

# Fast architecture generation and evaluation techniques for the design of large power systems

L. Laguna, R. Prieto, J. A. Oliver, J. A. Cobos  
Centro de Electrónica Industrial (CEI)  
Escuela Técnica Superior de Ingenieros Industriales  
Universidad Politécnica de Madrid  
José Gutierrez Abascal 2, Madrid, Spain  
Tel.: +34 913363191 Fax: +34915645966

H. Visairo-Cruz  
Systems Research Center, Mexico  
Intel Corporation  
Guadalajara, Mexico  
e-mail: horacio.visairo-cruz@intel.com

**Abstract**—This paper presents a methodology used to simplify the design of power supply systems based on high level of abstraction simulation models. This approach allows the designer to make decisions concerning the power system architecture and its components. This paper is focused on the techniques to identify topology candidates that can be built with available voltage regulator technologies and find the solution with best trade-off among energy efficiency, size and cost. In order to solve the problem when the number of possible options is very large, metaheuristic algorithms are used.

## I. INTRODUCTION

The design of computing platforms requires fast simulation tools to accurately analyze the system behavior while making quick trade-off analysis of important parameters that affect the entire platform. Long simulation times of today's modeling and simulation tools based on complex models could represent an issue at the initial stage of the design since designers are required to quickly arrive at practical solutions to significantly reduce the time and cost associated with a new product development. Therefore, a design methodology is required to accelerate the time to market of new products by finding the best solution in the shortest time.

Some of the scenarios that a power systems designer must face are:

- Evaluation of a new power conversion architecture that can be adopted in the current design.
- Developing new designs from the ground up that incorporate new and existing technologies.

Both cases involve the evaluation of a significant number of variations to know with good certainty whether the selected architecture and components meets the system requirements. The critical design factors are:

- **Energy efficiency:** minimizing the energy lost in a power architecture is a key factor. The power loss of the architecture can ultimately affect the thermal dissipation and in the case of mobile devices, the autonomy.
- **Size:** Especially in mobile applications, size allowances for power architectures are very constraining. In most cases, the amount of area used by the power architecture is an important consideration.

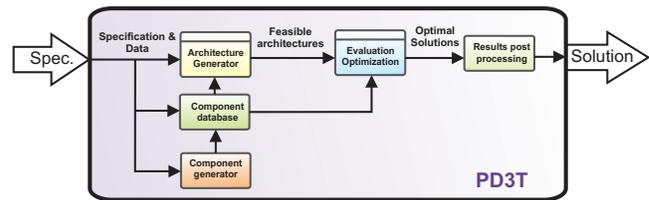


Figure 1. Modules of the Power Delivery and Distribution Design Toolkit.

- **Cost:** The monetary cost of the architecture is a fundamental criteria for judging power architectures.

In [1] has been introduced a methodology to automatically design and optimize the power distribution system. This methodology is implemented in the Power Delivery and Distribution Design Toolkit (PD3T). The modules that comprehend the PD3T are shown in figure 1. The design approach used by the PD3T consists in the following steps:

- 1) Capture a database of converter models.
- 2) An architecture generation algorithm is run to obtain all suitable ways to connect the converters in the database based on the platform specifications (power sources and loads).
- 3) Perform a final optimization in order to select the most appropriate converters based on the platform specifications.

At the end of this process the designer obtains a list of solutions that includes the options with the smallest size, cost and losses, and also the ones with the best trade-off in between these extremes. The final step is the selection of the preferred solutions in order to perform a more detailed simulation with any other specialized simulator or create physical prototypes.

This toolkit helps the designer to make decisions on the main aspects that affect the overall performance (cost, size, efficiency) of the platform: the selection of the architecture and its components.

