



**HAL**  
open science

# Optimizing energy management in buildings with Neural Networks

Guillaume Foggia, Thi Thu Hà Pham, Ghaith Warkozek, Frédéric Wurtz

► **To cite this version:**

Guillaume Foggia, Thi Thu Hà Pham, Ghaith Warkozek, Frédéric Wurtz. Optimizing energy management in buildings with Neural Networks. 10th International Workshop on Optimization and Inverse Problems in Electromagnetism (OIPE 2008), Sep 2008, Ilmenau, Germany. pp.Pages 113-114. hal-00326448

**HAL Id: hal-00326448**

**<https://hal.science/hal-00326448>**

Submitted on 3 Oct 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Optimizing energy management in buildings with Neural Networks

G. Foggia, T. T. Ha Pham, G. Warkozek, F. Wurtz

Grenoble Electrical Engineering (G2ELAB)  
CNRS – INPG / UJF

BP 46 - 38402, Saint-Martin d'Hères, France

E-mail: [Guillaume.Foggia@g2elab.inpg.fr](mailto:Guillaume.Foggia@g2elab.inpg.fr)

**Abstract:** Neural networks algorithms are tested to replace an optimization module for the energy management of a residential grid-connected PV system with storage. This study shows multiple problems of the implementation of these tools, despite their theoretical promising interests.

**Keywords:** Energy management, neural networks, optimization, multi-layer perceptron.

## I INTRODUCTION

The rapid development of renewable energy in the residential area and the focus on load management open new possibilities of energy management for the final user, in grid-connected multi-source system with large energy contribution. Such problems meet high uncertainties degrees and computation speed limits and need new tools.

## II GENERAL SET UP AND MODELING

### II.1 Model and optimization background

In a previous work, a power flow management module was proposed to optimize residential grid-connected PV system with storage: processing weather, load, prices data, and parameters of the lead-acid battery model [1]. The module was able to calculate the optimal hourly charging or discharging set-points of the battery, by the use of a Mixed Integer Linear Programming (MILP) solver.

MILP is an extension of Linear Programming where a part of the decision variables are binary. In this way, it is possible to take into account discrete values and to express logical conditions of the problem. Basically, the methods applied to solve MILP problems in practice use a two-level algorithm, where at the upper level, the tree of the binary variables is explored by a Branch & Bound method.

### II.2 Limits of the optimization module

Therefore, MILP is NP-hard: the binary tree has an exponential size, and in most practical case a good solution rather than the global optimum is computed. By using cutting plane techniques, constraint propagation and even heuristics, today's commercial solvers have made impressive progress and are now able to solve in a reasonable time many problems; despite this evolution, the battery energy management is a high-dimensional mixed integer problem, because of the temporal resolution. Furthermore, the optimization has to be done several times per day, to take into account some modification in the inputs: forecasting errors, meteorological variation.

This challenge of time computation is addressed by choosing and testing a different energy management algorithm, based on the offline optimization of power flows for known conditions and the training of Artificial Neural Networks (ANN) to apply this experience on a real power data series (as seen in Figure 1).

### II.3 Objective of the use of ANN

ANN have been widely used for a range of applications in the area of energy modelling and electromagnetism. Some of their application areas are function approximation, pattern recognition, associative memories, generation of new meaningful patterns, optimization [2]. They are accepted as a set of tools offering an alternative way to tackle complex and ill-defined problems. They can learn from examples, are fault tolerant, can deal with non-linear problems and, once trained, can perform prediction and generalisation at high speed.

This study will allow to explore the advantages and inconvenient of the use of such tools, in order to ensure the function of optimized management, and in a second work step, the function of prediction tool.

## III IMPLEMENTATION OF ANN MODULE

### III.1 Architecture choice

The chosen type of ANN is Multilayer Perceptron (MLP). This type is suitable for our application because of its capacity to be a *universal approximator* of non-linear function; moreover, MLP can handle multiple inputs and outputs. MLP used supervised learning with gradient descent algorithm.

