Smart Schedules Streamline Distribution Maintenance

Tom Creemers,[1] Lluís Ros,[1] Jordi Riera,[1] Josep Roca,[2] Carles Ferrarons[2]

PLANETS helps optimize and coordinate the use of the power company's resources.

Imagine the following scene in the dispatching control room of a large electric company. Within the following hours, several scheduled maintenance activities are to be carried out on the medium-voltage network. To keep on supplying energy to some customers that are about to be cut off from the distribution system due to these activities, a planning engineer requests the dispatching of a team of technicians to close tie switch 2378. Soon afterwards, however, the engineer realizes that this costly displacement of manpower could have been avoided, if switch 2378 was not opened after yesterday's activities on a nearby line section.

This situation is far from exceptional. The scheduling and coordination of maintenance tasks and resource dispatching, with the aim of guaranteeing service to all customers, forms an essential part of the planning engineer's tasks. Least-cost solutions are key.

Use of constraint-programming methods simplifies the elaboration of optimal schedules for distribution-system maintenance and reconfiguration

This article features a novel application, based on a constraint-logic programming paradigm, for scheduling maintenance activities on large distribution networks. Examples are included from system tests performed on subsets of the distribution network of Empresa Nacional Hidroeléctrica del Ribagorzana (ENHER), a Spanish electric company.

Scheduling Problems and Solutions

Planning engineers are faced with a complex problem: the preparation of the maintenance schedules for the distribution network. The design of such a schedule involves basically two subproblems.

- An outage calendar must be elaborated. Given a list of maintenance and repair activities to be carried out during the coming week, one has to distribute these over time, specifying for each one of them at what time the affected area will be isolated (to be able to work safely in its interior) and when it will be reconnected again to the network. During the time it is isolated or out of service, it is referred to as an *outage area*.
- Optimal reconfiguration plans must be designed. One has to decide how the system's topology has to evolve along the calendar to guarantee the continuity of energy supply to a maximal number of customers affected by the disconnection of line sections, while minimizing the cost to realize the necessary topology changes.

Although we are dealing with two distinct subproblems, they are not independent at all. To find the solution to one of them, one has to take into account the solution space of the other, and, hence, it is appropriate to tackle both simultaneously as one single global optimization problem.

Mathematically speaking, the latter can be classified as a highly combinatorial NP-complete problem, which makes

[1] Institut de Robòtica i Informàtica Industrial, Spain
[2] Empresa Nacional Hidroeléctrica del Ribagorzana, S.A., Spain

 ISSN 0895-0156/98/$10.00©1998IEEE

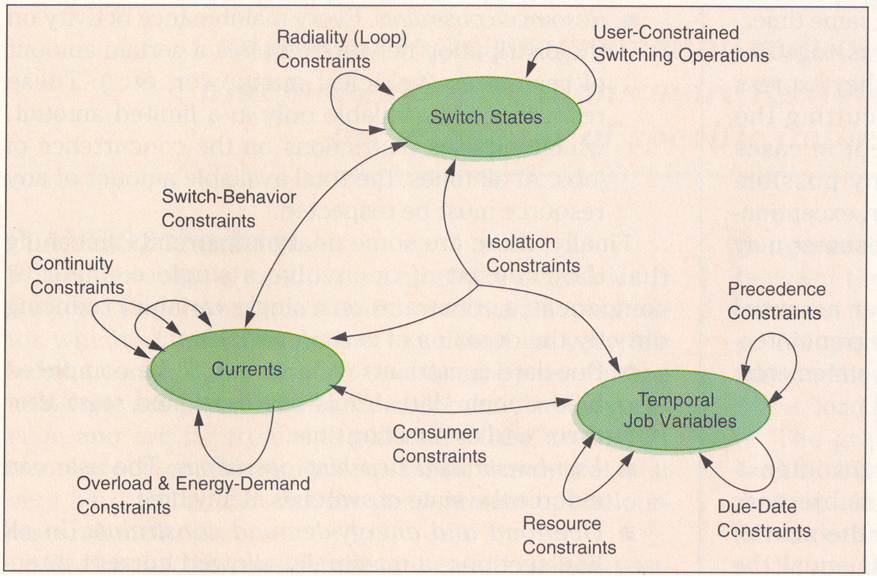

Figure 1. High-level representation of the constraint network

it a tough candidate to be tackled by classical search and optimization methods. In the following, we expose the variables involved in it, the constraints that delimit its search space, and the cost function we intend minimizing.

