance is fulfilled, i. e. before $s_{u,t} \geq s_{sp}$. For this plan it is only required that the partial plan remains $0, s_{max}$-bounded and that Requirement 1 is fulfilled. Hence, it is sufficient to produce a first part of the charge plan that only contains the operation $+1$ until $s_{u,t} \geq s_{sp}$.

D. Algorithm for the planned plan

As above, let t_{sp} be the time interval after that s_{sp} has been reached. By Requirement 1 it is ensured that $s_{u,t_{sp}} \geq s_{sp}$.

The algorithm for calculation of the planned plan is based on dynamic programming [12]. For this we observe that

$$\begin{aligned} V^*_{s,t} = \max\{V^*_{s,t-1} + \bar{p}\delta_{bat}(o, s') - \delta_{evse}(o, s')p(o, t) \mid \\ s' + \delta_{bat}(o, s') = s, o \in O\} \end{aligned} \tag{8}$$

holds for any given charge state s and any given 15 minute interval t, $t_{sp} \leq t \leq t_{end}$.

Using (8) we can build a table of reachable charge states and the associated cost for each 15 minute interval t. An entry $s \to (o, v)$ in this table at time t represents the fact that charge state s is reachable at time t by performing operation o at that time and the resulting plan has a value v. The table is seeded by an entry $s_{u,t_{sp}} \to (+1, V_{u,t_{sp}})$ at time t_{sp}, representing the state that is reached after spontaneous planning. At each iteration the time is increased by one unit, and the next row of the state table is calculated. To calculate the next row each reachable state stored in the last row is considered and all states reachable from there are calculated. In case the same state is reached multiple times only the path with the highest value is left in the state table. Since (8) relates optimally reachable states at time t to optimally reachable states at time $t - 1$ the resulting entries in the state table at t also carry the optimal value. The calculation of the state table is illustrated in Figure 2.

Algorithm 1 Building a Statetable

$m \leftarrow []$; $t \leftarrow t_{sp}$
$m[t] \leftarrow \{s_{u,t_{sp}} \rightarrow (+1, V_{u,t_{sp}})\}$

while $t < t_{dep}$ **do**
  $t \leftarrow t + 1$; $m[t] \leftarrow \{\}$
  for all $(s \rightarrow (\_, v)) \in m[t-1]$ **do**
    for all $o \in \{-1, 0, +1\}$ **do**
      $s' \leftarrow s + \delta_{bat}(o, s)$; $v' \leftarrow v + \delta_{bat}(o, s)\bar{p} - \delta_{evse}(o, s)p(i)$

      if $s' < s_{sp}$ or $s' > s_{max}$ **then**
        Skip this operation
      end if

      $use\_plan \leftarrow true$
      for all $(s \rightarrow (\_, v)) \in m[t]$ with $s \in [s' - \varepsilon, s' + \varepsilon]$ **do**
        if $v \geq v'$ **then**
          $use\_plan \leftarrow false$
        end if
      end for
      if use_plan **then**
        $m[t] \leftarrow \{(s'' \rightarrow (o, v)) \mid (s'' \rightarrow (o, v)) \in m[t] \wedge s'' \notin [s' - \varepsilon, s' + \varepsilon]\}$
        $m[t] \leftarrow m[t] \cup \{s' \rightarrow (o, v')\}$
      end if
    end for
  end for
end while

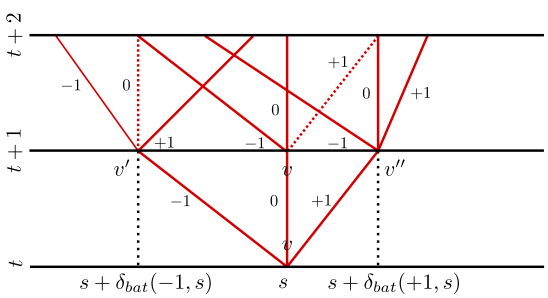

Figure 2. Illustration of the computed charge state table. At time t only the value v is stored for charge state s. At this time it is possible to charge, discharge or do nothing. At time $t+1$ the table will contain the value $v' := v + \delta_{bat}(-1, s)\bar{p} - \delta_{evse}(-1, s)p(-1, t+1)$ at position $s + \delta_{bat}(-1, s)$, the value v at position s and the value $v'' := v + \delta_{bat}(+1, s)\bar{p} - \delta_{evse}(+1, s)p(+1, t+1)$ at position $s + \delta_{bat}(+1, s)$. If multiple operations lead to the same charge state from a previous state, only one will be used (indicated by dotted lines).

