A Problem Classification Scheme for Finite Domain Constraint Solving

H. Simonis
COSYTEC SA
4, rue Jean Rostand
91893 Orsay Cedex
France
helmut@cosytec.fr

1. Abstract

In this paper we give a classification of problems solved with finite domain solvers in Constraint Logic Programming (CLP). This scheme tries to explain which types of problems can effectively be solved with CLP and which problems are best solved with competing techniques like integer programming or local search. The survey is based on industrial studies and applications developed over the last ten years. We also try to find explanations for failures in some projects and give a list of critical issues in problem solving with constraint logic programming.

2. Introduction

This paper gives an overview of application oriented research and end-user applications developed with finite domain solvers inside the Constraint Logic Programming framework. We classify these studies in different problem categories and evaluate the difficulties and successes in handling these problems. The aim is to give a problem classification scheme which helps to decide whether a particular problem is a likely candidate for constraint logic programming with finite domain constraints.

2.1 Approach

The approach used in this survey is case based. Starting from a set of application studies, we try to find explanations why some problems are more easy to solve than others. We should contrast this to a more systematic approach which would try to find solvable problem classes based on mathematical properties of the problems and of the finite domain problem solvers. This later approach would be much more interesting with its deep insights into problem structure. But since finite domain solvers now use a variety of methods and algorithms, it seems unlikely that such an analysis is within reach of current research.
It should also be noted that the author is much more familiar with results using CHIP of COSYTEC than with other tools. As many of the successful applications are not described fully in the open literature, this may lead to a certain bias towards applications developed by COSYTEC or its customers.

2.2 Influence of tool

The influence of the tool used should not be underestimated. While most finite domain solvers contain simple syntactic [Com95] constraints like arithmetic, they differ widely in the availability and strength of so-called global constraints [Bel94]. These global constraints, like the cumulative [Agg93] or cycle [Bel94] constraints in CHIP, allow a much more concise model of a problem as well as more powerful

constraint handling. The propagation is not limited to simple methods like forward checking or lookahead, but uses combinations of techniques from Operations Research, mathematics and artificial intelligence.
Many of the applications mentioned in this paper rely on this power of the global constraints of CHIP.

2.3 Alternative models

For many problems, there is no unique best constraint model. The same problem can be expressed with different types of constraints. In many practical problems, models can also differ based on which constraints are included in the model. Often, some constraints are considered as preferences, which may or may not be satisfied. Different users may have different opinions on which constraints are essential to the problem and which constraints can be ignored in a model.
For that reason, it is very difficult to say that the best model of a problem has been found or even that some type of problem has been completely solved using CLP.

2.4 Difference between study and application

Many of the problems presented in this survey are application studies, performed to test a solver against a particular problem. Other examples are full featured end-user applications, used in day-to-day operations. The difference between these two types of solutions can not be overestimated. It is quite common to perform a preliminary study within a week or two, concentrating on the modelling aspects and a single data set. End-user applications, on the other hand, typically need development times between 6 months and a year. Much of the development effort is spent on interfaces and data management. Stability over all data sets and explanation facilities in case of inconsistency become important development tasks for the problem solver.

2.5 Scalability

A number of application studies mentioned in this overview do not scale properly. This means that the model used to express the problem is either too complex or not powerful enough to handle large, realistic data-sets. Sometimes, a change in the model can solve this difficulty. In other cases this is an indication that constraint programming may not be the right tool to express the problem. We will try to indicate such cases in the overview.
In a similar way, there can be successful end-user applications, which only solve trivial constraint problems. This does not really matter. The aim of an end-user application is not to apply some complex techniques, but to improve a business process (and eventually earn some money).

3. Related research

There is little systematic research into application domains suitable for finite domain constraints. [VH89] presents a variety of application studies which are used to show the usefulness of the finite domain concept. The paper [Din88a] gives an overview of early attempts to solve problems. But at that time no end-user applications had been developed, so that most studies were comparisons with benchmarks against other tools.
The survey article [Jaf94] mainly describes the foundations of CLP techniques and different problem solvers. The constraint tool comparison [Cra94] concentrates on the features of the tools. Applications are not discussed in detail. The paper [Sim95]

describes a number of applications, but concentrates on development issues rather than classification. The survey article [Wal95] discusses many issues also handled here. But the viewpoint is more on the applied research aspect, showing how a number of open questions can lead to new research in the constraint domain.
The user group meetings of the different suppliers [COS94] [ILO95] [COS95] mention a number of applications which are not accessible from other literature.
The PAP and PACT conferences have shown a significant number of papers on application studies, less so on end-user applications.

4. Problem classification

We now begin with our application classification.

4.1 Overview

Each of the application classes is described in a similar way. We first briefly present the problem, explaining the main constraints. We then discuss the different programs which fall within this category. In some cases, we present one or two examples of end-user applications in this domain. Finally, we try to evaluate the usefulness of constraint logic programming for this type of problem.

4.2 Hardware design

Hardware design was one of the first application domains explored for constraint logic programming [But87][Sim92]. Many of the early applications use demon-based local propagation or Boolean unification [Sim93]. One application study of Dassault Aviation and ECRC [Gra89] combines the rational and the finite domain solvers of CHIP. An industrial tool for circuit verification, the CVE system from Siemens, using an extended Boolean unification, is described in [Fil91].

4.2.1 Problem

Two of the many design problems which can be handled with finite domain constraints should be mentioned here.

Channel routing is the problem of connecting some terminals on two sides of a rectangular area by horizontal and vertical line segments in such a way that no line segments occupy the same place. This problem occurs in the layout of integrated circuits. The details of the problem vary according to the technology used. This problem can be expressed as a 3-dimensional placement problem with additional constraints. It is described in [Sim92] [Zho95] [Duc96]. While results are not quite as good as those obtained by dedicated routing programs, they show the ease of solving such problems with constraints.
Another application type is the design of digital signal processing hardware. Certain aspects of this hardware are easily expressed as resource restricted scheduling problems. Very good results are obtained in [Bel94][VH94] with minimal effort.

4.2.2 Evaluation

Hardware design is a rather narrow field with few major tool suppliers. Most applications in this domain are custom coded for optimal performance, with a significant loss of flexibility. CLP can bring a number of benefits to this domain, but it is quite difficult to replace very well performing existing tools.

4.3 Placement

4.3.1 Problem

The problem in its most simple form consists in placing several n-dimensional objects into a restricted n-dimensional space so that objects do not overlap each other. Expressing this non-overlapping constraint only with syntactic constraints like inequality leads to a large number of disjunctive constraints which are very difficult to use for propagation. The introduction of the diffn constraint [Bel94] has made this domain much more accessible.

4.3.2 Studies

One of the earliest applications in this domain was part of the HIT container harbour scheduling project of ICL [Per91]. It decided on the best area to store containers between arrival and departure at the harbour. While not solving the individual placement problem for each container, it looked at the available space in different areas in the storage space before unloading ships.