Variables

Three types of variables are used to characterize a maintenance schedule in an unambiguous way:

- *Temporal job variables* represent the start and end times of the jobs to be scheduled. They define the look of the outage calendar. Their values range from 1 to the number of time slots. In the case examined later in this article, a 5-day week is discretized in 15 available start/end slots, with three working periods per day, in accordance with ENHER's operation policy.
- *Topological variables* represent the states of all operable switches in the network in each time slot of the calendar. A state can be either *open* or *closed*. The states of all switches define the topological configuration of the distribution network and allow to formulate any reconfiguration of it.
- *Electric variables* are the current intensities in all line sections and switches in each time slot of the calendar. They define the flow of current through the network along the considered calendar.

Constraints

Among the three variable types, there are many constraints. They constitute a dense network relating variables to many others and forming the basis for effective constraint propagation. A high-level representation of this network, dividing constraints in various classes, is represented in Figure 1.

- *Isolation constraints* express the requirement that during maintenance, for safety reasons, the work area must be electrically isolated from the rest of the distribution network. Currents in the area are constrained to 0, and surrounding switches must be open in the corresponding time slots. These are the most important constraints, since they relate (con-

Constraint Logic Programming

In recent years, the application of artificial-intelligence techniques to planning and scheduling problems has experienced considerable growth.

In particular, the relatively new methodology of constraint logic programming (CLP) has proven to be quite suitable for coping with discrete, combinatorial, NP-complete problems. The goal is always the same: to find, in a discrete space, a point satisfying a set of constraints.

CLP was developed with the aim of reducing development time and giving more flexibility to the applications, while preserving the same efficiency of specific implementations. This was achieved by embedding consistency techniques inside the classical logic programming framework. Thanks to the nondeterminism of logic programming, the user is freed from the tedious task of developing tree-search algorithms. The expression of constraints is straightforward thanks to its relational form, and its declarative semantics make the code easy to modify and extend.

Consistency techniques use constraints in an active way. They are used to prune the search space a priori, by removing values from the domain of a variable that is inconsistent with all the remaining values in the domain of another variable, and hence cannot appear in any solution. Basically, the search is an iteration of two steps:

- Propagation of constraints as much as possible (the reduction of variable domains based on consistency)
- Assumption of values for some variables.

until the problem is solved. In the second step, two choices have to be made: determining the next variable to be instantiated, and choosing a value from its remaining domain.

For Further Reading

P. Van Hentenryck, *Constraint Satisfaction in Logic Programming*, MIT Press, 1989.

A.K. Mackworth, "Consistency in Networks of Relations," *Artificial Intelligence*, Volume 8, pages 99-118, 1977.

strain) variables of all three types at the same time.

- *Consumer constraints* constrain currents together with temporal variables. Qualitatively, they express the fact that we do not ever allow cutting the power supply to any consumer, except in cases where there really does not exist any possible reconfiguration to avoid so. In the latter, exceptional case, the current flowing to the consumer may be temporarily 0.
- *Switch-behavior constraints* are another essential class, relating a switch's state S to its current intensity I. They express the two conditional statements:

  If $|I| \geq 0$, then S = closed

  If S = open, then $I = 0$.
- *Continuity constraints*, to which current intensities in different line sections are subjected, express Kirchoff's law of continuity: the sum of all incoming currents in a node must equal the sum of all outgoing currents.
- *Radiality constraints* are imposed on operational policy. At all times, the distribution network must have a radial (tree shaped) topology, i.e., there may not exist any closed loop. At least one of the switches in a possible loop must be open at any time.

