Noname manuscript No.
(will be inserted by the editor)

Constraint Programming and Operations Research

J. N. Hooker and W.-J. van Hoeve

November 2017

Abstract We present an overview of the integration of constraint programming (CP) and operations research (OR) to solve combinatorial optimization problems. We interpret CP and OR as relying on a common primal-dual solution approach that provides the basis for integration using four main strategies. The first strategy tightly interweaves propagation from CP and relaxation from OR in a single solver. The second applies OR techniques to domain filtering in CP. The third decomposes the problem into a portion solved by CP and a portion solved by OR, using CP-based column generation or logic-based Benders decomposition. The fourth uses relaxed decision diagrams developed for CP propagation to help solve dynamic programming models in OR. The paper cites a significant fraction of the literature on CP/OR integration and concludes with future perspectives.

Keywords Constraint Programming, Operations Research, Hybrid Optimization

1 Introduction

Constraint programming (CP) and operations research (OR) have the same overall goal. They strive to capture a real-world situation in a mathematical model and solve it efficiently. Both fields use constraints to build the model, often in conjunction with an objective function to evaluate solutions. It is therefore only natural that the two fields join forces to solve problems.

J. N. Hooker
Carnegie Mellon University, Pittsburgh, USA
E-mail: jh38@andrew.cmu.edu

W.-J. van Hoeve
Carnegie Mellon University, Pittsburgh, USA
E-mail: vanhoeve@andrew.cmu.edu

Attempting to unify CP and OR might be unwise if they relied on entirely different solution methods. However, their methods are not only related, but complementary, due to the contrasting intellectual origins of the two fields. This has allowed integrated methods to outperform those that rely solely on CP or OR techniques in a wide variety of problem areas, sometimes by orders of magnitude. Furthermore, the potential benefits of integration are, arguably, only beginning to be reaped, which suggests that CP/OR integration will continue to be an active research area.

Both CP and OR use what the OR community might call a primal-dual approach, which combines search with some kind of inference. Search solves the primal problem of finding a feasible solution (one that satisfies the constraints), while inference solves the dual problem of proving that a solution is optimal, or that no solution is feasible. Search frequently takes the form of a branching mechanism, at least in the context of exact methods. The two fields diverge when it comes to inference. In OR, it typically appears as problem relaxation, strengthened by such inferred constraints as cutting planes. In CP, inference appears as constraint propagation and domain filtering. Both relaxation and propagation can help find feasible solutions as well.

Operations research is strongly influenced by its historical roots in linear programming (LP), which formulates problems using inequality constraints. Much of the field today is based on inequality-constrained mathematical programming models, including those of nonlinear programming (NLP), mixed integer/linear programming (MILP), and mixed integer/nonlinear programming. A model is almost always relaxed by reducing it to a simpler inequality-constrained model, such as an LP or a convex NLP model, which can be solved with highly developed methods that exploit its special structure. Relaxation is essential because it allows the solver to infer a bound on the optimal value, which reduces branching. The relaxation is often strengthened by valid inequalities that are inferred from the constraint set.

Operations research is, of course, broader than mathematical programming, as it encompasses dynamic programming, queuing theory, simulation, and other areas. Although we focus primarily on mathematical programming, we will see that dynamic programming, as well as network and matching theory, also play a significant role in CP.

Constraint programming has roots in logic programming, where a model has both a declarative and a procedural interpretation. A model is declarative because its statements can be read as logical propositions that describe the problem, and it is procedural because the statements can be processed as instructions for how to find a solution. Something similar to this dual interpretation survives in today's CP. The statements in a model impose constraints that describe the problem, even while they invoke algorithms, such as domain filtering, that lead to a reduction in branching.

Due to these contrasting origins, OR and CP process a model differently as they conduct a search. OR solves an inequality-constrained relaxation of the model as a single problem, while CP processes the constraints of the model individually. This allows OR to combine information from the entire model while inferring a bound, but relaxation sacrifices much of the combinatorial complexity of the problem. The CP approach captures much of the combinatorial complexity of individual constraints while inferring reduced domains, but it must resort to the propagation of domains from one constraint to the next to obtain a global view. OR partially compensates for the weakness of its relaxations by strengthening them with valid constraints that capture some of the special structure of groups of constraints. CP partially compensates for the weakness of constraint propagation by defining high-level global constraints that represent a group of simpler constraints.

