A Genetic Algorithm to the Strategic Pricing Problem in Competitive Electricity Markets

Wagner Pimentel
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
Unidade de Ensino Descentralizada
Estrada de Adrianpolis, 1.317, Santa Rita
26041-271 Nova Iguau, RJ, Brasil
Doutorando PESC - linha de pesquisa: Otimização
COPPE/UFRJ
C.P. 68530, 21945-970, Rio de Janeiro, Brasil
wagner.pimentel@cefet-rj.br, pimentelwagner@cos.ufrj.br

Marcia Fampa
Instituto de Matemática-DCC and COPPE
Universidade Federal do Rio de Janeiro
C.P. 68530, 21945-970 Rio de Janeiro, Brasil
fampa@cos.ufrj.br

Resumo

In this paper, we present the problem of strategic bidding under uncertainty in a wholesale energy market, where the economic remuneration of each generator depends on the ability of its own management to submit price and quantity bids. This stochastic problem is highly non-convex and due to its difficulty, there has been an intensive search for efficient algorithms to solve it. We present a bilevel formulation for the problem and propose a genetic based algorithm for its solution, where the individual represents the decision of the leader of the bilevel problem. For each individual the linear programming formulation of the follower problem is considered and its exact optimum solution is obtained in a very efficient way. We also consider a local search strategy to improve the solutions generated after crossover and mutation on the genetic algorithm. The results obtained with the application of the algorithm to random instances of the problem are presented and compared with the results obtained by the solution of a mixed integer linear reformulation of the problem.

Keywords: metaheuristics, genetic algorithm; bilevel problem; electricity pool market; strategic pricing.

1 Introduction

In the strategic pricing problem the traditional planning of the operation, based on a centralized optimization has been replaced by decentralized procedures, based on market functioning. In this problem the private agents, generators, compete for contracts for power sales to distribution companies and free consumers. The generators can freely make their price offers for energy production. The units are then loaded in order of increasing supply of a minimum unit price until demand is met. All generators dispatched receive the unit price charged more expensive, which corresponds to the marginal cost of short-term or spot price of the system, Hunt(2003).

One of the basic features of the deregulation process is the creation of a wholesale energy market, or short-term electricity market, where all electric power purchase and sale transactions take place. In a simplified way, the wholesale energy market works as follows:

-All generators freely bid prices for their energy production, typically price-quantity bids on an hourly basis for the next day;

-A system operator dispatches the units by increasing price until demand is met. The dispatched generators are paid the price of the most expensive loaded unit, which corresponds to the system short-run marginal cost, or spot price. This corresponds to the well known uniform-price auction format, which is generally adopted in electricity markets and will be considered in this paper as well.

Different approaches have been used to solve the strategic pricing problem. In Fampa(2008), for example, the problem is formulated as a special class of bilevel program known as taxation problem, Labbé(2000), and then reformulated as a mixed integer linear program (MILP). In Pereira(2005) a MILP formulation is also presented for the problem. The exact solution of MILP is presented on both papers for some small instances. However, due to the size of the instances in real applications, especially when a reasonable number of scenarios is considered, this approach may not be suitable. The authors also present in Fampa(2008) some heuristics for the problem, which due to the deterministic behavior are suitable of converging to local optimal solutions.

In this paper we propose a genetic based algorithm (GA) for the strategic pricing problem in competitive electricity markets presented in Fampa(2008) and compare the results obtained with the GA to the solutions of the MILP formulation presented in Fampa(2008), for some randomly generated instances. Genetic algorithms have been widely used in the solution of combinatorial optimization problems, Pimentel(2001), and are theoretically and empirically proven to provide robust search in complex spaces, Goldberg(1989). Some recent papers also apply genetic algorithms to solve bilevel programming problems. For example, Marinakis(2007) propose a genetic algorithm for the vehicle routing problem modeled as a bilevel problem and Kuo(2010) uses a genetic algorithm with particle swarm to solve a bilevel linear programming problem for supply chain management. As it was done in Marinakis(2007), we use the mathematical formulation of the follower problem to obtain its solution for each individual of the population of the genetic algorithm, which represents itself the decision of the leader of the bilevel problem.