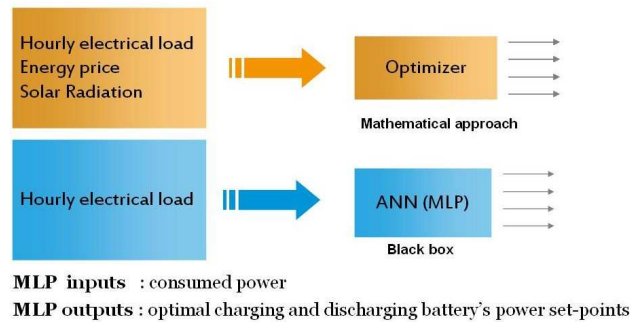


Figure 1: Principle of ANN module

The construction of MLP with *MATLAB Neural Network Toolbox*, as in general, is constituted by two phases: the choice of the building parameters, then the learning phase with appropriate learning data set. This learning data set is built up as following: curves of hourly electrical load for one day are simulated on *MATLAB®*; then, the load data is used as inputs of the optimization module, calculating the power flows and battery storage strategy. These outputs of the optimizer, associated with the corresponding load data, are then used to teach the MLP as reference points. This data base is separated into two subgroups, one for learning phase (30 days), and another for testing.

In fact the most critical part is the first one: indeed, there are no definite methods or bases to describe what would be the optimal MLP architecture for each application case; the existing rules of construction are only empiric. So we have to find out the most appropriate architecture through a preliminary study about the influence of the architecture parameters.

### III.2 Influence of the architecture parameters

We fixed the MLP with the most common number of layers: three layers including the inputs and outputs one's, and the hidden layer. For the inputs and outputs layers, the number of neurons is already fixed: one neuron for each hourly power data, which means 24 neurons in input layer and so like in the third one. What we must find, is now the suitable neurons number in hidden layer. We test the influence of neurons number by varying this last from 20 to 180 neurons. We assume that the most appropriate neuron's number in the hidden layer is around 60 neurons, although learning performance, even in this case, wasn't as expected theoretically.

The influence of the step of the gradient descent is studied too. We made it vary in a large interval (0,02-0,4). As expected, a mean value of gradient descent step (~0.1) is preferable to optimize the performance of learning. Another parameter is the activation function associated to each neuron: we test two types, one with tangent sigmoid and another with linear function; the impact in time learning was obvious for the sake of tangent sigmoid, in addition to better performance. Finally, the influence of initial values of weight is evaluated too.

At the end of this study, the learning performance was not as good as we were expecting; so we have to investigate our approach and the data base to find out the reasons of such low learning performance.

### III.3 Interpretation of the results

Different possible reasons of the low learning performance are investigated. Firstly, it was found that the optimizer module used some stochastic internal variables, which affect partially the outputs. This variation factor was not taken in consideration as inputs of our MLP, perturbing the generalization process by ANN.

Another possible error source is a strong correlation between daily electrical load curves in the learning data set: a new learning strategy is applied, selecting only the "significant days": that means not correlated with the already used data set. A second factor is the possibly not sufficient learning data set: a strategy with an evolving data set is applied, increasing the number of days as long as the performance was not acceptable.

## IV CONCLUSIONS

The ANN module, developed to reduce the computational cost inherent to the optimization, has not shown its potential yet: the major problem met is that the rules to build the structure are mainly empirical. Future improvements of the ANN module are possible, including a prediction function and an on-line adaptive training.

## REFERENCES

- [1] Wurtz F., Bacha S., Ha. Pham T. T., Foggia G., Roye D., Warkozek G., "Optimal Energy Management in buildings: sizing, anticipative and reactive management", Energy Management system Workshop – Torino 24 - 25 May 2007
- [2] Simpson P. K., "Artificial neural systems : foundations, paradigms, applications and implementations". Elmsford, NY : Pergamon press, 1990.