Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 54 (2012) 1058 – 1067

Procedia
Social and Behavioral Sciences

EWGT 2012

15th meeting of the EURO Working Group on Transportation

E-mobility Fleet Management Using Ant Algorithms

Felix-Alexander Döppers[a], Sebastian Iwanowski[b,*]

[a]*MHP – A Porsche Service Company, Renteillichtung 1, D-45134 Essen, Germany*
[b]*FH Wedel, University of Applied Sciences, Feldstr. 143, D-22880 Wedel, Germany*

Abstract

This paper identifies the crucial tasks of e-mobility fleet management which is to be understood the operation of a fleet of vehicles powered by electricity. It is shown that the essentials can be mapped to a quadratic assignment problem. This makes the use of well-known ant algorithms available in order to solve the e-mobility fleet management problem even in the dynamic case. A simulator using efficient concurrency management was built in order to validate the results.

Keywords: Fleet Management and Transportation Sharing; Operating Models for E-Mobility; Traffic Management Simulator; Transportation Planning; Dynamic Assignment Problem; Ant Algorithms; Search Techniques; Concurrency Management

1. Motivation and scope of this paper

E-Mobility means mobility using e-vehicles, i.e. vehicles that are powered by electrical energy only. The motivation for e-mobility is a better usage of energy resources and a better chance to use green technology for the production of the energy to be consumed. In order to fulfill this goal, e-mobility should not only be restricted to single vehicles, but to the operation of fleets comprising e-vehicles only.

Fleet management is a big challenge in general. But using e-vehicles only, fleet management faces some demands exceeding these general challenges: Due to the shorter operating range of e-vehicles, the dispatched vehicles are changing users more frequently, and for optimizing the availability of e-vehicles satisfying the special needs of potential users, a concise management of battery charges including reloading policies is necessary.

This paper shows how to translate economical needs into precise problem formulations suited for software-based treatment. As a result, a dedicated assignment problem is formulated covering all parameters necessary for a successful real-life fleet management.

* Corresponding author. Tel.: +49-4103-8048-63; fax: +49-4103-8048-910-63.
E-mail address: iw@fh-wedel.de.

1877-0428 © 2012 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of the Program Committee
doi:10.1016/j.sbspro.2012.09.821

We apply and adapt ant-based approaches to solve this problem and validate our solution with a specially designed simulator.

Due to space limitations, some details are only sketched in this paper. They can be read more precisely in the thesis [9] belonging to this paper.

2. Challenges of e-mobility fleet management

The purpose of fleet management is to grant mobility for a limited number of users in a restricted region. Managing a fleet is a versatile challenge involving different tasks such as vehicle procurement, maintenance, customer administration, billing, etc. But the central task is to allocate vehicles to users satisfying all requested needs, and this is what we focus on in this paper. Every potential user of a vehicle has got his own personal mobility needs. It is a major target for fleet management to fulfill all (or almost all) personal mobility needs with the vehicles available. Additional targets may be to use a minimal number of cars or to decrease the overall operating costs.

Electric driving mechanisms cause some problems which do not apply to vehicles conventionally driven. Our scenario had to fulfill the guidelines developed by the Department of Trade and Industry of the German Federal State of Baden-Württemberg [1]. We identify the major challenges:

The range of electrically driven vehicles is the major bottleneck to satisfy many mobility needs. Electrically driven vehicles can cover a distance of 175 kilometers with one battery charge. This range is rather small compared to conventional vehicles.

Another problem of electrically driven vehicles is the recharging time of the battery. Refueling a conventional vehicle requires just some minutes. Modern battery charging technology needs at least one hour to fully charge one battery. This is only possible when a fast charging mechanism is applied. But such fast charging mechanisms reduce the maximum capacity of the battery. Normal charging mechanisms not baring this disadvantage need three to ten hours for fully recharging the battery. The use of different charging mechanisms is a special challenge for the fleet management of electric vehicles.

Batteries lose their maximum capacity after a number of cycles consisting of charging's and dischargings. This is why the number of chargings and dischargings cycles should be minimized by the fleet management. Thus, the usual procedure of conventional car rental agencies of always filling up the tank between two different customers is not a clever strategy in e-mobility management.

A clever recharging policy is very much limited by the little availability of electric service stations, specially on the road. This must also be considered by e-mobility fleet management.