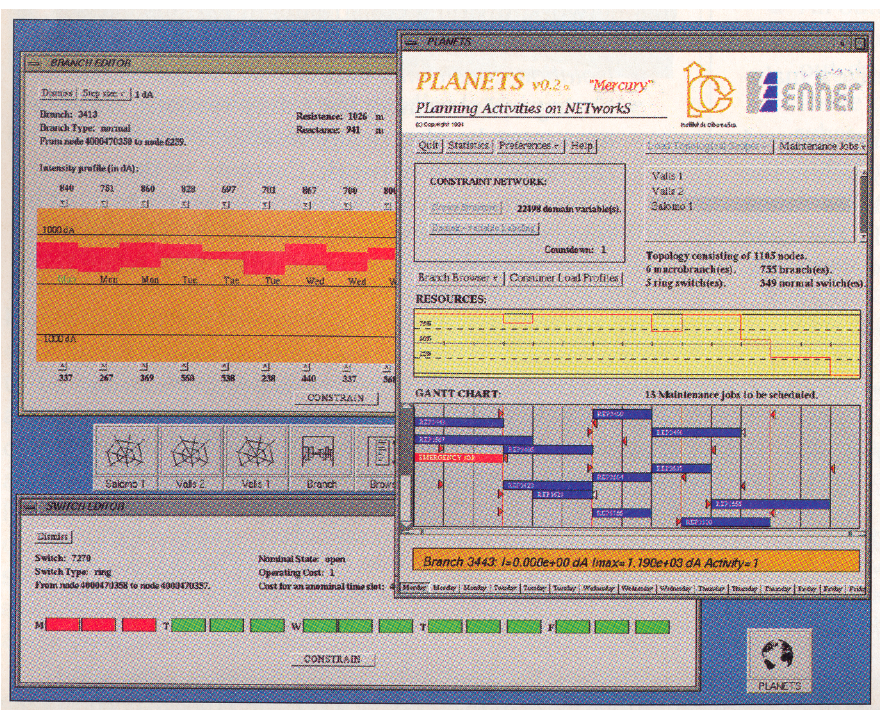

Figure 2. View of PLANETS' interface, main window and editors

The following two constraint classes involve only job-time variables:

- *Precedence constraints*. Activities on parts of the distribution tree closer to its root are to be performed before activities on descendant parts of the same tree. Additionally, arbitrary priorities might be assigned to jobs.
- *Resource constraints*. Every maintenance activity on the distribution network requires a certain amount of resources (vehicles, manpower, etc.). These resources are available only in a limited amount, which imposes restrictions on the concurrence of jobs. At all times, the total available amount of any resource must be respected.

Finally, there are some unary constraints (meaning that they consist of or involve a single element or component; a constraint on a single variable) reducing directly the domains of certain variables:

- Due-date constraints. All jobs are to be completed before their "latest finishing time" and start after their "earliest starting time."
- *User-constrained switching operations*. The user can enforce the state of switches at any time.
- *Overload and energy-demand constraints*. In all line sections, a maximally allowed current intensity is imposed. In lines connected to consumer nodes, a time-dependent minimal value can be imposed, reflecting the energy-demand profile at these nodes.

Note that the existence of all these classes of constraints prunes the space of possible outage calendars and, at the same time, the space of possible reconfiguration plans. This fact justifies once more the simultaneous tackling of the two previously defined subproblems.

Quality of the Adopted Schedule

Due to the fact that many companies' distribution systems are not yet telecontrolled, and, therefore, teams of personnel almost always have to be dispatched to carry out any topological change, the companies want to minimize the cost associated with these displacements. By adequately rearranging the outage calendar and imposing intelligent network reconfigurations, many redundant operations can be avoided. The total cost to carry out all necessary switching operations forms the objective function to be minimized. For this purpose, every individual switch is labeled with three attributes:

- Nominal state
- Cost to operate it (to change its state)
- Cost to leave it out of its nominal state during one time unit or time slot.

Optionally, we might consider the due-date constraints as *soft* ones, allowing them to be violated in certain cases, but assigning an additional cost to any delayed completion or early start. These costs can then be incorporated in the objective function.

Constraint logic programming reduces development time and makes applications more flexible, while preserving the efficiency of specific implementations

PLANETS Scheduler

The elaboration of optimal maintenance schedules is still, to a large extent, an iterative and manual process, for which planning engineers can count on very few helpful tools. The solutions proposed by the human scheduler are principally based on accumulated experience and are far from optimal due to the problem's tremendous combinatorial explosion, which makes it very hard to handle even for conventional operations-research methods.

