

Keep the balance with EMMA II.

Attila Bódis

Software Information Technology BSc, 5th semester

Consultant: **Tamás Gergely**, PhD – assistant professor

Extern consultant: **Péter Bajor** – professor's assistant
(Széchenyi István University, Győr)

University of Szeged – Faculty of Science and Informatics,
Department of Software Engineering

Abstract

In this paper, I am going to describe the problems of the electric grid, and show the results of the EMMA project, which is about optimizing the grid by wire-logistics viewpoints. My primary object is to find solutions for the suppliers' difficulties. I will introduce the EMMA Concept, the model of an industrial system including the usage of wind and photo-voltaic energy, electric vehicles, electricity storage, and of course the national grid.

Then I will describe the EMMA Vision system, which is created with the ability of optimizing the system. In this model, I have separated the system into two main parts: Hardware and Software configurations. Because of the lot of parameters, and the fact that even an objective function is not trivial, it is quite difficult to optimize any of these two modules. I will describe the created objective functions, and I will show genetic algorithms for the optimization of both of them. Finally, I will describe how these methods can be used in the real-life in the industrial facilities.

Contents

1.	Introduction	1
2.	Background	2
2.1.	Problems of the grid.....	2
2.2.	GRidfriEENd – concept.....	3
2.3.	Possible solutions.....	4
3.	EMMA Concept – The model of an industrial system.....	5
3.1.	The basics of the model	5
3.2.	The StadE.....	5
3.3.	Electricity Consumption (EC).....	6
3.4.	National Grid (NG)	7
3.5.	Wind and Photo-Voltaic energy (W and PV)	8
3.6.	Electric Vehicles (EV)	9
3.7.	Electricity Storage (ES)	10
4.	EMMA Vision – The complex optimization system	11
4.1.	The architecture	11
4.2.	The possibilities of the EMMA Vision system.....	12
5.	Genetic algorithms	13
5.1.	Genetic algorithms in general	13
5.2.	The currently used genetic algorithms	14
6.	Genetic algorithm for Software	15
6.1.	The individuals and the fitness function	15
6.2.	Population	17
6.3.	Mutation.....	18
6.4.	Selection.....	19
6.5.	The best results	21

7.	Genetic algorithm for Hardware	22
7.1.	The individuals and the fitness function	22
7.2.	One-Level Genetic Algorithm	23
7.2.1.	Population.....	23
7.2.2.	Mutation.....	24
7.2.3.	Selection	25
7.2.4.	The best results	26
7.3.	Multi-Level Genetic Algorithm	28
8.	Conclusions, future plans	31
9.	References	32
9.1.	Conference Proceedings.....	32
9.2.	Books	32

1. Introduction

In this paper, I am going to show the new results of the EMMA, which is about optimizing the electric grid by wire-logistics viewpoints with analysis of measured data and using a demand-side management concept. This paper is an extension of my previous essay, created for the Scientific Student Conference (A. Bódis et al., 2012). I will describe the basics of the project, as I did in that paper, and introduce the new developments: I tested the different settings of the optimization algorithm of the Software module described earlier, and I created an algorithm to optimize the Hardware, another module of the system.

In Section 2., I give a short description of the background of our topic. I describe the problems of the grid, which I want to solve. Then I mention some possible solutions from the scientific literature, including the GRIdfriEENd – concept (P. Földesi et al., 2011), which is created to define the objects of this research.

In Section 3., I introduce the EMMA Concept, the model of a theoretic industrial system. Firstly, I show how the system works in real-life. Then I describe the parts of this model. I define the StadE, the balance, on which all the incoming and outgoing energy are connected, and which always should be in equilibrium. I mention the role of the national grid in this balancing problem. I write about the usage of wind and photo-voltaic energy, and also about the effects of electric vehicles. I discuss the importance and dangers of electricity storage.

In Section 4., I introduce the EMMA Vision system, created with the ability for optimize the parameters of the system. In this model, I have separated the system into two main parts, the Hardware and the Software configurations. The foregoing one determines what kinds of devices are included in the system, while the latter one is about how to use the existing devices to get a GRIdfriEENd solution. I describe the structure of this model, and the optimization and prediction possibilities of the system. I consider the usage of the EMMA Vision system in industrial environments. I describe how it can be used to help making short – and long-term decisions to find more GRIdfriEENd solutions.

In Section 5., I write about the genetic algorithms in general. The reason of this is that I use them to optimize both the Software and the Hardware modules.

In Section 6., I introduce the genetic algorithm used for optimize the Software configuration. I discuss the implementation of the individuals, and the fitness function. Then I show the results of the tests made for determine the best parameterization of the algorithm.

In Section 7., I describe two genetic algorithms for Hardware optimization. I write about the representation of the genes, and about the fitness function. Then I present the results I got with the different parameters of the algorithms.

Finally in Section 8., I make the conclusions and write about the future plans of the development of the system.

2. Background

2.1. Problems of the grid

One of the most crucial supplies in our life is the electricity. We can basically find it everywhere, it is part of our everyday habits. It is even more important in industrial environment, because it is essential for the operation of factories and warehouses. The most of the machines (including producing equipments, forklifts) cannot work without proper electricity supply. Also, the lights, the air-conditioners and the computers need electricity. So we can understand why this service makes up such a large part of the costs at an industrial facility. In this research, my object is to reduce the costs of the suppliers by optimizing the consumption profile. Hopefully, the reduced costs can have an effect on the costumers' prices.

The electric grid is a special wire-logistic system. Similarly to the gas and water supply, the costumers' demand has to be satisfied immediately, because we never want to wait until we get water from the tap, or until the lights are turned on. However, unlike theses systems, there are usually no significant storage capacities in the electricity system. This means that the production and the transportation have to be flexible enough to satisfy the always changing demands. The system operator has to handle this problem by integrating different technologies to the system, like the base-load, the regulated and the non-regulated power stations, or even the fluctuating wind and photovoltaic energy (P. Földesi et al., 2010). In the next paragraphs, I will describe why it is difficult to follow the demands, what are the problems of the grid.

One of the main problems is the fact that the daily consumption profile is highly fluctuating. We use a lot more energy during the day than at night, so the supplier has to reduce the production in the evening, and increase again in the morning. Furthermore, the profile is different for every costumer, and this is extremely true for the industrial facilities, because of the possibly different operation times.

This fluctuation in the profile means a lot of problems for the grid. One of them is the peak consumption. The system infrastructure must be sized for this level to ensure the ability of meeting the demands. However, it is used only occasionally on this level. So we have to maintain an expensive and highly unutilized system.

Another problem is the minimum consumption. If this is on a very low level, a lot of power stations have to be switched off, and switched on again when the consumption is increasing. This is not possible for every type (e.g. Base-load power stations, like nuclear ones). Moreover, the lifetime of the devices are decreased by the lot of switching cycles.

The large gradient changes are also a big problem, because the system has to answer these demand-changes immediately, so the supplier has to switch on and off the devices very often. It takes a lot of time for the most of the types of power stations, so we have to use ones which can answer the changes quickly, but these types are more expensive.

Nowadays, a very popular solution is the usage of the renewable energy sources, like wind and photo-voltaic energy. They are usually told to be a cheap alternative of the traditional power stations, but they also cause several problems for the grid. The weather is mostly unpredictable, so the system operator cannot calculate exactly with the energy produced by the renewables. However, the object is to use as much renewable energy as possible because of financial and environment-friendly reasons. So the operator may try to use this energy always if it is possible, and this causes further fluctuation in the system from the viewpoint of the traditional power stations.

2.2. GRidfriEENd – concept

In the remainder part of this paper, I am investigating possible solutions of the mentioned problems. My object is to make the profile better from the viewpoint of the supplier according to the concept, called GRidfriEENd (Figure 1.). This concept contains two main behavioural patterns.

The first one is the Transfer method, which is about moving the peak-consumptions to the valleys. This is very efficient in decreasing the fluctuation, because the maximum level can be reduced and the minimum level is increased, too.

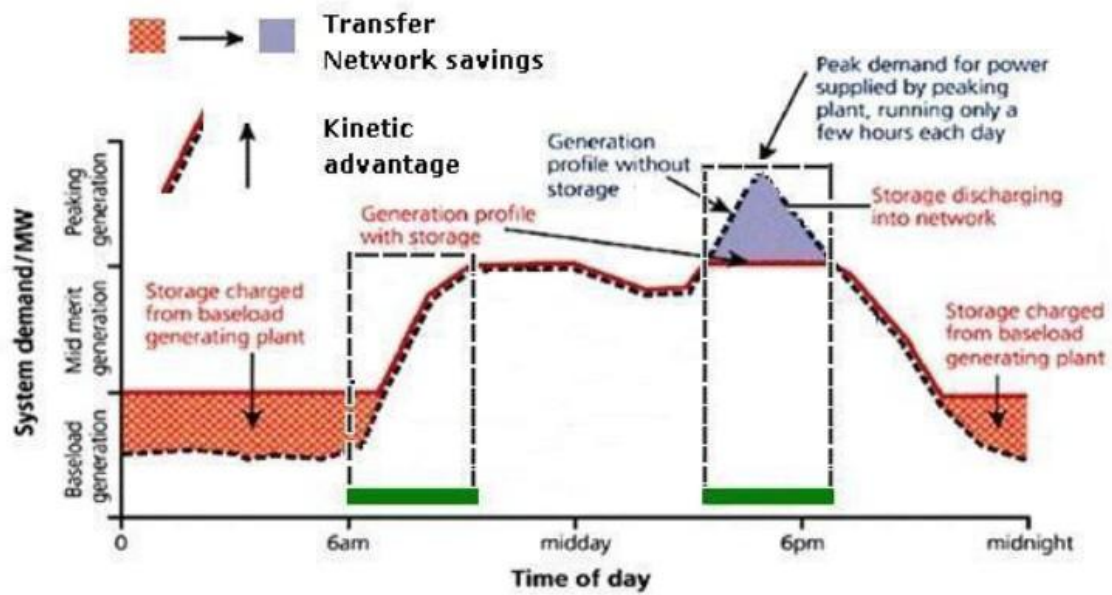


Figure 1. The average daily electricity load profile with ‘GRidfriEENd’ behaviour.– (P. Földesi et al., 2011)

The other main method, called Kinetic, is about to decrease the gradient changes during the day. This results that the operator does not have to switch the device so quickly, so the percentage of the most expensive devices can be reduced.

