A CONSTRAINT BASED APPROACH TO THE MAINTENANCE SCHEDULING ON ELECTRICAL POWER DISTRIBUTION NETWORKS

Raul Pinheiro, Nuno Gomes, Zita Vale
GECAD - Knowledge Engineering and Decision Support Group
Institute of Engineering, Polytechnic of Porto
Porto, Portugal
raul, ngomes, zav@dee.isep.ipp.pt

Abstract – Power distribution network exploration involves the scheduling of multiple maintenance and unforeseen repair tasks. The scheduling of these tasks is subject to topological, economical and electrical constraints. By is nature, problem modelling is a hard task. Traditional methods have great difficulties on dealing with such complex problems. Constraint Logic Programming (CLP) has great modelling capacities and so it seams the right tool to deal with them. On this work, we propose a new CLP method to solve the problem. This method deals with the electrical constraints on a different manner, avoiding many load flow calculations by means of performing what is known as contingency analysis on electrical networks, reducing computational complexity and speeding the calculations.

Keywords: Distribution Networks, Maintenance Scheduling, Combinatorial Optimisation, Electrical Constraints, Constraint Logic Programming, Contingency Analysis

I. INTRODUCTION

In recent history, the electricity market has suffered profound changes. Liberalization processes, globalization, appearance of small private producers and the increasing institutional investment in renewal energy among others has increased the interest in finding new ways to reduce costs. More than economical factors, modern companies need to provide high quality service in order to maintain the competitiveness. Maintenance Activities Scheduling on the Distribution Network (MASDN) is an important issue on power system operations and also an important competitiveness factor.

The main objective of the MASDN problem consists on determining, for each predicted maintenance task, a specific start time in the scheduling horizon (e.g. a year), while satisfying the system constraints and maintaining system reliability. This objective should be accomplished while some criteria are optimized

The large number of activities and especially the large number of the constraints and their complexity makes the MASDN problem and hard to solve.

For example, the network operator must take into account the limited resources available to perform the maintenance tasks. Basically, equipment, mobile technical staff and mobile generators are limited resources that shall be used wisely avoiding waste. This can be achieved by proper scheduling. Moreover, the network operator also has to consider the dynamic behaviour of the energy demand along the week. As an example, maintenance on network's main lines carrying high current values shall not be done during peak hours. The overload on neighbour lines would possibly become unacceptable. Additionally, he has to meet temporal constraints such as due dates, and apriori fixed dates. On some cases, to carry out maintenance tasks, it is first of all necessary to isolate the affected line section by opening all nearest surrounding switches. Then the network operator has to decide which of the many possible alternative paths of the network will feed the consumers affected by the maintenance task. The number of disconnected consumers has to be minimized. Disconnecting a costumer translates directly into company loss of profit and into a decrease on quality of service indicators. Therefore, this situation is only acceptable whenever it does not exist any reconfiguration that is able to sustain the supply. On the electrical side there are also restrictions. Network operator has to reconfigure the network trying to keep its radial operation (avoiding closed loops) and ensuring the line limits are not violated. Also, bus voltages must remain in a given tolerance interval.

If we keep in mind that electrical distribution networks easily achieve a great number of lines and buses (let's say 2000 nodes), we understand the combinatorial complexity of the optimisation problem.

In order to solve such a complex problem great modelling and solving capabilities are needed, especially if we keep in mind that flexibility is an important issue. Among the actual tools Constraint Logic Programming (CLP) seams to be the right choice.

On CLP the problem model is built in terms of domain variables and constraints. Recent CLP systems as [1] provide a large set of expressive constraints that makes the modelling process an "easy" issue. Problem models are small and very close to a natural description, and therefore simple to modify and maintain. CLP relays on branching to explore the search space, seeking for the optimal solution, and relays on Constraint Propagation to remove infeasible values on the variable domains, avoiding inconsistent solutions and speeding the search.

On this work we propose a CLP model to the MSADN problem. We specifically relay on the description of a propagation scheme for Electrical Constraints as those avoiding closed loops, islands, and that ensures the line limits and bus voltages.

