SmartSolarGrid
Deciding what to do with Solar Energy production

Diogo Morgado and Paulo Ferreira

INESC-ID / Technical University of Lisbon
diogo.morgado@ist.utl.pt,paulo.ferreira@inesc-id.pt

Abstract. Solar energy has been subject of great development in the past years, which led to the concept of Solar Roads: photovoltaic panels along the highways and roads. SmartSolarGrid is the merge of Solar Roads with Smart Grids, a new electrical distribution grid with improved efficiency and control. The goal of this work is to develop a software tool that further improves the efficiency of the electricity produced by automatically deciding in real time its destination: i) store the energy, ii) sell it to the global national-wide electric company, iii) sell it to the local electric company, etc.. In addition, we developed a software tool for electric cars which gives its driver suggestions about what he can do with the remaining energy stored in the car batteries (e.g. sell if there's enough for that) or where to charge up the battery (e.g. if there's not enough to get to the destination).

Keywords: Smart grid, Decision making, Solar energy, Solar road, SmartSolarGrid

1 Introduction

Mankind is facing a threat from the effects of global warming[1]. Now, more than ever, renewable energy sources should be used instead of fossil fuels, in an effort to fight global warming [1]. Of those types of energy, solar energy is one of the most popular, mostly due to the advances in solar photovoltaic panels technology [2]. The panels are arrays of solar photovoltaic cells that convert the sunlight into electrical energy taking advantage of the photoelectric effect.

One of the disadvantages of solar energy is that, to be produced at an efficient level, a large number of photovoltaic panels has to be used, thus requiring a large area. To tackle this problem, a solution has been proposed to make an effective use of the available area through out the country. It consists in deploying the solar panels in the shape of a tunnel around the highways and roads spread out around the country. This solution is called *Solar Road*. Figure 1 shows examples of such roads.

Together with a renewable power source like the *Solar Road*, *Smart Grids* [3] represent a big improvement over the older grids. A Smart Grid is an improved

[1] http://www.grida.no/publications/other/ipcc_tar/?src=/climate/ipcc_tar/wg1/247.htm

Fig. 1. Examples of possible Solar Roads

electricity distribution grid that manages in a very efficient, reliable, sustainable and economic way the electricity flowing through the system. This management, consists in controlling the electricity flow by gathering information from all the participants of the grid: the suppliers and the consumers. In Europe, Smart grid policy is organized as the Smart Grid European Technology Platform.[2]
To further improve smart grids, we must take into account the destination of the energy being produced by the grid. There are several options about what to do with this energy. In the context of the *Solar Road*, it can be stored in batteries for later use, sold to the nearest electric vehicle (EV) charging station, sold to the main distribution grid, etc.. In addition, taking advantage of the *Solar Road* concept, the new generation of electric vehicles can benefit greatly from this technology. If EV charging stations are spread along the highways, cars can use it to charge their internal batteries or to sell it to the grid in case there's room for it.

1.1 Goal and Requirements

The goal of this work is to come up with two software tools (the SmartSolarGrid Panels (SSG Panels) and SmartSolarGrid Cars (SSG Cars), which will be generically known as SmartSolarGrid.
SSG Panels objective is to automatically decide what to do with the energy produced given a few selected criteria. The alternatives include, for example: sell the energy to the nearest EV charging station, sell to the electric grid company, store in batteries, etc. The criteria to be taken into account is, for example: weather prediction, car traffic prediction, electric energy price, etc.
SSG Cars objective is to give suggestions to the driver about what to do with the energy left in case there's room to sell it, or where to charge it up if the remaining energy is not enough to make it to the destination. The criteria to be taken into account criteria are: the distance to the next EV charging stations, distance to destination, energy costs, etc.

[2] http://www.smartgrids.eu/

The requirements for SSG Panels are: decide in real time and in a semi automated way about what to do with the energy being currently produced; provide remote monitoring, control and configuration operations and provide authentication for the all the operations.
For the SSG Cars the requirements are: give suggestions to the driver about what to do with the stored electricity in a fully automated way and provide a route considering a possible chosen location to trade energy as a way point in the route.
Also, both tools are required to have a good degree of scalability, flexibility, system portability, adequate response time, user friendly interfaces, be able to run in real or simulation mode and use open source technologies.