The appropriate selection of the architecture is very important. Consider the following example: supply 1.1 V to a microprocessor from a battery of 12 V. Typically this is done by using a single stage of 12V to 1.1 V, but in the state

of the art have been presented solutions with two stages that present better performance [2], [3]. These solutions use a high efficiency pre-regulator that allows to using as second stage a converter with more efficiency. In [2] is presented a switched capacitor converter with a very small size and cost for light load applications. On the other side, in [3] it is presented a converter with coupled inductors (more size) yet capable of provide more output power than the first one. The selection of the architecture (single stage or two stages) and the converters (switch capacitor or coupled inductors) will depend completely on the platform characteristics. The PD3T is ideal to help the designer to make such decisions.

This paper is focused on the architecture generation problem, that is: given as input a database of models obtained from different power converter technologies (like switched capacitors, linear regulators and switching regulators) and the specifications of all the different loads and sources in the platform; all possible ways to connect the converters (all architectures) that are feasible in terms of losses, size and cost must be found.

The main difficulty in this type of analysis is that for real platforms the number of architectures that it is possible to build its very large. In most cases if the number of loads and converters in the database is large it is not possible to evaluate every single option because it could take a very long time, and in some cases the problem may be intractable<sup>1</sup>. In order to obtain interesting solutions in a short period of time, metaheuristic optimization algorithms are used to handle the complexity of the problem.

Section II summarizes the modeling and performance evaluation methodology of a power architecture. Section III presents the approach used to solve the architecture generation problem. Section IV shows a complete example of the design of a real size platform and presents the experimental results. Conclusion of this work is presented on Section V.

## II. MODELING METHODOLOGY AND EVALUATION OF POWER ARCHITECTURES

The presented methodology, as mentioned before, consists on the automatic evaluation of large number of power architectures. The main problem encountered is that conventional modeling and simulation tools used in power conversion systems are usually time consuming due to the use of complex models. In addition, simulation time can increase as the number of combinations to analyze increases. In order to quickly evaluate the feasibility of a power system, the behavioral modeling approach is used. This approach has the advantage that simulation models are simpler, therefore faster simulations can be performed.

The typical blocks of power architectures are: converters, loads and sources. Other blocks like filters and protections are considered as part of the converter. In order to calculate the

<sup>1</sup>A problem is considered intractable when, using the existing algorithms, the problem is still so big that cannot be solved even using a very powerful computer. Refer to [4] for more information.

energy efficiency and battery life it is necessary to define a loss model for every block of the power system.

The main idea behind the behavioral models is to create a simplified model that mimics the response of a system for a defined number of input and output variables. In dc-dc converters the losses can be modeled by the equation 1.

$$v_{in} \cdot i_{in} = v_{out} \cdot i_{out} + losses(i_{out}, v_{in}) \quad (1)$$

In typical converters, when the influence of the input voltage is negligible, the *losses* function can be approximated by a polynomial function (eq. 2).

$$losses(i_{out}) = \sum_{i=0}^n a_i \cdot i_{out}^i \quad (2)$$

Other effects can be added to the losses function if required, ex. dependency of temperature. A more detailed explanation of the behavioral modeling and the evaluation of the architectures can be found in [1], [5], [6].

The behavioral modeling approach has the advantage that it is not necessary to know details about the converter; the model can be obtained from various sources based on the datasheets, experiments, simulations or analysis. If a higher level of accuracy is required, it is possible to use more sophisticated models in each case at the expense of more simulation time.

The cost and area values are considered relative because it is very difficult to calculate with the given information the actual area and cost that the final prototype will present. In this case the size and cost of an architecture can be approximated by simply adding the cost and size of the individual components.

Once it is possible to model and evaluate the performance of a power architecture, the second step is to find the feasible power architectures to solve the problem.

## III. ARCHITECTURE GENERATION

The architecture generation problem can be solved for architectures with a few loads by using combinatorial techniques. But as stated before, for many loads the problem becomes more difficult, making necessary to use another smarter approach.