II. PREVIEWS WORK

The MSADN problem is an old problem among the Scientific Computer community. Some works date from early 70's [2], and later in 80's [3]

More recent and closer to nowadays reality works are from William Langdon, [4], [5], [6], [7] and Creemers and all [8], [9], [10], [11]. On the first work Langdon uses Genetic Algorithms to solve a real MSADN problem from the South Wales Region of the National Grid. The problem has been simplified in terms of constraints, since the approach only considers demand and electrical constraints. These constraints are included in the fitness function trough a penalty for each constraint violation. The articles report that the electrical constraints violations are evaluated using the DC load flow. However, nothing is said about implementation details, neither how the DC load flow is used. Despite not making any result comparison, the proposed method success among the Genetic Programming community indicates a good performance.

Differently, Creemers and all use Constraint Logic Programming to develop an application to solve the MSADN problem. They named the application PLANETS and they report their use on the ENHER's dispatching centre. The MSADN problem model considered all type of constraints referred above. The constraints are modelled using Global Constraints [12] dedicated to scheduling and usually available in CLP systems like ECLIPSE [1]. Although, Creemers and all have extended the MSADN problem considering at the same time of the schedule generation the network reconfiguration problem. In order to do that they use three type of domain variables, namely: temporal, topological and electric. The temporal domain variables represent the starting times of the jobs to be scheduled. The topological domain variables represent the states of all network switches in each time slot. Finally, every branch (and switch) of the network needs electric domain variables representing the currents that flow on them. The importance of the current variables lies on their lower and upper limits which will propagate changes to the domains of other current variables, as well as to the domains of topological and temporal variables. This modelling approach requires a lot of computational memory and penalizes the application performance. As an example they refer that for a power-distribution network of about 1200 nodes and 400 operable switches, and 15 maintenance jobs to be scheduled, the system creates about 22000 domain variables.

III. THE CONSTRAINT LOGIG PROGRAMMING APPROACH

Since the beginning of the 1980's a lot of work has been done in constraint based scheduling. An historical perspective is outlined in [13]. Constraint logic programming showed to be suitable for modelling and solving problems on electric circuits. However, the case of distribution networks is particularly difficult due to the large number of nodes, branches and switches

Presently, electric companies are spending a lot of human resources on dealing with the maintenance scheduling problem. However, the search for electrically and economically acceptable schedules is still generally slow, manual and iterative process mainly based on the experience of the network engineers.

One can say that the CLP approach to the maintenance scheduling problem on electrical distribution networks is an optimisation problem that tries to minimize a given cost function subject to some restrictions or constraints. The cost function can be related to energy loss, overloads on the lines, maintenance itself costs or other or even a combination of these. The constraints or restrictions are of different types and can be summarized as following:

- ☐ Resource Constraints – The available amount of resources must be respected at all times. Some of the resources are: maintenance teams, vehicles, tools, replacement parts, etc.
- ☐ Time Constraints – These constraints impose, for each unit to be maintained, the valid maintenance periods. For example, in some periods of the winter some units should not be main-

tained due to the possible bad weather conditions.

- Precedence Constraints – Jobs on ancestor branches must be carried out before jobs on their descendant branches.
- Capacity Constraints – These constraints impose that the production capacity of each unit is never exceeded.
- Demand Constraints – The network shall supply the demand for all consumers.
- Consumer constraints – At all times there must flow a minimal non zero current to all consumers
- Continuity constraints - There must be an electrical path from all nodes to all nodes - no islands.
- Overload constraints - Current on lines must respect its physical limits.
- Bus voltage constraints - Bus voltages must remain in a tolerance interval around their nominal values.

Similarly to [8] these constraints can be easily grouped under three major categories: economical, topological and electrical. The electrical constraints: line overload and bus voltage constraints are the most difficult to model on a constraint logic programming approach, since they involve the calculation of a load flow for each network possible configuration. Load flow calculus is an iterative nonlinear process that slows down the overall CLP algorithm, once it needs to be performed for every network configuration being evaluated. On this article, we propose a new method to consider these electrical constraints based on what is known as contingency analysis, which avoids many of these calculations, speeding the overall algorithm. Before, we will present the CLP model for all the other constraints.