In the next part, I will show some solutions for make the system operating on a more GRidfriEENd way.

2.3. Possible solutions

One simple solution would be to create a tariff system with consideration of the time when we use electricity. If we use it at a peak-time period, the price would be higher, and if we use it in a less frequently period, the price would be lower. On this way, the costumers would be motivated to use electricity smartly, because it would cost less for them. However, there are possible negative effects of this solution. The research of A. Dán et al. (2011) about modelling the consumer behaviour gave the result that the peak periods can be reduced on this way, but large gradient changes are observed because of the behaviours of the rational costumers: they would like to minimize the consequences of the rescheduling of the electricity usage, so they switch on their devices as soon as the prices get lower. So this has a big Transfer advantage, but a very big Kinetic disadvantage.

Another solution is the usage of electricity storage. This would make possible to shift the demands in time, and it would create a Transfer and Kinetic advantage as well. However, it is very important to use it with a proper strategy, otherwise it can even mess up the all profile making more difficult the situation for the suppliers. Furthermore, if there is storage on more levels of the system, a new problem occurs: the so called ‘Bullwhip-effect’. This is a problem of the traditional logistics systems and supply chains. It means that the real costumer demands are far away from the production because of the more level storage. As the electricity storage systems are getting more popular this new challenge will probably also appear. To avoid this problem, it is important to create exact rules about the storing strategies by the governments.

3. EMMA Concept – The model of an industrial system

3.1. The basics of the model

The main idea of the EMMA project is that industrial facilities could help a lot for the grid if they worked on a GRidfriEENd way. They can use renewable energy sources or they can even have electricity storage, and optimizing the usage of these devices can result a cheaper infrastructure.

As first step, I have created the EMMA Concept, which is a model of an industrial system (P. Földesi et al., 2011), used to calculate the effects of the usage of different devices and configurations. Before describing this model, let me briefly introduce how the electric system works at a facility.

Every electric device is connected to the grid, and gets the energy from it. However, the facilities can use for example renewable energy sources for decreasing its costs. These devices may produce energy even if the facility does not use at that moment. In this case the produced energy can be stored for later usage or sent back to the grid. So the system is always balanced because of the nature of the electricity. This is provided by the national grid, where the lack is satisfied from, and where the extra energy runs back.

3.2. The StadE

In the model, I defined the StadE as the line on which all devices, producers, and consumers, are connected (Figure 2.). As mentioned earlier, this always has to be balanced. As the balance must be kept continuously, it is unacceptable to calculate with average values. So I

use 5-minute long periods as the basis of the calculations. This is sufficiently short period that there are not significant changes in the system during this time. Fortunately, I have the needed measured data about the weather and the consumption from a specific facility for a one-year period.

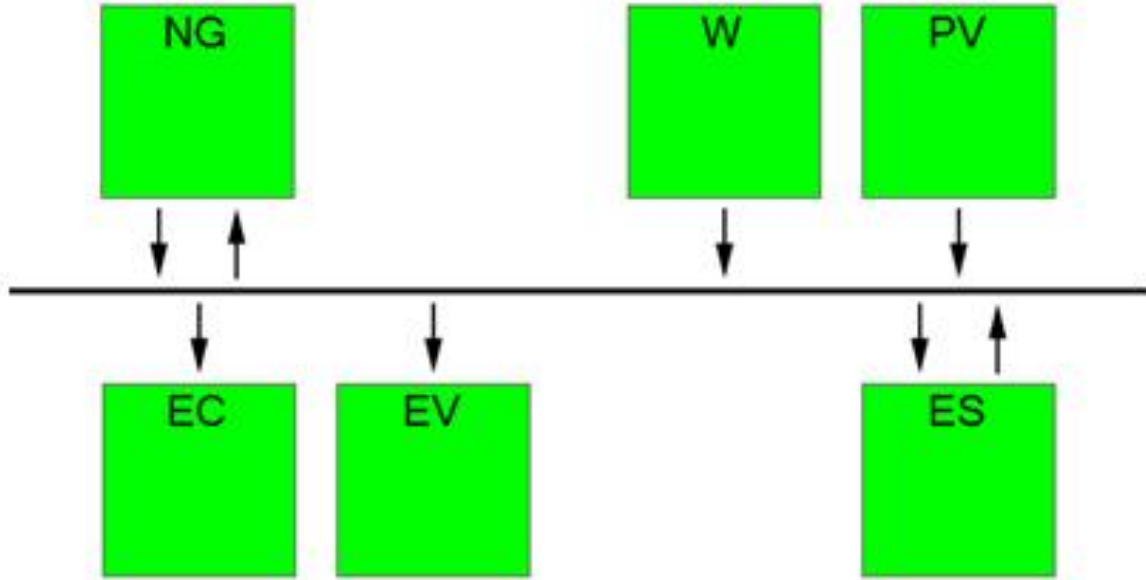


Figure 2. EMMA Concept – The model of the electricity system of a facility.

Considering the usage of Wind (W) and Photo-Voltaic (PV) energy, Electric Vehicles (EV), Electricity Storage (ES), the general Electricity Consumption (EC) and of course the National Grid (NG), the following equation must be satisfied for all the 5-minute long periods:

$$EC + EV + ES_{charging} = NG + W + PV + ES_{discharging}$$

3.3. Electricity Consumption (EC)

The electricity consumption includes the usage of all kind of electric devices, like the lights, computers, air-conditioners and producing machines. These data are available for the model from the on-site measurements.



Figure 3. The Electricity Consumption profile on a summer day at the facility.

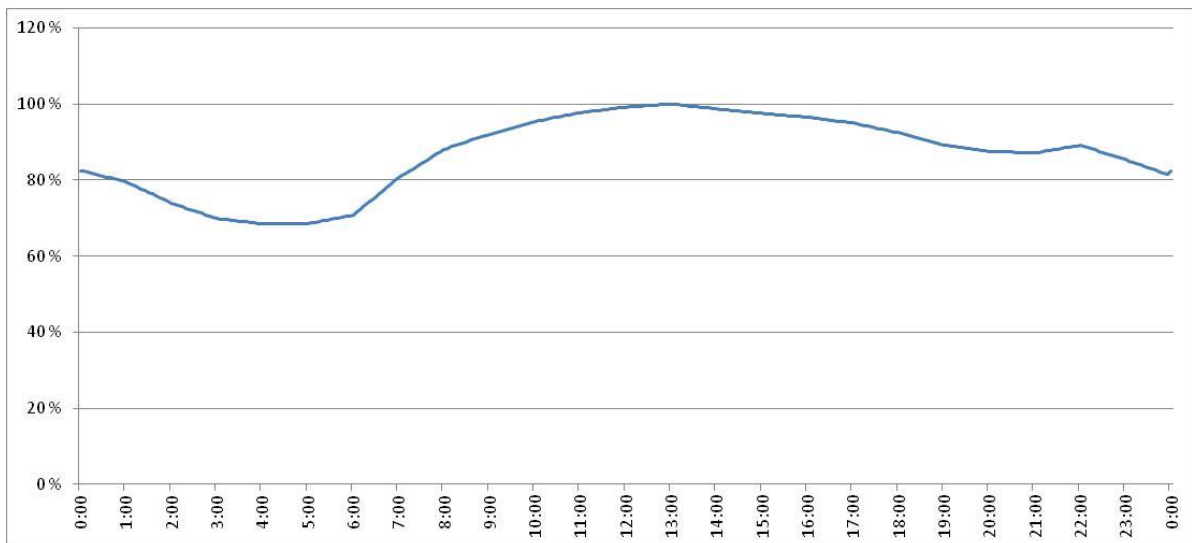


Figure 4. The Hungarian consumption profile on a summer day. (data from Mavir)

3.4. National Grid (NG)

Nowadays, the electricity consumption is usually satisfied by only the national grid. However, the facility has to pay for this service, of course. The tariff system can be different at the suppliers, but basically there are two main parts of these expenses: the first part is paid

after the ordered quantity which is the predicted consumption value, the other part is paid after the real consumption. Usually an extra fee must be paid if, either more or less energy were used, because the bad ordered quantity makes difficult to predict the operation of the system.

The usage of the national grid hardly depends on the part of the day, so the model handles different parameters for every 3-hour periods in one day. The facility has to say the Ordered Quantity for these periods, and the supplier determines the prices. The price is calculated from 4 parameters: Order-Price is the cost/kWh until the Ordered Quantity, Higher-Price is the cost/kWh over the Ordered Quantity, Lower-Price is the cost/kWh for the unused part of the Ordered Quantity and Sell-Price the income/kWh for the quantity sold to the supplier.

3.5. Wind and Photo-Voltaic energy (W and PV)

For the calculations with wind and photo-voltaic energy, the operation characteristics of these devices must be considered. This is basically the same for all of them: they start working on a specific level of the associated weather data (*start*), and then they reach their peak-production (*peakP*) on another level of the measured data (*peak*), and stops working, when this data is so high, that it endangers the device itself (*stop*). (Figure 5.)

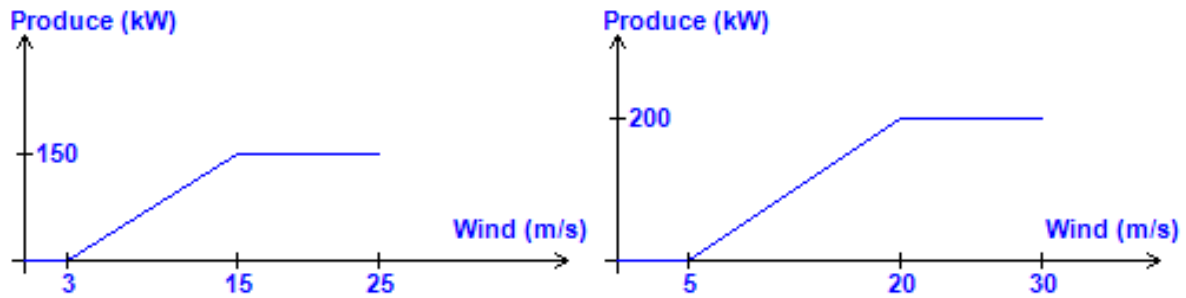


Figure 5. Production characteristics for Vestas V15 and Vestas V27 Wind turbines.