This paper is organized as follows: Section 2 presents an overview of the strategic pricing in competitive electricity markets. In this section the mathematical formulation of the strategic bidding problem under uncertainty is presented. Section 3 describes the genetic based algorithm proposed for the problem. Section 4 presents the numerical results of our computational experiments and compare them with the optimal solutions of the random instances considered, provided by the MILP formulation of the problem. Section 5 concludes the paper.

2 Strategic Pricing in Electricity Markets

In deregulated electricity markets, generators submit a set of hourly generation prices and available capacities for the following day. Based on these data and on an hourly load forecast, the system operator carries out the following economic dispatch at each time step, Fampa(2008):

$$
\begin{array}{llll}
 & & \text{dual variable} & \\
\text{Minimize}_{g_j} & \sum_{j\in J} \lambda_j g_j, & & \\
\text{subject to} & \sum_{j\in J} g_j = d, & \pi_d & \\
 & g_j \leq \bar{g}_j, & \pi_{g_j} & j \in J, \\
 & g_j \geq 0, & & j \in J,
\end{array}
\tag{1}
$$

where the input data d, λ_j and $\bar{g}_j$ represent, respectively, load (MWh), price bid $(\$/MWh)$ and generation capacity bid (MWh) of generator j and the variable g_j represents the energy production of generator j (MWh). The optimal value of the dual variable π_d is considered as the system spot price. The profit of each generator j, in each time step, corresponds to: $(\pi_d - c_j)g_j$, for $j \in J$, where c_j represents its unit operating cost. Note that c_j may be different from λ_j, its price bid.

The net profit of a generation company E, which may be a utility or an independent power producer that owns several different generation units, is given by:

$$\sum_{j\in E} (\pi_d - c_j) g_j,$$

where E is also used to denote the set of indexes associated to the plants belonging to the company E $(E \subset J)$.

Company E aims to determine a set of price bids $\lambda_E = \{\lambda_j, j \in E\}$ and quantity bids $\bar{g}_E = \{\bar{g}_j, j \in E\}$ that maximize its total net profit.

2.1 Bidding Strategies Under Uncertainty

In this paper we consider the Bertrand scheme for the optimal bidding problem, where the quantity bid of each generator of company E is fixed as $\bar{g}_j^*$ and the problem consists in determining the price bids of the company in order to make it receive the maximum profit. The complexity of this problem is greatly compounded by the fact that the calculation of π_d and g_j in the dispatch problem (1) depends on the knowledge of price vectors for all companies, as well as their generation availability and system load values. However, this information is not available to any single company at the time of its bid. Therefore, the bidding strategy has to take into account the uncertainty around these values, Fampa(2008). An approach used to deal with the uncertainty on the data of the problem is to define a set of scenarios for the remaining agent's behavior and maximize the profit of the company over all scenarios, in a classical strategic bidding under uncertainty problem. In this case, the bids from generators not belonging to company E and the load are considered uncertain, and represented by a set of scenarios indexed by s, which occur with exogenous probabilities $\{p_s$, s=1,...,S$\}$. The bilevel formulation for the problem is given by

$$
\begin{array}{lll}
\text{Maximize}_{\lambda_E} & \sum_{s\in S} p_s \sum_{j\in E}[\pi_d^s - c_j]g_j^s, & \\
\text{subject to} & & \\
& \begin{array}{lllll}
\text{Minimize}_{g_j^s} & \sum_{s\in S}\sum_{j\in E}\lambda_j g_j^s + \sum_{j\notin E}\lambda_j^{*s} g_j^s, & & \\
\text{subject to} & \sum_{j\in J} g_j^s = d^s, & & s\in S, \\
& 0 \le g_j^s \le \bar{g}_j^*, & j\in E, & s\in S, \\
& 0 \le g_j^s \le \bar{g}_j^{*s}, & j\notin E, & s\in S.
\end{array} &
\end{array}
\qquad (2)
$$

The first level of problem (2) represents the interest of company E (maximize expected profits), while the second level represents the interest of the system operator (minimize operational costs). The company is classified as leader of the bilevel program and controls the variables λ_j, for $j \in E$, while the system operator is classified as follower and controls the variables g_j^s for $j \in J$, $s \in S$. Once the variables of the leader are determined, the linear programming (LP) follower problem can be solved by any LP solver. Due to its structure, this LP problem can be solved separately for each scenario and the subproblems for the scenarios can be solved by the polynomial procedure presented in Figure 1, which returns the energy production of all generators and the spot price in all scenarios. We note that in order to satisfy the objective of leader, where we maximize the profit of company E, if a generator in E bids the same price as a generator not in E, then the generator in E should be considered first by the follower on the dispatch procedure. Furthermore, if two generators in E bid the same price, the one with smaller cost should be dispatched first.

