

# The AgentMatcher Architecture Applied to Power Grid Transactions

Riyanarto Sarno<sup>1</sup>, Lu Yang<sup>2</sup>, Virendra C. Bhavsar<sup>2</sup>, and Harold Boley<sup>3</sup>

<sup>1</sup>*Faculty of Information Technology  
Sepuluh Nopember Institute of Technology  
Surabaya, 60111  
Indonesia  
riyanarto@informatics-group.com*

<sup>2</sup>*Faculty of Computer Science  
University of New Brunswick  
Fredericton, NB, E3B 5A3  
Canada  
{o6c11,bhavsar}@unb.ca*

<sup>3</sup>*Institute for Information Technology e-Business  
National Research Council of Canada  
Fredericton, NB, E3B 9W4  
Canada  
Harold.Boley@nrc-cnrc.gc.ca*

## Abstract

*The number of complex transactions in a power grid to determine economical ways of satisfying power demands given a set of power plants has increased significantly. Knowledge-based solution and intelligent computational grids can help in these transactions. This paper presents the application of the AgentMatcher architecture for similarity, pairing and negotiation in order to optimize the transactions in these virtual markets. We use tree similarity and agent pairing to obtain for each buyer (power distributor, a ranking table of candidate sellers (power plants). A similarity based on weight variance and a negotiation algorithm based on priority groups of sellers are also proposed.*

**Keywords:** AgentMatcher architecture, unit commitment, multi-agent system, intelligent agent, similarity

## 1. Introduction

Electricity markets are complex and operate under a variety of rules in different schedules. Electricity markets all over the world are being deregulated these days. Many electricity markets are undergoing a transition from monopoly-like structures to market structures, which are usually identified as power grid

virtual markets [5, 6, 15]. In these virtual markets, the price of electricity fluctuates according to the current supply and demand. Therefore, the amount of transactions among sellers and buyers has increased significantly. Transactions consist of determining the most economical power plants to satisfy electricity demands and operating constraints.

Optimizing transactions is regarded as one of the major problems in power system operation, and is referred to in the literature as the unit commitment problem [14]. The unit commitment is a complex, large scale, mixed-integer, and non-linear optimization problem. Since it is a combinatorial problem, it has non-polynomial (NP) time complexity [13].

Besides the complexity of the solutions, power grid transactions are often handled from several wide areas having interconnection of electricity networks. In this regard, best solutions for the transactions can be obtained by using computational grids. The computational grid has evolved from systems sharing distributed resources to service oriented architectures for transparent and reliable distributed systems, which have high-end computational capabilities [7].

The unit commitment problem has been solved using several optimization methods [4, 12], which can be categorized into three types: (1) mathematical methods, such as integer and mixed-integer programming, branch-and-bound methods, Lagrangian relaxation, augmented Lagrangian, and dynamic programming; (2) heuristic methods, which include

genetic algorithms, genetic programming, simulated annealing, fuzzy logic; and (3) artificial intelligent methods, which involve artificial neural networks, expert systems, and multi-agent systems.

Since only few knowledge-based methods have been implemented in solving the unit commitment problems, this paper proposes the application of AgentMatcher architecture for computing similarity, pairing, and negotiating between power sellers and power buyers in virtual markets. The AgentMatcher architecture has been developed for e-business and e-learning applications [1].

The paper first provides some background on the AgentMatcher architecture, reviews the similarity algorithm and describes the pairing process. Then, the similarity based on weight variance, which is the extension of the similarity algorithm, is proposed in Section 3. Section 4 discusses the unit commitment problem and also proposes the negotiation algorithm. Finally, the conclusions are presented in Section 5.

## 2. AgentMatcher

In a multi-agent system architecture like ACORN [9], buyer and seller agents can carry the information for buyers and sellers. Buyer agents and seller agents need to communicate with each other through a middle agent to finalize their transaction. In this paper, the sellers are power plants and the buyers are electricity distributors, which will sell the electricity to consumers, such as hotels, shopping malls, industries, public buildings, and houses.

In order to implement the complete transaction, we propose an architecture called AgentMatcher that is composed of three components; the similarity computation of agents, the pairing process of the buyer and seller agents based on their similarity values, and the negotiation process. Thus, the AgentMatcher architecture covers the whole process after the buyer and seller agents enter the market place until they finish their negotiation.