Metaheuristic algorithms, like Harmony Search, Genetic Algorithms, Simulated Annealing, Ant Colony, etc. have proven to be very useful in the optimization of many engineering problems [7]. These kinds of algorithms are problem dependent; therefore it is not possible to guarantee that one algorithm is better than another when it is applied to any problem. Using a Metaheuristic algorithm does not solve the problem by default, it is necessary to define the appropriate optimization variables or a data structure representation. If an incorrect data structure is used, the algorithm may not have good convergence or may become stuck in a local optima without finding a good solution. Like in the case of Evolvable Hardware [8], where the typical approach is the Cartesian Genetic Programming [9] that eases the representation of digital blocks in terms of evolutionary algorithms.

The proposed architecture searching algorithm can be summarized in the following steps:

- 1) Classification of converters into categories
- 2) Calculation of branches
- 3) Search for architectures

These steps are detailed in the following subsections.

### A. Classification of converters into categories

Considering that in the database there is more than one converter with similar or very close characteristics (input voltage, output voltage, maximum output power and efficiency), these converters can be grouped to simplify the architecture search. This is typical when converters from different manufacturers are used because they provide products to target specific applications.

The first classification criterion is the input/output voltage ratio. If the converter has variable output voltage different models are considered for each configuration. Once the converters are grouped by voltage ratio a clustering algorithm (k-means) [10] is used to classify the converters by maximum output power and efficiency at full load. This way all converters from different manufacturers or with different design are grouped and the total number of converters used to search architectures is reduced.

It should be noticed that at this step, a group of converters is represented by a single generic converter. The selection of the specific converter that is going to be used is performed in a further step of the methodology by the Evaluation/Optimization module.

### B. Calculation of branches

At this step, all possible ways to supply a given voltage to each individual load are calculated based on the reduced set of converters. This step is performed using a combinatorial search. In this case, the combinatorial search is viable because the number of possible branches for a single load is constrained. Branches are the basic elements to build architectures. Figure 2 shows two pairs of branches for two different loads. Using these branches four different architectures can be built without creating buses.

Typically in power supply systems, more than three cascade stages are not recommended, this limits the maximum length of a branch. This parameter can be defined by the designer in case a specific number of stages is needed.

Once all the basic elements are defined (all branches), the following step is to select the branches in an appropriate way in order to obtain all the feasible architectures.

### C. Search for architectures

The number of architectures that can be built with the defined branches (for any real application) is very large. At this step the metaheuristic algorithms are used to find optimal architectures among the wide solution space. The evaluation criteria of an architecture is based on the following rules:

- All converters must operate within its recommended range.
- There should not be over-dimensioned voltage rails.

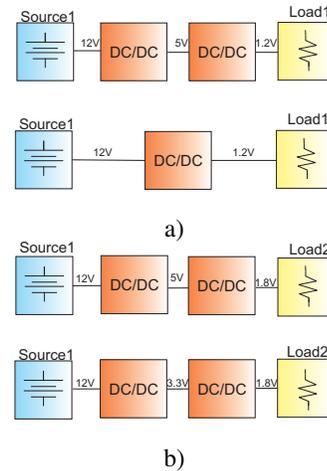


Figure 2. First step of architecture generation algorithm: calculation of the branches. a) Branches for Load1. b) Branches for Load2.

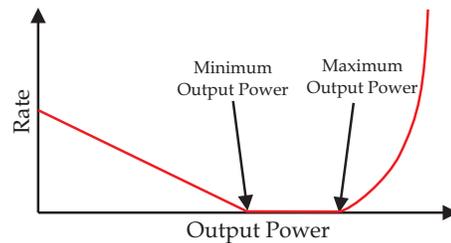


Figure 3. Shape of the function used to rate the architecture stages.

- The architectures should not exceed the maximum number of processing stages.

Based on these rules, a numeric value can be defined to rate each architecture. An example of the used function to rate each converter is shown in figure 3. It can be seen that the function has a range where the rate is zero. This range is defined between the maximum output power of the converter and a minimum that can be defined by the designer as a percentage. Using this rating an architecture is considered good candidate if the rate is very close to zero. This value is the objective function to minimize using the evolutionary algorithm.