The lorry loading application of EBI [COS95] solves a problem of stacking boxes in a lorry. It uses a number of heuristics to fill the container taking into account the order in which boxes must be unloaded at the different destinations.

In a similar application at Michelin, boxes are stored into a set of containers, which adds another degree of freedom. We not only have to place the boxes into the containers, but also assign them to a container.

A classical placement problem is the map labelling problem. For a two dimensional map, the labels of the different locations must be placed close to the location without overlapping other labels. There are preferred locations of the labels, but it is impossible to achieve a solution using these preferences only. This problem was first studied at ECRC, then at Bull [Cra94a]. [Cra94a] explains the difficulty encountered when trying to solve the problem with syntactic constraints only. A solution of the over-constrained problem with diffn is now a standard CHIP demo.

4.3.3 Evaluation

We must distinguish between two dimensional and higher dimensional problems. The 2D case is quite well understood [Agg91] [Bel94]. Using global constraints, even over-constrained problems with several hundred items can be easily solved with high quality. The three or four dimensional problems can be much more difficult. Good results have between obtained for some packing problems [Bel94]. But the heuristics required are non trivial and require a certain programming effort. In addition, some of the constraints encountered are quite difficult to express. If we require that a box is supported either by the bottom or by other boxes, we have to determine the centre of gravity of a stack of boxes. On the other hand, there exist many simpler placement problems which can be solved in a straightforward manner.

4.4 Cutting problems

4.4.1 Problem

Cutting problems are related to placement problems. The task is to cut (usually in two dimensions) pieces of given size from larger pieces or from a continuous roll. Their difference from placement problems lies in the constraints on the type of cut. Often, only Guillotine cuts are allowed, cuts which cut along the full length of the larger piece. The material (say glass) may impose additional constraints.

4.4.2 Studies

An early comparison of constraint techniques with IP was given in [Din92]. It handles the cutting of wood pieces from rectangular pieces. A similar problem has been solved with Prolog III. A glass cutting problem has been tried with Oz [Wue95].
An actual application of cutting problems is found in the Made system of Dassault [Cha94]. As part of a detailed scheduling problem, the program has to decide the cutting of complex aircraft pieces from sheet metal. Different pieces can be cut together from one sheet, imposing constraints on the scheduling part of the program. The complex shapes are approximated by a combination of rectangles. As the cutting is performed by a press, the usual constraints on cutting do not apply. The problem becomes more a placement problem.

4.4.3 Evaluation

This type of problem can be solved with constraints, but the competition from IP is very strong. If the constraints can be expressed by a set of inequalities alone, then constraints do not seem to bring advantages over the integer programming method. For more flexible situations, like in the Made application, constraints can offer a better environment.
Note that all methods described above work on rectangular shapes or shapes which are unions of rectangles. For cutting irregular pieces (clothing, leather) heuristic methods seem to work very well, while the constraint model lack the constraints even to express the problem properly.

4.5 Stand allocation

4.5.1 Problem

The problem consists in assigning parking locations to aircraft at an airport. Obviously, the same place can not be assigned to two aircraft at the same time. Other constraints impose limitations on possible parking stands according to aircraft type, airline or customs status. Preferences express constraints like ease of boarding or simplification of transfers. Due to delays and changes, an existing plan must be amended (re-scheduled) on a constant basis.
The same type of problem is encountered for platforms and trains at train stations, although the need to reschedule is often less pronounced.

4.5.2 Studies

The first application in this domain was the HIT system developed by ICL[Per91], which assigns berthing locations at a container harbour in Hong Kong. Additional constraints in this case are the availability of cranes and other resources, as well as manoeuvring space required for the ships.
The APACHE demonstrator [Din91] was used for solving the stand allocation problem for terminal 2 of Paris Roissy airport. Numerous additional constraints could be expressed, like the towing of aircraft between parking stands in order to free up space next to the terminal.
Similar systems have been developed by Siemens and Ilog for platform assignment at train stations.

4.5.3 Evaluation

The basic constraint underlying the stand allocation problem is a set of disequality constraints in a special form. Seen as a Gantt chart, we note that the stand allocation problem is a interval graph colouring problem. This problem can be solved in polynomial time. Given additional constraints, the problem typically becomes more difficult. But very good propagation results can be derived either from a diffn constraint expressing the non-overlapping condition, or from a set of alldifferent constraints (one per time interval).

If we take the airport example, it is also clear that the problem must be resolved whenever a delay or change of flight plan changes the constraint set. Fortunately, we can resolve even large instances in a few seconds. A typical requirement is the need not to change a solution too much when rescheduling, even if the cost is higher than for a completely new solution.

A proof of optimality for the solution can usually not be obtained by enumeration. The problem contains many symmetries (similar stands or aircraft) which are very difficult to control. We may require a different, cumulative constraint model to obtain lower bounds for the solution.

In general, this type of problem is well suited for constraints. But other techniques are also applicable, which makes it more difficult to justify the choice of constraints in this case.

4.6 Air traffic control

4.6.1 Problem

The general air traffic control (ATC) problem is quite difficult to express. Each aircraft is described by a four dimensional trajectory in time and space. The requirement is that no two trajectories are ever “close” to each other. At any given moment, a box of certain width, height and length is drawn around the aircraft. This box should not contain another aircraft. Since aircraft speed is a variable, this is an inherently non-linear problem. Constraint programming can be applied to special cases of this general problem.

4.6.2 Studies

Thomson [Fag95] has used finite domain constraints to express ATC problems for landing approach. In this case, the constraints are simplified to a sequencing problem, with constraints on distances and variations in airspeed and aircraft size. Incremental rescheduling becomes an important aspect of solving this problem.

A study by CENA [Che95] describes a capacity planning problem for ATC. Rather than following each trajectory, the program expresses hourly capacity limits on different control sectors. This study was done in competition with an IP approach, where both methods obtained quite good results.

Matra-Cap [COS95] describes an air traffic control problem for military mission planning.

4.6.3 Evaluation

The general ATC problem is clearly beyond the possibilities of current constraint methods. Special cases like the ones described above clearly show some promise.

4.7 Frequency allocation

4.7.1 Problem

The problem occurs in military radio networks or mobile telephone networks. In these networks, each node must be assigned one or several frequency bands for transmissions. The frequencies for neighbouring stations must be chosen in such a way that they do not disturb each other. Depending on the properties which are taken into account, this may mean that the same frequency can not be re-used, or that the frequencies must have a given distance, or that certain harmonics must not occur. In a first approximation, the problem can be seen as a graph colouring problem with stations being nodes and frequencies being the colours. The objective can be to minimise the number of frequencies or to get the best possible separation from a given set of frequencies.

4.7.2 Studies

Frequency assignment was chosen by Celar, a French military agency, for a benchmarking test of different CLP tools. The test data are available via http://web.cs.city.ac.uk/archive/constraints/constraints.html, the comp.constraints home page.

Frequency assignment in military networks was also studied by Thomson [Fag95].

An application for mobile telephone systems was presented in [Car93]. This paper gives an idea of the amount of work required to integrate a constraint program in a larger system.