### 2.1. Architecture

The matchmaking scenario of e-learning proposed in [1] could be adopted to our power plant application. Buyer and seller agents enter the power grid market to conduct their transaction as shown in Figure 1.

After power plant agents and power distributor agents enter the power grid market, similarity values are computed for every pair of agents. These similarity values are the prerequisites for pairing.

During the pairing process, our algorithm recommends an appropriate power plant for every

power distributor. The results of this pairing process are the prerequisites for negotiation.

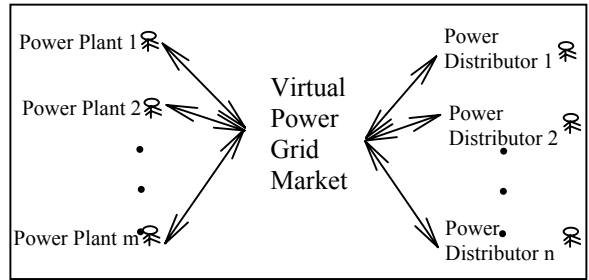


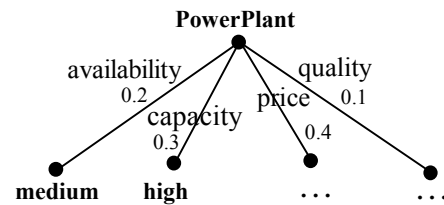
Figure 1. A virtual power grid market.

Whenever the power plant and power distributor decide to negotiate with each other, the negotiation between them could be conducted based on the negotiation algorithm described in Section 4.

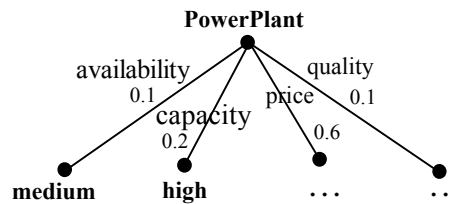
In this paper, when many buyers and sellers are in the market place, we use  $s_i$  ( $1 \leq i \leq m$ ) and  $b_j$  ( $1 \leq j \leq n$ ) to represent different sellers and buyers, respectively.

### 2.2. Review of Tree Similarity Algorithm

The information that buyer agents and seller agents carry is represented by node labelled, arc labelled and arc weighted trees. A tree similarity algorithm that computes the similarity between these trees is provided in [1], which is quite different from those developed before [11]. Figure 2 shows two example trees for the power plant application.



(a) Seller Tree.



(b) Buyer Tree.

Figure 2. Example trees of power plant application.

At every level of a subtree, all the arc labels are in alphabetical order and all arc weights add up to one. The similarity value of every pair of subtrees falls into the real interval [0,1]. The values of “0” and “1” mean they are totally different and identical, respectively. The depths and breadths of trees are not limited.

The tree similarity algorithm recursively traverses every pair of trees top down and starts computing the similarity bottom up when it reaches leaf nodes. The similarity value of every pair of upper level subtrees is computed based on the similarity of their lower level subtrees.

During the computation, the contribution of arc weights is also taken into account. The weights are averaged using the arithmetic mean,  $(w_i + w'_i)/2$ , and the recursively obtained similarity  $s_i$  of trees  $t_i$  and  $t'_i$  — adjusted to  $A(S_i)$  by an arc function  $A$  — is multiplied by the averaged weight. Finally, on each level the sum of all such weighted adjusted similarities,  $A(S_i)(w_i + w'_i)/2$ , is divided by the sum of all averaged weights:

$$\sum (A(S_i)(w_i + w'_i)/2) / \sum (w_i + w'_i)/2 \quad (1)$$

In the special case that weights on some level of both trees add up to 1, the denominator of Equation (1) becomes 1. Hence, if the weights on each level of both trees add up to 1 (a reasonable assumption used throughout this paper), Equation (1) will be simplified:

$$\sum (A(S_i)(w_i + w'_i)/2) \quad (2)$$

This algorithm also deals with several special cases. For example, missing subtrees, and subtrees only have identical root node labels, etc.

Based on the characteristics of node labelled, arc labelled and arc weighted trees, weighted extension of Object-Oriented RuleML [2, 3] is proposed to serialize the trees. The hierarchical structure of XML reflects the shape of these normalized trees and XML attributes are used to serialize the arc labels and weights. The tree in Figure 1 (a) is serialized as shown in Figure 3.

```
<cterm>
  <_opc><ctor>PowerPlant</ctor></_opc>
  <_r n="availability" w="0.2"><ind>medium</ind></_r>
  <_r n="capacity" w="0.3"><ind>middle</ind></_r>
  <_r n="price" w="0.4"><ind>high</ind></_r>
  <_r n="quality" w="0.1"><ind>average</ind></_r>
</cterm>
```

Figure 3. Symbolic tree representation.