Because of the battery model, multiple charge plans may reach very similar states, but not exactly the same state. For example, if the battery model takes dissipation between charge station and the battery into account, a sequence of charging and discharging may not reach the same state as doing nothing two times in the row, as each charging and discharging may reduce the battery state somewhat due to dissipation. Hence, keeping all reachable charge states in the state table may lead to a combinatorial explosion of the number of states at each further time step. To reduce this problem an approximation strategy is proposed. This strategy consists of not only comparing entries that have exactly the same charge state when removing other entries upon calculation of a new reachable state, but also other states, which only differ by a given ε. This method is similar to the common method of comparing floating point values by checking if the difference is smaller than a fixed ε. If this approximation is performed, however, it cannot be guaranteed that the resulting plan will be optimal. An experimental evaluation of this strategy is presented in Section V-B. The algorithm is presented in Algorithm 1.

To calculate the final plan from a state table first the charge state s^* with the highest value in last row of the table that is greater than s_{pl} has to be found. Then from this element in the table the plan can be traced back by following the stored operations. This method is similar to the trace-back portion of other dynamic programming algorithms.

E. Algorithm for the refining plan

For refinement planning only Requirements 4 and 5 have to be considered. These requirements can be translated directly into an algorithm. For this at each time step each operation is simulated and tested whether the requirements hold. In case neither charging nor discharging lead to an increase of the value or they would lead to a plan that is not s_{rf}, s_{max}-bounded, no charging or discharging is performed.

F. Reaction to updated price predictions

In the current field trial, price predictions are based on weather forecasts and provided for at least the next $24h$. Weather predictions are both based on predictions from a meteorological service as well as local measurements. Weather data from the meteorological service is updated every four hours, local measurements are taken in 15 minute intervals. In case weather predictions change unexpectedly the price predictions are updated and transmitted to all vehicles. On receiving new price predictions, all plans are recalculated and charging continues based on the updated plans. All considerations of optimality are based on the behaviour for an unaltered prediction.

V. Evaluation

A. Theoretical Evaluation

For our analysis it is assumed that the prices in the price list can be accessed in $O(1)$, e.g. by storing prices in an array. Furthermore, it is assumed that the battery model can be computed in $O(1)$ as well. In case of more complex implementations of the price list or the battery model, additional factors may need to be included.

The calculation of the spontaneous plan can be done with time and space complexity $O(t_{sp} - t_{start})$. This is optimal, as at least $t_{sp} - t_{start}$ charge steps are needed to reach the spontaneous distance given by the user.

The analysis for calculation of the planned plan is first carried out for $\varepsilon > 0$. For this let

$$n := \frac{s_{max} - s_{sp}}{\varepsilon} \tag{9}$$

denote the maximal number of entries in each row of the state table. Then the loop in line 6 will be repeated $O(n)$ times. The most complex operations in this loop are the retrieval of possible states that have to be removed in line 15 and replacement of non-optimal states in lines 21 and 22. Using a balanced binary tree to store the calculated states all these operations can be performed in $O(\log n)$. Hence, the loop at line 6 can be computed in $O(n \log n)$. This loop will be invoked once for each iteration of the loop in line 4. This loop will run exactly $m := t_{dep} - t_{sp}$ times. Therefore, the total time complexity of the algorithm is $O\left(m * n \log n\right)$. The algorithm requires the complete state table to be stored for the trace-back portion, so the space complexity is $O(m * n)$.

In case $\varepsilon = 0$ the above analysis is invalid. In this case depending on the battery model, the algorithm may have to deal with combinatorial explosion. Hence, the number of stored states in iteration i of the outer loop will be $O(3^i)$. Again we assume that a balanced tree is used for storage of each line, so that all operations are in $O(log(N))$ with N the size of the tree. Therefore, the total time complexity of the loop starting in line 6 is $O(log(3^i)) = O(i)$. The total time complexity of the algorithm is $O(\sum_{i=1}^{m} 3^i * i) = O((2m-1)3^m)$. The space complexity for the complete state table is $O(3^m)$.

The refining plan is calculated in space and time complexity $O(t_{end} - t_{dep})$, as each time step after the departure time has to be checked at least once. This part of the algorithm is also optimal.

B. Experimental Evaluation

As indicated in Section I, the final algorithm has to run in a constrained environment on an embedded device within the car. A full update of the charge plan is required for each user input, to provide predictions of the current price. Thus, it is necessary to test the algorithm in this environment. This way the influence of the factor ε can also be estimated. As the execution time is mostly accounted for by the calculation of the planned part of the charge plan, only this portion of the full algorithm is analyzed here.