4.7.3 Evaluation

Frequency allocation can be a very difficult problem. It is usually not too difficult to find even good solutions, but proving optimality becomes quite hard. On one side ,the constraint propagation does not handle the graph colouring problem very well. Using only syntactic constraints, very little global information is obtained. On the other hand, no global constraint fits the problem very well.

Another problem is the reduction of symmetry, which is essential in reducing search complexity.

It seems that more work is required to find an adequate model of this problem, using results from graph theory in the constraint propagation.

4.8 Network configuration

4.8.1 Problem

Network configuration problems come in many different forms. Many of them are related to the warehouse location [Mir90] problem of deciding which locations to use to deliver goods to an existing customer base.

4.8.2 Studies

A good example is the load balancing application developed by ICON [Chi94]. The application decides where to place mainframe applications in the Italian inter-banking network to provide sufficient services for the clients. Network topology, size of the applications and application demand control the main constraints.

The Planets system developed for Enher by the University of Catalonia in Barcelona [Cre95] is used to reconfigure electrical power networks for maintenance. In order to

isolate areas on which maintenance is performed, power is re-routed using the redundancy in the network. Customer service disruptions must be minimised, while all electrical constraints (Kirchhoff's law) must be satisfied.
A follow-on project is now under way at UCB for water networks.
The LOCARIM system for France Telecom is used to design the cabling for telephone systems in large buildings. The telephone system connects terminals in the rooms via cables to concentrators, which in turn are connected to each other or a central switch. The cabling is obviously limited by the building topology, as well as rules on crossing walls or following cable ducts. The objective is to generate an estimate of the cost of the cabling, which is then used to select suppliers. LOCARIM is operational since 1993; its constraint solver uses finite domains.
The same type of problem has also been tackled using Prolog III and rational constraints [Nar94]. This approach is much closer to the OR approach of using mixed integer programming for solving warehouse location and related problems.

4.8.3 Evaluation

For this type of problems, mixed methods using both finite domains and rational constraints seem to offer very good possibilities. A number of applications which are currently operational show that even with strong competition from mixed integer programming, CLP offers some advantages for this type of problem.

4.9 Product design

4.9.1 Problem

Product design is the process of designing or configuring products according to special customer requirements. There are not too many examples for constraint use in this category. One major example was developed by Bull.

4.9.2 Studies

A system for designing lock and key systems was built by Bull for Vachette. The task is to design the locks and keys for a complete building, where keys for different persons should open multiple sets of doors. Parts of the keys are used to encode groups of locks which are opened. The objective is to provide a customised lock and key system for each customer without spending too much time to develop it.

4.9.3 Evaluation

This area clearly has potential for constraints. However, in many cases, the constraint problems seem to be quite small and isolated, so that very simple methods suffice for solving them. The Vachette example shows that even very complex design problems can be solved, given enough time and resources.

4.10 Production step planning

4.10.1 Problem

The problem is to plan the sequence of production steps which are performed to build a complex system like an aeroplane. Multiple steps can be performed in parallel, provided they do not disturb each other and do not require the same resources. Some steps must be performed before others. The access to locations may be limited, and safety rules may prohibit the concurrent work on some steps.

4.10.2 Studies

The COCA system of Dassault performs this production step planning for manufacturing aeroplane sections. As these aeroplanes are built in series, the same plan will be reused for several planes. The construction plan requires several tens of thousands of steps, but decomposition can be used to reduce the problem complexity.

4.10.3 Evaluation

This type of problem is basically a scheduling problem with strong resource constraints. It is interesting in its own right, but also since it provides much of the input data for the detailed scheduling phase of production.

4.11 Production sequencing

4.11.1 Problem

This problem is another sub-class of scheduling problems that we want to describe independently. The objective is to sequence tasks on one or several machines in such a way that sequence dependent set-up costs or time are minimised. Sequencing for one machine can be seen as a travelling salesman (TSP) problem, finding the minimal cost tour through all tasks. For multiple machines, it corresponds to finding minimal cost cycles in directed graphs. The set-up cost or times occur because machinery has to be cleaned or adjusted, with the cost not only related to the product which is to be produced, but also to the previous product.

4.11.2 Studies

Two typical examples of this category have been developed by Beyers and Partners for Glucose manufacturers in Belgium [COS95]. The products have different grades and qualities, and changing products may lead to loss of production and down-grading of quality levels, which incur a cost.
A similar problem underlies the MOSES system, a program developed by COSYTEC for animal feed production.
Another production sequencing problem is the car sequencing problem described in [Din88b] and [Bel94].

4.11.3 Evaluation

The problem of sequencing products requires rather strong constraints for expressing set-up values. The cycle constraint in CHIP provides a good basis [Sim95a]. A much weaker version can normally also be expressed with disjunctive inequalities. The problem becomes significantly more difficult if due-dates for individual tasks must be taken into account as well. Handling set-up constraints is quite difficult in most OR based scheduling systems. The ease with which they can be included in constraint based models is a clear advantage for this technique.

4.12 Production scheduling

4.12.1 Problem

By far the most widespread use of finite domain constraints can be found for production scheduling problems. Production scheduling basically consists in finding start dates for tasks, which may require different resources during their execution. Limited resource availability constrains the possible starting times for machines.

Numerous other constraints, like manpower, producer-consumer constraints for material, or alternative machine choices can occur in these problems. Often tasks have due dates that we wish to satisfy. Limited shelf life of products may enforce other constraints on the schedule.

4.12.2 Studies

A significant number of operational scheduling applications have been developed using finite domain constraints. We only list a few here:

The ATLAS system [Sim95b] developed by Beyers and Partners and COSYTEC is a complex production scheduling system for a chemical factory. It schedules a two-phase herbicide manufacturing process. The first phase, a batch based process, produces chemicals, which are then filled into bottles of different sizes on a number of packing lines. The scheduler is completely integrated in an interactive environment. The system also performs inventory control, making sure that enough raw materials and intermediate products are available at all times.

The MOSES system of COSYTEC is a similar scheduling system developed for animal feed production. The system is fully parameterized. A model of the factory is entered in a graphical form to describe the resource structure.

The ORDO-VAP application of COSYTEC schedules production of goblets in a glass factory. The scheduling process is a multi-step production with a percentage of production loss due to the material. The objective is to balance man-power utilisation and to minimise down-time due to set-up operations.

The MADE application of Dassault [Cha94] performs detailed scheduling of a flexible work cell in the aircraft manufacturing field. The scheduling part is linked to a placement application, which arranges complex metal pieces in pieces of sheet metal for concurrent processing.

The SAVEPLAN [COS94] system of Sligos is a scheduling package integrating a CHIP based constraint engine. It is used for example in the heavy industry.

The GMP package of Beyers and Partners [COS95] is another scheduling package integrating a CHIP based constraint engine. It has been used for a number of applications in the chemical field.

The PLANE system of Dassault [Bel92a] performs medium/long term production scheduling by controlling the speed of the production lines for different aircraft types.