### 2.3. Similarity and Pairing Results

Trees representing the information of buyers and

sellers will be much more complex than the trees shown in Figure 2, because of the complex situations of buyers and sellers in our real life, Figure 4 shows a tree that represents information carried by buyer or seller agents.

Based on the tree similarity algorithm described above, we work out the similarity value of every pair of trees. So, after a buyer agent and a seller agent enter the market place, the similarity value between

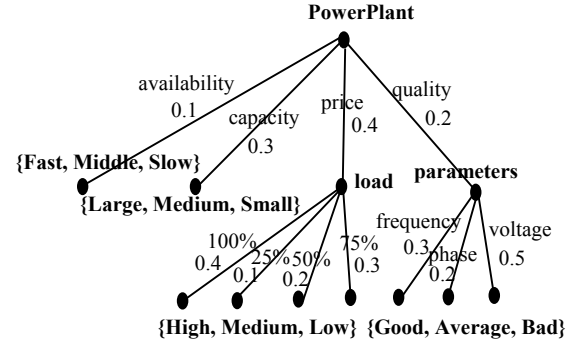


Figure 4. Tree form information that the buyer or seller agent carries.

them is computed and the AgentMatcher decides based on similarity thresholds, which is dynamically determined according to current demand, if they should start negotiating according to their similarity values. However, the problem is that not only one buyer and one seller are in the power grid market. Just like in our real life, there are many shops in a shopping mall, but there are many more buyers who go to the shopping mall. Generally, buyers visit more than one shop to select a product that could satisfy them. Meanwhile, sellers in the shop are also looking for the right buyers that could satisfy them. Actually, this is a matching process. Whenever the sellers and the buyers feel their preferences and interests are close, they could set up the negotiation and try to get as much benefits as possible from the negotiation.

Rank	b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>	b <sub>4</sub>
1	s <sub>1</sub> 0.84	s <sub>2</sub> 0.75	s <sub>4</sub> 0.96	s <sub>5</sub> 0.69
2	s <sub>4</sub> 0.80	s <sub>5</sub> 0.72	s <sub>1</sub> 0.87	s <sub>4</sub> 0.67
3	s <sub>2</sub> 0.63	s <sub>3</sub> 0.72	s <sub>2</sub> 0.80	s <sub>1</sub> 0.60
4	s <sub>3</sub> 0.55	s <sub>1</sub> 0.53	s <sub>3</sub> 0.71	s <sub>3</sub> 0.55
5	s <sub>5</sub> 0.41	s <sub>4</sub> 0.38	s <sub>5</sub> 0.67	s <sub>2</sub> 0.52

Figure 5. The most common case during matchmaking.

Suppose that there are 4 buyer agents and 5 seller agents in the power grid market. We further suppose that every buyer wants to evaluate the offer from every seller. So, for every buyer agent, we need to compute

the similarity between it and all seller agents. Therefore, a ranking table could be created for buyers. It is shown in Figure 5. For every buyer, our algorithm ranks the similarity values in descending order.

We could find, at the row of rank 1 in Figure 5, the similarity value of  $b_3$  and  $s_4$  is the biggest one. Intuitively,  $s_4$  should be recommended to  $b_3$ . After one buyer agent gets its recommendation, the buyer agent and the recommended seller agent in every table will be marked as unavailable, which is shown in Figure 6. "Unavailable" means that they will not take part in the matching process in this cycle. A cycle begins after the 4 buyer agents and 5 seller agents enter the power grid market, and ends when no more recommendations will be made.

Rank	$b_1$		$b_2$		$b_3$		$b_4$	
1	$s_1$	0.84	$s_2$	0.75	$s_4$	0.96	$s_5$	0.69
2	$s_4$	0.80	$s_5$	0.72	$s_1$	0.87	$s_4$	0.67
3	$s_2$	0.63	$s_3$	0.72	$s_2$	0.80	$s_1$	0.60
4	$s_3$	0.55	$s_1$	0.53	$s_3$	0.71	$s_3$	0.55
5	$s_5$	0.41	$s_4$	0.38	$s_5$	0.67	$s_2$	0.52

Figure 6. The table after  $s_4$  is recommended.