Summarizing, we identify two tasks to be performed in e-mobility fleet management: The first task is to administrate efficient chargings and dischargings of batteries at every fleet location. The second task is to allocate appropriate vehicles with appropriate batteries to the users asking for certain demands of mobility.

3. Task of this work

In the following we describe the confinements for the fleet management scenarios we want to solve in this paper. They are also the specification for the simulator which serves for validation of our approach.

Potential fleet users may specify their mobility needs to the e-mobility fleet management at any time prior to use. A concise mobility need is defined by a starting time, a distance to the destination, and the provision of spare time for charging the battery and a priority. The e-mobility fleet management may administer a lot of different mobility needs at the same time.

After a user has specified his mobility need, the e-mobility fleet management tries to assign a vehicle to fulfill this mobility need. The assignments must obey a lot of different criteria. In the following, we elaborate the major three of them:

The first criterion corresponds to a priority order among the mobility needs. We distinguish between two priority classes. The e-mobility fleet management may not be able to make an assignment for all mobility needs. But it has to make an assignment for every higher priority mobility need if it makes an assignment to a lower priority mobility need. The appearance of new announcements with different priorities may be highly dynamic. This problem is one of the big challenges of this paper.

The second criterion is the current state of charge of the battery of an electric vehicle. The fleet management has to know this from every vehicle. The fleet management even has to know a forecast of the state of charge for every battery after use. This is why we need to know the distance to the destination for every mobility need. The current and forecasted states of charge must be used to find an optimal assignment. The big challenge of this criterion is to consider every aftermath of every assignment decision.

The third criterion is the use of special time for charging the batteries. This special time can be used to charge a battery on the way to the destination or to charge it prior to the starting time of a mobility need. This criterion provides a lot of flexibility for the e-mobility fleet management.

The assignment of a vehicle should always guarantee the fulfillment of the respective mobility need. In order to achieve this, a forecast must be made. This forecast is based on the typical speed average of the electric vehicle. It further contains assumptions about the available recharging infrastructure. The actual execution of the mobility need must be monitored by the e-mobility fleet management, too. This monitoring is compared with the previous forecast and should be used in order to make more accurate assignments in the future.

This scenario will be simulated in a simulator. The main goal of the simulator is to test the result of the e-mobility fleet management with a lot of different numbers of electric vehicles in a lot of constellations of personal mobility needs. The simulator adopts parameters for the generation of mobility needs, the current electric infrastructure, the locations of the e-mobility fleet, special organizational parameters, and information about the fleet vehicles. Although this is beyond the scope of this paper, it should be mentioned that such a simulator can also be used to compute the optimum size of a fleet for typical series of mobility needs.

4. Transformation into a mathematical optimization problem

According to the previous section, the task of this paper is to construct a function mapping a mobility need to an electric vehicle. Thus, the definition domain is the set of mobility needs and the range is the set of available electric vehicles.

But the functions need to satisfy further requirements: Each mobility need has a time span that can be computed from its parameters using the forecast technique explained in the previous section. For this work it is totally sufficient to work with a discrete notion of time. This makes it also easier for the simulator. Thus, each mobility need is associated with a finite set of successively ordered time points.

Now we can formulate our problem more precisely: The task is to construct an assignment function for each time point such that the following holds:

- For each time point, the assignment is injective, i.e. an electric vehicle can only be assigned to one mobility need belonging to that time point.
- Even for different time points, an assignment must always assign the same vehicle to the same mobility need, i.e. changes during use by one customer are not permitted.

This explains which functions are feasible. Now we extend this to an optimization problem. In our experiments we were working with quite a few different optimization criteria, but here we mention just three of them:

The first criterion is the number of satisfied high priority mobility needs: The more are satisfied, the better is the assignment.

The second criterion is the number of all satisfied mobility needs. The more are satisfied, the better is the assignment.

The third criterion refers to the capacity utilization of assigned electric vehicles: The more uniform is the degree of capacity utilization, the better is the assignment.

Our optimization problem is related to the quadratic assignment problem which is well known in Operation Research: That problem (first introduced in 1957) considers the allocation of a set of facilities to a set of locations. For each pair of locations, a distance is specified. For each pair of facilities, a weight or a flow is specified. The goal of the quadratic assignment problem is to minimize the distances multiplied by the corresponding flows between the facilities for every assignment from a facility to a location [2].