A. Problem Variables and Domains

The main problem variables are the following:

S_i	Starting maintenance period of task i
Out_{bt}	Out_{bt}=1 if branch *b* is electrically isolated. Out_{bt}=0 otherwise.
E_{it}	E_{it}=1 if task *i* is in maintenance in period *t* and E_{it}=0 otherwise.

We also have a graph structure that keeps the network topology. The structure has all the network branches in the form $b(node_1, node_2, state)$, where $node_1$ is the source node, $node_2$ is the sink node, and *state* indicates if the branch is disconnected or not.

B. Problem Constraints

The constraints are imposed over the variables using basic pre-defined propagation algorithms and in some cases more complex ones.

1) Time constraints

Time constraints represent restrictions over the periods when the task can be executed. These restrictions are done over the S_i variable.

$$1 \le S_i + r_i \le T \qquad (1)$$

$$e_i \le S_i \le l_i \qquad (2)$$

Constraint (1) says that the starting maintenance period for task *i* must be greater than 1 and lesser than the last period minus the maintenance duration (r_i). Constraint (2) says that the starting maintenance period for task *i* must be greater than e_i and lesser than l_i, where [e_i, l_i] is the task time window. Other restrictions can be imposed.

2) Precedence Constraints

Precedence constraints establish order among tasks. For example usually is desirable, or even necessary, to execute a maintenance task in a descendent branch just after the ascendant branch is maintained. An extension to this can be the non-overlapping constraint where two constraints can not be executed simultaneously.

$$S_i + r_i \le S_j \qquad (3)$$

$$S_i + r_i \le S_j \vee S_j + r_j \le S_i \qquad (4)$$

Constraints (3) and (4) represent respectively the precedence and non-overlapping constrains.

3) Resources Constraints

Resources Constraints are often one of the most difficult to model and propagate. However they are essential on scheduling problems. This way, several specific propagation algorithms were developed for these constraints and some of them included in CLP systems. This constraint appears as a global constraint and has the following formalism:

cumulative(StartTimes, Durations, Resources, ResourceLimit)

The declarative meaning is: If there are N tasks, each starting at a certain start time, having certain duration and consuming a certain (constant) amount of resource, then the sum of resource usage of all the tasks does not exceed ResourceLimit at any time. This constraint propagation is time consuming, so it must be used wisely. The trade-off between time to propagate the constraint and search tree reduction should be rewarding for the second case.

Another alternative to the resource constraint is:

$$\sum_i E_{it} * rc_{ip} \le ar_{tp} \qquad (5)$$

Being rc_{ip} the resource consumption of type p by the task i and ar_{tp} the available resource of type p for the period t.

4) Electrical Constraints

We classify the remaining constraints as electrical constraints. However we subdivided these constraints in two classes, one handled by CLP only and the other handled by an external algorithm.

The CLP class includes the Continuity constraints. This constraint is modelled using a graph which represents the electrical network topology for each period. Whenever a branch is removed from service for maintenance purposes, a function is called to determine the "Critical Branches". A critical branch is one that, when removed, would make the graph become disconnected. This means that, in order to avoid "islands", and also to ensure that all customers are always supplied, none of the critical branches can be removed.

The class handled by an external algorithm includes the Capacity constraints, the Demand constraints, Overload constraints and Bus voltage constraints. The propagation algorithm for all these constraints is described in section IV and corresponds to the main contribution of this work.

C. Optimization Function

After instantiating all the variables, we end up with a non-optimal solution. In order to perform optimization we need to define a solution cost. The optimal solution is the one that, among all the solutions, that has best costs (minimal or maximal). On this work, cost is defined by the following function

$$Cost = \sum_{it} E_{it} * mc_{it} + Nsc \qquad (6)$$

Where mc_{it} is the maintenance cost for task i in period t and Nsc is the network security cost.