Because of the considerable, inherent constraint-satisfaction component, we prefer attacking the problem by means of more powerful and flexible techniques of constraint (logic) programming (CLP), resulting in a first prototype system named Planning Activities on Networks (PLANETS). The system has been successfully tested on several subsets of ENHER's distribution network. The system optimizes both the outage calendar and its corresponding reconfiguration plan. It makes use of CLP techniques in the field of distribution-system reconfiguration.

System Architecture.

The kernel of PLANETS is a constraint-propagation engine completely written in CHIP (Constraint Handling In Prolog), a language which, apart from its standard constraint-handling facilities, offers specialized scheduling constraints and optimization predicates based on branch-and-bound and simplex techniques. The problem constraints outlined earlier were modeled in a straightforward manner and posted declaratively using this language. The actual scheduler (apart from the graphical interface) occupies no more than 12 pages of Prolog code, a fact that illustrates the expressive power of this paradigm.

Although the electric laws that rule the behavior of the distribution system are already partly embedded in the problem's constraint model (mainly Kirchoff's law of continuity), the system has been interfaced to the load-flow C-library DISPOT, developed at ENHER, which allows a more accurate simulation of the network's operation, once the scheduling process has terminated. This makes it possible to obtain precise values for the currents and voltages in any time slot of the generated maintenance schedule. The connection between these two previous modules is established by means of the C Language Interface to CHIP (CLIC).

An up-to-date database provides the necessary information about the network topology as well as the predicted load for the coming week.

The graphical user interface (Figures 2 and 3), based on X Windows and developed with CHIP's XGIP module, allows the easy selection of those parts of the network that are to be considered in the construction of the maintenance schedule, as well as the easy introduction of all data relating to the activities that need to be scheduled. Results can be visualized on a Gantt chart representing the proposed maintenance schedule together with a profile diagram showing the resource usage over time. Clicking on any point of the time axis pops up graphical tree representations of the network's topology scheduled for that specific time slot. To maneuver the network from the topology in a given slot to the next, the system also provides a sequence of switching operations that guarantees a fail-safe transition (Figure 4).