At least four basic strategies for combining the complementary strengths of OR and CP have been developed in the literature. They can be summarized as follows.

Combine relaxation from OR with propagation from CP. This can be effective when some constraints "relax well" in the sense that they have a tight inequality relaxation, and others "propagate well." A relaxation is tight when its feasible set is similar to that of the original problem, or at least yields a similar optimal value. Constraints propagate well when their structure allows significant domain reduction when some variables are fixed (or their domains reduced), perhaps by branching. The so-called knapsack constraints of OR, which are linear inequalities with many nonzero coefficients, tend to relax well, because they serve as their own LP relaxation. Certain groups of constraints can also give rise to useful valid inequalities, such as the famous Gomory cuts, which are derived from constraints that are tight in the solution of the LP relaxation, or the valid cuts derived from subtours and "combs" in the solution of a relaxed traveling salesman problem. The classical "binary" constraints of CP, which contain only two variables, generally propagate well, because fixing (or reducing the domain of) one variable tends to have a significant effect on the domain of the other. High-level global constraints may also propagate well, assuming they have been analyzed and implemented in solvers. Examples include disjunctive and cumulative scheduling constraints, which have been deeply analyzed and help explain the success of CP in the scheduling domain.

Use OR methods for domain filtering in CP. Network and matching theory, as well as dynamic programming, are widely used to filter domains for a variety of global constraints. Edge-finding methods, originally developed in OR, are indispensable for domain filtering in disjunctive and cumulative scheduling problems. In addition, since achieving domain consistency for a global constraint is frequently an NP-hard problem, it can be helpful to use a more tractable OR-based relaxation of the constraint as a basis for filtering.

Decompose the problem into parts that suitable for OR and CP, respectively. This can be accomplished with two decomposition methods originally developed in the OR literature: column generation and Benders decomposition. Column generation accommodates CP by using it to generate columns in the pricing subproblem. Benders methods can accommodate CP if they are generalized to "logic-based" Benders decomposition, which allows the subproblem to be solved by CP.

Apply constraint propagation to dynamic programming models. If a problem can be given a recursive model as in dynamic programming, the state-transition graph for the model can be treated as a decision diagram, and arcs can be deleted from the diagram much as values from a domain in a conventional domain store. Normally a relaxed decision diagram is used rather than an exact one, which tends to grow exponentially. The relaxed diagram not only allows propagation that is stronger than propagation through domains, but it provides a valid bound on the optimal value that allows one to solve a dynamic programming model by branch-and-bound methods when no inequality-constrained relaxation is available.

After a brief historical overview of CP/OR integration in Section 2, we expand on the above strategies. Sections 3–5 deal with the first stratregy, the combination of OR-based relaxation with CP-based propagation. Section 3 first shows how relaxation and propagation can be tightly integrated in optimization systems. Section 4 then describes relaxations of global constraints that can be used for propagation, and Section 5 indicates how such relaxations can be defined and utilized relative to the entire CP model. The second strategy, the application of OR to filtering, is the subject of Section 6. It outlines how such OR methods as matching theory and network flows have been used to design filters for global constraints. Decomposition strategies are discussed in Section 7 (column generation) and Section 8 (logic-based Benders decomposition), both of which combine an OR-based master problem with a CP-based subproblem. The final strategy is covered in Section 9, which describes how relaxed decision diagrams that were developed in the context of CP have been used to solve problems that can be given dynamic programming models. The paper concludes with perspectives on the future of CP/OR integration.

2 Historical Overview

We begin with a brief overview of major milestones along the road to OR/CP integration. A fuller account of the early work can be found in [142].

Domain filtering and constraint propagation were foreshadowed in the OR literature as early as the 1960s, when Garfinkel and Nemhauser [107] used a technique then known as implicit enumeration to solve integer programming models of political districting and other problems. This was before relaxation methods developed in OR, and it was necessary to reduce branching in some other way. The OR literature explicitly mentioned CP as early as 1989 [49], while true integration began in the 1990s.

OR took an early step away from inequality-based modeling in 1990, when Beaumont [21] replaced integer variables with logical disjunctions and solved the problem by branching on disjunctions. After an early application to processing networks in [117], Hooker and Osorio [152] extended this approach to "mixed logical/linear programming." Meanwhile, the integration of propagation and relaxation in branch-and-cut methods was advocated by Hooker in

1994 [139] and further explored in a number of publications, such as [43, 141]. In the 2000s, it became the basis for integrated solvers like SIMPL [11, 252], which permits modeling with global constraints, and later the award-winning solver SCIP ("SCIP" is an acronym for "solving constraint integer programs") [1, 2].