Evolutionary algorithms need an objective function and a list of parameters to optimize. In the architecture generation what the evolutionary algorithm is trying to optimize is the appropriate selection of branches and bus (voltage rails) definition.

The steps performed by the algorithm can be summarized as follows:

- 1) Create an initial set of random architectures.
- 2) Select one architecture and apply a modification.
- 3) Evaluate the new architecture and based on the objective function keep the ones with best fit.
- 4) Repeat steps 2 and 3 until a termination criteria is met.

The step 2 is one of the key points of the algorithm; it takes one architecture, removes probabilistically some branches and

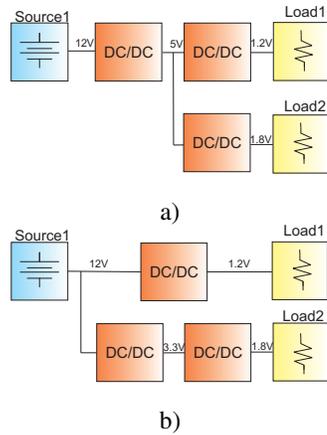


Figure 4. Second step of architecture generation algorithm: searching for architectures. a) Example with pre-regulator. b) Example with a single stage for Load1.

selects randomly new branches to replace those removed. Iteration by iteration the architectures with bad performance are replaced with new ones that present better characteristics. After finishing the process, the feasible architectures are obtained. The number of architectures obtained depends on how many architectures the designer wants to keep. A reasonable number is recommended (less than 1000) in order to analyze a large number without drastically affect the total evaluation time.

Figure 4 shows two architectures that can be built using the branches in figure 2.

#### IV. EXPERIMENTAL RESULTS

In this section a handheld device consisting in 63 loads is designed. The characteristics of the loads are shown in table I.

The database of converters consists in a total of 56 converters that include LDO and switching regulators from different manufacturers. The first step in the architecture search is the classification of the converters into categories. After running the first step, these 56 converters are classified into 33 categories. The details of the groups are presented in table II. It can be seen from the table that each category has defined a range of output power. This range goes from the 80% of the maximum output power to the 100%<sup>2</sup>.

After running the second and third steps of the architecture search, a total of 100 feasible architectures is obtained<sup>3</sup>. One of the 100 architectures is shown in figure 5.

The next step is to select the appropriate converter in each group in order to find the solutions that present good characteristics. With the 100 architectures obtained, around 1.2 millions of possible solutions can be built. Using multi-objective evolutionary techniques the solutions that

<sup>2</sup>The minimum percentage can be defined by the designer.

<sup>3</sup>In this example only 100 are kept, but this number can be defined by the designer.