Our problem is a special case of this quadratic assignment problem: The mobility needs correspond to the facilities, and the assigned vehicles correspond to the locations. The weight of two mobility needs is the priority of the first one. Since each mobility need appears twice in each pair (on either position), all priorities are considered evenly. The analogue to the distance between two assigned locations is the following: For each pair of cars, define the usage distance to be the difference of its usage period within the considered planning period! If the car is the same for both positions of a pair, the usage distance is not 0, but rather half of the planning period. The main goal of the e-mobility fleet management problem is to minimize the usage distance multiplied by the corresponding priority for every assignment from a mobility need to an electric vehicle.

The mapping to the quadratic assignment problem makes us believe that the e-mobility fleet management problem belongs to the most difficult class of optimization problems (NP hard) as the quadratic assignment problem does. However, this would have to be proven separately.

Although this result already gives reasons why classical algorithmic results are of little use for our problem, there is one more requirement which backs this insight even more:

Our scenario should be highly dynamic: The set of incoming mobility needs is not fixed, but may alter from time to time. This makes our problem far more difficult than the classical applications of the quadratic assignment problem.

5. Applying and modifying ant algorithms

Ant systems [5] are used to solve optimization problems on graphs. The current optimization problem is defined by a target function depending on edge costs and possibly further information about the graph. Ant systems are specially designed for the scenario that edge costs or even the topology of the graph change and that the target function varies during operation. They resemble the behaviour of natural ants when they seek for food. In the following, we will only describe artificial ant systems.

An (artificial) ant is a software unit which is continuously generated over time by the ant system. Each ant uses the current data of the graph, considers the current constraints to be solved at the time of its generation, and tries to find a single solution for this problem. The quality of the result influences the modification of pheromones which are dynamic information chunks placed onto the edges of the graph. The pheromones represent the collected memory of previous ants using the respective edge. Subsequently generated ants are biased by these pheromones for their own construction of a solution.

In general, ant systems use graphs for pathfinding problems since this will always enables them to complete partial solutions. The probability of selecting an edge for a path completion depends on the quality of pheromones put so far as well as on some heuristic value which is usually derived from the graph's cost function. Usually, this heuristic value is static, but for ant systems even that need not be. The trade-of between the dynamic pheromones and this heuristic value may vary depending on the stage of the process or on the application in general. The continuous generation of ants guarantees that the pheromone value is successively updated to the latest situation in an eager way. After the occurrence of a new dynamic event, the longer the ant system is running, the better do the pheromones reflect the current situation.

The construction of solutions is carried out in different phases. First, in the initiation phase, initial pheromones are distributed to all edges in the graph. Normally, all edges get the same pheromone value, but at initialization we could make that also dependent on the cost function. Then, in a loop, construction and coordination phase are executed in turn as long as the ant system is needed. In the construction phase, a certain number of ants is generated simultaneously. Each ant has to find a solution for the pathfinding problem. At each node which is reached by an ant during the construction of a new solution, the probability that this ant will use a certain edge leaving this node is directly proportional to the amount of the pheromone value. Thus, more ants will use the edges baring good pheromone values. When all ants that were generated in the construction phase have constructed a solution, the construction phase is finished and the coordination phase starts in which all pheromone values are updated:

First, all pheromones are decreased resembling evaporation. This makes the future results more decisive than the past ones. After evaporation all ants increase the pheromones on the edges they actually

used for their specific solution. The increase of the pheromones is inversely proportional to the real costs of the associated solution. In total, this makes pheromones of edges belonging to favourable solutions increasing and of disfavourable solutions decreasing. Note that the alternating phases of construction and coordination correspond to a discrete simulation of a continuous process.

If a current solution is requested, the system returns the solution obtained from the current pheromones. This enables an ant system to give a quick answer with a solution corresponding to the current status of the dynamic system and makes it suit well for the problem we have addressed above.

However, if we want to solve our e-mobility fleet management with an ant algorithm, we have to formulate our problem as a pathfinding problem on a graph.

The graph for the e-mobility fleet management problem is defined as follows: A node is a triple of mobility need, electric vehicle and a possible starting time of the mobility need. Such node represents an assignment from a mobility need to an electric vehicle. The nodes are connected by the following edges:

Every node has got an edge to every other node, that doesn`t try to fulfil the same mobility need.

The ant algorithm needs to find a path on this graph obeying the following restrictions:

- The solution has to contain a node representing an assignment for every relevant mobility need.
- The path representing the current assignments has to satisfy the restrictions of the e-mobility fleet management problem identified in Section 4.

For more details how the path finding problem is solved, look up in the thesis [9].

