A Review of Commercially Available Constraint Programming Tools[†]

Tim Duncan
AI Applications Institute,
University of Edinburgh

AIAI-TR-149
October 1993
(Revised: March 1994)

Here at AIAI we are fortunate enough to have access to three major constraint programming tools: CHIP, CHARME, and ILOG SOLVER. These systems are currently attracting much interest due to their ability to represent and solve Constraint Satisfaction Problems (CSP), and thus their suitability for dealing with scheduling problems.

What is a CSP?

A CSP is defined by a set of variables, each of which have a discrete and finite set of possible values (their *domain*), and a set of constraints between these variables. The solution to a CSP is a set of variable-value assignments which satisfy the constraints.

Since all three systems support the CSP paradigm they share certain similarities. In each case there is a language for modeling the problem which provides the ability to declare domain variables and their initial domains, and to place constraints. Each of the tools provides a set of builtin constraints. In order to search for a solution the system will generate values for the variables, propagating values through the constraints in order to prune parts of the solution space where inconsistencies are discovered. The basic method is therefore a backtrack search where the constraints allow the system to *look ahead* to the consequences of decisions and spot failure earlier. The search is therefore more efficient.

[†]A shorter version of this report appears in AIRING November 1993

Thumbnail Sketches

CHIP: Based on Prolog, therefore backtracking is well integrated. Prolog terms can be used for structuring data. The '`cumulative`' constraint is particularly useful for scheduling problems. Also contains constraint solvers for rational arithmetic and boolean constraints.

CHARME: A new language, small and easy to learn. Looks superficially like C but isn't. Block structure, conditionals, and iterative loops. Backtracking is well integrated and since a default generation strategy is provided, the user need not be aware of it. Data structures limited to variables and arrays.

ILOG SOLVER: An object-oriented library in `C++` and `Le-Lisp` versions. Many options and features available and therefore less easy to learn. Backtracking is provided by combining non-deterministic elements. Allows constraints to be inhibited and domains to be reset. Also contains domain variables over sets and an interval arithmetic constraint solver.

Searching for a Solution

There are two basic decisions within search, first selecting which variable to deal with, and secondly choosing a value from the variable's domain to try. These are referred to as the *variable-ordering* and *value-ordering* strategies. All of the systems provide a number of functions to support these.

Variable-Ordering Strategies: The functions provided embody several well known heuristics such as the *first-fail* principle, which selects the variable with the smallest number of elements in its domain. The reasoning behind this is that this will be the most difficult to satisfy, so we should try to do so now rather than wait until later when it will only be harder! As a corollary, if it does fail the failure is likely to be discovered quickly, thus avoiding wasted effort. Another heuristic which all of the systems use (sometimes in combination with first-fail) is to choose the variable which is involved in the greatest number of constraints. Picking the variable with the smallest value in its domain is a good heuristic for bin-packing or layout problems and is provided by two of the systems and can be programmed in the third. Interestingly, CHARME also offers the smallest (or largest) *range* as a criteria, *i.e.* the difference between minimum and maximum values in the domain. CHIP offers '`max_regret`' (well known in Operations Research circles) as a criteria. This selects the variable with the maximum difference between the smallest element in its domain and the second smallest. Although ILOG SOLVER offers a limited number of options (domain size or number of constraints) for selecting variables, the user may define a function which the system will use, selecting the variable which minimizes

this function. This flexibility allows the user not only to combine criteria (possibly with weighting factors), but also to take into account domain specific features of the problem (*e.g.* priorities).

Value-Ordering Strategies: Choosing a value from the selected variable's domain is much simpler. All of the systems provide the option of starting at the minimum value and working up if this fails, or of working from the maximum value downwards. In addition CHIP offers the option of starting at the middle (or other nominated value) and on backtracking selecting alternatively the next higher or lower value. CHARME and ILOG SOLVER offer *domain splitting* instead (although this could easily be programmed in CHIP also). Domain splitting means that rather than selecting a particular value and instantiating the variable, the system will replace the variable's domain by the upper (or lower) half of the domain and on backtracking replace it by the other half. This is a particularly useful technique where the domain is large, since it enables a large set of values to be rejected rather than testing them all one at a time. As with variable selection, ILOG SOLVER offers the facility for the user to define the criteria.