The CP community also pursued integrated methods during the 1990s, primarily by exchanging information between CP and LP solvers, as advocated by Little and Darby-Dowman [172]. Papers by Wallace, Novello and Schimpf [248] and Rodošek, Wallace and Hajian [213] described an implementation of this mechanism in ECL^{i}PS^e, which was perhaps the first general-purpose solver to combine CP and OR. Régin [208, 209] applied matching and network flow theory to filtering the all-different and generalized cardinality constraints, while Baptiste, Le Pape and Nuijten [18] applied edge finding to scheduling constraints. In later work, such OR ideas as reduced-cost variable fixing, linear relaxations of global constraints, and convex hull relaxations of piecewise linear functions were brought into CP-based algorithms [97–99, 195, 206, 207]. ILOG's OPL Studio [243] and Concert technology (introduced in ILOG Solver 5.0), as well as NICTA's G12 system [230], provided modeling languages that invoke CP and MILP solvers.

While this research was underway, two schemes were introduced for combining OR and CP in a decomposition method. One uses CP-based column generation in OR's well-known branch-and-price method for integer programming [157, 254]. Another development, logic-based Benders decomposition [140, 141, 156, 153], allows the Benders subproblem to be solved by CP methods and has become a widely-used vehicle for combining CP and OR.

The 2000s have seen the adaptation of binary and multivalued decision diagrams to discrete optimization [125, 126], particularly when the problem has a dynamic programming formulation, a much-studied topic in OR. Decision diagrams provide an alternative approach to global relaxation for bounding and propagation, as well as to branching [8, 37, 38].

OR and CP researchers were first brought together in 1995 at an International Joint Workshop on AI and OR, organized by M. L. Ginsberg and J. N. Hooker at Timberline Lodge on Mt. Hood, USA. The idea of a joint conference was revived in 1999 with the annual CP-AI-OR workshop (Integration of AI and OR Techniques in CP for Combinatorial Optimization), which first met in Ferrara, Italy. It is now an annual conference series with published proceedings. Papers on hybrid methods regularly appear in CP, OR and mathematical programming conferences. A department of a major OR journal, *INFORMS Journal on Computing*, is dedicated to CP/OR integration.

3 Combining Propagation and Relaxation

One general strategy for integrating CP and OR is to combine constraint propagation with relaxation. The two techniques are mutually reinforcing, because propagation can tighten bounds on the variables in an LP relaxation, while a relaxation can prove optimality or infeasibility of a problem obtained during CP-based search.

An early application of this strategy (1995) used an LP relaxation to prove the optimality of a solution obtained by CP in five minutes for a boat-party scheduling problem that MILP could not solve in five hours [228]. Other studies [206, 252] combined CP with convex-hull and assignment relaxations (special cases of LP relaxations), as well as with reduced-cost variable fixing, a propagation technique based on LP relaxation. This last technique is actually a special case of a general propagation method based on LP dual multipliers [149]. Propagation has been used with other kinds of relaxations as well, such as Lagrangean relaxation [223, 55, 162], semidefinite programming relaxation [119, 244], and linear quasi-relaxation [46] (a quasi-relaxation excludes some feasible solutions but no optimal ones). All of these techniques achieved significantly better performance than the MILP or CP solvers of the time.

Relaxation combined with various kinds of propagation has seen a number of applications, and we provide only a sampling here. They include single-vehicle routing [218], truss structure design [46], processing network design [117, 152], resource-constrained scheduling [132, 81], multiple machine scheduling [44], shuttle transit routing [204], orthogonal Latin squares [10], and the multidimensional knapsack problem [194]. Convex hull relaxations of piecewise linear constraints have been used in a CP context to solve fixed-charge problems and transportation problems with piecewise linear costs [206], as well as production planning problems with piecewise linear costs [195, 196].