Input: Price and quantity bids of all generators in J for all scenarios in S. Load d^s for all scenarios in S.

for $s \in S$ **do**
- Let $\tilde{\lambda}_{j_1} \le \tilde{\lambda}_{j_2} \le \ldots \le \tilde{\lambda}_{j_{|J|}}$ be the ordered price bids of all generators in J in scenario s and $\tilde{g}_{j_1}, \tilde{g}_{j_2}, \ldots, \tilde{g}_{j_{|J|}}$ be the corresponding quantity bids.
- Let k be the maximum index such that $\sum_{i=1}^{k} \tilde{g}_{j_i} \le d^s$.
- **for** $i = 1, \ldots, k$ **do**
  - $g_{j_i}^s := \tilde{g}_{j_i}$.
- $g_{j_{k+1}}^s := d^s - \sum_{i=1}^{k} \tilde{g}_{j_i}$.
- **for** $i = k+2, \ldots, |J|$ **do**
  - $g_{j_i}^s := 0$.
- $\pi_d^s := \tilde{\lambda}_{j\,k+1}$.

return $g_j^s, \forall j \in J, s \in S$, $\pi_d^s, \forall s \in S$;

Figure 1: Polynomial solution of the follower problem

3 Genetic Algorithm

In the following we propose a genetic based algorithm (GA) to solve the bilevel strategic bidding problem. In the first level of the proposed algorithm, a genetic algorithm is used for calculating promising price bids for the generators of the company E. In the second level of the proposed algorithm, the follower problem of (2) is solved, independently for each individual of the population. In Figure 2 we present the general description of the GA.

```
Create the initial population of MAXPOP individuals, using the three options: Type 1
  solution, Type 2 solution or Type 3 solution.
Evaluate the fitness of each individual solving the follower problem of (2) by the
  procedure presented in Figure 1.
while stopping criterion do
    Select two parents from the current population using rank selection.
    Apply the CROSSOVER operator to the two parents, randomly selecting each gene
      from either one of the parents to be copied into the chromosome of the new
      individual.
    Repeat the previous two steps until 90% of MAXPOP individuals are generated.
    Apply the MUTATION operator to 2% of the offspring, randomly selecting a gene of
      each selected individual to be updated by another randomly selected price bid.
    Apply the LOCAL SEARCH PROCEDURE to two randomly selected individual
      from the offspring and from the set ELITE, and replace the individual by the
      solution obtained.
    Copy the best 10% of MAXPOP individuals (set ELITE) from the current
      population on the next population.
    Copy the 90% of MAXPOP generated individuals on the next population.
Apply the LOCAL SEARCH PROCEDURE to the best individual obtained by the
  genetic algorithm and return the solution obtained. return λ_j, ∀j ∈ E,
  g_j^s, ∀j ∈ J, s ∈ S, π_d^s, ∀s ∈ S;
```

Figure 2: General description of the Genetic Algorithm

3.1 Initial Population

We define an individual for the genetic algorithm as a vector of $|E|$ components, where the j-th component corresponds to the price bid λ_j of the generator $j \in E$.

It is straightforward to verify that there is always an optimal solution for the strategic bidding problem, where all the generators belonging to company E bid the same price as a generator not belonging to E in some scenario. Therefore, denoting by $\mathcal{L}$ the set of all possible values for each component of the individual of our proposed genetic algorithm, we initially define $\mathcal{L} = \{\lambda_j^{*s}, j \notin E, s \in S\}$. To improve the efficiency of the algorithm we can still eliminate some of the price bids from this set. Consider λ_{max}^s as the maximum value in $\mathcal{L}$ that can be assigned to the price bid of each generator of company E such that at the optimal solution of the economic dispatch problem (1) for the scenario s every generator of E generate all of its capacity. Now let $\lambda_{MIN} = min\{\lambda_{max}^s, s \in S\}$. Note that we can eliminate from $\mathcal{L}$ all the price bids that are smaller than λ_{MIN}, with no risk of cutting the optimal solution of the problem from the search space. Furthermore, consider λ_{min}^s as the minimum value in $\mathcal{L}$ that could be assigned to the price bid of each generator of company E with none of them being dispatched at the optimal solution of the economic dispatch problem (1) for the scenario s.