A special type of scheduling problem is found in satellite tasking. The problem is to schedule the operation of the satellite, say low orbit earth observation, in such a way that the limited resources aboard the satellite are used in the best possible way. An example of such a system is given in [Bel93].

4.12.3 Evaluation

Production scheduling is a very fertile field for constraint use. A full methodology for expressing different types of constraints with global constraints has been developed in [Sim95c]. A significant number of large, operational scheduling systems have been developed. It is interesting to note the integration of constraint systems into scheduling packages, which hide the constraint model from the end-users, who interact with the system in a graphical environment.

Practical scheduling problems, often mixing many different types of constraints, can be more easily expressed with constraints than with the more structured, but also rather inflexible methods of OR. But even for classical OR scheduling benchmarks, very good results have been obtained by using constraints [Pat84] [Cas94] [Bel96].

4.13 Maintenance planning

4.13.1 Problem

Maintenance planning is a special type of scheduling problem with particular constraints. A number of maintenance tasks must be scheduled close to the preferred date. Any deviation from the ideal date causes some cost. Resource constraints limit the amount of work which can be done at any one time. In some case, certain tasks may be postponed into the next period or completely cancelled.

4.13.2 Studies

A typical study of this form is shown in [Bel92]. It shows a way to express and control the additive cost function.
A number of other studies have been performed, for example scheduling rail track maintenance for the Hong Kong Public Transport Authority.

4.13.3 Evaluation

This type of problem can be rather difficult to solve, in particular if it is possible to postpone or cancel some operations. A promising approach is temporal decomposition. This consists in splitting the problem into independent sub-problems for shorter time periods. These sub-problems can be solved independently and can then be combined to produce a solution to the complete problem.

4.14 Product blending

4.14.1 Problem

The problem consists in mixing different ingredients with different prices and properties in order to make a finished product which satisfies a set of required properties at minimal cost. This is a classical linear programming problem. Additional constraints may impose integer values for certain variables. While most of the constraints will be over a continuous domain, there can be a use of finite domains.

4.14.2 Studies

A number of studies have been performed for problems of this type, for example in the cosmetics area.
Two operational systems should be mentioned:
A rubber re-mix system at Michelin [COS95] uses CHIP with a combination of finite domain and rational constraints. The system determines the optimal re-use of rubber rework of different qualities for a tyre manufacturer.
The FORWARD system [Sim94b] uses continuous constraints for optimised blending of finished products or crude oils in oil refineries. It provides explanation facilities based on meta-programming in case the original constraint system is overconstrained.

4.14.3 Evaluation

It is very hard to compete against mixed integer programming solutions in this domain. MIP provides very good solutions for this problem with a small modelling effort. The constraint model is basically a set of inequalities, all known a priori. Finite domain solvers do not perform significant propagation on such constraint sets. Constraints can be used successfully, if they are integrated with other facilities like explanation tools. In this case, the flexibility of programming with constraints gives possibilities which are not so easily achieved using standard LP packages.

4.15 Time tabling

4.15.1 Problem

The typical time tabling problem is finding an assignment of classes, teachers, rooms and/or special equipment to a time table in such a way that all required constraints are satisfied:

- Teachers can not teach two classes at the same time.
- A room may only be used by one course at a time.
- There should be no gaps in the courses of a class.

The problem usually occurs at schools and universities, but is also found for industrial training centres or institutions. An important requirement is the need to balance the time table, so that nobody has a much better (or worse) schedule than everybody else. Preferences for certain resources and/or times also often lead to conflicts.

4.15.2 Studies

A large number of experimental systems have been presented to perform school or university time tabling, see for example [Fra95][Rou95][Yeu95][COS95].
An interesting variant is described in [Bou94], which is used to schedule exams at a university. The major difference to the usual time tabling problem lies in the fact that different exams can be placed in the same room, if the room is large enough to hold all students.

4.15.3 Evaluation

Most of these systems are experimental, or purpose-built for a particular environment. A common weakness is the lack of stability. Small changes in the input data may make a previously easy problem very hard to solve. The constraint models are currently rather weak. They can not detect inconsistency at an early point in time to avoid deep, useless backtracking.

4.16 Crew rotation

4.16.1 Problem

The problem of finding crew rotations is most complex for airlines, but is also found for other transport operators. The objective is to assign personnel to flights, satisfying the requirements of each flight in terms of qualifications, safety rules etc. Personnel is based at a home station, where it starts work and where it returns after work. In order to cover all flights, it may be necessary for people to stay overnight at some location or to travel as passengers to a certain location (called passive transport or “dead-heading”). The total working time is restricted by various rules: e.g. breaks and rest periods must be inserted at regular intervals. For social reasons, the schedule must be balanced, so that everybody is performing roughly the same amount of work. People have preferences, which may or may not be taken into account.

The crew scheduling problem is typically handled in two phases, a longer term planning (often four weeks) and an operational re-planning stage. In re-planning, the existing plan must be corrected to handle changes, delays, illness or other unavailability, without disrupting the whole existing schedule.

4.16.2 Studies

The Pilot system [Bau94] [COS95] handles the day-to-day re-planning for short and medium haul flights of SAS. It tries to cover “open flights”, flights for which resources are missing, by re-organising the existing schedule.
The DAYSY Esprit project, a joint development by Lufthansa, Sema Group, the University of Patras and COSYTEC, builds a day to day management system for Lufthansa.
A system developed by PrologIA for Air Littoral [Gue95] re-implements a Simplex based set covering system with a special 0/1 Simplex constraint solver.
Similar crew rotation systems have been developed for railway operators as well.
Another Esprit project, SuperBus, (PrologIA, Brunel Univ.) [Vet96] aims at crew scheduling for public transport systems.

4.16.3 Evaluation

The current tools for handling long term planning are OR based, combining a generate&test procedure with a large scale set partitioning solver. The size of the problem handled and the structure of the constraints make it unlikely that constraints will become competitive in that domain (at least for large scale operators). The day-to-day planning and rescheduling are much more likely candidates. Constraints are very complex, and may change from one year to the next. They are quite difficult to express. The sequence constraint in CHIP [Bel96a] has been designed with this type of problem in mind.
The problems of balancing the schedule and of taking preferences into account, where possible, require sophisticated heuristics and strategies.

4.17 Aircraft/train rotation

4.17.1 Problem

Similar to the crew assignment, this problem also requires the assignment of aircraft or trains to cover all specified services. The important concept here is location continuity, so that an aircraft landing at an airport must pick up another service departing from the airport. Maintenance requirements impose regular stops at certain locations. Often, there are constraints on a balanced usage of the material. Passive movement, i.e. flying an aircraft empty from one airport to another, must be avoided if at all possible.

4.17.2 Studies

There have been two projects for SNCF, both of which looked at train rotations throughout France. One project was done by Bull to look at capacity planning of the different train depots. The objective was to decide how many trains to move between the different depots during the night, in order to have all required material available in the right place at the next morning. A prototype was built using constraints, but later replaced by a specialised max-flow algorithm to solve the problem.