After one recommendation is made, we move to the first available seller in every buyer's table. Repeating the above process, we find that  $s_1$  should be recommended to  $b_1$ . Marked table is shown in Figure 7.

Rank	$b_1$		$b_2$		$b_3$		$b_4$	
1	$s_1$	0.84	$s_2$	0.75	$s_4$	0.96	$s_5$	0.69
2	$s_4$	0.80	$s_5$	0.72	$s_1$	0.87	$s_4$	0.67
3	$s_2$	0.63	$s_3$	0.72	$s_2$	0.80	$s_1$	0.60
4	$s_3$	0.55	$s_1$	0.53	$s_3$	0.71	$s_3$	0.55
5	$s_5$	0.41	$s_4$	0.38	$s_5$	0.67	$s_2$	0.52

Figure 7. The table after  $s_1$  is recommended.

Continuing this process,  $s_2$  should be recommended to  $b_2$ . Figure 8 shows the table after this recommendation.

Rank	$b_1$		$b_2$		$b_3$		$b_4$	
1	$s_1$	0.84	$s_2$	0.75	$s_4$	0.96	$s_5$	0.69
2	$s_4$	0.80	$s_5$	0.72	$s_1$	0.87	$s_4$	0.67
3	$s_2$	0.63	$s_3$	0.72	$s_2$	0.80	$s_1$	0.60
4	$s_3$	0.55	$s_1$	0.53	$s_3$	0.71	$s_3$	0.55
5	$s_5$	0.41	$s_4$	0.38	$s_5$	0.67	$s_2$	0.52

Figure 8. The table after  $s_2$  is recommended.

Then, the last buyer  $b_4$  can only get seller  $s_5$ . Until now, all buyers get their recommendations. So, all of them are marked as unavailable in Figure 9.

Rank	$b_1$		$b_2$		$b_3$		$b_4$	
1	$s_1$	0.84	$s_2$	0.75	$s_4$	0.96	$s_5$	0.69
2	$s_4$	0.80	$s_5$	0.72	$s_1$	0.87	$s_4$	0.67
3	$s_2$	0.63	$s_3$	0.72	$s_2$	0.80	$s_1$	0.60
4	$s_3$	0.55	$s_1$	0.53	$s_3$	0.71	$s_3$	0.55
5	$s_5$	0.41	$s_4$	0.38	$s_5$	0.67	$s_2$	0.52

Figure 9. Table after all buyers got their recommendations.

A complete cycle is finished now. During the match making process, we could set a threshold for the similarity value. For example, only the similarity value that is above 0.5 is counted as available. Thus, if all the similarity values in all tables are above 0.5, all of the four buyers could get their recommendations after one cycle because there are 5 seller agents in the market place. But if lots of similarity values are below the threshold, there might be some buyers that could not get a recommendation. Those buyers that do not get recommendations will enter the power grid market again to try and find appropriate sellers. Our agent matchmaking algorithm could compute how many cycles every agent needs to get its recommendation.

Figure 5 only shows the conflict-free case of the match making. However, there are some conflicting cases. For example, in Figure 10, at rank 1, two pairs of agents,  $b_2$  and  $s_1$  and  $b_4$  and  $s_3$ , have identical biggest similarity values. In this case, we need to decide if we should recommend  $s_1$  to  $b_2$  or  $s_3$  to  $b_4$ . But we find that  $s_1$  and  $s_3$  are not identical, so we recommend  $s_1$  to  $b_2$  and  $s_3$  to  $b_4$  at the same time. Working in such a parallel way, the efficiency of the algorithm is improved and buyers could get their best recommendations.

Rank	$b_1$		$b_2$		$b_3$		$b_4$	
1	$s_1$	0.84	$s_1$	0.88	$s_4$	0.61	$s_3$	0.88
2	$s_4$	0.80	$s_3$	0.79	$s_5$	0.58	$s_4$	0.67
3	$s_2$	0.63	$s_5$	0.77	$s_1$	0.43	$s_2$	0.60
4	$s_3$	0.55	$s_4$	0.68	$s_3$	0.34	$s_1$	0.55
5	$s_5$	0.41	$s_2$	0.63	$s_2$	0.20	$s_5$	0.52

Figure 10. Two different power plants have identical similarity value with two different buyers.