In the model, beside the mentioned 4 parameters I handle the *price* and *maintenance costs*, too. In case of wind turbines, there is another important parameter: the *height* of the device, because the wind-speed differs on different heights and the device may work on other height than the data measured. For this correction, I use the *alpha* parameter determining the type of the environment (e.g. forest, hills, plain, buildings).

3.6. Electric Vehicles (EV)

In industrial environments, the transportation, both on the site and to the partners, is always a crucial and highly energy-intensive part of the processes. The electric vehicles technology is already available, but in the near future it will probably get more widespread because of financial and environmental advantages. This provides a green and cheap alternative of the traditional vehicles. Furthermore, there are specific places where the usage of the fossil fuel-powered ones is not possible (e.g. small area, cooled warehouses, etc.).

Because of the importance of this technology I cannot leave it out of the model, of course. Actually, these vehicles offers a very good solution for making the consumption profile more ‘GRidfriEENd’, because they can be charged in the valley periods increasing the minimum level of the profile. In the model, the charging profile can be defined with the following parameters. Firstly, we have to define the summarized capacity of the vehicles (*FullNeed*). Then the profile is described by more charging periods. We specify the percent of capacity charged (*Psum*), the start (*startH*, *startMin*) and end time (*endH*, *endMin*) of the charging periods. Also, the way of charging (*mode*) is required for every period, which can be Rising, Falling or Constant mode. (Figure 6.)

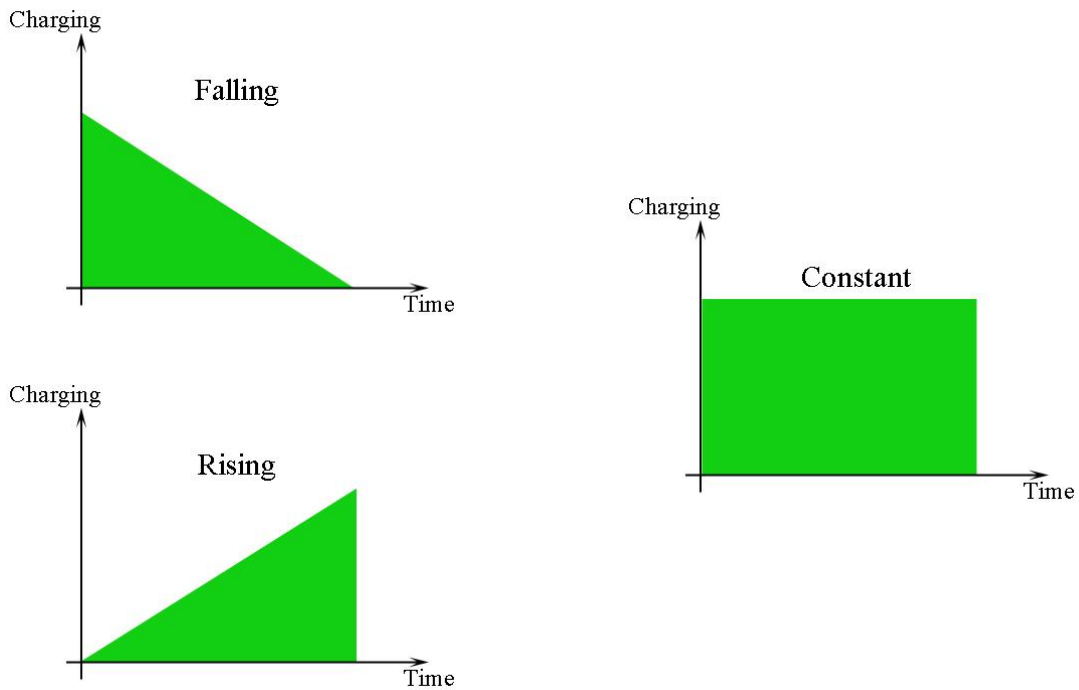


Figure 6. Possible modes of EV charging.

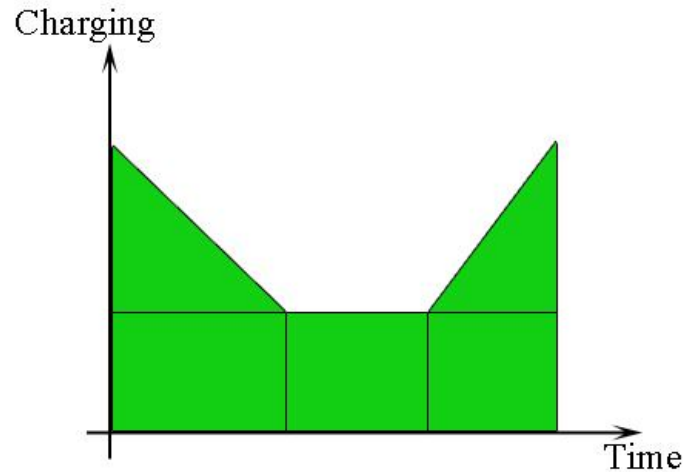


Figure 7. One possible EV charging profile

3.7. Electricity Storage (ES)

The next module of the EMMA Concept I write about, is Electricity Storage. As I already mentioned, this is one of the best way to make the costumers' behaviour more 'GRidfriEENd', because it ensures a lot of flexibility to shift the demands in the time. Currently, this technology is not too frequently used, but it is getting more popular, so it will have huge effect on the profiles in the near future.

In the model, I have to use several technical parameters for these devices. Not only the *Max. Capacity* and the *Efficiency* are required, but the charging speed is important parameter, too. The charging has two directions, so the devices have two charging speeds: *DownSpeed* means the charged quantity coming down from the grid, and the *UpSpeed* determines the quantity charged back to the grid in a 5-minute long period. Another parameter specifies the percent of the Max. Capacity used for Transfer purposes (*TransPercent*). Of course, the *price* and the *maintenance costs* are also handled. As already mentioned it is very important to use the storage capacity with good strategy to avoid the negative effects of the usage.

For this reason, I have introduced an off-line algorithm, called TransNetic Algorithm (TNA), which creates the optimal charging profile for the storage devices. This algorithm starts with Transfer-method by increasing the minimum consumption step-by-step until there is still free capacity in the device. After, the maximum is decreased on the same way. However, not all the capacity is used for this part, but only the quantity determined by the *Max. Capacity* and the *TransPercent* parameters together. After this method, the Kinetic-

method is started. First, I find the start and the end point of the longest period in the morning – because the most serious Kinetic-problem is at this time –, which can be improved with the quantity dedicated for Kinetic purposes. Then I am going to create a linear profile between the start and the end points, so I decrease the maximum difference from this line step-by-step to the second highest difference level. Finally, I apply the Kinetic-method for the evening period, because there is also a serious Kinetic-problem. So, the algorithm created a new profile, which has less Transfer and Kinetic problem, too. If the original level was under the new profile, it means the facility has to buy some more energy from the grid to charge the ES. If the original profile is above the improved one, they have to buy less, and use the stored energy.

4. EMMA Vision – The complex optimization system

4.1. The architecture

Based on the experiments of the EMMA Concept software, I have created a new architecture, called EMMA Vision (Figure 8.). In this part, I will introduce this system, its two modules: Hardware and Software, their relationship, their optimization algorithms, and the usage of this system.

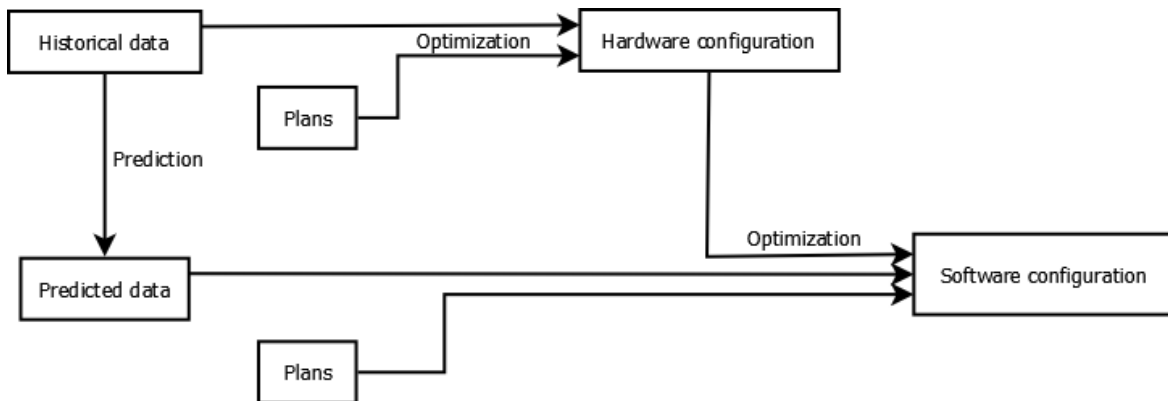


Figure 8. The model of EMMA Vision

The Hardware configuration determines what kinds of devices are used. This contains the physical parameters of the different equipments. The Wind turbines and the Photo-Voltaic cells have the following parameters in this module: *start*, *peak*, *peakP*, *stop*, *price*,

maintenance cost/year, and only for W the *height* and the *alpha* values. For the Electricity Storage devices, there are the following ones: *Max. Capacity*, *Efficiency*, *DownSpeed*, *UpSpeed*, *price*, *maintenance cost/year*. In case of the Electric Vehicles, the Hardware contains only the *FullNeed* parameter.