D. The search algorithm

Constraint propagation method is not complete. In fact, after propagating all the constraints we just have the guarantee that all the removed domain values are inconsistent with at least one constraint. However, we have no guaranties that all the remaining domain values are consistent. This way, in order to find a solution we need to sequentially select a variable and instantiate with a value from its domain, until all the variables are instantiated.

When all the variables are instantiated we have a valid solution. A new constraint is then imposed, corresponding to a solution cost upper bound. The search continues by trying alternative values for the variables. The general algorithm of our method is represented in figure 1.

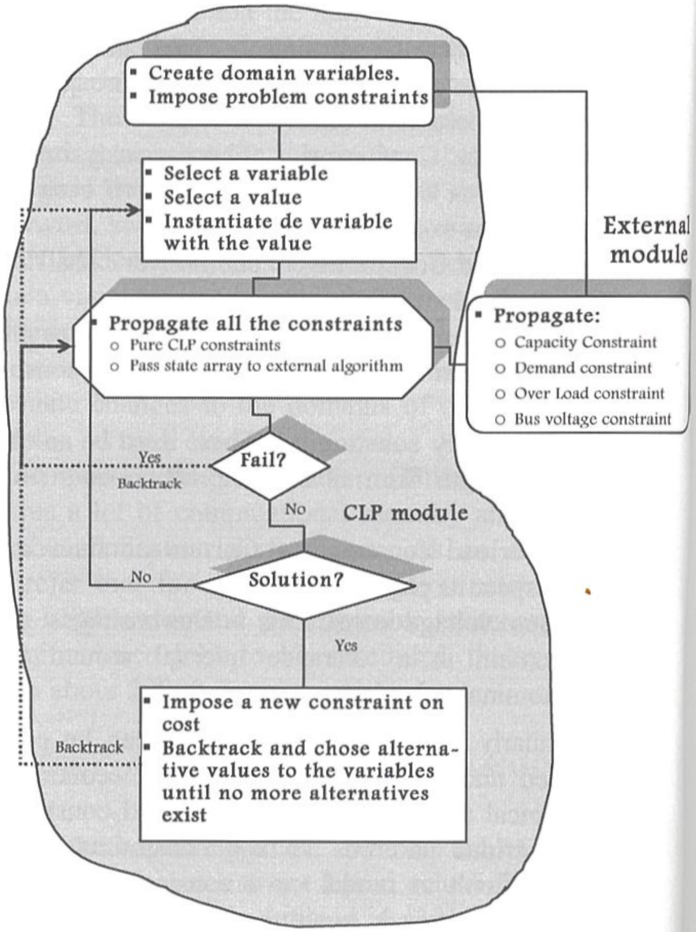

Figure 1- Method Scheme

IV. THE EXTERNAL ALGORITHM

As shown above the external algorithm is responsible for the propagation of several electrical constraints. The algorithm is called whenever a certain line (branch) is removed from the network and the network topology is changed. Then, the algorithm must verify the consistency of the constraints with respect to the changes and, make some *Forward Checking* and *Looking Ahead* [14].

The algorithm starts from a certain network state, where it is know that there are some lines that cannot be removed, to determine which of the remaining lines can be removed without violating any of the electrical constraints. Only these lines and corresponding maintenance tasks can be considered for maintenance in the given period. At the first sight, one has to perform a load flow for each possible line removal to determine which removals violate electrical limits. Is at this particular point that we propose a function that allows to determine which of the "possible to remove" lines can actually be removed respecting electrical constraints and performing just one load flow algorithm, instead of one per "possible to remove" line.

V. CONTINGENCY ANALYSIS

When a line is switched onto or off the system through the action of circuit breakers, line currents are redistributed throughout the network and bus voltages change. The new steady-state bus voltages and line currents can be predicted using what is called contingency analysis. Distribution factors and compensating currents are important concepts in contingency analysis that will be discussed further on this article

VI. DISTRIBUTION FACTORS

Contingency analysis uses two distribution factors named current-injection distribution factor K, and line-outage distribution factor L.