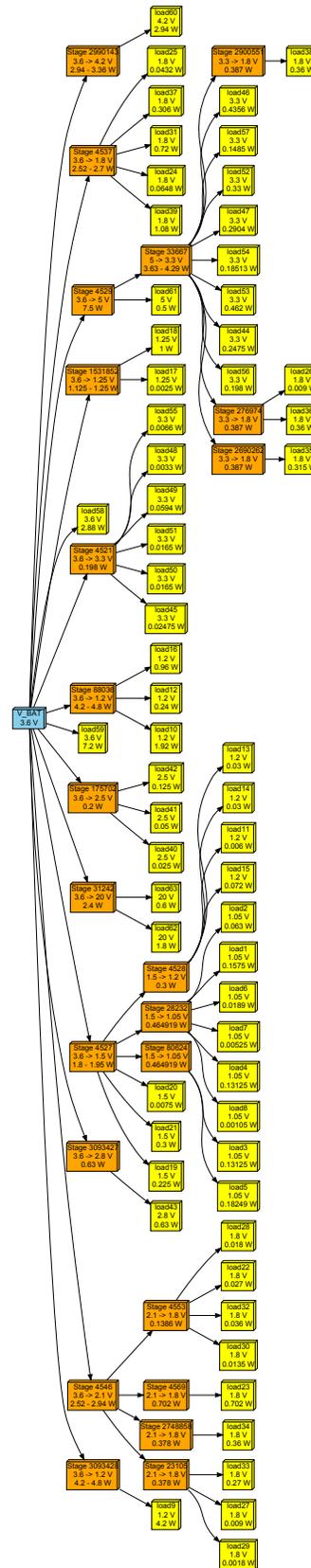


Figure 5. Example of one of the generated architectures in the example.

Table I  
CHARACTERISTICS OF THE 63 LOADS USED IN THE EXAMPLE

Load name	Voltage (V)	Power (W)	Load name	Voltage (v)	Power (W)
load1	1.05	0.1575	load33	1.8	0.27
load2	1.05	0.063	load34	1.8	0.36
load3	1.05	0.13125	load35	1.8	0.315
load4	1.05	0.13125	load36	1.8	0.36
load5	1.05	0.18249	load37	1.8	0.306
load6	1.05	0.0189	load38	1.8	0.36
load7	1.05	0.00525	load39	1.8	1.08
load8	1.05	0.00105	load40	2.5	0.025
load9	1.2	4.2	load41	2.5	0.05
load10	1.2	1.92	load42	2.5	0.125
load11	1.2	0.006	load43	2.8	0.63
load12	1.2	0.24	load44	3.3	0.2475
load13	1.2	0.03	load45	3.3	0.02475
load14	1.2	0.03	load46	3.3	0.4356
load15	1.2	0.072	load47	3.3	0.2904
load16	1.2	0.96	load48	3.3	0.0033
load17	1.25	0.0025	load49	3.3	0.0594
load18	1.25	1.0	load50	3.3	0.0165
load19	1.5	0.225	load51	3.3	0.0165
load20	1.5	0.0075	load52	3.3	0.33
load21	1.5	0.3	load53	3.3	0.462
load22	1.8	0.027	load54	3.3	0.18513
load23	1.8	0.702	load55	3.3	0.0066
load24	1.8	0.0648	load56	3.3	0.198
load25	1.8	0.0432	load57	3.3	0.1485
load26	1.8	0.009	load58	3.6	2.88
load27	1.8	0.009	load59	3.6	7.2
load28	1.8	0.018	load60	4.2	2.94
load29	1.8	0.0018	load61	5.0	0.5
load30	1.8	0.0135	load62	20.0	1.8
load31	1.8	0.72	load63	20.0	0.6
load32	1.8	0.036			

Table II  
GROUPS OF CONVERTERS USED IN THE EXAMPLE

Input Voltage (V)	Output Voltage (V)	Output Power Range (W)	Input Voltage (V)	Output Voltage (V)	Output Power Range (W)
1.5	1.05	0.13 - 0.163	3.6	1.5	0.24 - 0.3
1.5	1.05	0.05 - 0.063	3.6	1.5	1.44 - 1.95
1.5	1.05	0.372 - 0.465	3.6	1.8	0.245 - 0.36
1.5	1.2	0.005 - 0.006	3.6	1.8	2.016 - 2.7
1.5	1.2	0.24 - 0.3	3.6	2	1.44 - 1.8
1.5	1.2	0.336 - 0.42	3.6	2.1	2.016 - 2.94
1.8	1.25	0.002 - 0.002	3.6	2.1	1.68 - 2.1
1.8	1.5	0.18 - 0.225	3.6	2.5	0.16 - 0.2
2.1	1.8	0.562 - 0.702	3.6	2.8	0.504 - 0.63
2.1	1.8	0.111 - 0.139	3.6	2.8	1.12 - 1.4
2.1	1.8	0.302 - 0.378	3.6	3.3	0.158 - 0.198
3.3	1.8	0.31 - 0.387	3.6	4.2	2.352 - 3.36
3.6	1.2	3.36 - 4.8	3.6	5	1.4 - 1.75
3.6	1.2	3.84 - 4.8	3.6	5	6 - 7.5
3.6	1.2	1.92 - 2.4	3.6	20	1.92 - 2.4
3.6	1.2	1.44 - 1.92	5	3.3	2.904 - 4.29
3.6	1.25	0.9 - 1.25			