Combining relaxation and propagation can also form the basis of a general-purpose solver, along the lines developed in a series of papers in the 1990s and early 2000s [11, 43, 143, 145, 154, 195, 196]. The constraints are processed individually as in a CP solver, so as to filter domains and generate constraint-specific relaxations. A domain store for the discrete variables is maintained, along with the kind of a global LP relaxation one sees in an MILP solver. The domain store propagates the results of domain filtering for global constraints in the model. The global relaxation contains the inequality constraints in the original model along with the constraint-specific relaxations. In addition, a group of structured inequality constraints in the model, such as fixed-charge network flow constraints, can be represented with a metaconstraint (analogous to a global constraint). When processed, the metaconstraint generates cutting planes for the global relaxation that are specifically designed for the constraint's particular structure.

This general approach is implemented in SIMPL, mentioned earlier. It is partially implemented in SCIP, which allows the user to plug in "constraint handlers" that filter domains. Another solver, SCIL, is focused on MILP but borrows CP-style modeling by generating specialized cutting planes for metaconstraints [6].

Nogood constraints, which have long been used in satisfiability (SAT) solvers in the form of conflict clauses [20], can be used in an integrated solver as well. This is proposed in [141] and implemented very effectively in SCIP. When the LP relaxation is found to be infeasible at a node of the branching tree, the resulting dual multipliers are used to formulate a nogood based on the variable assignments that gave rise to the infeasibility. A similar approach that does not rely on dual multipliers or an LP relaxation is branch and check [141, 234], a variant of logic-based Benders decomposition. In fact, conflict clauses are special cases of Benders cuts [150,151], and the SAT solvers that use them can be interpreted as specialized implementations of logic-based Benders decomposition.

There is perennial debate over whether it is better to combine CP and OR in a single general-purpose solver, or keep them separate in highly-tuned dedicated solvers. Advocates for separation argue that the extraordinary efficiency of a dedicated solver, particularly an MILP solver, outweighs the difficulty of incorporating constraints that are better suited for the other type of solver. Advocates for integration argue that the jury is still out, because much of the efficiency of dedicated solvers is due to decades of intense engineering effort that has not been lavished on an integrated solver. MILP solvers, for example, have improved by several orders of magnitude using the same basic branch-and-cut technology. An integrated solver that does not benefit from this kind of intensive development can already outperform MILP on a variety of problems [1,2,11,252].

Even if it proves best to exploit the power of dedicated solvers, one can still profitably combine OR and CP by linking the solvers in a decomposition method such as CP-based branch and price or logic-based Benders decomposition. As discussed in Sections 7 and 8, these techniques, particularly the latter, can achieve orders-of-magnitude speedups over either of the dedicated solvers used alone.

4 Relaxation of Global Constraints

Many constraint programming languages allow the definition of global constraints that represent NP-complete problems. For example, one of the earliest constraint languages, Alice [170], included the 'circuit' constraint to state that a set of variables represent a Hamiltonian circuit in a graph. Since establishing domain consistency for such constraint would be NP-hard, it is natural to design propagation algorithms based on a relaxation of the constraint. In particular relaxations stemming from OR, such as linear programming and Lagrangian relaxations, have been used for this purpose.

One of the first systematic applications of linear relaxations in global constraint propagation was developed in a series of papers by Focacci, Lodi, and Milano [97,101,100,102]. Using the traveling salesman problem with time windows as illustrative application, they develop optimization-oriented global constraints that 1) use the linear programming bound to tighten the domain of the variable representing the objective, and 2) apply reduced-cost based variable fixing to filter sub-optimal domain values. Reduced costs can also be applied to guide and decompose the CP search [186]. In these applications, the global constraint provides an interface for the finite-domain variables in the

CP model to the continuous variables in the associated linear programming model.

The network-flow constraint is an example of a global constraint that, by definition, has an immediate linear programming model that can be exploited in CP. It was first proposed in [45] and implemented in the CHIP solver. A specific application to fixed-charge network flows was developed in [163]. A generic network flow constraint, that embeds a network simplex algorithm, was implemented in the Jacop solver [229]. In that work, the flow constraint is also used to implement flow-based propagators for all-different, cardinality constraints, and their soft versions. Other implementations of network flow constraints are presented in [108, 87], and extended to explain infeasibilities, or generate nogoods, based on the linear programming model.