Finally, let $\lambda_{MAX} = max\{\lambda^s_{min}, s \in S\}$. Note that we can also eliminate from $\mathcal{L}$ all the price bids that are larger than or equal to λ_{MAX}. Therefore, in the following we consider the set $\mathcal{L}$ of all possible values assigned to the price bids of the generators of company E on our genetic algorithm defined as

$$\mathcal{L} = \{\lambda^{*s}_j, j \notin E, s \in S | \lambda_{MIN} \leq \lambda^{*s}_j < \lambda_{MAX}\}. \quad (3)$$

We consider three different solution types to generate the initial population of the algorithm.

The solution of type 1 is based on the idea of initializing the population with the solution where company E generates all of its capacity on a randomly selected scenario s. We consider $\lambda_j = \lambda^s_{max}$ for all $j \in E$, and therefore we have $g^s_j = \bar{g}^*_j$ for all $j \in E$.

The solution of type 2 corresponds to the case where no plant belonging to company E is dispatched on a randomly selected scenario s. We consider $\lambda_j = \lambda^s_{min}$ for all $j \in E$, and therefore we have $g^s_j = 0$ for all $j \in E$. Since we assume that the remaining plants can always meet the demand on all scenarios, the procedure always generates a feasible solution.

At last, the solution of type 3 is based on a polynomial algorithm to solve the strategic bidding problem for a randomly selected scenario s. In Fampa(2008) it was proven that, when there is only one possible scenario, there is an optimal solution for the strategic bidding problem, where all plants belonging to company E bid the same price. In Figure 3 we present the procedure used to generate the solution of type 3.

Input: A randomly selected scenario s.
1 Let $\mathcal{L}^s = \{\lambda^{*s}_j \in \mathcal{L} | \lambda^s_{max} \leq \lambda^{*s}_j < \lambda^s_{min}\}$.
2 Let the price bids of all generators of company E equal to each price bid in $\mathcal{L}^s$. Compute the corresponding profit of company E solving the economic dispatch problem (1) for scenario s.
3 Determine as the price bids of the generators of company E, the price bid that leads to the maximum profit of E on scenario s. **return** $\lambda_j, \forall j \in E$;

Figure 3: Procedure to generate Type 3 solution

3.2 Local Search

The goal of the local search procedure is to improve a solution obtained by the genetic algorithm, by an iterative process. Without lost of generality, consider that the indexes $j = 1, \ldots, |E|$ correspond to the plants belonging to company E and satisfy the relation $\bar{g}^*_1 \geq \bar{g}^*_2 \geq \ldots \geq \bar{g}^*_{|E|}$. Consider $\beta_1 < \beta_2 < \ldots < \beta_n$ as the different price bids in $\mathcal{L}$ (3). The local search procedure considers a initial solution for the strategic bidding problem (2), given by $\tilde{\lambda}_j$, for $j \in E$, $\tilde{g}^s_j$, for $j \in J, s \in S$ and $\tilde{\pi}^s_d$, for $s \in S$ with objective function value $\tilde{z}$, and iteratively test if there is a better value β_j, $j = 1, \ldots, n$ for the price bid of each generator of E, starting the procedure from a randomly selected generator. In Figure 4 we present our local search procedure, that returns the best price bid obtained for each generator of company E.

Input: $\beta_1 < \beta_2 < \ldots < \beta_n$, $\tilde{\lambda}_j, \forall j \in J$, $c_j, \forall j \in E$, $p_s, \forall s \in S$, $\tilde{z}$.