Scheduling Example

Figure 5 visualizes parts of the results of an example

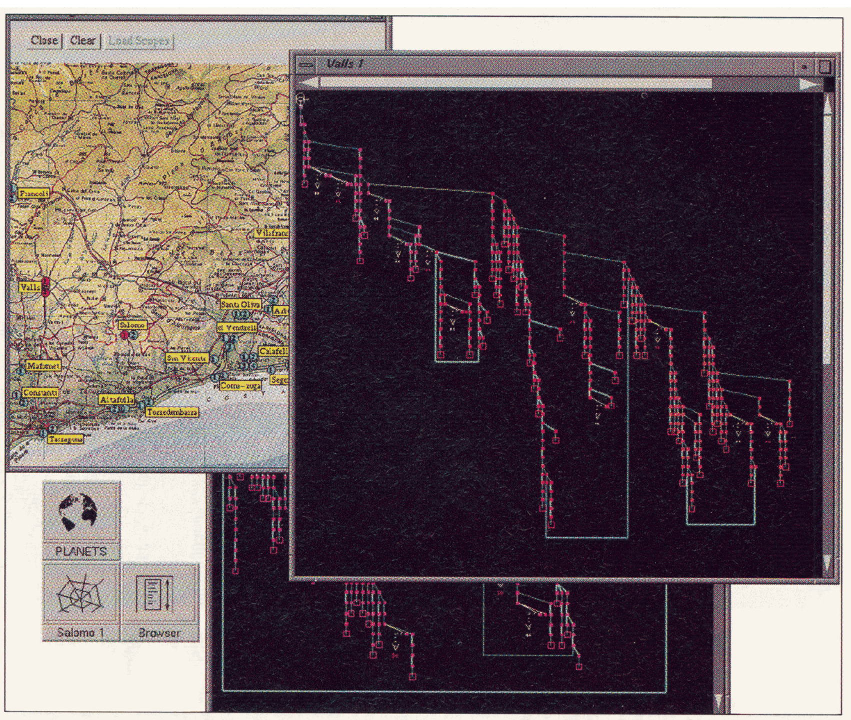

Figure 3. Graphical representation of selected parts of the network

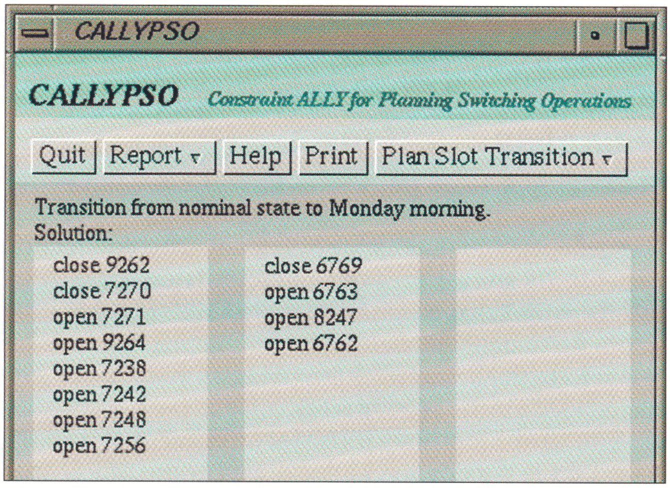

Figure 4. Sequence of fail-safe switching operations to maneuver between two topologies

run for a particular case requiring the scheduling of 15 maintenance and repair activities. The jobs are to be carried out during the coming week and will take place in three interconnected feeder areas (numbered 50, 51, and 32).

Assuming the dispatching center has previously evaluated the costs related to the operation of all switches, as well as the penalties corresponding to the violation of due dates, the planning engineers have to supply the system with the data related to the list of activities to be scheduled. For each activity, one needs to specify the line sections to be isolated, the amount of resources required (of diverse types), and, optionally, the job's priority, earliest starting time (EST) and latest finishing time (LFT). The user can choose between expressing the EST and LFT either as hard or soft constraints.