A similar situation arose for a train engine rotation system developed by Ilog. The initial constraint prototype was later on replaced by a hard-coded travelling salesman type algorithm.

4.17.3 Evaluation

The constraints of this type of problem are quite similar to the ones encountered for crew rotation, although the most complex work rules do not appear. Even though the aircraft rotation problem is simpler than the crew scheduling problem, the projects above were not successful as constraint applications. Both were attempted without high-level global constraints. The limited propagation performed by syntactic constraints was not sufficient to solve large scale problems, although they showed promise for small example problems. We will return to this effect in the main evaluation section 5.

4.18 Transport

4.18.1 Problem

The problem is to schedule the transport of goods between known locations by a set of lorries with fixed capacity. Lorries typically start at a central depot, to which they return at the end of a tour. A tour may consist of one or several deliveries. Drivers have restrictions on the number of working hours, on the maximal time between breaks, or on the total mileage travelled in a day. There may be restrictions, called time windows, when goods may be picked up or delivered to the customer.

4.18.2 Studies

The EVA application [COS94], developed by GIST for EDF, the French electricity utility, schedules the transport of nuclear waste between the power stations and the re-processing site in France. The program is used to optimise the use of the special transport containers and to minimise reactor downtimes due to transport delays.

EBI in Turkey has developed a program to schedule lorry transport [COS95] between warehouses and customers, which combines lorry capacity planning and route optimisation.

The TACT system [Sim95], developed by COSYTEC, solves a rather complex transport problem. A number of interacting solvers perform capacity planning, scheduling and assignment of lorries, drivers and other resources for a set of transport orders between farms and a number of food industry factories. Additional constraints are imposed by the operations on farms and at the factories, so that the resulting problem is very difficult to solve by hand. The TACT application has completely replaced the manual scheduling process, and has been in daily use since February 95.

The PASZA project of COSYTEC handles lorry transport of animal feed from feed mills to farms, combining trips and optimising the use of different lorry compartment configurations.

4.18.3 Evaluation

Most constraints of transport applications can be easily expressed with the cycle and other global constraints. Most applications have a important scheduling component, and may be further complicated by work rules that must be respected by personnel. Producer/consumer constraints [Sim95b] play an important role in “just in time” (JIT)

schedules, where deliveries are made as late as possible in order to minimise stock levels.
The general transport problem of satisfying an order book of transport requests between any two different locations, while loading multiple orders on one lorry up to the lorry capacity, is quite complex. Local search methods or agent based systems seem to perform quite well in that environment.

4.19 Personnel assignment

4.19.1 Problem

Personnel assignment problems are related to resource scheduling problems. The objective is to assign personnel to tasks, which are usually fixed in time. Not all persons can handle all types of tasks, and each person has a limited work period during which tasks can be assigned. Rest periods, days off and holidays must be respected. Overtime or other exceptions will incur extra cost.

4.19.2 Studies

A rather large personnel assignment system was build for Servair by GSI, ITMI and COSYTEC. It handles the assignment of bar and restaurant personnel to a part of the TGV high speed train system in France. In a first step, a crew pairing module finds tours for the service. These tours are one or two day journeys starting and ending at the home location. In a second personnel assignment phase, actual employees are assigned to the tours for a four week planning horizon.
The OPTISERVICE system developed by GIST and COSYTEC for the French TV/radio station RFO is used to assign technicians and journalists to the required transmissions. It is used to minimise over-time work and to balance the work load among the personnel.
A system for the Banque Bruxelles Lambert [BBL95] is used to assign bank personnel to shifts. The system is implemented using Ilog solver.
Another personnel assignment system was developed at Grenoble [Heu96] for nurse scheduling, again using Ilog solver.
Finally, we should mention a resource assignment system developed by Bull [Cra94] for the Olympic winter games in 1992.

4.19.3 Evaluation

The problem is quite similar to the crew rostering problems described above. Work rules, union agreements and government regulations impose many complex constraints, which can be quite hard to express in a constraint solver. The assignment problem is somewhat easier if all activities are at the same location. In that case, the location continuity does not impose additional constraints.
A rather difficult property is the requirement for balancing workload, which can not easily be expressed inside the constraint model. It is best handled as a strategy in the search procedure together with some post-processing.

4.20 Personnel requirement planning

4.20.1 Problem

Personnel requirement planning is a slightly easier problem which tries to find the required number of personnel to handle a set of tasks. The tasks and their personnel demands are given. A choice exists as to when the personnel should start work.

4.20.2 Studies

A good example of personnel requirement planning was given in [Arn94]. The service demand for an emergency response centre over a period of 24 hours must be satisfied by having different numbers of personnel start at different times of the day. The solution was built using MIP techniques in the 2LP system.
An operational planning tool for personnel requirement planning was developed by Sligos for the staffing of a French credit card service centre.
A similar system for determining the demand of ground crews at an airport is under development for Havas in Turkey.

4.20.3 Evaluation

This is another area where the traditional IP approach is quite strong. The problem can be easily expressed as a set of linear inequalities. With finite domain constraints, it is possible to use the cumulative constraint to express the resource requirements. Due to the typically small problem size, different methods will give satisfactory solutions.

5. Evaluation

We now try to extract some general rules from the overview on the last section. We will first discuss failures of constraint programming to solve some problems. We then give a check list of constraints or features which often cause trouble in constraint solutions. In the last part, we then focus on four application areas, where we think CLP has shown its usefulness and where it is among the best methods currently known to solve these problems.

5.1 Failure

In the literature, there is not much information on failed attempts to apply constraint logic programming. In part, this is due to the normal scientific procedure to only publish successful results. The presentation of J.Y. Cras at ILPS 94 is the only one to discuss a number of failures and try to explain what went wrong. We will return to some of the problems mentioned in this paper later on.
In general, we must distinguish between two types of failures. An application development project may fail, even though the constraint solver inside works perfectly. We describe this type of problem under project failure.
Interestingly, a project may still succeed, even if the constraint solver inside does not work. An alternative way of solving the problem can be substituted, for example a heuristic method or an IP program and the end-user may be very satisfied with the result. But if the constraint model does not work, we should class this under the heading of problem failure. We will try to explain various reasons for such failures.

5.1.1 Project failure

IT projects fail for a variety of reasons, most related to project management. The result can be that no working system is delivered, or that time and/or cost limits are not respected.

As this paper is not focused on project management topics, we will just mention a few problems which may be encountered.

Project management can fail by defining a project which is technically too ambitious, i.e. which tries to solve too many things at once. This may happen if the limits of constraint technology are not considered carefully during the definition of the project.

Another common problem is the under-estimation of costs due to commercial or management pressure; due dates may not be achievable. Often, the development of the problem solver is not the critical issue here, but rather the development of interfaces and links to other applications.

A last, but very important reason for project failure is missing end-user acceptance. The best tool can not succeed if the users are not willing to use it. Fortunately, constraint programming with its rapid development cycle is ideally suited to early end-user involvement and iterative refinement of the program to suit the user's needs.