The current-injection distribution factor $K_{ij,m}$ relates the variation of current on line *i-j* with the variation of injected current on node *m*. (see equation (7))

$$K_{ij,m} = \frac{\Delta I_{ij}}{\Delta I_{m}} = \frac{Z_{im} - Z_{jm}}{Z_{c}} \qquad (7)$$

The remaining *Z* elements on the expression refer to the elements of the *Z* matrix obtained considering the impedances of the generators. It is slightly different from the *Z* matrix used for most load flows, because on these we consider the generator buses to be of PV type, where voltage is specified and controlled. On a distribution network, one can consider the injection points to be of infinite short-circuit power and therefore these impedances to be infinite. On this case, the *Z* matrix remains the same.

The line-outage distribution factor $L_{ij,mn}$ relates the variation of current on line *p-q* due to an outage on line *m-n* (see eq. (8))

$$L_{pq,mn} = \frac{\Delta I_{pq}}{I_{mn}} = -\frac{Z_{a}}{Z_{b}}\left[\frac{\left(Z_{pm} - Z_{pn}\right) - \left(Z_{qm} - Z_{qn}\right)}{Z_{th,mn} - Z_{a}}\right] \qquad (8)$$

where Z_a is the *p-q* line actual impedance, Z_b the *m-n* outage line impedance. $Z_{th,mn}$ is the Thevenin impedance seen from *m* and *n* nodes and is given by eq. (9):

$$Z_{th,mn} = Z_{mm} + Z_{nn} - Z_{mn} - Z_{nm} \qquad (9)$$

The other *Zs* are elements of the *Z* matrix.

Both *K* end *L* factors are easy to calculate directly from the Z matrix and the impedances.

So based on the line-outage distribution factor *L* one can compute which lines one can put out of service for maintenance purposes, without overloading the remaining ones.

VII. CURRENT CONSTRAINT

Let us assume that our algorithm receives, besides network data, a vector (size = number of lines) where each element is associated with an electrical line and can assume one of three values:

- Value of –1 – If the line was already removed from the original network
- Value of 0 – If the line cannot be removed from the network
- Value of 1 – If the line can be removed from the network

On a given point of the overall algorithm, some lines might already have been put out of service and we still want to know if there are any more that can also be removed. This is the reason to include the –1 elements on the vector.

The analysis of the constraints other than electrical, might lead to the conclusion that, independently of the electrical constraints, some lines cannot be put out of service. These are labelled 0.

The remaining lines, labelled 1, if put out of service the network still satisfy the non-electrical constraints, and therefore the electrical constraints must be studied.

On the presence of these data, the algorithm computes the *Z* matrix for the actual network. This calculation can be done starting from the "all connected network" *Z* matrix (which can be stored in memory) and then updating it by removing the –1 kind of branches. In our case, as speed is important we shall not build the *Z* matrix from scratch.

Then we must perform a load flow calculation. This can be done using a Newton-Raphson or Gauss-Seidel method. If the network is radial we can use proper methods to compute the load flow, but we must keep in mind that the final operating network shall be radial, but it is not necessary that the one the algorithm is processing to be so. As a matter fact, some maintenance scheduling algorithms assume the network is fully connected in the beginning of the process, and then lines are being disconnected until the network becomes radial. Also, special methods for radial networks are usually faster since they don't have to compute the *Z* matrix. On this case the *Z* matrix will need to be calculated further in the process, and this advantage is therefore lost.

Now the algorithm must modify the *Z* matrix by inserting the generators branches impedances as discussed before.

Then, it calculates the $L_{ij,mn}$ line outage distribution factors for all pairs of lines regarding:

Lines *m-n* are of type 0

Lines *i-j* are of type 0 and 1

Also keep in mind that it makes no sense to compute $L_{ij,ij}$. Following, compute the products:

$$\Delta I_{ij} = L_{ij,mn} \times I_{mn} \tag{10}$$

And finally the new currents if line *m-n* is removed:

$$I_{ij}' = I_{ij} + \Delta I_{ij} \tag{11}$$

For each *m-n* line, compare all the corresponding I_{ij}' with the maximum current allowed on line *i-j*. If the maximum is exceeded, the *m-n* line cannot be removed for maintenance purposes. Otherwise at least one of the other lines will overload.