Likewise, one can derive Lagrangian relaxations for global constraints, as was first done by Caseau and Laburthe [63]. That is, the global constraint now embeds a Lagrangian relaxation of the combinatorial structure it represents. Lagrangian relaxations can be appealing for this purpose, as they sometimes allow very fast combinatorial algorithms instead of linear programming – in the case of [63] the Held-Karp Lagrangian relaxation is applied as a filter for weighted Hamiltonian circuits. Moreover, for integer optimization problems the Lagrangian bound can potentially be stronger than the linear programming bound. Lastly, Lagrangian relaxations can also be used for variable fixing, based on the dual multipliers. Sellmann provides a formal study of the use of Lagrangian relaxations in CP, and in particular shows that sub-optimal Lagrangian relaxations may yield stronger domain filtering [222]. Recent perspectives on CP and Lagrangian relaxation appear in [15, 111, 104].

Successful applications of Lagrangian relaxations in CP include the traveling tournament problem [30], capacitated network design [225], automated digital recording [224], network design [77], improved arc-consistency for constraint satisfaction problems [162], resource-constrained shortest path problems [109, 121], personnel scheduling with regular expressions [182], multileaf collimator sequencing for cancer treatment [55], parallel machine scheduling [90], the traveling purchaser problem [56], the weighted-circuit global constraint [26], resource-constrained project scheduling [118], the AtMostNValue global constraint [52], propagation based on decision diagrams [34], and neural networks in empirical model learning [174, 175].

A recent development in this context is the application of *Lagrangian decomposition* to strengthen the optimization reasoning and constraint propogation in CP. In the variant of Lagrangian decomposition introduced by Guignard and Kim [122], a given problem is reformulated by introducing a copy of the variables for pre-specified subproblems, while equality constraints are added on these variables to ensure that they are assigned the same value. The Lagrangian decomposition then follows from dualizing these equality constraints with associated Lagrangian multipliers. This perspective can be directly applied to CP models, because constraints are individually propagated – as if defined on a copy of the variables. The implicit equality constraints between variables in different constraints of a CP model can therefore be exploited by a Lagrangian decomposition, as first proposed, simultaneously, in [33] and [123]. Constraint propagation based on Lagrangian decomposition can be stronger than pairwise consistency [33]. The application of Lagrangian decomposition in CP was further studied by Chu et al. [69]. In that work, subproblems are solved via search rather than through a specialized propagator, which allows the application to arbitrary subproblems instead of those defined by global constraints.

Linear relaxations that define some or all facets of the convex hull of the feasible set have been developed for several global constraints, including the element constraint [141], the all-different constraint [141, 249, 149], and systems of all-different constraints [9, 40, 41]. A convex-hull linear relaxation for logical combinations of cardinality constraints is given in [251] and generalized in [16]. Various MILP models have been proposed for cumulative scheduling [64, 149, 242], and the integrality constraints can be dropped to obtain an LP relaxation. A relaxation that is not based on MILP, but defines a class of valid inequalities that are sometimes facet-defining, is presented in [149].

5 Linear Relaxations from CP Models

In the previous section, linear or Lagrangian relaxations are inferred from individual (global) constraints, which represent a specific combinatorial structure. This approach can be generalized to arbitrary subsets of constraints, or even the entire problem. That is, for a given CP (sub)problem, we can create a linear programming model that serves as a relaxation to the problem. Such linear model can then be maintained during search and applied for improved optimization bounds, reduced-cost based variable fixing, or guiding the search.

The first systematic approaches to automatically reformulate CP models into linear programming models were proposed for this purpose by Rodosek et al. [213] and Refalo [207]. The approach was further developed and implemented in the eplex library of the constraint logic programming system Eclipse in [227]. Belov et al. [25] present a related work that automatically translates CP models in MiniZinc to equivalent linear MIP models, to be solved by MIP solvers. However, such generic transformations may lead to poor LP relaxations, especially when many 'big-M' constraints are needed. Stronger linear models may be derived by taking into account the semantic information in CP models. In particular, Laborie and Rogerie present an automatically generated linear relaxation for advanced scheduling models, as used in IBM ILOG CP Optimizer. This LP relaxation can be particularly helpful for complex objective functions [167].

Naturally, the strongest possible LP relaxations can be derived for specific applications. In addition to a tailored linear model, this also allows the addition of problem-specific cuts to strengthen the relaxation. Example applications in which dedicated LP relaxations are embedded as a global constraint in CP models include multi-agent scheduling [136], integrated employee timetabling and job-shop scheduling [12], and time-dependent sequencing problems [164].

6 OR-Based Filtering Methods

OR methods have made major contributions to domain filtering for global constraints in CP. Outstanding examples include the all-different constraint, the generalized cardinality constraint, disjunctive and cumulative scheduling constraints, the sequence constraint, and the stretch constraint.