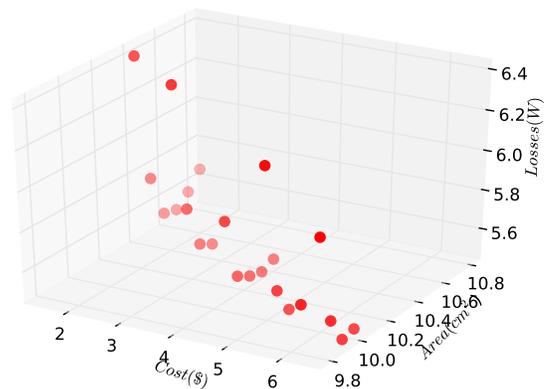


Figure 6. Plot of the results returned by the PD3T. Cost vs Area vs Losses.

belong to the pareto front are obtained. The results are summarized in figure 6 in a 3D view, and the corresponding projections of the three axis are shown in figure 7. In figures 7 the size of the markers represent the value of the third axis.

Figure 7.a shows that solutions with small area and losses tend to be more expensive. In figure 7.b it can be seen that an inexpensive solution and with small losses can be obtained at the expense of larger area. From figure 7.c we can conclude that for solutions that have very similar cost, if the area is reduced, the losses are increased.

The selection of the final solution needs to be performed by the designer according to the platform design priorities. For this purpose a new function that assigns weight to each objective (cost, area, losses) can be defined to simplify the selection task. In the presented methodology the weight function approach is not used because it is very sensitive to scaling and some interesting solutions may be lost.

## V. CONCLUSIONS

This paper presents a new approach to automatically design power architectures considering the critical design aspects like size, efficiency and cost. This methodology has been used in the design of power supply systems for mobile computing platforms where the constraints are very strict.

The presented approach allows to:

- Analyze a huge number of solutions in a short time.
- Eliminate solutions that will not provide good results in early analysis to save computation time.
- Select the most appropriate candidates to accomplish a good trade-off among cost, size and energy efficiency.

The presented architecture search algorithm makes possible to design power supply systems of any size. Conventional