Based on the previous case, a more complex case may occur. In Figure 11, we can find at rank 1,  $s_3$  has the same similarity value with both  $b_1$  and  $b_3$ . It's

difficult to decide which buyer should get the recommendation. One way is that the buyer that enters the market place should get it. If we follow this way, and if we also assume that  $b_3$  enters earlier than  $b_1$ , so  $s_3$  should be recommended to  $b_3$ . However, after this recommendation,  $b_1$  will never get a good recommendation because the next available seller for it has a very low similarity value 0.50 compared to 0.89. While for  $b_3$ , it could get another good recommendation  $s_5$  with relatively much higher similarity value 0.88 if we recommend  $s_3$  to  $b_1$ . In order to achieve over all buyer and seller satisfaction, our algorithm will recommend  $s_3$  to  $b_1$ . Thus, both  $b_3$  and  $b_1$  are happy with the recommendation.

Rank	$b_1$		$b_2$		$b_3$		$b_4$	
1	$s_3$	0.89	$s_1$	0.84	$s_3$	0.89	$s_3$	0.76
2	$s_4$	0.50	$s_3$	0.79	$s_5$	0.88	$s_4$	0.70
3	$s_2$	0.47	$s_5$	0.77	$s_1$	0.76	$s_2$	0.67
4	$s_1$	0.39	$s_4$	0.68	$s_4$	0.73	$s_1$	0.55
5	$s_5$	0.28	$s_2$	0.63	$s_2$	0.65	$s_5$	0.52

**Figure 11. Two identical sellers have identical similarity values with two different buyers.**

After  $s_3$  is recommended to  $b_1$ , in Figure 12, the first available seller in every buyer's table shows that there is no conflict because all of their similarity values are different. So, we could continue the recommendation using the method described based on Figure 5.

Rank	<del><math>b_1</math></del>		$b_2$		$b_3$		$b_4$	
1	<del><math>s_3</math></del>	0.89	$s_1$	0.84	<del><math>s_3</math></del>	0.89	<del><math>s_3</math></del>	0.76
2	<del><math>s_4</math></del>	0.50	<del><math>s_3</math></del>	0.79	<del><math>s_5</math></del>	0.88	<del><math>s_4</math></del>	0.70
3	$s_2$	0.47	<del><math>s_5</math></del>	0.77	$s_1$	0.76	$s_2$	0.67
4	$s_1$	0.39	$s_4$	0.68	$s_4$	0.73	$s_1$	0.55
5	$s_5$	0.28	$s_2$	0.63	$s_2$	0.65	$s_5$	0.52

**Figure 12. Confliction resolved after the first recommendation.**

### 3. Similarity Based on Weight Variance

The tree similarity algorithm that utilizes averaged weights has been presented in Subsection 2.2. The present section proposes an extension in which the variance of every pair of weights is taken into account. Then, the variances are averaged. For a special case when the weights at a level of both trees add up to 1 (see Equation 2), the variance is computed as follows:

$$\overline{\text{variance}} = \left\{ \sum_1^n ((w_i - \bar{w}_i)^2 + (w_i' - \bar{w}_i')^2) \right\} / n \quad (3),$$

where  $\bar{w}_i = (w_i - w_i') / 2$ , and

$n$  = the number of weight pairs.

The average of variance is very useful to select the most preferable agent when several of the agents have same similarity values. The smaller variance of the similarity value indicates the more preferable agent or selection.

Figure 13 describes the advantage of the averaged variance as follows. For simplicity, all  $A(s_i)$  is equal to 1. In Figure 13 (a) and (b),  $t_1$  represents the weighted tree of a buyer, while  $t_2$  and  $t_3$  are the weighted trees of two different sellers. Figure 13 (a) shows that the similarity between  $t_1$  and  $t_2$  equals 1, while Figure 13 (b) describes that the similarity between  $t_1$  and  $t_3$  is also equal to 1. In this case the buyer is ambiguous to select a seller between the two sellers, because they have the same similarity value. However, it can be seen clearly that the seller represented by  $t_2$  in Figure 13 (a) is more preferred than the seller represented by  $t_3$  in Figure 13 (b), because the corresponding weights of  $t_1$  and  $t_2$  are same. This ambiguity can be solved by employing the averaged variance. The averaged variance of the similarity of the trees in Figure 13 (a) is 0, while those of Figure 13 (b) is equal to 0.18. Furthermore, seller  $t_2$  is selected because the averaged variance is smaller.