for $j = 1 \ldots, |J|$ **do**
  $\lambda_j := \tilde{\lambda}_j$.
$I = \emptyset$.
for $i = 1, \ldots, |E|$ **do**
  Randomly select a generator j from $E - I$.
  for $k = 1, \ldots, n$ **do**
    $\lambda_j := \beta_k$.
    Solve the follower problem of (2) with the procedure presented in Figure 1 and obtain the solution g_j^s and π_d^s, $j \in J, s \in S$.
    $z := \sum_{s \in S} p_s \sum_{j \in E} (\pi^s - c_j) g_j^s$
    if $z > \tilde{z}$ **then**
      $\tilde{\lambda}_j := \beta_k$.
      $\tilde{z} := z$.
  $\lambda_j := \tilde{\lambda}_j$. $I := I + \{j\}$.
return $\tilde{\lambda}_j, \forall j \in E$;

Figure 4: Local Search Procedure

4 Numerical Results

In this section we present computational results obtained by the proposed GA when applied to randomly generated instances of the strategic bidding problem. Our code was implemented in C and compiled with gcc (GNU COMPILE C). All runs were conducted on a 2GB Ram, 2.13GHz Intel Core processor running under Linux Ubuntu, Version: 9.10. The solver CPLEX, v12.2, Gay(2009), was used to obtain the optimal solution of the instances, considering the MILP formulation of the problem presented in Fampa(2008). In all tests we limited in 10800 seconds the CPU time used to obtain the optimal solution of the instances by CPLEX and in 1800 seconds the CPU time to obtain the solution by the GA.

All test problems considered in our numerical experiments were randomly generated instances where:
$|J| \in \{20, 30, 40, 50\}$, $|E| \in \{9, 10, 11, 12, 13, 15, 20, 25, 40\}$, $|S| \in \{5, 10, 15, 20, 25, 36, 72\}$, $g_j^* \in [50, 100]$, $c_j \in [0, 25], j \in E$, $g_j^{*s} \in [50, 100]$, $\lambda_j^{*s} \in [0, 31]$, $j \notin E, s \in S$, and $d_s \in [\delta^s/2, \delta^s], s \in S$, where δ^s is the sum of the generation capacities of the generators that do not belong to E in scenario s. Four instances were generated for each combination of values for $|E|$, $|J|$ and $|S|$. We ran the GA five times for each instance and computed the relative gap between each solution obtained and the optimal solution.

Tables 1 and 2 present average results for each instance size. In Table 1 we consider only the instances with known optimal solution. In the first column of the table we indicate the size of the instances considered. In the line corresponding to instance $Inst_{|J|,|E|,|S|}$, we present the mean values for four instances with $|J|$ generators on the total, $|E|$ of them belonging to company E, and $|S|$ scenarios, of the average gap ($\bar{x}_{gap}$), the standard deviation of the gap (σ_{gap}), the average CPU time ($\bar{x}_{time}$) and the standard deviation of the CPU time (σ_{time}) for the five runs of the GA. In the last column of Table 1 we also present the average CPU

Tabela 1: Instances with known optimal solution

Instance	Gap		CPU Time		
$Inst_{\|J\|,\|E\|,\|S\|}$	$\bar{x}_{gap}$	σ_{gap}	$\bar{x}_{time}$	σ_{time}	$MILP$
$Inst_{20,09,05}$	0.0000	0.0000	1.8190	1.2772	73.2500
$Inst_{20,09,10}$	0.0000	0.0000	1.1485	0.9105	260.7500
$Inst_{20,10,10}$	0.0000	0.0000	2.6885	2.7154	991.7500
$Inst_{20,11,05}$	0.0000	0.0000	1.8305	1.1038	32.2500
$Inst_{20,12,05}$	0.0000	0.0000	2.2135	0.8856	239.0000
$Inst_{20,13,05}$	0.0000	0.0000	4.2990	4.5225	759.2500
$Inst_{20,15,05}$	0.0000	0.0000	2.9815	1.9558	5134.7500
$Inst_{30,12,05}$	0.0000	0.0000	8.2975	5.8677	4516.5000
$Inst_{40,09,05}$	0.0000	0.0000	0.6740	0.3935	434.5000
$Inst_{40,09,10}$	0.0000	0.0000	3.3575	2.8401	929.0000