A lot of research has been done for the classic quadratic assignment problem using ant algorithms ([2], [6], [7], and [8]). For achieving faster convergence, some work has been done to integrate other ("classical") optimization techniques.

Some authors describe methods combining the probabilistic search of ant algorithms with standard local search techniques ([4], [5], [6]). Such local search techniques try to find better solutions by the local transformation of currently known solution. The local search stops the search for a new solution, when the current best solution can`t be improved by any transformation of the solution. Such local search techniques have got one major disadvantage: The first initial solution has got a big influence on the best solution found. A technique trying to overcome this disadvantage is the tabu search. The tabu search prevents certain transformations for some iterations.

We tested the integration of tabu search into an existing ant framework. This required a modification of the construction phase of the ant algorithm: The ant algorithm first creates an initial solution using the steps of the normal construction phase. Then, a tabu search is applied. The following coordination phase updates the pheromones according to the result of the tabu search.

Christian Blum [3] developed a different ant approach for the classical quadratic assignment problem. His approach used a combination with classical tree search techniques. In particular, he applied beam search. The

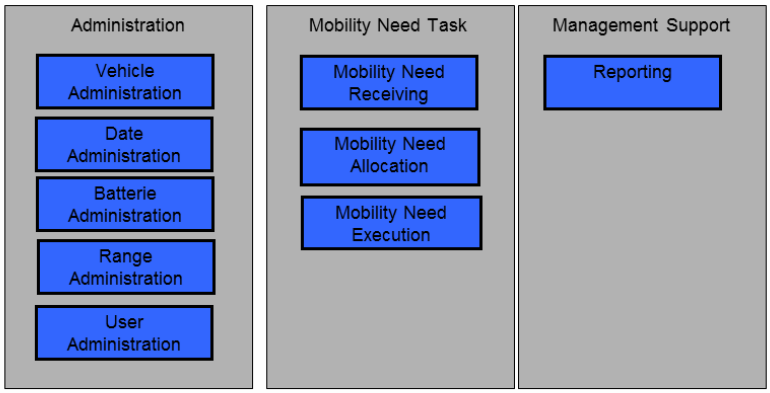
main idea of beam search is to expand a lot of different solutions from the currently best partial solutions. The beam search creates always n new solutions where n is a fixed number.

Also the integration of beam search into ant algorithms requires a modification of the construction phase: The construction phase always tries to create n new solutions. If the parameter n is increased, the probability that one of the new solutions is a better one is increased, too.

This is just an outline how to solve the e-mobility fleet management problem with an ant algorithm. The key observation is the close relation to the quadratic assignment problem. The integration of tabu and beam search serves for improving the performance. More details on this can be read in the thesis [9].

6. Building a simulator for validation of this approach

According to [1] which is the guideline for this work, e-mobility fleet management consists of the following tasks:



Task and Jobs of e-mobility Fleetmanagement

Our simulator also covers the administration tasks which are the following: The vehicle administration monitors the current state of the electric vehicle. The date administration prioritizes the mobility needs and collects the set of all mobility needs currently relevant. The user administration just checks the authorization of the fleet users. The administration of batteries and ranges are the specials tasks and jobs, which have to be done for e-mobility only. The main task of the battery administration is to monitor the current charging state for every battery. Another job of the battery administration is to predict the charging state for every battery in the future. The main task of the range administration is to optimize the usage of every battery for the execution of mobility needs. Another job of the range administration is the prediction how much battery power will be used up during the execution of a mobility need.

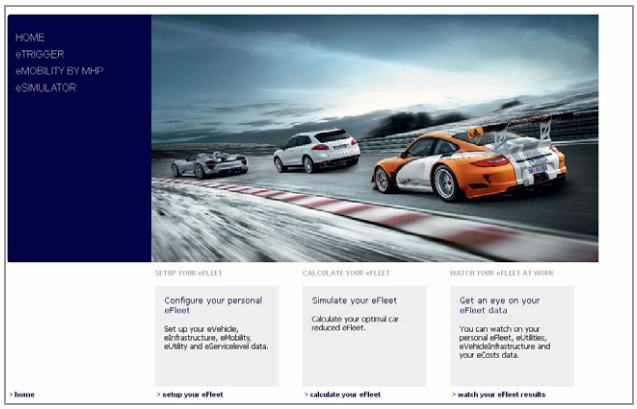
The mobility need task is the main task to be simulated in our simulator. The first step is the mobility need receiving. In order to simulate a realistic scenario, a lot of different personal mobility needs should be considered. Each mobility need has to be explicitly generated for the simulator. This generation can be standardized using typical mobility parameters. For simplification, the mobility parameters are structured in mobility profiles. A mobility profile describes a group of mobility users bearing the same mobility need behavior.