Optimization

Optimisation is frequently a requirement in scheduling and other Operations Research problems. All of the systems described here provide functions for optimization based on the *branch-and-bound* method. The result is the solution for the specified goal which minimizes the value of an expression (the objective function). This works by searching for a solution to the problem and then adding a further constraint that any new solution must be better than the previous one. CHARME offers two variants one simply adds the constraint which initiates backtracking, while the other restarts the search with the new constraint added. Both will find the optimum, but depending on the problem characteristics one or the other variant may converge on the solution more rapidly. CHIP and ILOG SOLVER both allow the user to specify the amount by which any new solution must be better than previous solutions. This is often quicker since it avoids finding many solutions which are only marginally better. However the solution is only be guaranteed to be within the specified step of the optimum. In addition CHIP allows the user to define upper and lower bounds on the solution (any solution with a lower value will end the search), and/or a timeout where the system returns the best solution found within the time specified.

Constraints

Each of the systems provides basic constraints for equality, disequality (*i.e.* $\neq$), and the inequalities (*i.e.* $>$, $<$, $\geq$, $\leq$). Inequality constraints are often used as *precedence* (*e.g.* one task comes before another) or *distance* constraints (*e.g.* task X must come at least 2 hours after task Y). All of the systems provide an '`all_different`' constraint

which ensures that a set of domain variables all have distinct values. Although provided as a builtin this constraint is made up of a series of simpler constraints and could be programmed by the user placing not-equal constraints on each pair in the set.

Origins

CHIP was originally developed at ECRC in Munich, a European research centre jointly funded by Bull, Siemens, and ICL. The results were exploited by Bull in 1989 when they re-engineered the system and released CHARME, which they have continued to develop. The results were also been taken up by ICL in a system called DecisionPower. In 1990 the team which had designed CHIP left and set up a company called COSYTEC where they continued to develop CHIP. ILOG SOLVER however developed along a different route. This was closely tied to the development of `Le-Lisp` at INRIA (a French national research centre), and the original system (then known as PECOS) was a library for `Le-Lisp`.

CHARME offers a number of constraints on arrays of variables: '`sum(Array,Sum)`' ensures that `Sum` is equal to the sum of the variables in `Array`. If `Sum` is a domain variable CHARME will ensure that its domain is updated as information becomes available about the variables in `Array`. There is also a constraint for the product of a list. The '`distribution`' constraint offers a way to place *quotas* on the number of times particular values may occur in an array of variables. This is well suited to resource allocation problems but is also applicable in a wide range of other problems.

Constraints in ILOG SOLVER are well integrated into the object-oriented scheme. There are ways of defining constraints on a class of objects so that all instances inherit the constraint. Similarly constraints can be placed on the value of an object's slot, and objects may themselves be used as values in a domain. Both CHIP and ILOG SOLVER provide an '`element`' constraint relating a value to its index in a list of variables. This provides a way of symbolically linking two variables (since the value may be a domain variable).

CHIP offers a wide range of builtin constraints for various purposes. The '`circuit`' constraint for example ensures that the elements of a list form a Hamiltonian circuit. This is ideally suited to *Travelling Salesman* type problems. The '`cumulative`' constraint has been designed for scheduling and packing problems. The basic form specifies the start times and durations of a series of tasks, together with the amount of resource that they use, and the maximum resource available at any one time. The left hand diagram in Figure 1 illustrates an example where three tasks on the same resource must not overlap. Where a set of resources possess the same capabilities or the resource is capable of processing more than one item at a time the last parameter (maximum resource available) could be increased. However the constraint can also be applied to bin-packing or layout tasks (see right hand diagram in Figure 1).

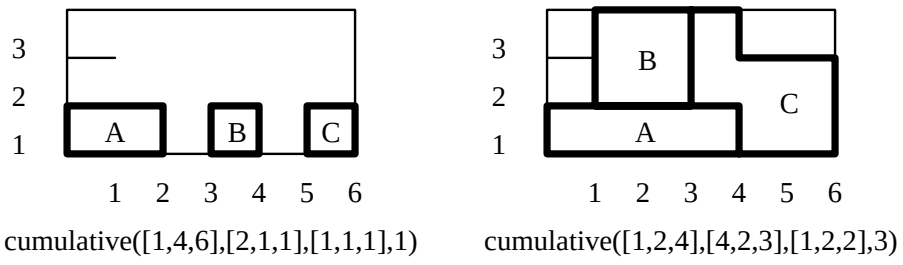

Figure 1: The '`cumulative`' constraint

CHIP also offers several more general constraints: '`distance`' allows the user to constrain the absolute difference between two domain variables; '`atmost`' and '`atleast`' constrain the number of variables in a list which have a particular value; '`minimum`' and '`maximum`' constrain a domain variable to be the minimum or maximum value from a list of variables. Naturally the system will update the domain variable as information is inferred about the variables in the list.