1.2 Outline

This article is organized in sections: section 1 presents the motivation to the topic and describes the goal and requirements of this work. Section 2 presents the most relevant related work for this article. Section 3 describes the high level architecture of the system. Section 4 presents the evaluation and section 5 is the conclusion of the article.

2 Related Work

The main aspect of SmartSolarGrid is deciding what to do with the energy produced by taking into account multiple parameters like the destination of the energy (Sell to the grid, store it, etc..) and criteria like weather and traffic prediction, electricity price, battery level, etc.. As such, the area that presents relevant work to solve this problem is Decision Making. Multiple Criteria Decision Making (MCDM) is the name given to the techniques used to solve decision making problems. It consists in the study of methods and procedures by which concerns about multiple conflicting criteria can be formally incorporated into the management planning process, as defined by the International Society on Multiple Criteria Decision Making[3].

There are several MCDM techniques that try to give the best result possible given input from the user. The Weighted Sum Method (WSM) and Weighted Product Method (WPM) are the most simple ones, where WSM accepts only units of measurement of the same type and WPM supports any units [4]. Although, they over simplify the problem and are not appropriate when taking into account a decision as complex as SmartSolarGrid requires. SMART [5,6] (Simple Multi-attribute Rating Technique) is also a popular technique due to its simplicity in the user's input required but we ruled it in favour of other techniques out because it involves too much steps. Then we have methods like the Elimination and Choice Translating Reality (ELECTRE) [7], the Technique for Order

[3] http://www.mcdmsociety.org/

A	1	B	4
A	4	C	1
B	9	C	1

Table 1. An example pairwise comparison

	A	B	C	Normalized Principal Eigenvector
A	1	1/ 4	4	0,2200
B	4	1	9	0,7132
C	1/ 4	1/ 9	1	0,0669

Table 2. The resulting matrix

Preference by Similarity to Ideal Solutions (TOPSIS) [8], the Multi Attribute Utility Theory (MAUT) [4,9], Compromise Programming (CP) [10] and Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE) [11,12], which are methods optimized for specific situations. For these reason, we don't take them into account to SmartSolarGrid. The one we favoured the most is the Analytical Hierarchy Process (AHP) [13,14] due to its flexibility, simplicity and basic user input.

2.1 AHP

AHP decomposes a decision problem into a hierarchy with the objective at the top level, the criteria at the middle level and the different alternatives at the bottom. The alternatives are compared two at a time to access their relative preference with respect to their impact on the criteria. To access the preference between two elements, the decision maker should use their judgement about the element's relative meaning and importance, but concrete data can also be used. It is the essence of AHP that human judgements can be used to perform the evaluation. [15].

The judgements are performed using Saaty's fundamental scale of 1-9 [4], where 1 means equal preference, 3 means moderately more preference about one of the elements, 5 strongly preference, 7 very strongly preference and 9 extremely more importance. The 2, 4, 6 and 8 values are used to express intermediate preference. The pairwise comparison is made in such a way that, for example, to compare alternative A against B taking criterion C1 into account, the decision maker assigns one of the previous values to the preferred option and 1 to the least preferred.

Table 1 shows an example where alternatives $\{A, B, C\}$ are measured against each other in a pairwise manner, taking criterion $C1$ into account. After dealing with the pairwise comparison of each element, the information is converted to matrices, from which the weights will be extracted. The weights are calculated using the matrices principal right eigenvectors. Table 2 shows an example using the comparison from table 1. This technique can be fully applied to make the decisions that the requirements state, where the outcome of the decision process is the destination of the electricity. As such, we chose AHP as the MCDM technique used in SmartSolargrid.

3 Architecture

We developed two tools: SSG Panels is responsible for the Solar Road infrastructure and SSG Cars takes care of the system that is used by the electric vehicles.