VIII. COMPENSATING CURRENTS

When considering line additions to or removals from an existing system it is not always necessary to build a new *Z* matrix, especially if the only interest is to establish the impact of the changes on the existing bus voltages and line flows. An alternative procedure is to consider the injection of compensating currents to account for the effects of the line changes. A detailed discussion on adding and removing multiple lines on a system by means of compensating currents can be found at [15]. On our case, one just needs to study the single line case:

For our study, let us suppose impedance Z_a is to be added to the network between nodes *m* and *n*. Removal of a line can be considered as inserting a line between the same nodes with the same but negative value for the impedance. This is, removing a line of impedance Z_b between nodes *p* and *q* is the same as inserting a line of impedance $-Z_b$ between the same nodes.

Due to the insertion of Z_a, the bus voltages *V* will change of an amount *? V*:

$$\Delta V = V' - V \tag{12}$$

where *V* represents the bus voltage vector before the insertion and *V'* the same vector but evaluated after the insertion.

The current I_a in the inserted branch is therefore:

$$I_a = \frac{(V_m' - V_n')}{Z_a} \tag{13}$$

This can be rewritten as follows:

$$Z_a \times I_a = A_c \times V' \tag{14}$$

Where A_c is an incidence matrix (horizontal vector on the current case), whose values are all zeros with exception of the m_{th} ($value = 1$) and n_{th} ($value = -1$).

Compensating currents can be expressed on the form:

$$I_{comp} = -A_c^T \times I_a \tag{15}$$

which is a vector of all zeros but the m_{th} ($value = -I_a$) and n_{th} ($value = I_a$). Therefore:

$$\Delta V = V' - V = -Z A_c^T I_a \tag{16}$$

$$V' = V + Z I_{comp} = V - Z A_c^T I_a \tag{17}$$

$$Z_a I_a = A_c V - A_c Z A_c^T I_a \tag{18}$$

$$Z_a I_a = A_c V - Z_{th,mn} I_a \tag{19}$$

where:

$$Z_{th,mn} = Z_{mm} + Z_{nn} - Z_{mn} - Z_{nm} \tag{20}$$

So, one can compute the I_a current on the base of known values:

$$I_a = \frac{A_c V}{Z_a + Z_{th,mn}} \tag{21}$$

$$I_a = \frac{V_m - V_n}{Z_a + Z_{th,mn}} \tag{22}$$

One can now use I_a value in equation (17) to find the new voltage values on all buses (vector *V'*).

IX. BUS VOLTAGE CONSTRAINTS

To deal with the bus voltage constraints, one has to compute all the new bus voltages resulting of the removal of each of the type 1 lines. The compensating current approach is suitable to do so; regarding that one must use for the Z_a value the negative of the line impedance being removed. If for a given type 1 line removal, the voltage at any of the buses violates the accepted tolerance, then this line cannot be removed.

X. CONSIDERATIONS

One can consider this proposed scheme to be a function of an overall constraint logic programming algorithm. As a matter fact, the function receives as input a vector that explains the state of each electrical line on a given stage of the problem solving. It then evaluates the possibility of line removal due to electrical constraints. This evaluation represents knowledge to the constraint logic programming algorithm's next step, since it can know what lines is can remove without having to compute the corresponding scenario (including a load flow) for each one. Therefore, the function shall return as an output a vector saying which lines can be removed and which cannot. Assuming the same encoding is used, the output vector is the same as the input, but with some of the lines originally labelled 1 now are labelled 0. These lines are the ones that if removed the system still obeys the non-electrical constraints, but doesn't obey the electrical ones.

This function has been implemented in GNU-Octave language (the same as MATLAB, but free). It has been tested on a 24 bus network and shown good performance.

XI. CONCLUSIONS

We modelled the MSADN problem using a CLP approach. CLP has proved to be a powerful tool to deal with this kind of problems. To deal with the electrical constraints part of the problem we propose CLP to use an external function that makes some *Forward Checking* and *Looking Ahead*, returning to the main algorithm the possibilities regarding line removals.