The Software module describes how to use the existing devices (the Hardware), and what will be the result of the operation, so it contains the operative parameters. For the Electric Vehicles, there are *PSum*, *startH*, *startMin*, *endH*, *endMin*, and *mode* for every charging period. I use only the *TransPercent* parameter for the storage devices in this module. In connection with National Grid, the Software contains the *Ordered Quantities* for every 3-hour period. The calculation starts with the renewable sources (W and PV) and the EV. Then the operation on the ES is calculated using TNA. So we get the final profile to buy from the National Grid, and this determines the price for the specific periods.

4.2. The possibilities of the EMMA Vision system

As mentioned earlier, the EMMA Vision architecture has the ability to use different optimization algorithms, so it is possible to not only examine fixed configurations, but to find the optimal ones. The optimizations of the two modules have completely different meanings, and they are used in different situations.

The Hardware optimization helps the facility to make strategic decisions, like what kind of wind turbine or ES-device should be bought. This is usable before investments to calculate the advantages of the alternatives, and helps the long-term planning. This optimization includes the Software optimization, of course, because we have to know how would work the devices for being able to say how good they are for the current objects.

However, the Software optimization is important separately, too. The result of this is a set of operative decisions to reach the optimal utilization of the devices and the best possible consumption-profile. This helps the facility in the short-term planning but also required for making long-term decisions, because the usage of the devices is important for the Hardware optimization.

Furthermore, the prediction of input data is needed for the operative calculations. This forecast is based on the measured historical data and the plans of the facility (e.g. planned stops, holidays, etc.). Instead of creating weather forecast, which would be very difficult, we are going to use meteorological sources to get the needed data. So we have to predict only the consumption.

The EMMA Vision system will provide a complex solution for industrial usage in order to reduce the costs by improving the effectiveness of the infrastructure. The first step of the installation of this service is finding the best hardware configuration. At this phase, we answer the question, what devices should the facility buy to become more GRIdfriEENd. For this purpose, the Hardware configuration must be optimized based on the Historical data, measured at the site, and the plans of the facility (e.g. maximum price of the devices).

After the Hardware is chosen, the optimal usage of the existing system has to be found, so we have to optimize the Software configuration for the daily operation. This phase answers the question, how to use the infrastructure. This method has the following 3 steps:

1. Predict data for the next day
2. Calculate the optimal Software configuration based on predicted data
3. After every 5 minutes modify the prediction according to the measured values and recalculate the optimal Software configuration

5. Genetic algorithms

5.1. Genetic algorithms in general

Before introducing the actually used genetic algorithms (GAs) for optimizing the modules of EMMA Vision, let me write shortly about them in general.

They can be used to search in a search space. The basic idea of them is to handle the possible solutions as natural individuals and use the evolution to find the best ones. More precisely, they do this search by changing the genotype of these individuals, and giving some kind of priorities for the best ones.

Although, there is no definition of genetic algorithm accepted by all in the evolutionary-computation community that differentiates GAs from other evolutionary computation methods, but it can be said that most methods called GAs have at least the following elements in common: populations of chromosomes, selection according to fitness, crossover to produce new offspring, and random mutation of new offspring. (M. Mitchell, 1998)

The individuals are often represented by bit strings and the evolutionary modifications are made on the zeros and ones. For compare the differences between the specific individuals, all the GAs need to have a good fitness function determining how good a specific individual is.

The genetic algorithms use 3 basic operators to find better and better solutions for the problem.

The *Selection* operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce. The *Crossover* operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. Finally, the *Mutation* operator randomly changes some of the genes in a chromosome. (M. Mitchell, 1998)

5.2. The currently used genetic algorithms

As mentioned, the individuals are often represented as bit strings. However, I will use another representation, because of several reasons, like real values and changing numbers of chromosomes.

Finding a proper fitness function is one of the most crucial points of creating optimization algorithms. In case of electric grids, it is quite difficult to find this function because there are several viewpoints from which the system should be optimal. For example, the supplier probably want other properties of the system to be optimal, than the consumer. So I have to use a function which is flexible enough to be able to create a common object for both parties. To simply solve this problem I have introduced a Balanced Score Card (BSC) system such that it shows how much the profile (and the all configuration) became better than the original one (EC). The BSC system means that I created percent-like indicators about the operation of the devices, and classified them into 4 groups: Financial, Partnership, Internal and Learning & Growth. Then the fitness function is the weighted average of the values of these indicators. So the main objects of the parties can be determined by defining the correct weights of the indicators. One of the biggest advantages of the BSC system is that not only the financial aspects are taken into consideration. This is important, because the electric grid is a system, where the current costs do not necessarily show the quality of the configurations.

I use a quite simple *Selection* method. After sorting the individuals in decreasing order of BSC values, I select the ‘neighbours’ for recombination until a given percentage of the population, so the 1st individual is paired with the 2nd, the 3rd one with the 4th, etc. The *Crossover* algorithm selects a random locus on the shorter chromosome, and exchanges the genes, located before the locus, with the other individual. After this the worst 50% of the population is deleted. So we always get the same size population as the original one. The *Mutation* operator changes every gene of every individual with a given percentage.

6. Genetic algorithm for Software

In this section I will describe the genetic algorithm used for optimizing the Software module with a fixed Hardware. I will show the structure of the used individuals and the fitness function. Then I will introduce the results of tests for different parameterizations of the algorithm. These tests are made such that I always changed only one parameter and left the others unchanged. I run the algorithm for every setting 5 times and I calculated the average of the best fitness values. So I was able to compare the effects of the different settings. Finally, I will show the effects on the profile of one of the best Software configurations.

6.1. The individuals and the fitness function

First of all, I have to define the genes for the algorithm, which are the parameters included in the Software module. This means that the genes are different-typed values. In connection with the NG, I use 8 genes for *Ordered Quantities*: 1 real value for every 3-hour periods. The EV Charging profile is a complex parameter containing more charging periods. For one charging period, I have the genes shown in Table 1:

Number of genes	Type of the gene	Parameters' name
1	Real value	<i>PSum</i>
2	Integer from 0 to 23	<i>startH, endH</i>
2	Integer from 0 to 59	<i>startMin, endMin</i>
1	3 possible values: 'Rising', 'Falling', 'Constant'	<i>mode</i>

Table 1. The genes for Electric Vehicles charging periods

There are as many genes for ES *TransPercent* parameter (real value), as ES devices are in the Hardware configuration.

As I mentioned, the fitness function is based on Balanced Score Card system, which has to be able to show differences between the individuals. I use the indicators for the Software module described in Table 2. (Note: They are calculated for one-day periods, so in the formulas I use 288 as the number of 5-minute periods.)

Financial	
Cost Change	$-\frac{cost - cost^*}{ cost }$
Consumption Change	$\frac{\sum_{i=0}^{287} NG_i - \sum_{i=0}^{287} NG_i^*}{\sum_{i=0}^{287} NG_i}$
Partnership	
Maximum Change	$\frac{max - max^*}{ max }$
Minimum Change	$-\frac{min - min^*}{ min }$
Maximum-Jump Change	Let $jump = NG_i - NG_{i-1}$ Then the indicator is: $\frac{jump_{max} - jump_{max}^*}{jump_{max}}$
Internal	
EV Utilization	$\frac{\sum_{i=0}^{287} EV_i}{EV_{capacity}}$
ES Charging Speed Percent	$\frac{\frac{\sum_{i=0}^{287} ES \text{ Charged}_i}{288}}{ES \text{ Charging Speed}}$
ES Discharging Speed Percent	$\frac{\frac{\sum_{i=0}^{287} ES \text{ Discharged}_i}{288}}{ES \text{ Discharging Speed}}$
ES Max Capacity Utilization	$\frac{ES \text{ Level}_{max}}{ES \text{ Capacity}}$
ES Charging Cycles Percent	Define $x_1, x_2, \dots, x_m$ such that $ES \text{ Charged}_{x_i} > 0$. Then the indicator is: $\frac{m}{288}$
ES Discharging Cycles Percent	Define $x_1, x_2, \dots, x_m$ such that $ES \text{ Discharged}_{x_i} > 0$. Then the indicator is: $\frac{m}{288}$.
RES Production Percent	Define $x_1, x_2, \dots, x_m$ such that $RES \text{ Produced}_{x_i} > 0$. Then the indicator is: $\frac{m}{288}$.
Learning & Growth	
Profile-Width-Change	$\frac{(max^* - min^*)}{(max - min)}$

Table 2. The Balanced Score Card indicators for the Software module.

6.2. Population

For determine the best population size I used a fixed selection rate of 100%, which means that all the individuals took part in the recombination, and a mutation rate of 20%, meaning every genes of every individual were changed with probability 20%. I have tested populations of 10, 20 and 50 individuals, and the average results are shown on Figure 9. and 10.

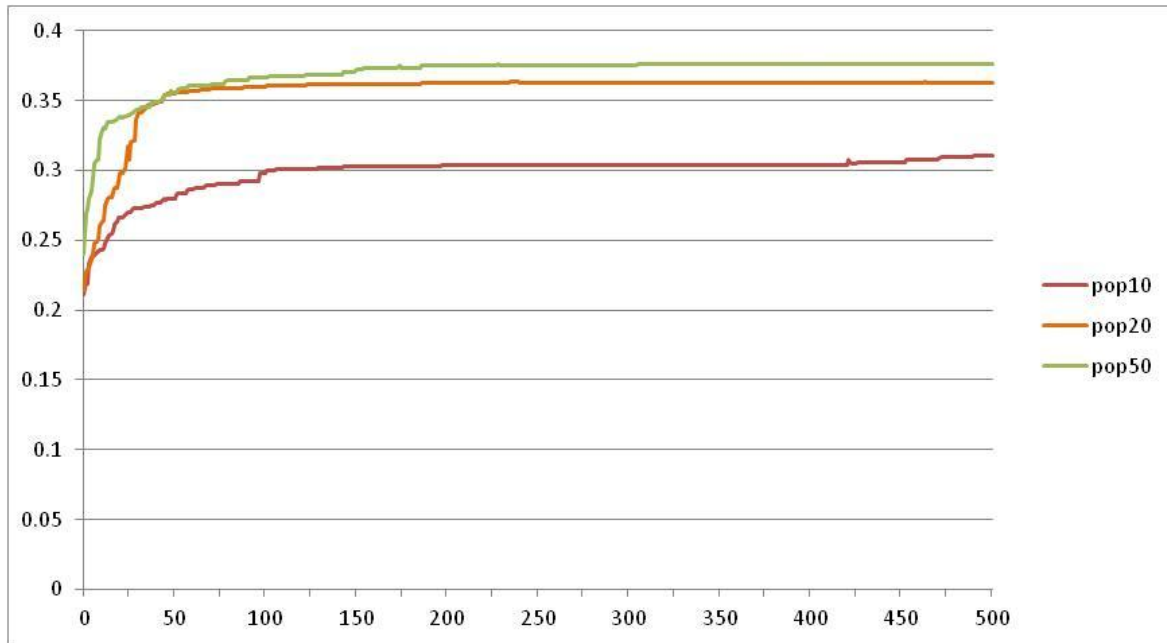


Figure 9. The best BSC values over the generations for different population sizes. (average of 5 tests)