5.1.2 Problem failure

Constraint models may fail for a variety of reasons. Some of these failures are easily corrected; some may prove fatal to a project. We now present four types of failures, and some remedies against them.

5.1.2.1 Wrong problem

We may be solving the *wrong problem*. In some cases, this is literally true. Due to misunderstandings or a lack of analysis, a constraint model may not express the needs of the user. Some important constraints are not respected or not formalised correctly.

There are other problems which normally should not be attacked using constraints. If *specialised polynomial algorithms* exist to solve a pure problem, then it is often pointless to use a (weaker) constraint model. The main advantage of constraints in this case is the ease of adding new, unforeseen constraints.

Another difficult situation may occur if *constraint relaxation* is essential to solving the problem. If no constraint of the problem is hard, then the constraint solver has no initial propagation to guide the search. For many problems, ad-hoc constraint relaxation schemes have been successfully used. But coming up with a general scheme is still very much a research topic.

Constraint models can also fail because the model tries to solve a problem which is *too generic*. Constraint propagation very much depends on using particular properties and problem specific knowledge.

5.1.2.2 Wrong solver

Another reason for failure is use of the wrong constraint solver. We distinguish two problems here:

On one side, we may be using the *wrong type* of solver. Using a continuous domain where discrete solutions are expected, is a typical problem, or we may be using a complete solver for a problem where we are only interested in one particular solution.

Even if the problem is well-suited to the finite domain approach, we may be using the *wrong tool*. If the constraint solver only provides simple, syntactic constraints, we should not expect to be able to solve very hard problem instances. The global constraints were introduced for that reason.

5.1.2.3 Wrong model

Often, a constraint program will not work because the model used is not well chosen. A typical problem is the bad choice of decision variables, for example using many 0/1 domain variables rather than a few variables with a bigger domain. Another problem occurs when user-written constraints express the right declarative semantics, but do not propagate enough. In this case the solver may make wrong choices in the search tree without recognising the failure in time. The same problem occurs when the search strategy is not selected correctly. It is important to use problem specific information to guide the constraint solver. A particular case is the use of a weak cost model, which becomes important when a significant amount of search is performed or when the optimal solution is required.

5.1.2.4 Wrong test case

The last source of problems is the selection of test data for the problem solver. It is important that the solver can handle full-size problems. As finite domain problems are normally combinatorial in nature, it is not enough to test only small sample test sets.
It is also worthwhile to see whether the current solution, obtained without constraints, satisfies the constraint model. Very often failure to do this points out problems in the constraint model. This is true in particular, if conflicts in the preferences require the relaxation of certain constraints.
For practical problems, it is crucial to test the program with a large variety of test data, covering variations for example due to seasonal changes in demand, peak and non-peak business periods or special cases like holidays. Full testing early in the development will avoid expensive errors discovered only when the application is in daily use.
Obviously, this requires that test data be collected over a period of time, a non-negligible additional task in the project.

5.2 Modelling checklist

We now want to discuss some issues which often cause problems in finite domain constraint programs. In the current generation of tools, only limited support exists to handle these problems in a systematic way. For many, there is active research going on to find general solutions which can be applied in an application independent way [Wal95].

5.2.1 Soft constraints

In many problems we encounter soft constraints or preferences. These are constraints that we would like to satisfy in a solution, but where we know that it is impossible to satisfy all such soft constraints. These soft constraints cause several problems.

- As not all constraints can be satisfied, we have to search for a combination of constraints which are relaxed and not taken into account in the solution. This adds another dimension to the original search problem. Not only do we have to search among variable assignments which satisfy the constraints, we also have to search among the constraints to find a relaxed subset.

- Not all soft constraints have the same weight in the problem. Some preferences are more important than others. This leads to the introduction of constraint weights or hierarchies. Finding a good solution requires an optimisation procedure to find the minimum weight relaxation.
- If all constraints in the problem are considered as soft, then there is no initial propagation. This means that the initial search steps will be performed blindly, losing the advantages of the constraint approach.

In many of the application examples shown above, soft constraints occur. For most, the solution consists in handling soft constraints as part of the heuristics, and to avoid searching over alternative constraint relaxation at all.

5.2.2 Over-constrained problems

A related problem is the case of over-constrained problems [Jam96]. Taking all constraints which are initially specified into account, there is no possible solution at all. The current solution, not obtained by the constraint program, also does not satisfy the constraints. In this case, the constraint system usually returns the answer ‘No’. This will not satisfy the end-user. The application must therefore treat some constraints as soft, and provide explanations why certain combinations of constraints do not allow solutions.
In some applications, like the TACT system, there exist specific constraint modules only used to detect and explain such inconsistent problems [Sim94a]. In others, like the Forward system [Sim94b], inconsistency is handled via meta-programming on top of the main constraint program.

5.2.3 Balancing

A common requirement in many personnel related problems is the need to balance the resource use over all resources. Expressing this condition as a constraint rarely leads to satisfactory solutions. Stating a lower bound constraint on a required minimal resource use will often lead to failure, as the constraint is not handled in an active manner. It is often better to impose only upper bound constraints and to apply some post-processing to balance the work-load in the end.

5.2.4 Non-local cost

For many optimisation problems, the cost function will be expressed as a weighted sum of different cost factors. This type of cost does not allow useful pruning in the search tree and can lead to excessive backtracking.
Even if the original cost is expressed as such a non-local cost, it is often better to impose a cost constraint on individual cost factors. [Bel92] shows an example where instead of minimising overall delay of a set of tasks, it is much better to impose constraints on the maximal delay of each task separately.

5.2.5 Planning problems

The constraint approach requires that variables and constraints are defined before the search. If, for example, we represent tasks in a scheduling problem by domain variables for their start, this means that we have to know which tasks we have to schedule from the beginning. In planning problems we often do not know how many tasks we need. We can overcome this problem by introducing a number of dummy tasks which are only used when required, but this often leads to awkward models with

difficult conditional constraints. It may be better to change the model completely and to ignore a task structure.
Many capacity based planning problems are best expressed as sets of inequalities and solved with integer or mixed-integer programming.

5.2.6 Passive transport

For transport problems, we often face the problem of passive transport. Besides the transport orders that we know, it can be necessary to move resources from one place to another in order to pick-up work at the new location. This may mean that a lorry is driving empty or that a pilot is transported as a passenger from one airport to another. The total amount of this passive transport must be minimised for cost reasons, yet it may not be possible to avoid it completely. Similar to the planning problem, we face the difficulty of not knowing which and how many variables we need in the model. In some situations it is possible to include passive transport into the constraint model without introducing new variables [Sim94].

5.3 What really works

We now discuss four application fields again, for which constraint programming with finite domains offers the appropriate tool. Another important criterion is the availability of a methodology to express and solve such problems.