The all-different constraint first appeared in 1978 [170]. Filtering algorithms that achieve domain consistency for all-different were derived in the early 1990s [208] using results from matching theory in the OR literature [155,177,75], which is in turn based on classical network flow theory. The OR literature also provided the basis for achieving bounds consistency [181], namely a result for convex graphs [112]. The generalized cardinality constraint is filtered using a network flow model [209], and bounds consistency achieved using a flow-based algorithm that again exploits convexity of the graph [161].

In the 2000s the network-flow based propagation was extended to cost-based global constraints, by representing them with minimum-cost network flows. This was first done to establish domain consistency for weighted cardinality constraints [210,211]. Minimum-cost network flows have also been applied to establish domain consistency on soft global constraints for which one aims to minimize the violation, as was first done for the the soft all-different constraint [245]. This approach was generalized in [137] and applied to soft cardinality and soft regular constraints. An overview of soft global constraints can be found in [135]. Other global constraints that use minimum-cost network flows include the soft sequence constraint [179], the soft all-different constraint with preferences [183], the soft cardinality and soft regular constraints with preferences [184,185], soft global constraints for weighted CSPs [171], and soft open global constraints [178].

Disjunctive and cumulative scheduling represent one of the key successes of OR/CP collaboration. It began with the edge-finding algorithms of Carlier and Pinson, published in the OR literature [58–61]. These algorithms reduce the time windows within which tasks must execute, based on the fact that they cannot overlap, and thereby accelerate the search for a feasible schedule. The technology then passed over to the CP community, which further developed edge-finding methods for disjunctive scheduling [18,190,193] and extended them to allow incremental updates [62] and setup times [50,103]. These were followed by not-first/not-last rules, which achieve some bound tightening missed by edge finding [17,84,237]. In the meantime, the cumulative scheduling constraint was introduced [3], which along with its variations, became a major component of CP's powerful scheduling technology. A number of edge-finding algorithms for the constraint appeared [18,62,191,192], along with "extended" edge finding [18,190], not-first/not-last rules [191,192], and energetic reasoning [92,93]. Much of this work is described in [18]. Although these contributions advanced substantially beyond the original edge-finding methods, they owe their intellectual inspiration to ideas that came out of the OR literature.

The sequence constraint [23] also illustrates a remarkable linkage of OR and CP. While there are elegant polynomial-time filters for achieving domain consistency that do not rely on OR methods [48, 246, 138], a competitive polytime filtering method [179] is grounded in deep results from integer programming. An integer programming model for the constraint has a coefficient matrix that exhibits the consecutive ones property, which means that the matrix is totally unimodular, and the problem can be solved by LP alone. Furthermore, it is known that such a problem can be given a specially-structured LP formulation, namely a rather unobvious network flow model [4, 247]. This provides the basis for an efficient polytime filtering algorithm.

Dynamic programming, a classical OR technique that dates back to the 1950s [24], can also be used to filter global constraints. For example, it can be applied to knapsack constraints [239, 240]. Constraints of this form are traditionally propagated using simple bound consistency, which is rather weak. Dynamic programming can be used instead to establish domain consistency. Another example is the stretch constraint [200], which is designed for employee shift scheduling and related applications. It can be efficiently filtered with a dynamic programming algorithm that is carefully tailored to exploit the combinatorial structure of the constraint [131]. It can be generalized to the regular constraint [201] which is applied to represent strings that belong to the language of a regular expression.

7 Column Generation

Some linear programming models consist of a huge number of variables, as compared to the number of constraints – perhaps the size of the model even exceeds the memory of the computer. It is still possible to solve such LPs efficiently, by recognizing that an optimal solution only needs at most as many non-zero variables as the number of constraints. Namely, we can start with an initial (small) subset of variables that permits a feasible LP solution. After solving the LP, we identify a new variable that may improve the current solution by evaluating its reduced cost. We then add the new variable to the LP model and repeat. If there are no variables with a negative reduced cost (for a minimization problem), the current solution is optimal, by LP theory [71].

This decomposition approach is called column generation, as variables correspond to columns in the matrix representation of LP models. The LP defined by the current set of columns is called the *(restricted) master problem*. Finding a new variable consists of finding the entries of its column, i.e., the coefficients of the linear constraints in which the variable appears, which is done in the *pricing problem*. Column generation can also be applied to integer linear programming models, by embedding the procedure in an enumerative search called *branch-and-price* [19]. It is one of the most important and widely used OR techniques for large-scale optimization.