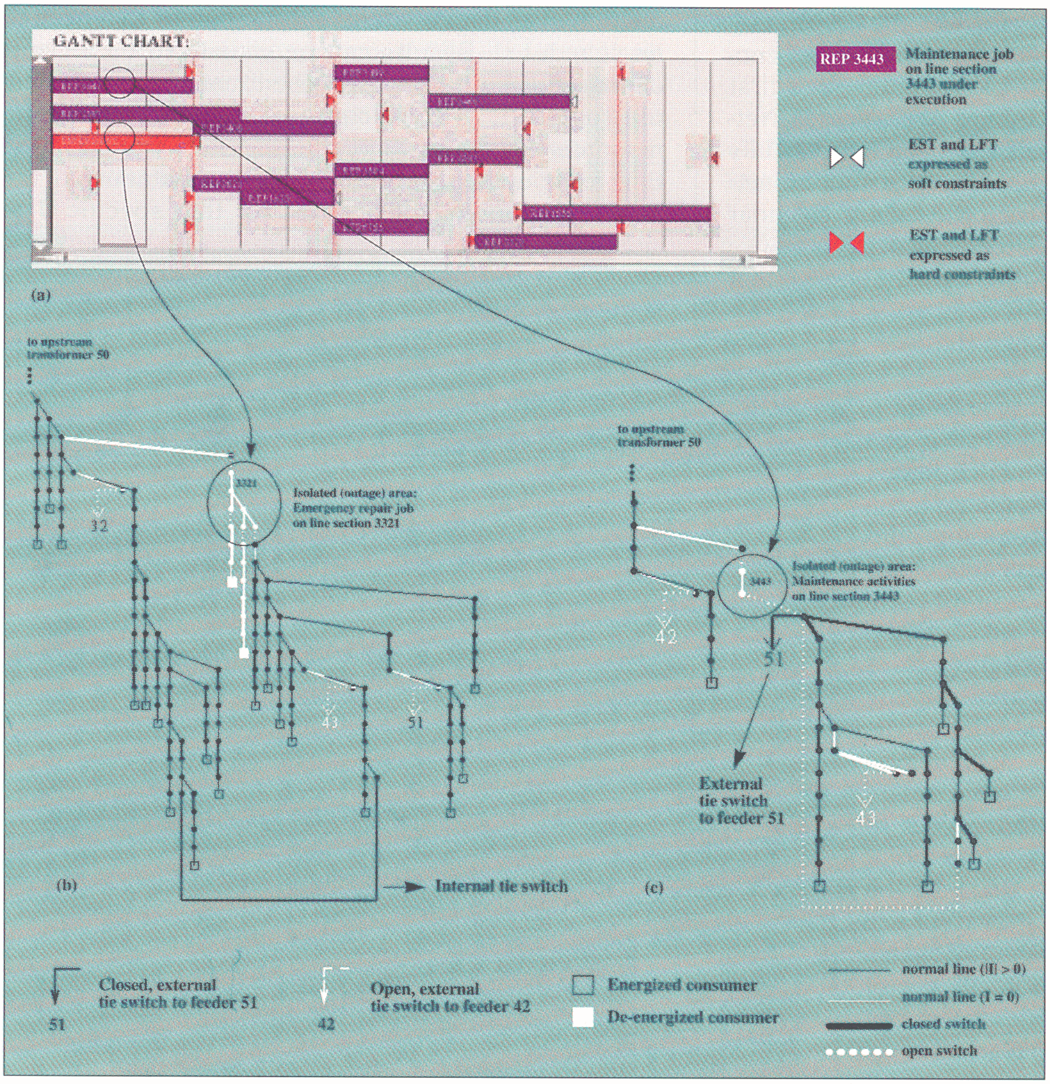

Figure 5. The optimal schedule is represented by means of (a) a Gantt chart. Topological changes, during the second time slot, proposed by the system to perform maintenance activities on (b) line section 3321 (emergency job) and (c) line section 3443 of feeder 50. (The simultaneous maintenance job on line section 1567 of feeder 32 is not shown.)

Outage Calendar

Figure 5a displays the optimal outage calendar proposed by the PLANETS system. The time axis is discretized in 15 relevant time slots. Jobs are represented by horizontal bars. The small triangles at both sides of the bars denote the EST and LFT.

Reconfiguration Plan

Figure 5 (b and c) shows the necessary topological reconfigurations proposed by the system to maintain power supply to all consumers situated in portions downstream of the zones under maintenance. Nodes at zero voltage and line sections carrying no current are painted white.

Only a small subset of the distribution network under consideration is shown in this figure. One can imagine a transformer at the root of both trees connecting the high-voltage transmission network to the medium-voltage distribution network. Consumers are situated at *leaves* of the trees and are depicted as boxes.

Switches connecting two parts of the same tree are called internal tie switches. If they connect two distinct trees, they are called external tie switches.

Figure 5b represents the proposed solution to perform the emergency job on line section 3321. Urgent repair activities like this one are due to unforeseen, sudden failures and are scheduled with top priority. As can be seen, in order to de-energize line section 3321, all nearest surrounding switches are opened. In this particular case, the latter results in the temporary cutoff of two consumer nodes because there are no switches available to connect them to neighboring energized lines. However, the major part of the downstream subtree has successfully been recovered thanks to the closing of an internal tie switch. Note that there are two other switches (those connecting with neighboring feeders 43 and 51) that could have been chosen to be closed for this same purpose. However, as a result of the optimization process, the system has opted for the internal tie switch, due to the lower cost associated with it.

Bear in mind that the proposed solution for a particular outage always takes into account (and is influenced by) all other simultaneous, past or future jobs. This is a consequence of the connectivity of the constraint network in topological, electric, and temporal dimensions.

Figure 5c shows the topology adopted to energize the area downstream of line section 3433. In this case, the affected subtree will be fed from a different transformer in the neighboring feeder 51.

There are many other cases in which the recovery of a subtree is not solved by the mere closing of one single tie switch. For example, if the subtree is too big and contains a high load of consumers, it is possible that, when trying to transfer many of these to a neighboring energized area, the current-intensity limits on certain line sections are exceeded. In such cases, the proposed solutions are more sophisticated than the ones outlined in Figure 5, and imply the division of the subtree in different partitions and the load transfer of each one of these to a different neighboring feeder.