Thus, this averaged variance assures that the smaller averaged variance means the more preferred matching or selection. This example is a special case, and the averaged variance can be utilized in any cases with same similarity values.

## 4. Negotiations

### 4.1. Unit Commitment

Solving the unit commitment problem requires minimization of an objective function subject to a variety of system constraints and unit constraints. The objective function represents the total production cost of electricity. The system constraints include power demand, spinning reserve requirements, transmission and environmental constraints. The unit constraints comprise all generating limitation, such as maximum capacity, minimum up time, minimum down time, and ramp rates.

In the unit commitment, the solution method selects a set of power plants ( $m$  units) in order to supply the electricity demand. Then, the production capacity of the selected power plants is adjusted to match the electricity demand with the minimum cost of

operations for the generating units. This adjustment is commonly referred to in the literature as Economic Dispatch [14].

A typical example of objective function can be described mathematically as follows:

$$\text{Obj} = \text{Minimize} \left[ \sum_{i=1}^m F_i(P_i) \right] \quad (4)$$

and the constraints are the following:

$$\sum_{i=1}^m P_i = P_L + P_{TL} \quad (5)$$

and

$$P_i^{\min} \leq P_i \leq P_i^{\max} \quad (6)$$

where:

Obj : objective function;

$P_i$  : output of power plant  $i$  at period  $t$  (MW);

$F_i(P_i)$  : fuel cost of power plant  $i$  when its output power is  $P_i$  (\$);

$m$  : total number of power plants;

$P_L$  : total demand (required capacity);

$P_{TL}$  : total power loss in transmission.

The fuel cost function can be represented by a polynomial function as follows:

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i$$

This example can be solved by utilizing a mathematical, a heuristic, or an artificial intelligent method.

## 4.2. A Negotiation Algorithm

As described in Section 2, the AgentMatcher architecture is comprised of three components; that is, the similarity computation of agents, the pairing process of agents based on their similarity values, and the negotiation process of agents. The similarity computation and the pairing process have been implemented on five power sellers and four power buyers. In order to finalize the transactions between buyer and seller agents, a negotiation algorithm is proposed, and described in Figure 16.

We assume that the number of buyers is less than the number of sellers, therefore a buyer can deal with several sellers (see Figure 14). It is common because the buyers are power distributor companies which will sell the electricity to small consumers. Therefore, the pairing procedure discussed in Subsection 2.3 is extended to result in priority groups of sellers, which are categorized based on their pairing iteration. Figure 14 shows three priority groups of sellers, which are derived by employing pairing process three iterations.

Figure 3 shows that a power plant described by four labeled arcs; however, in this negotiation algorithm only two labeled arcs (the price and the capacity) are

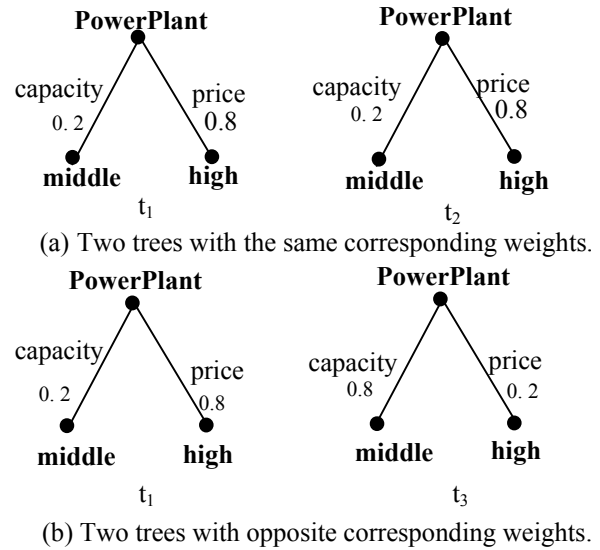
considered because their weights are dominant and the solution can be visualized. In the future work, we will consider all of the labeled arcs.

Figure 14 describes four buyers and twelve sellers, and their similarity values. The first row, obtained from Figure 12, represents a group of four seller agents having the first priority to be bought by the buyer. Subsequently, the second row represents a group of the other four seller agents having the second priority to be bought by the same buyers. Each priority group is obtained by using the pairing process described in Subsection 2.3.

Furthermore, the priority 2 and 3 are derived subsequently by employing the same pairing process. It is noted that the higher priority groups of sellers are not included in every iteration of the pairing process.