5.3.1 Scheduling

Scheduling is probably the most successful application domain for finite domain constraints. A methodology for expressing and solving a variety of scheduling problems using global constraints is available [Sim95c]. For a number of benchmark problems, constraint programming provides competitive results [Agg93] [Bel94], [Bel96] [Cas94]. Different strategies, both from the OR and the AI field, can be easily expressed as search procedures in CLP. It is interesting to note that many advanced methods for solving scheduling problems in OR use domain-like structures, see [Pin88] [Pat84] [Got93], and are propagation based.
It should be noted that end-user scheduling projects can still be quite complex. This is mainly due to the required interfaces and the necessary integration, but also to the need to evaluate strategies and to test different data sets.

5.3.2 Allocation

Allocation problems like aircraft stand allocation [Din91] are typically not as difficult as scheduling problems as the time dimension is usually fixed. The problem is mainly expressed by disequality constraints or more complex aggregate constraints (alldifferent, diffn) [Bel94]. For these constraints the propagation results are very good and some complete methods for sub-problems exist. We also find that re-solving the problem can be done very rapidly, within a few seconds.
But it should be recognised that allocation problems can also be solved quite well using other techniques, so that competition is much stronger.

5.3.3 Transport

Transport applications can be expressed and solved with finite domains if the global constraints like diffn and cycle are available. These constraints handle the important location continuity property [Bel94] [Sim94] [Sim95]. The problem size is limited to a few hundred nodes (cities, orders, etc.), so that the approach is more suitable to difficult, but medium size problems. A clear advantage of constraints in this domain is their flexibility to handle additional, problem-specific constraints without revising the overall program.

5.3.4 Crew rotation

Probably the most complex application domain solved with finite domain constraints is the crew rotation problem for airlines and other transport services. Due to the very complex structure of the constraints for personnel, it becomes a challenge to express the constraints in a constraint system at all. The sequence constraint in CHIP [Bel96a] has been particularly developed for this purpose. A number of applications show that constraint programming can be a useful alternative to integer programming for this problem, in particular for day-to-day operational management.

6. Summary

In this survey we have presented a classification scheme for applications using the finite domain solver in constraint logic programming. Based on an overview of application studies and end-user applications, we have grouped them into categories with similar properties. For some of these categories, constraint programming is shown to be an appropriate choice, for others CLP does not compete with the best alternative techniques. We have presented some reasons why areas like scheduling or allocation fit well into the finite domain framework, and why transport and crew assignment applications also are likely candidates. We have presented some typical causes for failures of constraint based projects and shown a list of features which cause trouble in constraint applications.
This overview should be helpful for identifying new problem areas susceptible to the constraint approach or deciding whether a particular problem should be attacked with finite domain constraint programming.

7. Acknowledgement

This study relies on input from many persons in the constraint community and a lot of discussions with members of the COSYTEC team. Special thanks go to Michael Jampel and Jennifer Burg for detailed language corrections on the draft version of the paper.

8. Bibliography

[Agg93] A. Aggoun, N. Beldiceanu
Extending CHIP in Order to Solve Complex Scheduling Problems
Journal of Mathematical and Computer Modelling, Vol. 17, No. 7, pages 57-73
Pergamon Press, 1993

[Arn94] D. Arnow, K. McAloon, C. Tretkoff
Parallel Integer Goal Programming
ILPS Post Conf.. Workshop on Constraint Languages/Systems and their Use in Problems, Ithaca, NY, Nov 1994 (ECRC Technical Report 94-38)

[Bap94] P. Baptiste, B. Legeard, M. Manier, C. Vernier
A Scheduling Problem Optimisation Solved with Constraint Logic Programming
2nd Conf Practical Applications of Prolog, London, April 1994

[Bau89] G. Baues
Einsatz der Constraint-Logic-Programming Sprache CHIP bei der Loesung eines Job-Shop Scheduling Problems

Diplomarbeit, TU Muenchen, Mai 1989

[Bau94] G. Baues, P. Kay, P. Charlier
Constraint Based Resource Allocation for Airline Crew Management
ATTIS 94, Paris, April 1994

[BBL95]
The time tabling problem at the BBL
PACT 95, Paris, April, 1995

[Bel92] N. Beldiceanu, H. Simonis
Aircraft Maintenance Scheduling
COSYTEC Technical Report, Nov 1992

[Bel92a] J. Bellone, A. Chamard, C. Pradelles
PLANE -An Evolutive Planning System for Aircraft Production.
First International Conference on the Practical Application of Prolog. 1-3 April 1992, London.

[Bel93] J. Bellone, J.M. Pieplu
Resource Management with Constraint Logic Programming for Satellite Constellations
4th Workshop on AI and KBS for Space
ESTEC, Noordwijk, May 1993

[Bel94] N. Beldiceanu, E. Contejean
Introducing Global Constraints in CHIP
Journal of Mathematical and Computer Modelling, Vol 20, No 12, pp 97-123, 1994

[Bel96] N. Beldiceanu, E. Bourreau, D. Rivreau, H. Simonis
Solving Resource-Constrained Project Scheduling Problems with CHIP
Fifth International Workshop on Project Management and Scheduling, Poznan, Poland, April 1996

[Bel96a] N. Beldiceanu, E. Contejean
Introducing Constrained Sequences in CHIP
COSYTEC Technical Report, January 1996

[Bou94] P. Bouzimault, Y. Delon, L. Peridy
Planning Exams Using Constraint Logic Programming
2nd Conf Practical Applications of Prolog, London, April 1994

[But87] W. Büttner, H. Simonis
Embedding Boolean Expressions into Logic Programming
Journal of Symbolic Computation, 4:191-205, October 1987

[Car93] M. Carlsson, M. Grindal
Automatic Frequency Assignment for Cellular Telephones Using Constraint Satisfaction Techniques
Proc 10th Int Conf Logic Programming, Budapest, Hungary, pp 647-665, 1993

[Cas94] Y. Caseau, F. Laburthe
Improved CLP Scheduling with Task Intervals
Proc 11th ICLP 1994,
Italy, June 1994. MIT Press

[Cha94] A. Chamard, F. Deces, A. Fischler
A Workshop Scheduler System written in CHIP
2nd Conf Practical Applications of Prolog, London, April 1994

[Che95] D. Chelma, D. Diaz, P. Kerlirzin, S. Manchon
Using CLP(FD) to Support Air Traffic Flow Management
PAP 95, Paris, April 1995

[Chi94] C. Chiopris, M. Fabris
Optimal Management of a Large Computer Network with CHIP
2nd Conf Practical Applications of Prolog, London, April 1994

[Com95] H. Comon, M. Dincbas, J.P. Jouannaud, C. Kirchner
A Methodological View of Constraint Solving
LRI Technical Report, 1995

[COS94] COSYTEC
CHIP User's Group Meeting
November 1994

[COS95] COSYTEC
CHIP User's Group Meeting
November 1995

[Cra94] J-Y. Cras
A Review of Industrial Constraint Solving Tools
AI Perspectives Report
Oxford, UK, 1994

[Cra94a] J.-Y. Cras
Using Constraint Logic Programming: A few short tales
Proc ILPS 94, Ithaca, NY, November 1994

[Cre95] T. Creemers, L. R. Giralt, J. Riera, C. Ferrarons, J. Rocca, X. Corbella
Constrained-Based Maintenance Scheduling on an Electric Power-Distribution Network
PAP95, Paris, April 1995

[Din88] M. Dincbas, P. Van Hentenryck, H. Simonis, A. Aggoun, T. Graf and F. Berthier.
The Constraint Logic Programming Language CHIP.
In Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'88), pages 693-702, Tokyo, 1988.

