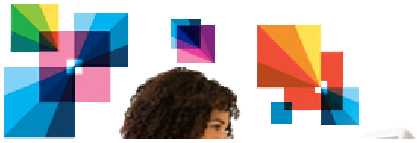
IBM Software > Products > Prescriptive Analytics > CPLEX Optimization Studio >

Constraint Programming Technology



What is constraint programming technology?

Constraint programming technology is used to find solutions to scheduling and combinatorial optimization problems. It is based primarily on computer science fundamentals, such as logic programming and graph theory, in contrast to mathematical programming, which is based on numerical linear algebra.

Constraint programming is invaluable when dealing with the complexity of many real-world sequencing and scheduling problems. Whether the problem at hand is to schedule people, machines or jobs of processes, you need constraint programming when there are complex logical and arithmetic relationships between decision variables, activities and resources.

A constraint programming model is expressed in a declarative fashion, using decision variables, constraints and objectives that must be minimized or maximized, just as in mathematical programming; Lexicographical multi-criteria objectives are possible in ILOG CPLEX CP Optimizer. IBM's OPL modeling language and integrated development environment (IDE) and ILOG Concert Technology interfaces, also part of IBM ILOG CPLEX Optimization Studio, can be used to formulate constraint programming models as well as mathematical programming models. This gives the option to combine the technologies in a multi-model framework. For example, constraint programming can be used as a heuristic to find solutions for mixed integer programs.

Constraint programming looks first to reduce the set of possible values of the decision variables which will satisfy all the constraints using logical, graph-theoretic, arithmetic and other arguments. Once the deduction that some values from the decision variable’s domain are not possible, this information is propagated through the constraints perhaps enabling further deductions. Various search strategies are also used until a value is assigned to every decision variable, that is, until a solution is found. Once a first solution is found, the search proceeds to find further solutions with better objective values.

The power of modeling in constraint programming

Modeling in constraint programming revolves around the details of what is possible. For example, if you have to schedule a large number of resources and activities that respect capacity limitations, operational sequencing requirements, and business policies while meeting individual customer service goals, these modeling features are available to you:

Optional tasks

- For modeling activities or processes that may or may not be executed in the final schedule

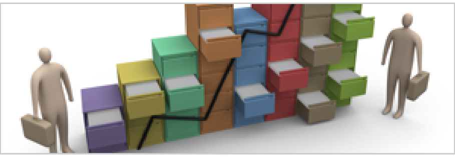
Contact IBM

Considering a purchase?

✉ Email IBM

☎ Or call us at: 1-877-426-3774
Priority code: Commerce

Optimization Forums



Technical discussion around ILOG Optimization products including CPLEX

→ Go to the forum

Related links

- → Prescriptive analytics
- → ILOG CPLEX Optimization Studio
- → ILOG CPLEX Optimization Studio Interfaces
- → IBM Academic Initiative

- Tasks can be grouped to match the work breakdown structure of a problem

Precedence constraints

- Can model dependencies between tasks
- Can include a delay
- Can be applied to group of intervals

Expression on interval properties

- Express typical scheduling costs such as tardiness costs, completion costs or total duration
- The presence status of an optional interval can be used to express completion costs or resource costs
- Can be applied to group of intervals

Finite capacity reservoir and resources

- Specify limits on the number of tasks that can be performed in parallel with a common resources
- Set constraints on inventory levels

Set-up times and batches

Calendars

- State that some tasks cannot start or end on some dates
- State that some tasks that cannot overlap some dates
- Define resource breaks
- State that resource productivity changes over time

Resource states

These modeling features can be used for combinatorial optimization problems
Arithmetic linear and non-linear constraints
Logical constraints

Specialized constraints and expressions:

- The all-different constraint: enforces uniqueness for each variable in an array
- The pack constraint: packs items into containers with finite capacity in one dimension (time, weight, budget etc.)
- The lexicographic constraint: enforces a lexical ordering between groups of decision variables and is convenient to break symmetries
- The count expression
- The standard deviation expression

Compatibility and incompatibility constraints:

- Define possible assignments for arrays of decision variables. They can be used, for instance, to model allowed transitions in a sequencing problem.

The model-and-run constraint programming optimizer

Three concepts are fundamental to the way constraint programming generates workable schedules and solutions from such complexity:

- **Awareness of time..** IBM ILOG CPLEX CP Optimizer algorithms can determine the best sequence of activities that satisfy capacity limitations, set-up times, maintenance requirements, availability restrictions, individual delivery due dates, and more by exploiting the information in the specialized scheduling and combinatorial optimization modeling constructs
- **Aggressive elimination of possibilities.** Domain reduction is the internal mechanism that makes difficult allocation, sequencing and scheduling problems manageable. Every resource or activity to be scheduled has a certain number of possible values that can be systematically reduced until a solution is reached. Every reduction makes the problem easier since the optimizer will propagate the results of the decision throughout the model
- **Rapidly traversing the decision tree.** The concept of systematically exploring a decision tree for feasible and efficient solutions relates to domain reduction. One benefit of this structured approach is the ability to move flexibly throughout the search space and to backtrack when early choices turn out to be dead ends.
- **Adaptive Search.** The default search consists of several search techniques which are dynamically changed during the search adapting to the problem at hand. The different search techniques include but are not restricted to LNS (large neighborhood search) and GA (genetic algorithms).

These concepts are built into the model-and-run optimizer of ILOG CPLEX CP Optimizer but are also available to build a customized search. The optimizer

- Solves a large range of problems with default settings
- Automatically removes redundant constraints
- Reformulates the model to use constraints that are more efficiently propagated
- Identifies conflicting constraints
- Quickly finds feasible solutions for use in multi-model architectures such as column generation

The optimizer is tested **on an extensive library of models and the search engine can be tuned via parameters or callbacks.**

Additional resources

Data Sheet: IBM ILOG CPLEX Optimization Studio

Build and deploy optimization-based models quickly and efficiently.

Read data sheet