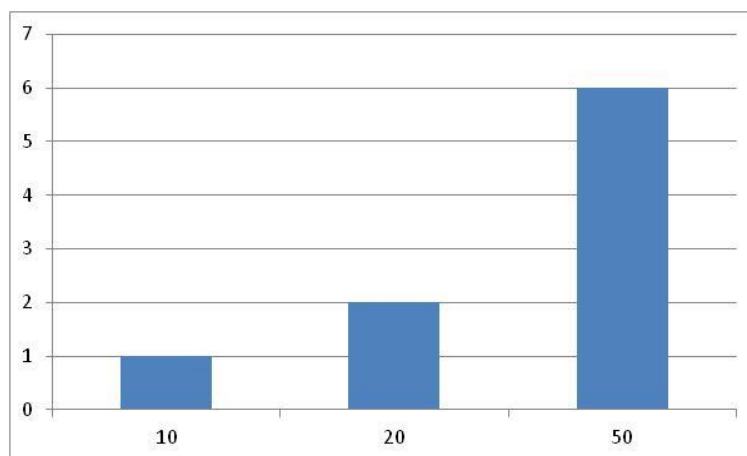


Figure 10. The average running time of one generation for different population sizes (rounded to seconds)

As we can see, the best convergence occurs with 20 and 50 individuals. However, the running time of the generations is highly influenced by the population size. So I decided to use 20 individuals in the populations of the further tests, because this gives mostly the same as the tests with value of 50, but with much less running time.

6.3. Mutation

In order to find the best mutation rate, a fixed population size of 20 and selection rate of 100%. I tested the algorithms with 5%, 10%, 20% and 30% mutation, and I got the following results.

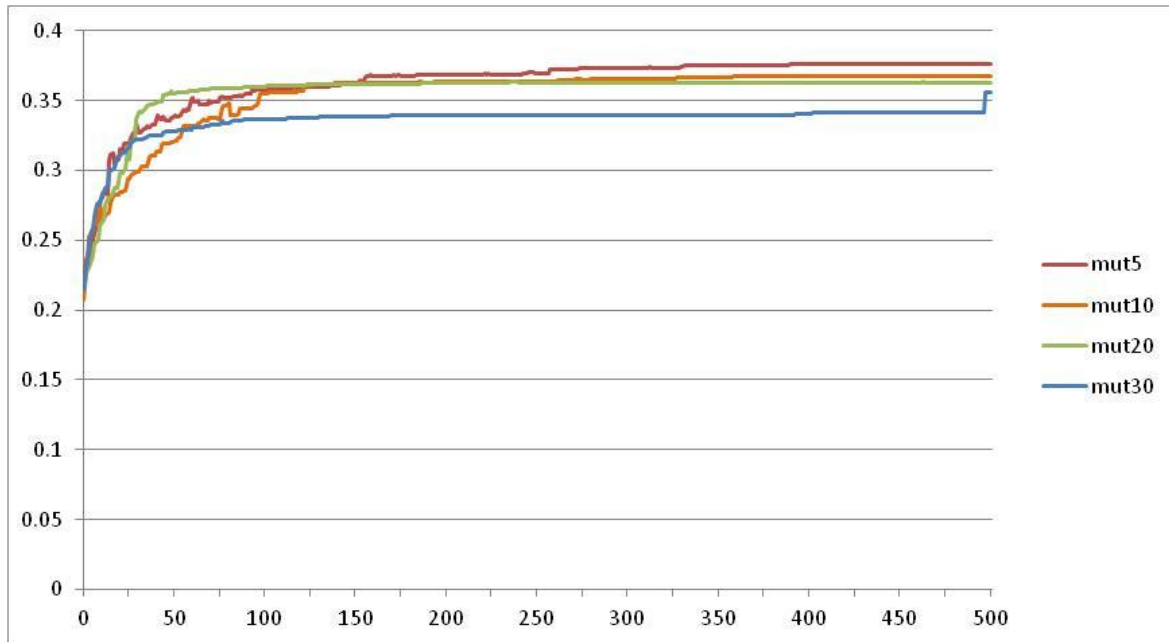


Figure 11. The best individuals over the generations for different mutation percentages.
(average of 5 tests)

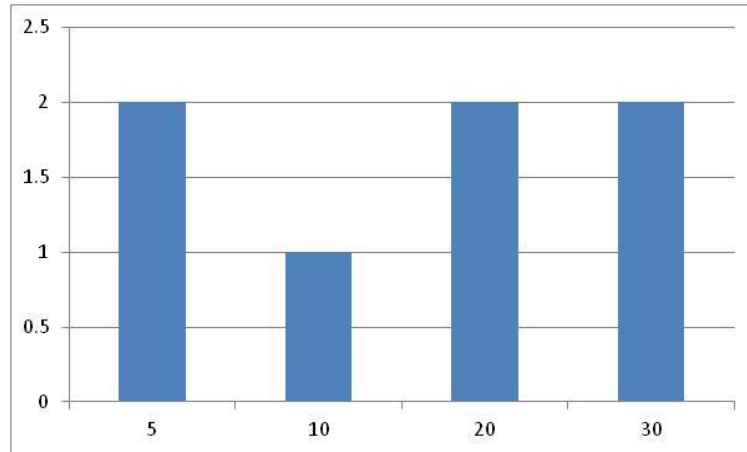


Figure 12. The average running time of one generation for different mutation percentages. (rounded to seconds)

As we can see on Figure 11. and 12., The convergence of the average of the best individuals is the best for the case of 5%, while the different mutation rates does not really affects the running time of the generations. So I will use 5% mutation rate for the further tests.

6.4. Selection

For the tests of the selection rate, I used fixed population of 20 individuals and 5% mutation according to the earlier results. I have tested 10%, 30% 50%, 70% and 100% as selection rate. This means that this percentage of the individuals took part in the recombination (the best ones).

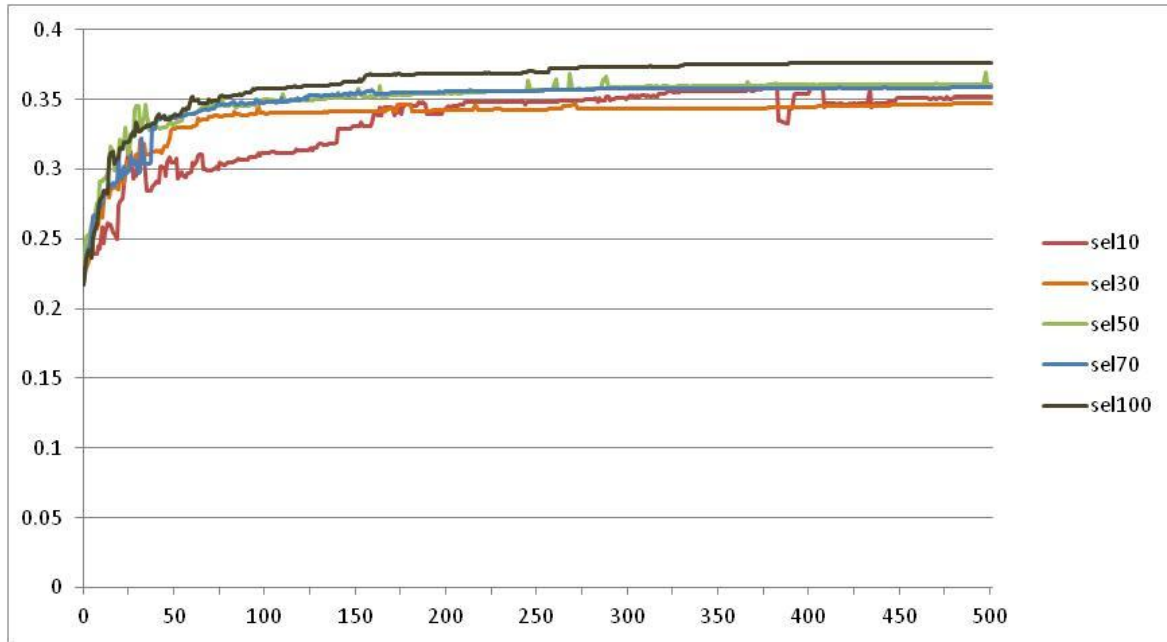


Figure 13. The best individuals over the generations for different selection percentages.
(average of 5 tests)

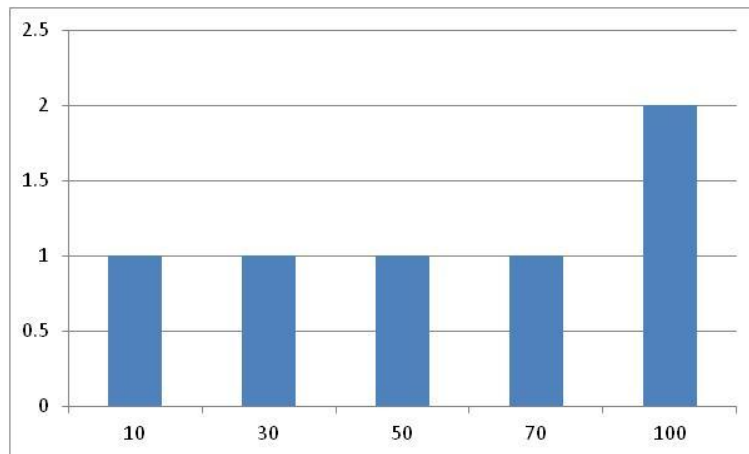


Figure 14. The average running time of one generation for different selection percentages.
(rounded to seconds)

As we can see on Figure 13. and 14., I got the best convergence when the selection rate was 100%. Since the running time still reasonably good for this case, I decided to use this value in the algorithm.