3.1 SmartSolarGrid Panels

The photovoltaic panels are organized by what we call **Production Sites (PS)**. A Production Site is an agglomerate of photovoltaic panels seen in the system as a single production entity. This means that, for the system, the amount of energy being produced by each PS is equal to the sum of the energy produced by each individual photovoltaic panel contained in that PS.
The infrastructure that supports SSG Panels is a N level hierarchy of servers, where the typical value for N is 3. The top level is composed by the **Central Server (CS)**, which is responsible for computing the decisions based on input from a human operator. The middle level contains the **Zone Servers (ZS)**, which are responsible for relaying any messages received from the CS to the panels. The bottom level is composed by what we call **Location Servers (LS)**, which are in charge of managing one Production Site based on information received from the CS. Figure 2 depicts the architecture described. The PS is composed by hardware dependant on the infrastructure operator, and as such is out of the scope of this article. We abstract this fact and assume the connection between the LS and PS is already in place and working. All the communication is performed through the internet, using a custom and secure (authentication) string based protocol. A private network can be used, since the software is designed to handle any kind of communication network, as long as it supports the typical TCP/IP stack. The Central Server is the core of the system. The main functionality and the decision algorithm is implemented here. We use a straightforward implementation of the AHP algorithm. This means that for each PS, there will be a set of criteria (e.g. Weather Prediction, Electricity Price, etc.) and alternatives (e.g. Sell the energy to the distribution grid, Store in batteries, etc.), that in conjunction with decision input from an operator, will allow the CS to calculate the decisions. To accomplish the flexibility requirement, for each PS, both the criteria and alternatives can be dynamically added or removed and can exist in any number. This way, different Production Sites can have different criteria and alternatives. To help the operator filling the decision matrix required by AHP, each criterion will have an associated value. The values' only purpose is to help the operator filling the decision matrix. However, this introduces a flexibility problem, because each criterion will have different ways of being updated. This problem is partly solved by the use of plugins. This means that each time a criterion is added, if the operator wants its value to be updated, a plugin to update the criterion is necessary. If it is already installed then the criterion is automatically updated. Otherwise, a plugin has to be coded and installed. The application automatically compiles and loads plugins when starting.

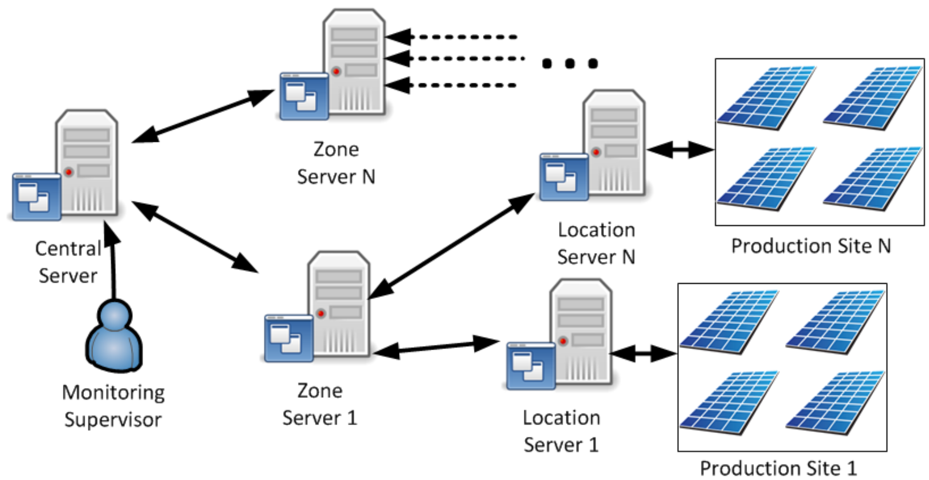

Fig. 2. Overview of the panels' system architecture. The dashed lines mean hypothetical connections.