Since the pairs of power seller and power buyer agents have been categorized into several groups having sorted priority indexes, the negotiation process can be carried out step by step according to the priority groups. Figure 16 shows the negotiation algorithm which can be described as follows.

The first step carries out the pairs of power seller and power buyer agents from the first priority, and compares the total demand with the total supply. The total demand is accumulated from the individual required capacities of the buyer agents, while the total supply is aggregated from the individual offered capacities of the seller agents. We define Residue which is equal to the total supply subtracted by the total demand. When the Residue is greater than or equal to zero, the unit commitment problem or the transactions can be solved. However, the second step is handled if the Residue is less than zero.



**Figure 13. Trees with extreme corresponding weights.**

In the second step, the second priority of the power sellers are taken and the Residue is calculated and accumulated with that from the previous step. The steps are further carried until the Residue is greater than or equal to zero.

When the Residue is greater than zero, the prices of the current priority group are sorted in ascending order. Then, a clearing price is determined based on the intersection between the supply curve and the total demand curve. Figure 15 is used to give better explanation of the proposed algorithm.

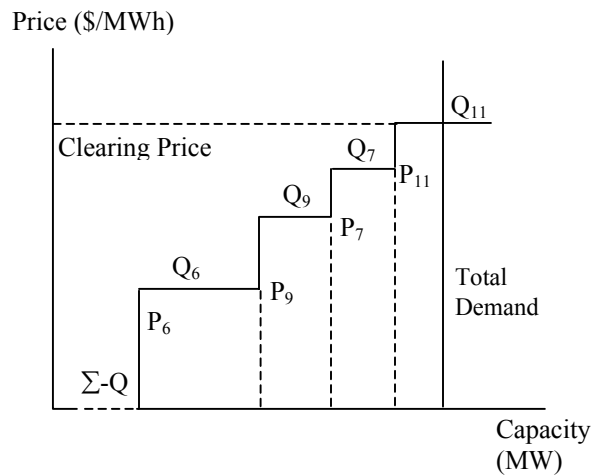
Finally, the clearing price is used to select the power plants. The power plants, whose offered prices are lower than or equal to the clearing price, are selected to supply.

Priority	b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>	b <sub>4</sub>
1	s <sub>3</sub> 0.89	s <sub>1</sub> 0.84	s <sub>5</sub> 0.88	s <sub>4</sub> 0.70
2	s <sub>2</sub> 0.80	s <sub>12</sub> 0.79	s <sub>10</sub> 0.85	s <sub>8</sub> 0.67
3	s <sub>6</sub> 0.75	s <sub>9</sub> 0.74	s <sub>7</sub> 0.76	s <sub>11</sub> 0.64

**Figure 14. Results for extended tree similarity algorithm.**

In Figure 15, the axis presents the capacity in megawatts (MW), whereas the ordinate describes the price in dollars/megawatts-hour (\$/MWh). The offered capacities of the sellers from Priority 3 Figure 14 (s<sub>6</sub>, s<sub>9</sub>, s<sub>7</sub> and s<sub>11</sub>) are aggregated to form the total supply curve. The total demand curve is derived from the summation of required capacities of buyers.

Further, a clearing price is determined based on the intersection between the supply curve and the demand curve, see Figure 15. All of the power plants whose offered prices are lower than or equal to the clearing price are scheduled to supply.



**Figure 15. Determination of the clearing price.**

### Begin

- For  $i = 1, \dots, n$  ( $n$  is the number of power distributors):

Calculate Total demand (required capacity) = aggregation of the subdemands of power distributor  $b_i$

- For  $k=1, \dots, p$  ( $k$  is the priority index) do:

For  $i=1, \dots, n$  select the buyer-seller pair ( $b_i, s_j$ ), where  $j = f(i)$ ,  $f(i)$  being the index of a seller that is paired with  $b_i$ ; in our case these are pairs of power distributors  $b_i$  and power plants  $s_j$ , respectively.