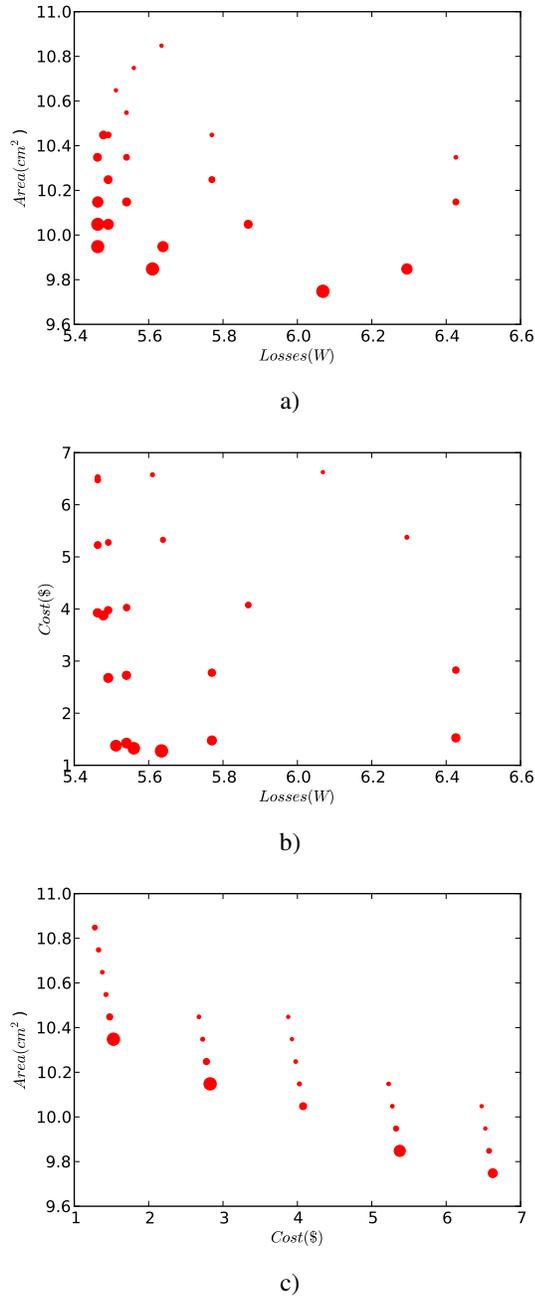


Figure 7. Graphic representation of the results obtained in the presented example. A designer can choose, among these valid architectures, the option that best suits the specifications. a) Cost vs. Area b) Area Vs. Losses c) Losses vs. Cost.

search techniques, like combinatorial search, fail to achieve useful solutions in power supply systems with a large number of loads.

This same approach can be used in the design of many different types of platforms like mobile devices, web servers, aerospace and automotive systems. Benefits of the presented approach have been demonstrated through simulation results of complex architectures used as an example.

## REFERENCES

- [1] L. Laguna, R. Prieto, J.A. Oliver, J.A. Cobos, H. Visairo, and P. Kumar. Power conversion modeling methodology based on building block models. *Energy Conversion Congress and Exposition, ECCE*, pages 3404–3410, 2009.
- [2] Julu Sun, Ming Xu, Yucheng Ying, and F.C. Lee. High power density, high efficiency system two-stage power architecture for laptop computers. In *Power Electronics Specialists Conference, 2006. PESC '06. 37th IEEE*, pages 1–7, 18–22 2006.
- [3] M.C. Gonzalez, P. Alou, O. Garcia, J.A. Cobos, and H. Visairo. Two-stage power architecture for voltage regulator application based on coupled magnetic structure. In *5th International Conference on Integrated Power Electronics Systems (CIPS 2008)*, 11–13 2008.
- [4] Herbert S. Wilf. *Algorithms and Complexity*. A.K. Peters, Natik, Massachusetts, 2 edition, 2002.
- [5] J.A. Oliver, R. Prieto, V. Romero, and J.A. Cobos. Behavioral modeling of dc-dc converters for large signal simulation of distributed power systems. *Applied Power Electronics Conference and Exposition, APEC*, 2006.
- [6] R. Prieto, L. Laguna, J.A. Oliver, and J.A. Cobos. Dc/dc converter parametric models for system level simulation. *Applied Power Electronics Conference and Exposition, APEC*, 2009.
- [7] Z. Michalewicz. *Evolutionary Algorithms in Engineering Applications*. Springer-Verlag New York, Inc., 1997.
- [8] A. Thompson, editor. *Hardware Evolution: Automatic design of electronic circuits in reconfigurable hardware by artificial evolution*. Distinguished dissertation series. Springer-Verlag, 1998.
- [9] Julian F. Miller and Peter Thomson. Cartesian genetic programming. In Riccardo Poli, Wolfgang Banzhaf, William B. Langdon, Julian F. Miller, Peter Nordin, and Terence C. Fogarty, editors, *Genetic Programming, Proceedings of EuroGP'2000*, volume 1802 of *LNCIS*, pages 121–132, Edinburgh, 15–16 April 2000. Springer-Verlag.
- [10] A. Nayak, A. Nayak, A. Nayak, I. Stojmenovic, I. Stojmenovic, and I. Stojmenovic. *Handbook of Applied Algorithms (Solving Scientific, Engineering, and Practical Problems)*. 2007.