3.2 SmartSolarGrid Cars

The Cars' application consists of a traditional Client-Server architecture. The client is executed on a portable device in electric cars, while the server can be executed any where (e.g. in the cloud). As with SSG Panels, the communication is performed through any TCP/IP network using the same string based protocol. The necessity of a server arises from the intensive calculations and data transfers necessary to compute the suggestions for the driver. A portable device like a tablet or a smart phone poses several limitations regarding battery consumption, CPU power and memory size and as such a server is necessary.
As with the SSG Panels, the availability of the system wasn't a primary concern, but this tool is also flexible enough to allow an integration with existing solutions. When using the application for the first time, we request the user to introduce the price he pays per kWh at home. We store this value and use it later to aid the suggestions algorithm. He then has two options: get driving directions either with or without suggestions. With this we mean energy trade suggestions, that is, sell or buy energy. We implement an algorithm to calculate them. Either way, the request always goes to the server which responds with the route or the suggestions. If suggestions are requested, the application will give one of three possible options: buy energy, sell energy or do nothing. If the suggestion is to buy or sell energy, a list of recharging stations will be presented on the map and the user can choose which one he will use. After that, the application shows the route taking the chosen station as a way point. To search for EV charging stations, the application is flexible enough to use any kind of searching service.

We use Google's Places API[4], which has limits regarding the number of queries per second. This has a tremendous impact on the algorithm's performance. As such, when the system is used in a real business situation, a professional service to search for the stations should be used instead.
For paths that take more than one battery recharge (recharging step), we calculate the next station when the user reaches the previous one. This repeats until the last recharging step. In case that, while the user is driving, the battery consumption rate changes more than 5% in comparison to the value used to compute the suggestions, we present the user with two alternatives: a safe-to-use-while-driving option that requires him to just push one button and the system recomputes the suggestions and automatically chooses the best one, or the normal way, where the user must choose the station from the map. By best station we mean: if he's buying energy, the one where the amount of money he'll have to spend is the lowest or, if he's selling energy, the one where the amount of money he'll receive is the greatest.

4 Evaluation

We adopted a Test-Driven-Development (TDD) process in which we thought of a testing case for a new function or functionality and then wrote the code in order to pass the test. This means that many implementation errors that might occur on normal development are taken care of and as such, the main evaluation of our software will be performance tests on the decision algorithm for the SSG Panels and on the suggestions algorithm for SSG Cars. The testing set-up, consisted on an Intel Core i7 720QM CPU with a 1.6GHz frequency together with 4GB of RAM memory at 1333MHz. We used the 64 bit Professional edition of Microsoft Windows 7, running JDK 6 Update 33 - 64 bit. It is worth noting that both algorithms should be executed on server grade machines, which implies more computational resources when compared to the machine where we performed the tests. This means that, in a real situation, the execution times are supposedly to be equal or better than the ones achieved in our tests.
The user interfaces were developed having one aspect in consideration: simplicity. Although, the system is flexible enough to allow changes to the interfaces without changing the logic of the tools. In the case of the CS, as the interface is made by html/php files, it is completely independent from the business logic. As such it's easy for a graphical designer to create more eye candy interfaces. Figure 3 show the decision input page of the SSG Panels and the screen to select the route for SSG Cars.

4.1 Decision algorithm

The decision algorithm has a computational time of $O(ca)$, where c is the number of criteria and a the number of alternatives. This time can easily turn into

[4] https://developers.google.com/places/documentation/

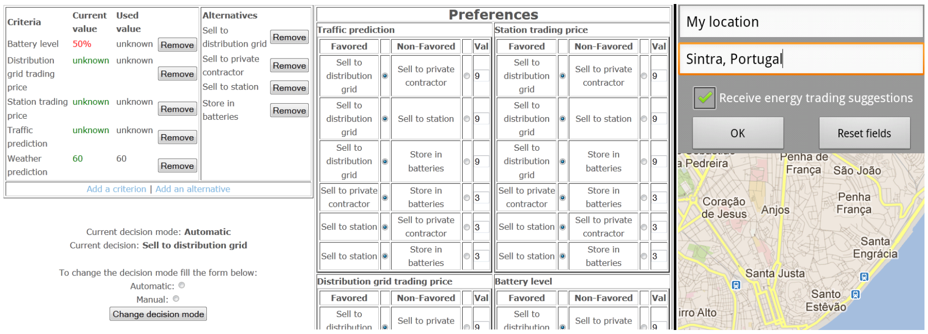

Fig. 3. Interface of the tools. On the left is the panels' and on the right the cars'