Each of the systems offers some facility for defining *demons*. These are essentially ways of activating a function in response to specified *propagation events*. These events are changes to the domain of a specified variable, and usually the user is given the option of reacting to changes in the minimum or maximum bounds, both, or other changes (*i.e.* when a value is removed from the domain), or when the variable becomes instantiated (*i.e.* is given a value). The user defined function is then called whenever the event occurs. The most obvious use for demons is to update graphical displays when values become known, but they can also be used to impose conditional constraints (see [Duncan & El-Said 92] for an example) or to allow the user to define their own constraints. CHIP also offers a builtin facility for defining conditional propagation of the form '`if` <*Cond*> `then` <*Pred-1*> `else` <*Pred-2*>'. This could be used for example to define an '`or`' over two constraints (*e.g.* `or(`$S2 > S1 + D1, S1 > S2 + D2$`)` *i.e.* two tasks must not overlap) such that when one is known to be false the other is enforced.

Other Features

In addition to the basic features of the CSP language the three systems each provide interfaces for graphics, relational database access, and some degree of embedability. CHARME comes with an X windows-based environment and access to X windows primitives. A separate package CHARME-PRO offers an SQL interface (allowing one to interface with ORACLE, INGRES, SYBASE and other databases), and the ability to extend CHARME with functions defined in `C` (known as "aliens"), or embed CHARME in another `C` program.

ILOG SOLVER is of course a library, so there is little difficulty in either writing `C` functions or embedding it within another `C++` application. ILOG offer a number of other packages

that may be used with ILOG SOLVER. These include ILOG VIEWS, an Motif/OpenLook/MS-Windows graphics package; ILOG BUILDER for generating OSF/Motif interfaces; ILOG DB LINK an SQL interface for relational databases; and ILOG RULES a rule based module.

COSYTEC offer packages for CHIP which provide X windows graphics (XGIP), an SQL interface (QUIC), and the ability to write external `C` functions (CLIC) or embed CHIP in a larger application (EMC). There is also an Openwindows environment for developing CHIP programs, known as Chiptool.

It may also be worth noting that all three systems are also available on 386+ PCs, although these will generally require a minimum of 16 MegaBytes memory, and benefit substantially from the faster 486 processors. CHARME offer runtime versions for UNIX or PCs.

Contacts

The facility for doing trials and evaluations of some of these products with the support of AIAI staff is available to our clients. For further details of the products discussed here please contact the distributors.

	UK DISTRIBUTOR	ELSEWHERE
CHIP	AI International The Chapel, Park View Road, Berkhamstead, Herts HP4 3EY	COSYTEC Parc Club Orsay-Université, 4 rue Jean-Rostand, 91893 Orsay Cedex France
CHARME	Bull HN Information Systems Ltd Maxted Road, Hemel Hempstead, Herts HP2 7DZ	Bull CEDIAG 68 route de Versailles, 78430 Louveciennes France
ILOG SOLVER	ILOG Ltd The Surrey Technology Centre 40 Occam Road, Guildford, Surrey, GU2 5YH	ILOG S.A. 12 avenue Raspail, BP 7, 94251 Gentilly Cedex France

References

[Duncan & El-Said 92] T. Duncan and A. El-Said. The Use of Charme to Represent Protein Structure Constraints. Technical Report AIAI-TR-125, Artificial Intelligence Applications Institute, University of Edinburgh, 1992.

Postscript

Since this report was written Bull have announced that they are no longer actively marketing CHARME as a product. They will be concentrating instead on offering complete solutions (which may optionally make use of CHARME). Under an agreement with Ilog, Bull will be able to market a version of ILOG SOLVER under the name "Bull Solver".

Ilog have released a `C++` library for representing finite-capacity scheduling problems. ILOG SCHEDULE is written in ILOG SOLVER and offers object classes for representing various types of resource and activity, together with a wide collection of scheduling constraints.

The following World Wide Web page gives access to more information about constraint programming tools and applications:

`http://www.aiai.ed.ac.uk/~timd/constraints/csptools.html`