6.5. The best results

According to the tests, I found that the best convergence occurs when I use population size of 20 individuals, mutation rate of 5% and selection rate of 100%. Finally, I got a Software configuration described in Table 3., which has overall Balanced Score Card value of 0.394609. This configuration created a fairly linear consumption profile as we can see on Figure 15.

NG: Ordered Quantities							
0:00-3:00	3:00-6:00	6:00-9:00	9:00-12:00	12:00-15:00	15:00-18:00	18:00-21:00	21:00-24:00
22	23	22	25	5	30	20	27
EV: Charging profile							
<i>PSum</i>	<i>startH</i>	<i>startMin</i>	<i>endH</i>	<i>endMin</i>	<i>mode</i>		
63	0	35	8	0	Constant		
37	17	18	0	39	Rising		
ES: TransPercents							
93							

Table 3. One of the best Software configurations



Figure 15. The change of the profile with the best configuration.

7. Genetic algorithm for Hardware

In this section I will introduce the genetic algorithm for Hardware optimization. I will describe two possible representations of the individuals and I will define the fitness function. Then I will show the results of the tests for both representations made on the same way as in case of Software optimization – I always changed only one parameter, and then I compared the average of the best fitness values. Finally, I will show the effects on the system of one of the best Hardware configurations.

7.1. The individuals and the fitness function

During the optimization of the Hardware module, it is important to note that the quality of the Hardware highly depends on how we use it, that is, what is the optimal Software configuration for them. Considering this I have implemented two different genetic algorithms.

In case of One-Level genetic algorithm, I optimize the Hardware and the Software module together, which means that the chromosomes are simple extensions of the ones used at Software optimization. So I have all the genes described at the Software module - *Ordered Quantities* for the National Grid, charging profiles for the Electric Vehicles and the *TransPercent* value for the storages devices – and I have genes for the physical parameters of the devices. For the latter ones, I have created a list of W, PV and ES devices, and the genes determine only which devices and how many of them are used in the system (*Id* and *Quantity*). I always take a fixed value for the *FullNeed* parameter of the Electric Vehicles module, because I take it that it is part of the facility.

For the Multi-Level implementation, I use only the genes for the Hardware module – *Id* and *Quantity* for W, PV and ES – because I run another genetic algorithm (described in Section 6.) to determine what is the good usage of the infrastructure so the genes for the Software module are used only in that algorithm.

The fitness function is the same for both implementations: I have all the BSC indicators from the Software module to determine how well the system works, and I have one more indicator in the Financial part to consider the return time of the investment to the devices according to the prices and maintenance costs. The formula of this indicator is in Table 4.

Financial	
Return Time	<i>Price</i>
	$(cost^* - cost) \frac{105120}{ Calculation\ Period } - \text{Yearly Maintenance Cost}$
	Where 10512 is the number of 5-minute periods in one year, and $ Calculation\ Period $ is the number of 5-minute period I am calculating with. (In these tests, it is 288 that is for one day)

Table 4. The Balanced Score Card indicators for the Hardware module extending the one for Software.

7.2. One-Level Genetic Algorithm

7.2.1. Population

For testing the population size parameter, I run the algorithm 5 times with values of 20, 50, 100, 200 and 300 with fixed mutation and selection rates. I used the best settings of the Software GA – mutation rate of 5% and selection rate of 100%.

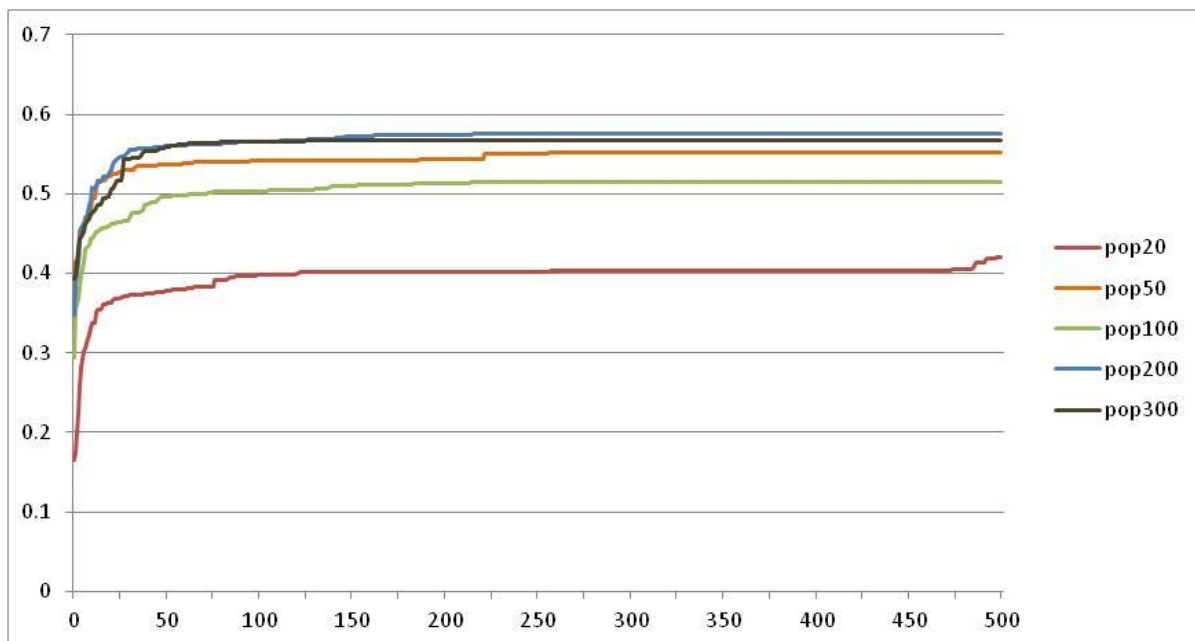


Figure 16. The best individuals over the generations for different population sizes. (average of 5 tests)

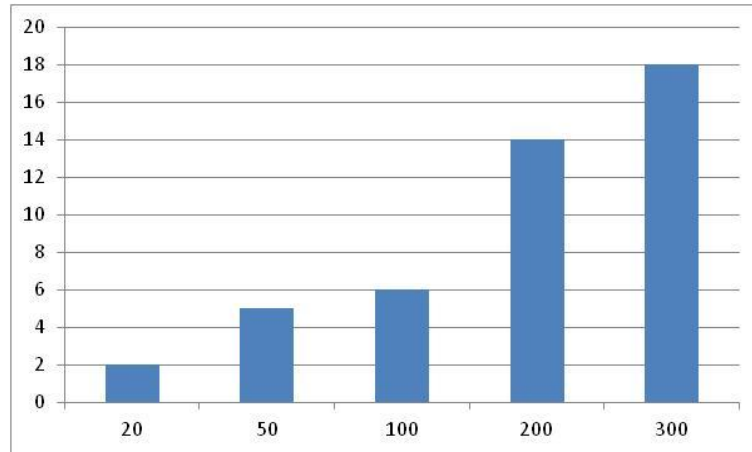


Figure 17. The average running time of one generation for different population sizes (rounded to seconds)

The results of the tests are shown on Figure 16. and 17. I experienced the best convergence for 50, 200 and 300 individuals. As the running time is highly influenced by the population size, I decided to use population size of 50 in the further tests.

7.2.2. Mutation

In case of mutation rate, I tested the values of 5%, 10%, 20% and 30% with 50 individuals and 100% selection rate.

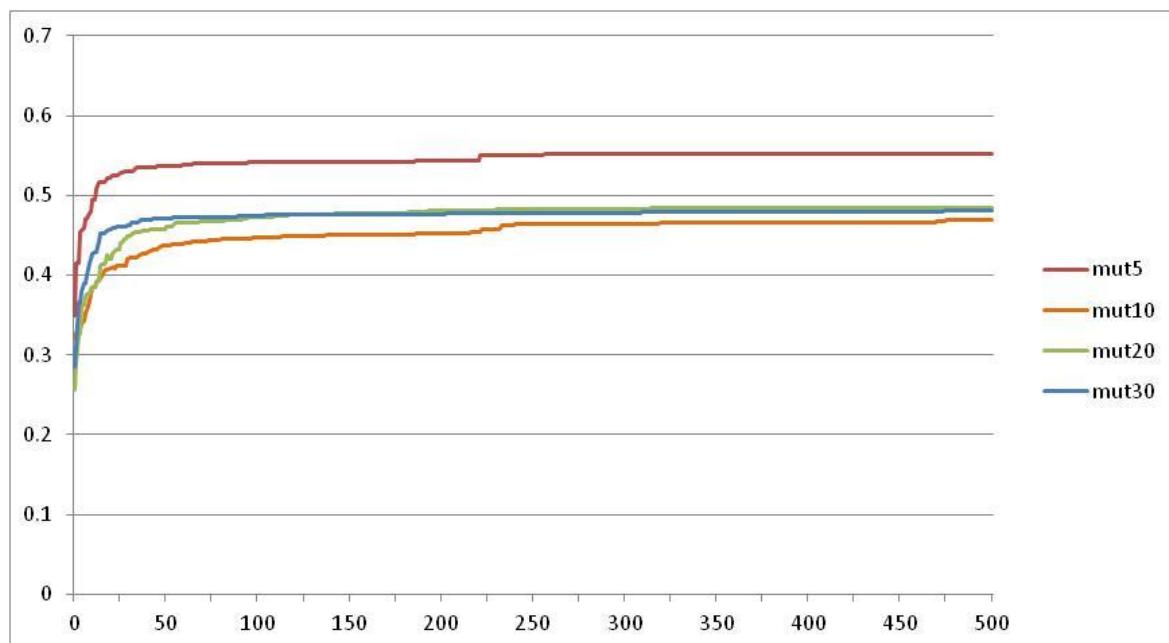


Figure 18. The best individuals over the generations for different mutation percentages. (average of 5 tests)