Tabela 2: Instances with the best solution in 3 hours MILP

Instance	$\text{Gap}_{(2\%ofbest)}$		$\text{CPU Time}_{(2\%ofbest)}$		Gap in 1800 sec.	
$Inst_{\|J\|,\|E\|,\|S\|}$	$\bar{x}_{gap}$	σ_{gap}	$\bar{x}_{time}$	σ_{time}	$\bar{x}_{gap}$	σ_{gap}
$Inst_{20,15,15}$	0.0108	0.0031	0.4060	0.2190	0.0000	0.0000
$Inst_{20,15,20}$	0.0093	0.0029	1.0450	0.7287	-0.0037	0.0000
$Inst_{30,20,15}$	0.0116	0.0043	8.0305	4.3748	-0.0025	0.0006
$Inst_{30,20,20}$	0.0061	0.0064	15.9010	15.0319	-0.0008	0.0007
$Inst_{30,20,25}$	0.0051	0.0057	5.7020	2.3627	-0.0139	0.0006
$Inst_{30,25,25}$	0.0062	0.0043	18.6960	9.0902	-0.0065	0.0000
$Inst_{50,20,36}$	0.0080	0.0056	12.1805	7.4679	-0.0063	0.0003
$Inst_{50,20,72}$	0.0003	0.0029	13.5480	5.1626	-0.0063	0.0003
$Inst_{50,40,36}$	0.0004	0.0059	39.4335	34.2748	-0.0370	0.0029
$Inst_{50,40,72}$	-0.0023	0.0097	69.2395	40.7482	-0.0301	0.0019

time to obtain the optimal solution of the instances with CPLEX ($MILP$). The CPU time is always given in seconds. The GA obtained the optimal solution of all instances generated for Table1 in a very small computational time. The algorithm has a very robust behavior, since the standard deviation of the gap is always equal to zero. In Table 2 we present similar statistics for instances for which we were not able to prove optimality in three hours of CPU time running CPLEX. In this case the gaps presented are related to the best solution ($best$) found by CPLEX. Two stoping criterion were used for GA: the solution obtained is at most 2% worse than $best$ and the CPU time is 1800 seconds. The results in Table 2 show again the good quality of the results given by GA, which are better than the results obtained by CPLEX for all groups of instances except the first one with $|J| = 20$, $|E| = 15$ and $|S| = 15$. In this case, GA obtained the same solutions of CPLEX, which we note that can be the optimal solutions of the instances.

5 Conclusion

In this paper we present a genetic based algorithm (GA) for the strategic bidding problem under uncertainty in a wholesale energy market, which is formulated as a bilevel programming problem. The variables controlled by the leader of the bilevel problem are determined by a detailed described genetic algorithm, combined with a local search procedure. For each individual of the genetic algorithm, the follower of the bilevel problem, a linear programming problem, is exactly solved by a polynomial algorithm. We present computational results obtained with the application of the GA to randomly generated instances of the problem and compare them to their best known solutions. The results show that the GA is very robust and obtain solutions of very good quality for all instances considered.

Referências

Fampa, M., Barroso, L. A., Candal, D., Simonetti, L. (2008). Bilevel Optimization applied to Strategic Pricing in Competitive Electricity Markets, *Computational Optimization and Applications* 39(2), 121-142.

Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison Wesley, Reading, 1989.

Gay, D. M. IBM ILOG CPLEX, User's Manual for CPLEX v12.1, 2009.

Hunt, S.*Making competition work in electricity.* John Wiley and Sons, 2003.

Kuo, R. J., Han, Y. S.(2010). Using Hybrid Metaheuristic Approaches to Solve Bi-Level Linear Programming Problem for Supply Chain Management, *Proceedings of the 2010 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 1154-1158.

Labbé, M., Marcotte, P., Savard, G.(2000). On a class of bilevel programs, *Nonlinear Optimization and Related Topics*, di Pillo and Giannessi (eds), Kluwer Academic Publishers, 183-206.

Marinakis, Y. , Migdalas, A., Pardalos, P. M.(2006). A new bilevel formulation for the vehicle routing problem and a solution method using a genetic algorithm, *Jornal Global Optimization* 38(4), 555-580.

Pereira, M. V., Granville, S., Fampa, M., Dix, R., Barroso, L. A.(2005). Strategic Bidding Under Uncertainty: A Binary Expansion Approach, *IEEE Transactions on Power Systems* 20(1), 180-188.

Pimentel, W., Figueiredo, C. M. H., Carvalho, L. A. V.(2001). Planejamento de rotas aéreas por algoritmos genéticos, *V Congresso Brasileiro de Redes Neurais (V CBRN)*, Rio de Janeiro, 1-6.