[Din88a] M. Dincbas, P. Van Hentenryck, H. Simonis, A. Aggoun and T. Graf.
Applications of CHIP to industrial and engineering problems.
In First International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Tullahoma, Tennessee, USA, June 1988.

[Din88b] M. Dincbas, H. Simonis, P. Van Hentenryck.
Solving the Car Sequencing Problem in Constraint Logic Programming.
In European Conference on Artificial Intelligence (ECAI-88),
Munich, W. Germany, August 1988.

[Din91] M. Dincbas, H. Simonis
APACHE - A Constraint Based, Automated Stand Allocation System
Proc. of Advanced Software Technology in Air Transport (ASTAIR'91)
Royal Aeronautical Society, London, UK, 23-24 October 1991, pages 267-282

[Din92] M. Dincbas, H. Simonis, P. Van Hentenryck
Solving a Cutting-Stock Problem with the Constraint Logic Programming
Language CHIP
Journal of Mathematical and Computer Modelling, Vol. 16, No. 1, pp. 95-105,
Pergamon Press, 1992

[Duc96] D. Duchier, S. le Huitouze
Channel Routing with CLP(FD)
PACT96, London, April 1996

[Fag95] F. Fages, J. Fowler, T. Sola
A Reactive Constraint Logic Programming Scheme
12th ICLP, Tokyo, Japan, June 1995

[Fil91] T. Filkorn, R. Schmid, E. Tiden, P. Warkentin
Experiences from a Large Industrial Circuit Design Application
ILPS, San Diego, Ca., October 1991

[Fra95] H. Frangouli, V. Harmandas, S. Stamatopoulos
UTSE: Construction of Optimum Timetables for University Courses - A CLP Approach
PAP 95, Paris, April 1995

[Fru92] T. Fruewirth, A. Herold, V. Kuchenhoff, T. Le Provost, P. Lim, M. Wallace
Constraint Logic Programming - An Informal Introduction
In Logic Programming in Action LNCS 636, 3-35, 1992

[Got93] Gotha
Les Problemes d'Ordonnancement
Operations Research vol27, 1, 1993, pages 77-150

[Gra89] T. Graf, P. Van Hentenryck, C. Pradelles, L. Zimmer
Simulation of Hybrid Circuits in Constraint Logic Programming
IJCAI, Detroit, August 1989

[Gue95] N. Guerinik, N. Van Caneghem
Solving Crew Scheduling Problems by Constraint Programming
Proc CP 95, Cassis, France, Sept 1995

[Heu96] K. Heus, G. Weil
Constraint Programming: A Nurse Scheduling Application
PACT96, London, April 1996

[ILO95] Ilog
Ilog User's Group Meeting
Paris, 1995

[Jaf94] J. Jaffar M. Maher
Constraint Logic Programming: A Survey
Journal of Logic Programming, 19/20:503-581, 1994

[Jam96] M. Jampel, E. Freuder, M. Maher (Eds)
Over-constrained Systems
Springer, LNCS 1106, August 1996

[Kay95] P. Kay, H. Simonis
Building Industrial CHIP Applications from Reusable Software Components
PAP95, Paris, April 1995

[Mir90] P.B. Mirchandani, R.L. Francis
Dircrete Location Theory
Wiley Interscience Series in Discrete Mathematics and Optimization
New York, 1990

[Nar94] G. Narboni
Optimisation of Concentrator Positioning in the Design of Private Data Networks
PAP 94, London, April 1994

[Pat84] J. Patterson.
A Comparison of exact procedures for solving multiple constrained resource project scheduling problem.
Management Science 30(7), pp 854-867, 1984.

[Per91] M. Perrett
Using Constraint Logic Programming Techniques in Container Port Planning
ICL Technical Journal, May, 1991, pp 537-545

[Pin88] E. Pinson
Le Probleme de Job Shop
These de Doctorat de Univ Paris VI, 1988

[Rou95] G. Roux, J.L. Bouquard, P.Y. Partant
Solving School Timetabling Problems with CHIP

PAP 95, Paris, April 1995

[Sim92] H. Simonis
Constraint Logic Programming as a Digital Circuit Design Tool
COSYTEC Technical Report, 1992

[Sim93] H. Simonis, M. Dincbas
Propositional Calculus Problems in CHIP
In A. Colmerauer and F. Benhamou, Editors,
Constraint Logic Programming - Selected Research, pages 269-285
MIT Press, 1993

[Sim94] H. Simonis
The Use of Exclusion Constraints to Handle Location Continuity Conditions
COSYTEC Technical Report, November 1994

[Sim94a] H. Simonis
Calculating Lower Bounds on a Resource Scheduling Problem
COSYTEC Technical Report, March 1995

[Sim94b] H. Simonis
Blend Optimisation in the Forward System
COSYTEC Technical Report, March 1995

[Sim95] H. Simonis
Application Development with the CHIP System
Proc Contessa Workshop, Friedrichshafen, Germany, September 1995, Springer LNCS

[Sim95a] H. Simonis
Modelling Machine Set-up Time with Exclusion Constraints
COSYTEC Technical Report, March 1995

[Sim95b] H. Simonis, T. Cornelissens
Modelling Producer/Consumer Constraints
Proc. Principles and Practice of Constraint Programming, Cassis, France, September 1995

[Sim95c] H. Simonis
Tutorial Scheduling and Planning
PAP95, Paris, April 1995

[VH89] P. Van Hentenryck.
Constraint Satisfaction in Logic Programming.
MIT Press, Boston, Ma, 1989.

[VH92] P. Van Hentenryck, H. Simonis, M. Dincbas
Constraint Satisfaction using Constraint Logic Programming
Journal of Artificial Intelligence, Vol.58, No.1-3, pp.113-161, USA, 1992

[VH94] P. Van Hentenryck.
Scheduling and Packing in the Constraint Language cc(FD)
in M. Zwieben, M. Fox(Eds): Intelligent Scheduling
Morgan Kaufmann, San Francisco, 1994

[Vet96] E. Vetillard
Applications de la programmation logique avec contraintes aux problemes de transport
JFPLC96, Clermont-Ferrand, 5-7 June, 1996, Hermes

[Wal95] M. Wallace
Survey: Practical Applications of Constraint Programming
Technical Report, IC Parc, September 1995

[Wue95] J. Wuertz
Personal Communication

[Yeu95] C.M. Yeung, S.M. Leung, H.F. Jeung
Applying Constraint Satisfaction Technique in University Timetable Scheduling
PAP 95, Paris, April 1995

[Zho95] N.F. Zhou
A Logic Programming Approach to Channel Routing
12th Int Conf on Logic Programming (ICLP), Tokyo, June 1995