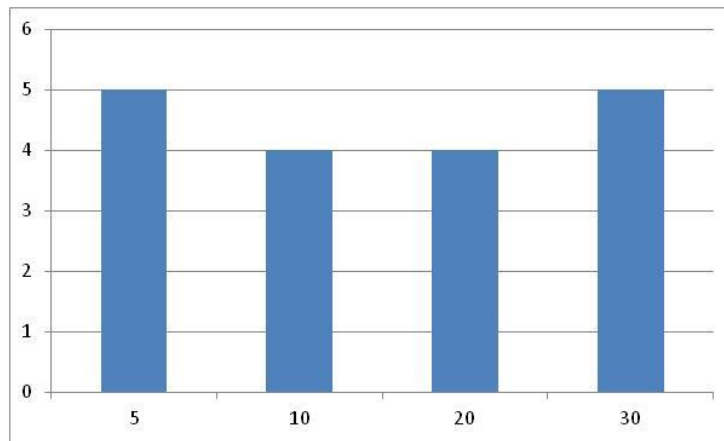


Figure 19. The average running time of one generation for different mutation percentages (rounded to seconds)

As we can see on Figure 18. and 19., the algorithms gives much better results with 5% mutation than any others, and the running time is not really affected by this parameter, so I will use this value.

7.2.3. Selection

For the selection rates I tested the values of 30%, 50%, 70% and 100% with earlier selected fixed parameters – 50 individuals and 5% mutation.

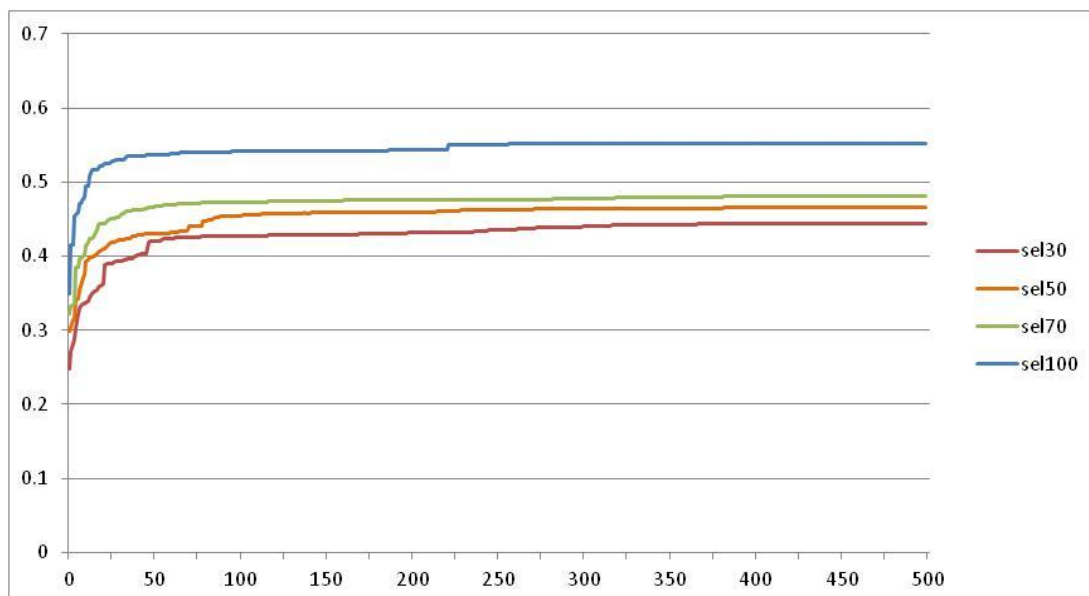


Figure 20. The best individuals over the generations for different selection percentages. (average of 5 tests)

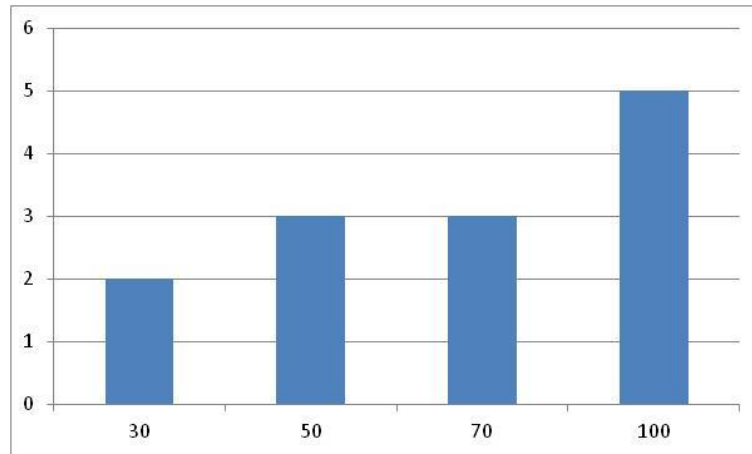


Figure 21. The average running time of one generation for different selection percentages (rounded to seconds)

The results on Figure 20. and 21. show that the algorithm created much better individuals with 100% selection than any others. Although the running time for this setting is worse, I decide to use this value, because of the advantages in fitness function.

7.2.4. The best results

According to the tests, I found that the best convergence occurs when I use population size of 50 individuals, mutation rate of 5% and selection rate of 100%. Finally, I got a Hardware-Software configuration described in Table 5., which has overall Balanced Score Card value of 0.645147. This configuration created a much more linear consumption profile than the original one was. (Figure 22.)

Wpark										
<i>Id</i>	Name	Quantity	Start	Peak	PeakP	Stop	Height	Alpha	Price	Yearly
-	-	-	-	-	-	-	-	-	-	-
PVpark										
<i>Id</i>	Name	Quantity	Start	Peak	PeakP	Stop			Price	Yearly
-	-	-	-	-	-	-			-	-
ESpark										
<i>Id</i>	Name	Max	Efficiency	DownSpeed	UpSpeed				Price	Yearly
2	ES-3	50	70	60	60				2000000	500000

NG: Ordered Quantities							
0:00-3:00	3:00-6:00	6:00-9:00	9:00-12:00	12:00-15:00	15:00-18:00	18:00-21:00	21:00-24:00
21	18	19	18	1	0	21	21
EV: Charging profile							
<i>P</i> Sum	<i>startH</i>	<i>startMin</i>	<i>endH</i>	<i>endMin</i>	<i>mode</i>		
77	8	14	6	3	Rising		
23	21	50	11	24	Falling		
ES: TransPercents							
99							

Table 5. One of the best Hardware-Software configurations by the One-Level algorithms

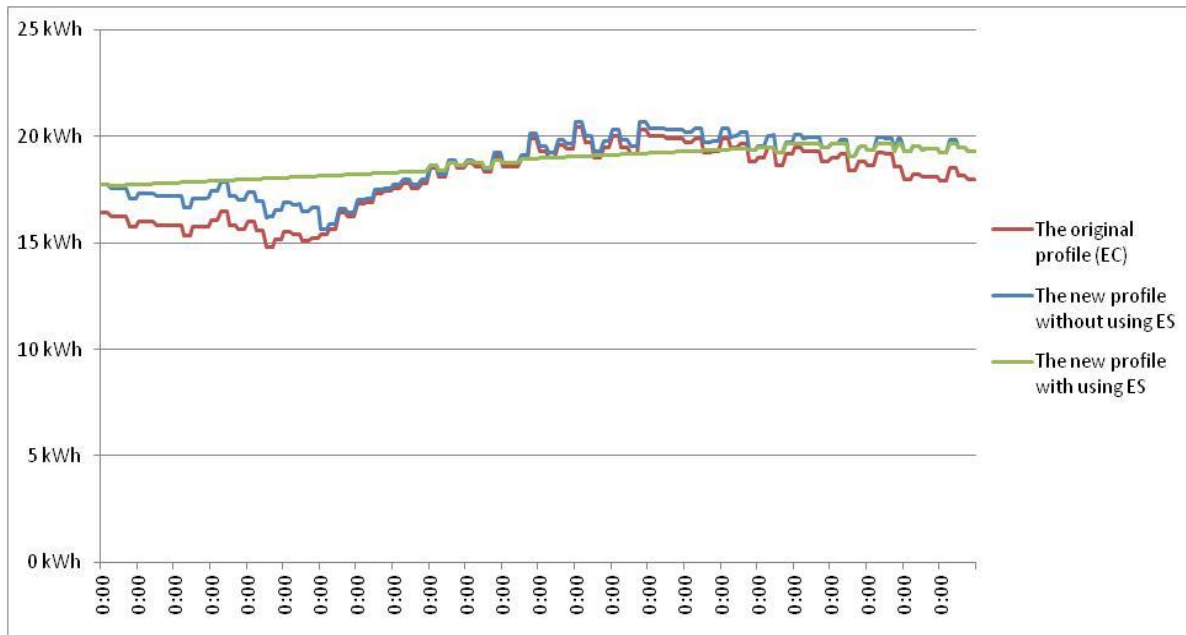


Figure 22. The change of the profile with the best configuration.

7.3. Multi-Level Genetic Algorithm

With this implementation, I try to find the optimal Hardware configuration with GA-in-GA method. This means that the fitness function is calculated such that I firstly run a genetic algorithm to find the good usage of the devices. For this purpose I use the algorithm described in Section 6. To make this faster, I calculate only until 20 generations, because I experienced that a reasonably good solution is already calculated this way, so I can compare the devices. Then I simply calculate the BSC value for the Hardware based on the Software-BSC. So I run a genetic algorithm by this value (mostly the result of another GA) as a fitness function.

The advantage of this method is that I calculate the fitness function with already good Software configurations. However, the running time is a big disadvantage of this implementation, because it is much slower to calculate the fitness on this way. Actually it takes approximately average 1800 seconds to calculate one generation, while this is only 5 seconds for the One-Level algorithm.