$O(n^2)$ if $c = a$, which is the worst case scenario. Using $c = a$, we performed a test to evaluate the response time and memory used by the algorithm. The test consists on measuring the time to compute the best alternative and the amount of memory used. We repeated this test increasing the number of criteria and alternatives by one until a maximum of 100 criteria and alternatives. Due to storing the persistent information in a database, the time taken to read the information from it to memory is also important and as such we also measured this time. When the system is deployed in the field and being used in a real situation, we have to take into account the number of Production Sites, because each PS has an independent set of criteria and alternatives. This means that the results presented in this evaluation have to be multiplied by the number of Production Sites, which is dependant on a specific application of the system to a real situation. Figure 4 shows the results of the tests. We can clearly see that reading the database is the event that takes the most time. However it is well within an acceptable time for the typical case of 5-6 criteria and 3-5 alternatives even when multiplied by a reasonable amount of Production Sites. For example, in the case of $c = a = 10$, that takes only a few milliseconds to read the DB, even if there are hundreds of Production Sites, it should not take more than a few seconds. Even more, we only read the database when the CS is first started and then only on specific events, like changing the preferences, etc.

The time to compute the decision proves to meet the requirements. Even with $c = a = 100$, it takes only about 0.05 ms to compute the decision, which is considered a real time response. The time interval between the several tests is so low, around 0.06 ms, that the variations that appear as huge spikes in the middle of the chart have no practical meaning. This means that the objective was accomplished and the algorithm meets the time requirements to be used in a real situation.

The amount of memory used is also within acceptable ranges for the typical use case described above: it takes only around 1 MB of memory to hold the data structures. Even multiplying by a large number of Production Sites it shouldn't need more than a few dozens of MBytes. Also, since the CS should be executed

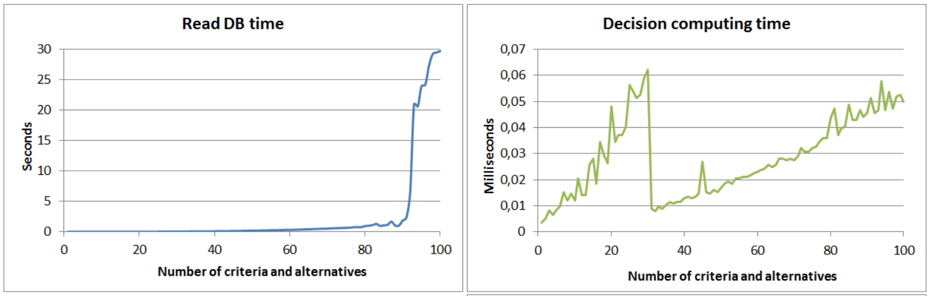

Table 3. Suggestions algorithm results

Locals	Distance (Km)	Time (Secs)	Memory (kB)
Sintra-Lisboa	31,331	2,409	50,227
Sintra-Faro	299,862	3,146	126,734
Sintra-Porto	336,713	3,788	180,680
Sintra-Bragança	541,408	17,852	267,164
Sintra-Madrid	647,858	19,259	224,320
Sintra-Bilbao	884,614	16,945	307,609
Sintra-Barcelona	1.276,972	20,717	386,180
Sintra-Paris	1.762,351	17,243	467,547
Sintra-Brussels	2.067,276	19,066	518,781

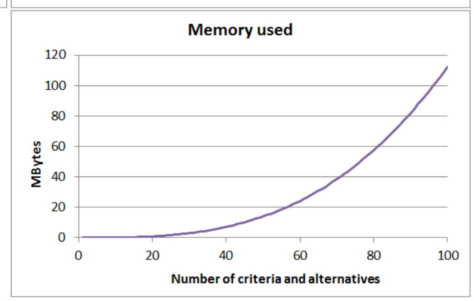

Fig. 4. Decision algorithm results

on a server machine that typically has dozens of GB, the amount of memory needed by the program isn't a problem.