Computational Cost

The scheduling example involved a subset of the ENHER distribution network containing about 2,400 line sections and 800 operable switches. Executed on a Sun SuperSparc 20 workstation running Solaris 5.5.1, the PLANETS scheduler required about 2 minutes of CPU time to produce an optimal schedule together with the corresponding topology reconfigurations. The major part of this time (about 70 percent) is spent on the construction of the underlying constraint network. The computational cost of the latter is almost linear with the size of the network. In the example case, the network comprises a total amount of 25,000 domain variables and a comparable amount of constraints between them.

Acknowledgments

This project is the result of a private research contract between ENHER, S.A. and the Institut de Cibernètica (UPC/CSIC). PLANETS is currently being integrated with the company's online databases and other information resources. Given the successful offline tests, PLANETS is expected to be taken into online use as soon as this integration period is completed.

We are most grateful to Manel Batlle from ORIGIN for his valuable help in the development of PLANETS' interface to the electric simulation library DISPOT.

For Further Reading

M. Dincbas, P. Van Hentenryck, H. Simonis, A. Aggoun, T. Graf, F. Berthier, "The Constraint Logic Programming Language CHIP," *Proceedings of the 1988 International Conference on Fifth-Generation Computer Systems*, edited by ICOT, pages 693-702, 1988.

T. Creemers, L. Ros, J. Riera, C. Ferrarons, J. Roca, X. Corbella, "Constraint-Based Maintenance Scheduling on an Electric Power-Distribution Network," *Proceedings of the Third International Conference and Exhibition on Practical Applications of Prolog*, Paris, 1995, The Practical Application Company, Blackpool, Lancashire, UK, ISBN: 0 9525554 0 9, pages 135-144. Also available through the Web, http://www-iri.upc.es/people/planets.

A. Aggoun, N. Beldiceanu, "Extending CHIP in Order to Solve Complex Scheduling Problems," *Journal of Mathematical and Computer Modeling*, Volume 17, Number 7, Pergamon Press, 1993, pages 57-73.

N. Beldiceanu, E. Contejean, "Introducing Global Constraints in CHIP," *Journal of Mathematical and Computer Modeling*, Volume 20, Number 12, Pergamon Press, 1994 pages 97-123.

Biographies

Tom Creemers is a computer science engineer at the Institut de Robòtica i Informàtica Industrial at the Universitat Politècnica de Catalunya-Consejo Superior de Investigaciones Científicas (UPC-CSIC), Barcelona, where his research is mainly focused on constraint logic programming techniques applied to advanced scheduling, planning, and decision-making. He received his MS degree in computer science engineering and his master of engineering degree in artificial intelligence from the Katholieke Universiteit Leuven, Belgium, in 1989 and 1990. Afterwards he formed part of the Expert Systems Application Development group of the chemical engineering department at the same university. Readers can reach him at tcreemers@iri.upc.es.

Lluís Ros is a research engineer at the Institut de Robòtica i Informàtica Industrial at UPC-CSIC, where he is involved in the development of the PLANETS system. Currently, he is also working towards his PhD in geometric constraint satisfaction algorithms. His research interests include combinatorial optimization, geometric constraint programming, genetic programming and their application to power systems engineering. He received his MS degree in mechanical engineering from the Polytechnic University of Catalonia in 1992. Readers can reach him at llros@iri.upc.es.

Jordi Riera is a research engineer at the Institut de Robòtica i Informàtica Industrial at UPC-CSIC and at the Institut de Cibernètica at UPC. His current research interests include power systems and nonlinear control. He received his PhD from the Polytechnic University of Catalonia in 1985. Readers can reach him at jriera@iri.upc.es.

Josep Roca is a senior engineer at ENHER S.A., Barcelona. He has a MS degree in power engineering from the Polytechnic University of Catalonia. He is involved heavily in distribution systems planning, control, and stability.

Carles Ferrarons is a senior engineer for ENHER S.A., Barcelona. He has a MS degree in telecommunications engineering from the Polytechnic University of Catalonia. He has been an assistant professor at the UPC for several years. After joining ENHER, he has been involved in power systems control, stability, and simulation.