Column generation is most effective when variables represent a combinatorial pattern, for example a work schedule for an employee, the locations on a route for a truck, or a cutting pattern to divide a bar of steel into sub-pieces. This means that the pricing problem is a combinatorial optimization problem – for example, find the minimum-cost schedule, route, or pattern. Therefore, we can apply a hybrid CP/OR approach: The master problem can be solved with linear programming, while the combinatorial pricing problem may be solved with CP. The general CP-based column generation framework was first proposed by Junker et al. [157] and Yunes et al. [254,255].

Given the popularity of column generation in OR, many dedicated approaches exist for solving the pricing problem. A major benefit of using CP for this purpose is the flexibility of generic modeling, which allows to easily adapt the pricing problem to different specifications. Another benefit is the computational efficiency, especially for highly constrained scheduling problems. A possible drawback, however, is that the stabilization problem of column generation may be worse when using CP, as discussed in [214].

CP-based column generation has been successfully applied to a wide range of applications, including airline crew assignment [157,66,166,94,226], the traveling tournament problem [88,89], vehicle routing with time windows [215, 216], network design [67], airline planning [115,105,116,106], urban transit crew management [255], employee timetabling [82,83], physician scheduling [110], multi-machine scheduling [219], two-dimensional bin packing [202], wireless mesh networks [57], radiation therapy delivery [55], graph coloring [120], technician dispatching [74], and operating room planning and scheduling [85, 86].

8 Benders Decomposition

Benders decomposition is designed for problems that yield a much simpler problem (the Benders subproblem) when certain variables are fixed. The subproblem is solved to obtain one or more Benders cuts that bound the cost of fixing variables to these or similar values. The Benders cuts are added to a master problem that is solved to find the next set of values for the fixed variables. The process is repeated until the optimal values of the master problem and subproblem converge. Thus the problem decomposes into two parts that communicate through Benders cuts.

In the original Benders method [27], the subproblem must be an LP, and the Benders cut is derived from the LP dual. Hooker [140,141] and Hooker and Ottosson [153] substantially generalized the classical method to logic-based Benders decomposition, in which the subproblem can in principle be any optimization or constraint satisfaction problem, and the Benders cuts are derived from an inference dual. Guidelines for applying the method can be found in [72,147,149].

Logic-based Benders decomposition (LBBD) provides a broad scope for OR/CP collaboration, because the master problem and subproblem can be attacked with different solvers, one from OR and one from CP. In most applications, the subproblem is a CP problem, perhaps a scheduling problem. Its combinatorial nature is no longer a barrier to generating Benders cuts. The master problem can be solved by whatever OR method is convenient, such as MILP or a heuristic method. Logic-based cuts must be developed anew for each problem class, unlike classical Benders cuts, which are always based on the LP dual in the same way. However, this provides an opportunity to exploit the special structure of the problem.

The computational advantages of LBBD have been demonstrated in a wide variety of applications. One of the first [156] was a planning and scheduling problem in which the master problem (solved by MILP) assigns jobs to machines, and the subproblem (solved by CP) schedules the jobs subject to time windows. The subproblem decouples into a separate problem for each machine, thus making the problem well suited to decomposition. Recently updated experiments [72] found that LBBD remains several orders of magnitude faster than the latest MILP technology for this problem, and the advantage over a dedicated CP solver even greater. Similar results have been obtained for various planning and scheduling problems [70,129,144,146,148,238].

Other successful LBBD applications that combine OR and CP include shop scheduling [14,65,128,218], shift scheduling [220], batch scheduling in chemical plants [180,235], computer processor scheduling [28,29,53,91,133, 173,176,217,221], facility location [95,96,232], concrete delivery [165], stock cutting [80], space packing [76], vehicle routing [73,212,250], network design [231], operator counts in planning algorithms [79], home health care [130], service restoration [114], queuing design and control [233], optimal control of dynamical systems [47], propositional satisfiability [13], and sports scheduling [205,241]. LBBD is compared with branch and check in [22] and is implemented in SIMPL [252] as well as MiniZinc [78].