The second step is the mobility need allocation. Fleet management tries to find an assignment from each mobility need to an electric vehicle. The step contains an optimization run of the specified ant algorithm. Mobility need allocation needs support from vehicle administration, date administration, battery administration, and rage administration. These administration jobs have got a big influence on the dynamic events that have to be considered in the assignments.

The last step of the mobility need task is mobility need execution. The execution may create several dynamic events for range administration and battery administration that has to be considered in subsequent mobility need allocations.

The management support function contains a reporting job. This reporting job helps to monitor the results of the ant algorithm. The report job collects data about the incoming personal mobility needs, the results of the mobility need allocation and the execution of a personal mobility need. This data will also be used to calculate a prediction of the cost for the e-mobility fleet.

The simulator is implemented as a distributed system, which has got typical the software fragments three software units frontend, middleware and backend. The front end provides a web interface for the simulator looking as follows:



User Interface Simulator

The application administers 5 parameter groups which may be set at the frontend. The first group describes the behavior of fleet users. Mobility profiles can be created for every mobility user. The second parameter group describes the locational infrastructure of the e-mobility fleet. The third parameter group describes the recharging capabilities of the infrastructure. The parameters for the electric vehicles are considered in the fourth group. In the last group some general administration parameters can be declared.

The simulator has been implemented with the rich internet application framework Ext Js [10]. The middleware offers an http- interface for every service of the simulators backend. The middleware is implemented as a typical Java servlet [11]. The middleware uses the remote method invocation (RMI [12]) to communicate with the simulator's backend. This backend has been developed with the scala concurrency concept [13] for performance reasons.

Further details about the simulator and about the software technology used for implementation is given in the thesis [9].

7. Test results

In the following, we specify a sample test scenario:

Every parameter group of the simulator has to be determined for this scenario. For the profiles of mobility users, we define 3 different user groups. The first user group "Manager" has got only a few members. A manager tells his mobility need to the fleet management two hours before he wants to start his execution of his mobility need. Managers get a high priority. Managers permit the fleet management only 5 minutes to use for charging (which is almost nothing).

The second user group "Junior Manager" has got a little more members. A junior manager tells his mobility need 6 hours before he wants to start the execution. A junior manager provides 15 minutes for charging the battery. A junior manager mobility need has got a normal priority.

The third group "Worker" has got most of the members. A worker announces his mobility needs 18 hours before intended execution. Workers have got a low priority. Workers provide 30 extra minutes for charging.

Every fleet member has one mobility need with a distance between 15 and 30 kilometers per day. This example contains only one fleet location with 2 Mitsubishi iMiEV. The location of the fleet provides 2 charging stations. Every fifth mobility need provides a possibility to charge the battery during operation. Three days are simulated in this test scenario.

Since this is the crucial objective of our approach, we show how the ant algorithm reacts on dynamic events such as a change of the charging state or a new incoming job. At a special simulation time point (4099) we got the following relevant mobility needs and currently planned assignments:

Mobility need identifier	Member of Usergroup	Current execution state	Electric vehicle	Start time	End time
1	Junior Manager	Executed	Vehicle 2	4009	4155
2	Junior Manager	Executed	Vehicle 1	4098	4199
3	Worker	Not Executed	Vehicle 1	4741	4813
4	Worker	Not Executed	Vehicle 2	4769	4857
5	Worker	Not Executed	Vehicle 2	4998	5115
6	Worker	Not Executed	Vehicle 1	5016	5107
7	Worker	Not Executed	No Vehicle		

At the time point 4500, we face the following situation:

- Mobility need 1 and 2 have been fully executed.
- The electric vehicle 1 and 2 have been recharged at the locations infrastructure
- A manager announces a new mobility (no. 8) need starting at 4687.
- Three more workers announce mobility needs (no. 9, 10, 11).