- Calculate the Total supply (offered capacity) = aggregation of the subsupplies of power plant  $s_j$
- Calculate Residue = Total supply – Total demand

Accumulate the Residues as  $\sum \text{Residue}$

If  $\sum \text{Residue} = 0$

then the unit commitment is solved

If  $\sum \text{Residue} < 0$

then the unit commitment cannot be solved

If  $\sum \text{Residue} > 0$

- Sort in ascending order the prices of all  $s_j$ , where  $j = f(i)$ ,  $i = 1, \dots, n$
- Determine the clearing price
- Select the power plants with offered prices lower than or equal to the clearing price

End

**Figure 16. The Negotiation Algorithm.**

## 5. Conclusion

The AgentMatcher architecture has been implemented to compute the similarity and pairing of the power buyer and seller agents. We have found that the proposed similarity based on weight variance selects a more preferred agent among agents having the same similarity value. A negotiation algorithm has been proposed and utilized to handle power transactions when a power distributor (buyer) has to deal with several power plants (sellers) because of capacity reasons.

The current tree similarity algorithm can only compute the similarity of pair of nodes by comparing their string attributes. We are extending the algorithm by considering all arc-labels in the negotiation. Also, we are modifying the similarity algorithm in different ways to take into account the semantic meaning of the string attributes.

## Acknowledgements

This research work is partially funded by the CANARIE eduSource project and NSERC grants of Bhavsar and Boley. We thank Michael M. Richter for discussions and comments about the tree similarity algorithm. We also thank Alexander Chaudhry for proof reading.

## References

- [1] Bhavsar, V.C., Harold B., and Yang, L., "A Weighted-Tree Similarity Algorithm for Multi-agent System in E-Business Environments", Proc. 2003 Workshop on Business Agents and the Semantic Web, Halifax, June 14, 2003, National Research Council of Canada, Institute for Information Technology, Fredericton, pp. 53-72, 2003.
- [2] Boley, H., The Rule Markup Language: RDF-XML Data Model, XML Schema Hierarchy, and XSL Transformations, Invited Talk, INAP2001, Tokyo, October 2001. LNAI 2543, Springer-Verlag, 2003.
- [3] Boley, H., "Object-Oriented RuleML: User-Level Roles, URI-Grounded Clauses, and Order-Sorted Terms", Proc. Rules and Rule Markup Languages for the Semantic Web (RuleML-2003), Sanibel Island, Florida, LNCS 2876, Springer-Verlag, October 2003.
- [4] Clayton, R.E., and Mukerji, R., "System Planning Tools for the Competitive Market", IEEE Computer Application in Power, Volume 9, No. 3, July, pp. 50-55, 1996.
- [5] Dunn, W.H. Jr., Rossi, M.A., and Avramovic, B., "Impact of Market Restructuring on Power Systems Operation", IEEE Computer Application in Power, Volume 8, No. 1, January, pp. 42-47, 1995.
- [6] Ede, S., Mount, T., Schulze, W., Thomas, R., and Zimmerman, R., "Experimental Tests of Competitive Markets for Electric Power", in 34th Hawaii International Conference on System Sciences, 2001.
- [7] Foster, I., Kesselman, C., "The Grid: Blueprint for a New Computing Infrastructure", Morgan Kaufmann Publishers, Inc., San Francisco, California, 1999.
- [8] Lassila, O. and Swick, R.R., Resource Description Framework (RDF) Model and Syntax Specification. Recommendation REC-rdf-syntax-19990222, W3C, February 1999.
- [9] Marsh, S., Ghorbani, A., and Bhavsar, V.C., "The ACORN Multi-Agent System", Web Intelligence and Agent Systems: An International Journal 1, pp. 65-86, 2003.
- [10] North, M. et al., "E-Laboratories: Agent-Based Modeling of Electricity Markets", J. of Algorithms, Vo. 16, No. 1, pp. 33-66, 1994.
- [11] Shasha, D., Wang, J., Zhang, K., "Treediff: Approximate Tree Matcher for Ordered Trees", October 25, 2001.
- [12] Sheble, G.B. and Fahd, G.N., "Unit Commitment Literature Synopsis", IEEE Transactions on Power Systems, Vol.9, No 1, February 1994, pp.128-135.
- [13] Tseng, C.L., Li, C. and Oren, S.S., "Solving the Unit Commitment Problem by Using A Unit Decommitment Method, Journal of Optimization Theory and Applications, Vol.105, No 3, pp.707-730, 2000.
- [14] Wood, A.J., and Wollenberg, B.F., "Power Generation, Operation, and Control", John Wiley & Sons, Inc., New York, 1996.
- [15] Wu, D.J., Kleindorfer, P. and Zhang, J.E., "Optimal Bidding and Contracting Strategies in the Deregulated Electric Power Market: Part I", in 33rd Hawaii International Conference of System Sciences, 2000.