For test this algorithm, I used the same parameterizations as for the previous two algorithms – the outer GA is calculated as in Section 7.2, and the inner one is calculated as in Section 6. – because it would take a lot of time to test all the parameters one-by-one for this implementation, and it basically does the same as the two other ones, it is a merge of them. So I run this algorithm 5 times with the parameters described in Table 6. To analyze the results of this algorithm I compare them with the results of the One-Level Genetic Algorithm.

Parameter	Outer GA	Inner GA
Population size	50	20
Mutation	5%	5%
Selection	100%	100%

Table 6. The parameters of the Multi-Level algorithm

As we can see on Figure 23. and 24., the Multi-Level Genetic Algorithm gives much better individuals, but it takes a lot more time. However, I prefer the Multi-Level algorithm against its slow operation, because, as mentioned in Section 4.2, the hardware configuration of a facility must be optimized quite rarely, only at the time of the investments, so it is acceptable to calculate the much better solution such slower.

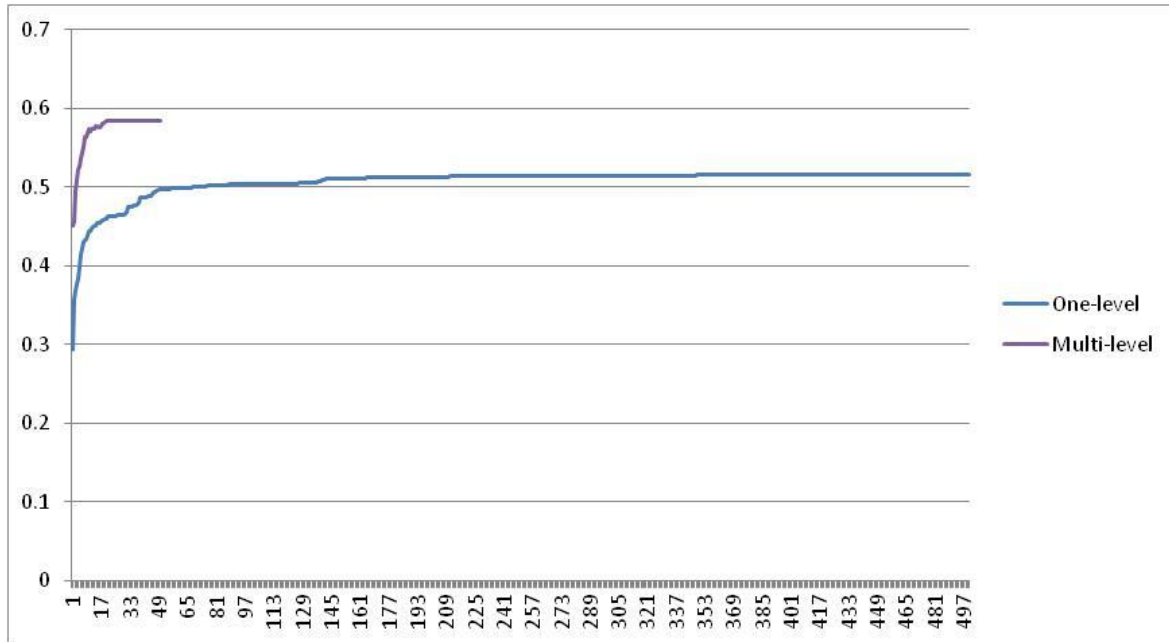


Figure 23. The best individuals over the generations for One and Multi-Level methods.
(average of 5 tests)

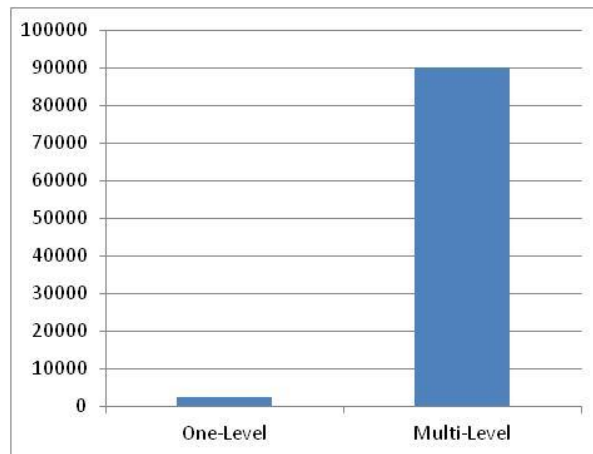


Figure 24. The full running time in seconds for the two genetic algorithms.
(calculated by average values)

Finally, I got a Hardware-Software configuration described in Table 7., which has overall Balanced Score Card value of 0.685617. This configuration created a fairly linear consumption profile as we can see on Figure 25.

Wpark										
<i>Id</i>	Name	Quantity	Start	Peak	PeakP	Stop	Height	Alpha	Price	Yearly
-	-	-	-	-	-	-	-	-	-	-
PVpark										
<i>Id</i>	Name	Quantity	Start	Peak	PeakP	Stop			Price	Yearly
-	-	-	-	-	-	-			-	-
ESpark										
<i>Id</i>	Name	Max	Efficiency	DownSpeed	UpSpeed				Price	Yearly
1	ES-2	200	50	40	40				2100000	350000

NG: Ordered Quantities							
0:00-3:00	3:00-6:00	6:00-9:00	9:00-12:00	12:00-15:00	15:00-18:00	18:00-21:00	21:00-24:00
35	52	19	91	168	34	30	25
EV: Charging profile							
<i>PSum</i>	<i>startH</i>	<i>startMin</i>	<i>endH</i>	<i>endMin</i>	<i>mode</i>		
86	14	46	0	17	Rising		
14	21	53	17	13	Falling		
ES: TransPercents							
95							

Table 7. One of the best Hardware-Software configurations by the Mutli-Level algorithms

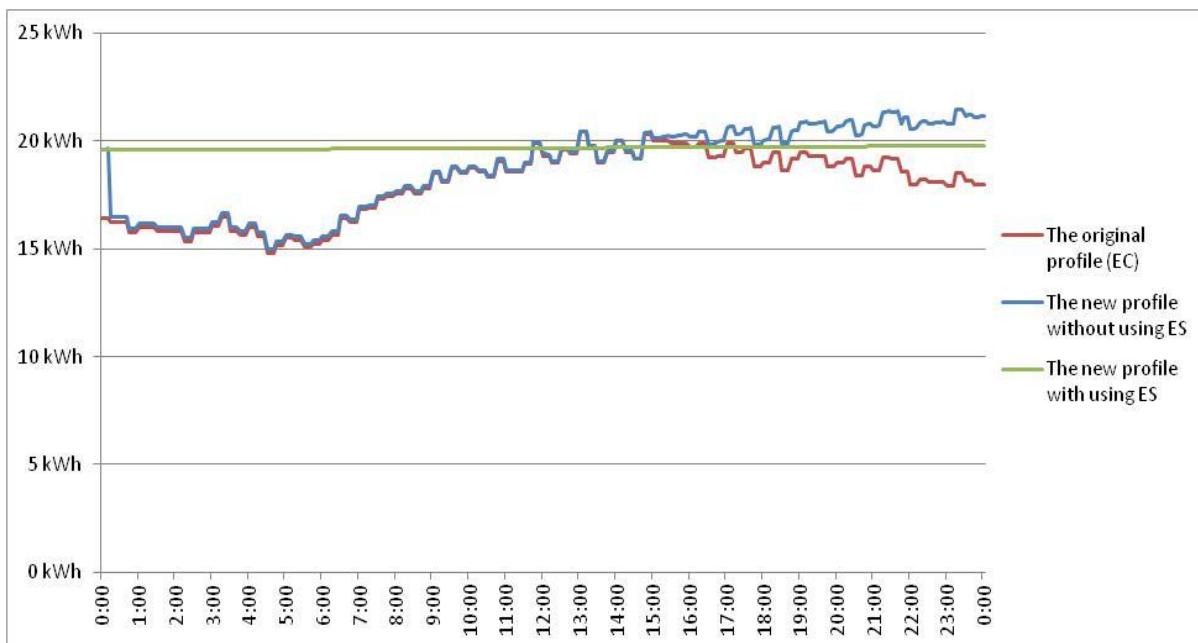


Figure 25. The change of the profile with the best configuration.

8. Conclusions, future plans

As conclusion, I can say that I have created a complex system in order to model and examine the problems of the electric grid, and to optimize the consumption profile from various viewpoints. According to the test results applied on real measured data, EMMA helps to find GRidfriEENd solutions by using genetic algorithms for the optimization of the Software and the Hardware modules. So it provides a useful service for both the suppliers and the industrial facilities.

I was managed to create a Software optimizing algorithm, which is quick enough to use making operative decisions by one 5-minute period to the other. For this purpose, it is crucial to have correct prediction for the consumption for the next periods and the all day. The next step in the developments will be to implement a prediction algorithm, after what the EMMA system will become an even more powerful solution.

9. References

9.1. Conference Proceedings

- A. Bódis, T. Gergely, P. Bajor: Keep the balance with EMMA, Scientific Student Conference, University of Szeged - Faculty of Science and Informatics, 2012.
- P. Földesi, L. Hegyi, P. Bajor: The “The minA Concept” in managing the cooling energy demand of logistics processes, Agricultural Logistics Conference, Novo Mesto – SI, 2011.
- P. Földesi; P. Bajor; A. Bódis; Z. Fűrész; A. Kollár: The GRIdfriEENd cooling I. project, in. 8th International Conference on Logistics and Sustainable Transport, Celje-SI, 2011
- P. Földesi; P. Bajor; M. Baricza; Cs. Kiss; O. Vas: The GreenTrucks Project, in. 7th International Conference on Logistics and Sustainable Transport, Celje-SI, 2010
- A. Dán, D. Divényi, B. Hartmann, P. Kiss, D. Raisz and I. Vokony: Perspectives of Demand-Side Management in a Smart Metered Environment in International Conference on Renewable Energies and Power Quality, 2011

9.2. Books

- M. Mitchell: An introduction to genetic algorithms (1998), pp. 6-8.