The ant algorithm computes new assignments reacting on these dynamic events. At time point 4686, the ant algorithm has computed the following assignment:

Mobility need identifier	Member of Usergroup	Current execution state	Electric vehicle	Start time	End time
8	Manager	Not Executed	Vehicle 1	4687	4878
3	Worker	Not Executed	Vehicle 1	4730	4802
4	Worker	Not Executed	No Vehicle		
5	Worker	Not Executed	Vehicle 2	5006	5123
6	Worker	Not Executed	Vehicle 1	5015	5106
7	Worker	Not Executed	No Vehicle		

9	Worker	Not Executed	Vehicle 2	5214	5285
10	Worker	Not Executed	Vehicle 1	5486	5554
11	Worker	Not Executed	Vehicle 2	5618	5682

As we can see, most of the requests can be fulfilled. All new works jobs can be executed by the fleet. The battery chargings help a lot to execute more mobility needs.

Besides this particular example on dynamic events, we have also performed long-term runs: 3 days have been simulated for several scenarios identified by [1]. We got the following results:

Percentage degree of capacity utilization first vehicle	Percentage degree of capacity utilization second vehicle	Percentage total executed mobility needs	Percentage high priority executed mobility needs	Percentage middle priority executed mobility needs	Percentage low priority executed mobility needs
47%	53%	79,49%	100,00%	66,67%	79,17%

This shows that even without the special consideration of dynamic events the ant algorithm has achieved very satisfying results: The degree of capacity utilization between both vehicles is almost uniform. Nearly 80 % of all requested mobility needs have been executed by the fleet. Every high priority mobility need has been executed by the fleet.

More details on the tests are given in the thesis [9]. The described scenario has also been tested with several variations of the ant algorithm parameters described in section 5. The results are given in the thesis, too.

8. Conclusion

E-mobility fleet management requires some special considerations which make the task more complicated than general fleet management. The main problems are caused by battery recharging policies and a limited infrastructure for that.

In this paper, we identified the consideration of dynamic issues to be an important distinction between real-life applications and mere academic considerations. We showed that our practical problem is similar to the well-studied quadratic assignment problem. In consequence, we adapted solutions published for that problem to our fleet management problem. These solutions also cope with the dynamic case which is has been proved in an extensive test phase.

Our test profit from the implementation of a performant simulator satisfying all needs identified for e-mobility by a national agency in Germany.

References

[1] Wirtschaftsministerium Baden-Württemberg / Landesagentur für Elektromobilität. *Systemanalyse BWe mobil*, Technical Report, 2010 (in German)

[2] R. Burkard, E. Cela, P. Pardalos, and L. Pitsoulis. The Quadratic Assignment Problem, *Handbook of Combinatorial Optimization*, Volume 3, pp. 241-337

[3] C.Blum. Beam-ACO—Hybridizing Ant Colony Optimization with Beam Search: An Application to Open Shop Scheduling, *Computers & Operation Researcher* 32 (2005): pp. 1565-1591

[4] S.Scheuerer. *Neue Tabusuche-Heuristiken für die logistische Tourenplanung bei restringierendem Anhängereinsatz, mehreren Depots und Planungsperioden*, PhD thesis, University of Regensburg 2004 (in German)

[5] M.Dorigo and T.Stützle. *Ant Colony Optimization*, 2004, MIT Press, ISBN 0-262-04219-3

[6] T.Stützle, M.Lopez-Ibanez, and M.Dorigo. A Concise Overview of Applications of Ant Colony Optimization, *Wiley Encyclopedia of Operations Research and Management Science*, pages 896-911, John Wiley & Sons, 2011
[7] L. M. Gambardella and M.Dorigo. Ant Colonies for the Quadratic Assignment Problem, *Journal of the Operational Research Society*, 50(2), 2004: pp. 167-176
[8] K.-C. Wu, C.J.Ting, G. Casanova, and F. Lionel. An Ant Colony Optimization Algorithm for Quadratic Assignment Problem, *12th Asia Pacific Industrial Engineering and Management Systems Conference*, 2011
[9] F.-A. Döppers. *Simulation und Optimierung von Abläufen des Flottenmanagements in der Elektromobilität* Masterthesis, FH Wedel 2012 (in German)
[10] J. Garcia. *Ext Js in Action*, Manning 2010, ISBN 1-93518-211-0
[11] C.Schalk, E. Burns. *JavaServer Faces - The Complete Reference*, Mcgraw Hill 2006, ISBN 0072262400
[12] W. Grosso. *Java RMI*, O`Reilly 2002, ISBN 1-56592-452-5
[13] M.Odersky, L.Spoon, Lex and B.Venners. *Programming in Scala*, Antima Press 2008, ISBN 0-9815376-0-1