4.2 Suggestions algorithm

The suggestions algorithm performance depends on trip's distance, because as it increases, so does the distance to search for EV charging stations. However, we implemented the algorithm to search only for stations that are within range of the car's current autonomy, which limits the searching distance in case the trip's longer than the autonomy range. This means that when the trip's distance is greater than the autonomy, the algorithm's performance depends on the car's autonomy only. We searched for examples of electric car's range: Nissan Leaf with 175 Km[5] and Tesla Roadster with 394 Km[6]. To perform the tests we used a value of 500 Km autonomy, because it is more than what electric cars can achieve at the present and as such, allows us to test the algorithm having some years in advance. The test consisted of a normal request to the server, using the 500 Km autonomy value and a source and destination city. We then repeated the test using ever increasing trips' distance. We measured the time it took for the algorithm to execute and the amount of memory used. As we mentioned on section 3 we use the Google Maps Places API to retrieve the EV charging

[5] http://newsroom.nissan-europe.com/media/articles/html/75281_1_9.aspx
[6] http://www.greencarmagazine.net/2009/07/tesla-motors-moving-quickly-to-commercialization-of-an-electric-car/

stations, which has usage limits and affects the algorithm's execution time. Table 3 shows the results. From there, we can see that indeed the calculation time increases when trip distance increases, with a considerable increase from around 350 to 500 Km. However, we can make a distinction between the trips. For the average person daily trip ($<$ 100 Km) the response time is within an acceptable range ($<$ 3 seconds). For longer trips, with a distance greater than the car's autonomy (500 Km), it can take up to 20 seconds to perform the calculations. This is not an optimal response time, but given the context of the application, we consider that this time is acceptable. Sometimes, it can take several minutes to acquire a GPS signal and users of this types of applications are used to waiting a few minutes. Also, these values will tend to be lower in cars with lower autonomy, which is the case today.

5 Conclusion

We present a new way to improve the efficiency of electric production in a Solar Road/Smart Grid context. The use of a Multiple Criteria Decision Making technique allow for a semi automated way of choosing the best decision regarding the destination of the energy being currently produced in the system. We also presented a way to give suggestions to drivers of electric cars about what they can do with the energy stored in their car's battery or where to charge it up. The results of the evaluation show that we achieved the objectives. We can conclude that the requirements were met.

References

1. Michael Hoel and Snorre Kverndokk. Depletion of fossil fuels and the impacts of global warming. *Resource and Energy Economics*, 18(2):115–136, 1996.
2. Richard M. Swanson. Photovoltaics power up. *Science*, 324:–892.
3. S. Massoud Amin and B.F. Wollenberg. Toward a smart grid: power delivery for the 21st century. *Power and Energy Magazine, IEEE*, 3(5):34–41, sept.-oct. 2005.
4. S.D. Pohekar and M. Ramachandran. Application of multi-criteria decision making to sustainable energy planning—a review. *Renewable and Sustainable Energy Reviews*, 8(4):365–381, August 2004.
5. George Wright Paul Goodwin. *Decision Analysis for Managment Judgement*. John Wiley and Sons, Ltd, 2004.
6. W. Edwards. Social utilities. *Engineering Economist*, 6:119–129, 1971.
7. B Roy. The outranking approach and the foundations of electre methods. *Theory and Decision*, 31(1):49–73, 1991.
8. C.L. Hwang and K. Yoon. *Multiple attribute decision making: methods and applications*. Springer-Verlag, 1981.
9. Ralph L Keeney and Howard Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge University Press, 1993.
10. M. Zeleny. *Multiple Criteria Decision Making*. McGraw Hill, New York, 1982.
11. J.P. Brans, Ph. Vincke, and B. Mareschal. How to select and how to rank projects: The promethee method. *European Journal of Operational Research*, 24(2):228 – 238, 1986.

12. J P Brans, Bertrand Mareschal, and Ph Vincke. *PROMETHEE: A new family of outranking methods in multicriteria analysis*, pages 477–490. 1984.
13. T L Saaty. Analytic hierarchy process. *Decision Analysis*, 50(1):579–606, 1980.
14. T.L. Saaty. *Decision making for leaders: the analytic hierarchy process for decisions in a complex world*. The analytic hierarchy process series. RWS Publications, 1990.
15. T L Saaty. Relative measurement and its generalization in decision making: Why pairwise comparisons are central in mathematics for the measurement of intangible factors. *Rev R Acad Cien*, 102(2):251–318, 2008.