Preliminary tests showed that this algorithm can be a good tool to efficiently solve the MSADN problem.

XII. FURTHER WORK

The immediate task will be to integrate the external algorithm or function with the overall CLP MSADN model. Then some tests and result comparisons shall be done. We are particularly interested on testing this method on large and real instances provided by EDIS (Portuguese electrical distribution company). Hopefully, the resulting overall method will perform faster and will show greater accuracy than the other cited methods, since it deals with the electrical constraints in a more proper manner. If so, we plan to develop an application based on the proposed for EDIS to use on their systems.

To integrate the CLP approach with other optimisation methods such as local search or linear programming could also improve the overall search performance and constitutes therefore a research line.

Also, it would be interesting to apply just the external algorithm to real network operational data, just to check the network reliability to unpredictable outages.

REFERENCES

[1] Mark Wallace, S. N. J. S., "ECLiPSe : A Platform for Constraint Logic Programming," London, 1997.

[2] Egan G., Dillon T., and Morsztyn K., An experimental method of determination of optimal maintenance schedules in power systems using the branch-and-bound technique *IEEE Transactions on Systems, Man and Cybernetics* , vol. 6, pp. 538–5471976.

[3] Yamayee Z., Sidenblad K., and Yoshimura M., A computationally efficient optimal maintenance scheduling method *IEEE Transactions on Power Apparatus and Systems*, vol. 102, pp. 330–3381983.

[4] Langdon W. B., Scheduling Planned Maintenance of the {National Grid}, *Lecture Notes in Computer Science*, vol. 993, pp. 132—153,1995.

[5] Langdon, W. B., "Scheduling Maintenance of Electrical Power Transmission Networks Using Genetic Programming,", *The 1st Online Workshop on Soft Computing (WSC1),* pp. 19--30, 1996.

[6] Langdon W. B., "Scheduling Planned Maintenance of Electrical Power Transmission Networks Using Genetic Algorithms," *Artificial Neural Networks and Genetic Algorithms in Power Engineering,* 1997.

[7] Langdon, W. and Treleaven, P. Scheduling Maintenance of Electric Power Transmission Networks Using Genetic Programming. In: *Artificial Intelligence Techniques in Power Systems*, .pp. 220-237, 1997.

[8] Creemers T., Ros L., Riera J., Ferrarons C., Roca J., Corbella X., "Constraint-based Maintenance Scheduling on an Electric Power-distribution Network," *{PAP}'95: Proceedings of the Third International Conference on Practical Applications of Prolog,* 1995.

[9] Creemers T., Ros L., Riera J., Ferrarons C., Roca J., Corbella X, "Programación de Tarefas de Mantenimiento Y Reconfiguración Sobre Redes de Média Tensíon," *4as Jornadas Luso-Espanholas de Engenharia Electrotécnica,* Porto, Portugal, 1995.

[10] Creemers T., Ros L., Riera J., Ferrarons C., Roca J, Constraint Logic Programming for Fault-Tolerant Distributed Systems, *Journal of Computer and System Sciences International* , vol. 37, pp. 689-699, 1998.

[11] Creemers, T., Ros, L., Riera, J., Roca, J., and Ferrarons, C., Smart Schedules Streamline Distribution Maintenance, *IEEE Computer Applications in Power*, vol. 11, pp. 48-53, 1998.

[12] Beldiceanu N., Introducing global constraints in CHIP *Mathematical and Computer Modelling*, vol. 20, pp. 97-123, 1994.

[13]Le Pape C., "Constraint-Based Programming for Scheduling: An Historical Perspective," *Operations Research Society Seminar on Constraint Handling Techniques,* London, United Kingdom, 1994.

[14]Haralick R. and Elliott G, Increasing Tree Search Efficiency for Constraints Satisfaction Problems, *Artificial Intelligence*, 14, pp. 263-313, 1980.

[15]Graninger J. and Stevenson W. *Power System Analysis*, McGraw-Hill, 1994.