To benchmark the algorithm $N = 10000$ inputs were generated randomly and the algorithm was run with different values of ε. A simple battery model was used with

$$\delta_{evse}(o, s) = \begin{cases} 500Wh & \text{for } o = +1 \\ 0Wh & \text{for } o = 0 \\ -500Wh & \text{for } o = -1 \end{cases}$$

and

$$\delta_{bat}(o, s) = \begin{cases} 500Wh * .95 & \text{for } o = +1 \\ 0Wh & \text{for } o = 0 \\ -500Wh/.95 & \text{for } o = -1, \end{cases}$$

i. e. the current state of charge was not considered and 5% dissipation between battery and charge station was taken into account. The maximum capacity of the battery was $16kWh$. The power, dissipation and maximum capacity were determined from the physical characteristics of the vehicle and charger currently used in the field trial.

To use different price predictions the start time of each charge process was varied in steps of $3h$ over the range of one week. To sample settings that were in accordance with observed user behaviour, the desired planned distance varied between $30km$ and the maximal available distance in $5km$ steps. The spontaneous distance was varied between $0km$ and $60km$ also in $5km$ steps. The departure time was between $1h$ and $24h$ in the future in $1h$ steps. The charge at arrival was between $0kWh$ and $16kWh$ varied in $4kWh$ steps. Settings for which no charge plan was possible, e. g. because the time until departure was insufficient, were excluded. Of all generated setting $N = 10000$ were selected randomly to be included in the analysis. The spontaneous and planned distances were chosen based on a preliminary analysis of user patterns in the field trial, i. e. they were chosen in the range of the 90th percentile of the actual settings used in the field trial. The maximum departure time was limited to keep the execution time of the optimal algorithm within an acceptable limit. However, other tests of the approximate algorithm with departure times further in future, showed that the execution times did not increase drastically after the chosen $24h$ limit.

The price data was taken from realistic price data produced by a municipal German energy provider as part of the project econnect (see Section IV-F).

The resulting execution times are shown in Figure 3a. A system is considered to respond instantaneously by users, if the response time is below $0.1s$ [13]. On average this was the case for $\varepsilon \geq 0.002Wh$. However, maximum response times were much higher at this epsilon. For $\varepsilon \geq 50Wh$ the algorithm responded instantaneously for all plans. As [13] indicates, at response times less than $1s$ no special user feedback is necessary to indicate the system is still running correctly. The response time of the system remained under this threshold for $\varepsilon \geq 0.002Wh$. For comparison of the approximation with the optimal algorithm the average fraction of execution time reduction is presented in Figure 3b. The fraction of execution time reduction was computed as $(t_{opt} - t_{approx})/t_{opt}$, where t_{opt} indicates the execution time of the optimal algorithm and t_{approx} indicates the execution time of the approximative algorithm.

No degradation in quality was detected as long as $\varepsilon \leq 50Wh$, i. e. the optimal plan was found for all settings. At $\varepsilon = 100Wh$ the optimal plan was still found for 82.3% of the

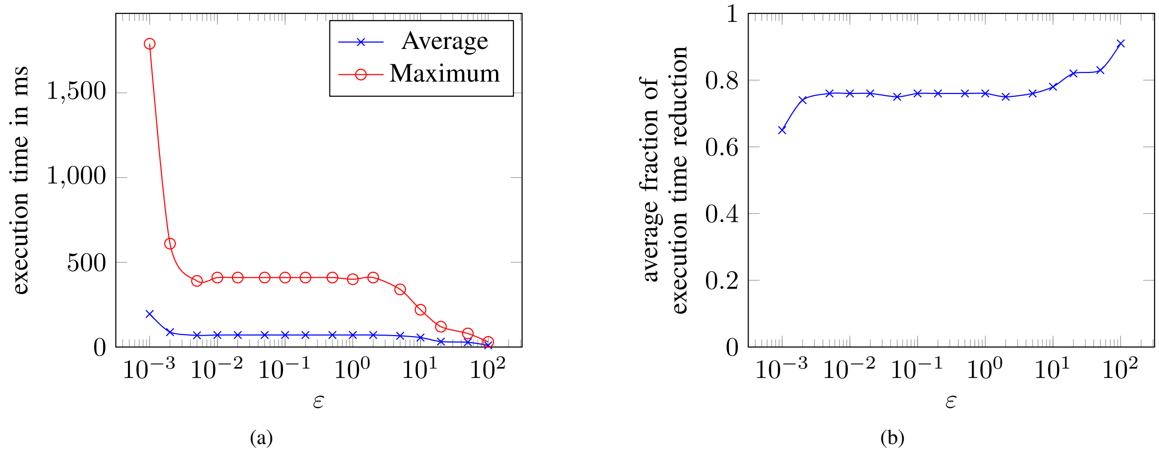

Figure 3. Execution time of the algorithm in relation to ε. Figure 3a shows the absolute execution time in ms. Figure 3b shows the average fraction of reduced execution time compared to the time required for calculating the optimal solution (i. e. $\varepsilon = 0$).

cases where the algorithm returned a result. In 0.48% of the testcases no plan could be calculated at $\varepsilon = 100Wh$, i. e. the algorithm failed to produce any plan for these testcases.

This analysis shows that the algorithm is usable for planning of charge processes on an embedded device within the EV. With the tested battery model an acceptable execution time was achieved without any quality degradation of the computed charge plans. It should be noted that the used battery model was kept as simple as possible; future tests should include more realistic battery models.

VI. Conclusion and Future Research

In this paper, a system was presented that was designed to investigate user acceptance for G2V/V2G technology. This system includes algorithms for decentralized charge planning on the vehicle. A theoretical and experimental evaluation shows, that these algorithms can be used on an embedded platform within the vehicle. The complete system is currently used in a field study aimed at evaluating user acceptance of G2V/V2G technology. The field trial will be completed by the end of 2014, results will be presented after data collection has been completed.

The algorithm was based on several basic decisions about factors that may influence user acceptance, which were not tested. Future studies on acceptance should test these assumptions, so that limiting factors for the large scale implementation of G2V/V2G technology can be clearly identified. Furthermore, based on these basic decisions and the goal of this study, the algorithm was designed to only take cost effective charging for the user into account. Properties of interest for grid operators were not investigated. Future research should test these properties and propose possible modifications, e. g. by adding a penalty term in the prices similar to the one in [9].

VII. Acknowledgment

We would like to thank Stadtwerke Trier, ABB AG and the department of Cognitive Psychology at Trier University for their cooperation in this project and Heinz Schmitz for his contribution to the initial idea for the algorithm.

References

[1] C. S. Thomas, "Transportation options in a carbon-constrained world: Hybrids, plug-in hybrids, biofuels, fuel cell electric vehicles, and battery electric vehicles," *International Journal of Hydrogen Energy*, vol. 34, no. 23, pp. 9279–9296, 2009.

[2] N. L. Panwar, S. C. Kaushik, and S. Kothari, "Role of renewable energy sources in environmental protection: a review," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 3, pp. 1513–1524, 2011.

[3] W. Kempton and J. Tomić, "Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy," *Journal of Power Sources*, vol. 144, no. 1, pp. 280–294, 2005.

[4] R. Linn, "Fahrer-Fahrzeug-Interaktion zum gesteuerten Laden und Rückspeisen von Elektrofahrzeugen," in *7. VDI-Tagung "Der Fahrer im 21. Jahrhundert"*, 2013.

[5] H. Wiegman, "Battery state estimation and control for power buffering applications," Ph.D. dissertation, The University of Wisconsin – Madison, 1999.

[6] T. Nett and J. Schneider, "Running autosar and linux side by side," in *Proc. of the 7th Junior Researcher Workshop on Real-Time Computing*, 2013.

[7] N. Leemput, J. Van Roy, F. Geth, P. Tant, B. Claessens, and J. Driesen, "Comparative analysis of coordination strategies for electric vehicles," in *2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe)*, 2011, pp. 1–8.

[8] Y. He, B. Venkatesh, and L. Guan, "Optimal scheduling for charging and discharging of electric vehicles," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1095–1105, 2012.

[9] Z. Ma, D. S. Callaway, and I. A. Hiskens, "Decentralized charging control of large populations of plug-in electric vehicles," *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 1, pp. 67–78, 2013.

[10] S. Kamboj, W. Kempton, and K. S. Decker, "Deploying power grid-integrated electric vehicles as a multi-agent system," in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 13–20.

[11] N. Rotering and M. Ilic, "Optimal charge control of plug-in hybrid electric vehicles in deregulated electricity markets," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1021–1029, Aug. 2011.

[12] R. Bellman, "The theory of dynamic programming," *Bulletin of the American Mathematical Society*, vol. 60, no. 6, pp. 503–515, 1954.

[13] J. Nielsen, "Response times: the three important limits," Excerpt from Chapter 5 of Usability Engineering by Jakob Nielsen, Academic Press, 1993. [Online]. Available: http://www.useit.com/papers/responsetime