In [168], Lam and Van Hentenryck integrate column generation, Benders decomposition and constraint programming in a branch-and-price-and-check framework for a rich vehicle routing problem with location congestion. The same authors present in [169] a branch-and-check approach for vehicle routing problems in which combinatorial cuts are found by general-purpose conflict analysis using a SAT solver.

9 Decision Diagrams and Dynamic Programming

Decision diagrams [5,7,51] have long been used for circuit design and product configuration. More recently, Hadžić and Hooker [124–126] adapted decision diagrams to optimization and, with Anderson and Tiedemann [8,127], showed that they can be an effective alternative to the traditional constraint store in CP. Rather than propagate through variable domains, one can propagate through a decision diagram that represents a discrete relaxation of the problem. The connection of decision diagrams to operations research is that they are well suited for the solution of optimization problems that have dynamic programming models [35, 39, 134, 113].

Dynamic programming models are normally solved by a recursive process that enumerates the state space at each stage of the recursion. Because the state space typically grows exponentially with the number of state variables, such techniques as state space approximation and approximate dynamic programming are often used to resist the "curse of dimensionality" [68, 187, 203]. Decision diagrams provide the option of solving the problem by a branch-and-bound technique, and in particular, one that branches on nodes of a relaxed decision diagram rather than on values of variables [38]. The bounding mechanism is based on relaxation values obtained from relaxed sub-diagrams rooted at branching nodes [42, 37], much as traditional branch and bound is based on relaxation bounds obtained from LP relaxations at nodes of the branching tree. This can lead to significant speedups relative to state-of-the-art MILP solvers on some problems that have a natural MILP formulation [38]. Its primary potential, however, is in the solution of dynamic programming models that are not readily formulated as MILP problems [35].

Viewing a dynamic programming model in terms of decision diagrams can occasionally lead to radical simplification of the problem [113]. This is accomplished by rearranging costs on the arcs of the decision diagram (which are immediate costs in the dynamic programming model) so that the diagram can be reduced to a much simpler diagram.

10 Future Perspectives

The integration of CP and OR has proceeded over a period of nearly three decades, first rather slowly, but at a gradually quickening pace. It has brought improved solution methods—sometimes radically improved—to a wide variety of problems, as well as advances in modeling. The perspective afforded by one field has lent new insight into the other, which in turn leads to still more effective methods.

Despite this considerable progress, there remains great potential for further integration, with the concomitant improvement in both modeling and solution methods. Any attempt to predict the direction of research is a fool's errand, but we can point out some current research activity that shows promise for further progress, as well as some possible areas for future research.

One active area of current research is the development of *advanced modeling systems* that invoke both CP and OR solvers. These include the OPL [243], miniZinc [188] and Savile Row [189] modeling languages (although the last, at this writing, invokes only CP and SAT solvers). These systems do not yet integrate CP and OR at a low level as does the research code SIMPL [253], but they are moving that direction. For example, miniZinc recently added SIMPL's capability to implement logic-based Benders decomposition [78].

A second active research area, discussed in Section 9, is the introduction of *decision diagrams* into constraint solving and optimization. They link the two technologies by providing both a relaxation and a propagation tool, not to mention a novel framework for branching. For example, in the context of constraint programming, decision diagrams provide a generic tool for modeling and propagating constraints and conjunctions of constraints [134, 32, 197–199]. In the context of integer programming, recent examples include the use of decision diagrams for generating cutting planes [236] and for representing nonlinear objective functions [31]. Lastly, due to the close connection between decision diagrams and dynamic programming, they also offer alternative methods for solving dynamic programming problems, such as a branch-and-bound search that uses bounds from relaxed decision diagrams.

A third area is the employment of *Lagrangian relaxation* in CP methods, as described in Section 4. Lagrangian methods have long been a staple of OR and can be very useful in CP as well, primarily to strengthen propagation. They can also improve bounds obtained from relaxed decision diagrams [36]. CP-based Lagrangian methods have already seen numerous applications and hold out significant potential for further contributions to CP technology.

Looking to the future, at least three areas of CP/OR collaboration may lie on the horizon. One is the development of a *highly-engineered solver* that is analogous to CP and MILP solvers but fully integrates the two technologies along the lines described in Section 3. Such a solver would be linked to a modeling system that conveys problem structure to the solver through the use of high-level or global constraints.

A second possible area of collaboration is *explanation*, which is conceptually related to duality in optimization. The dual solution of an optimization problem can be viewed as an explanation of why the solution obtained is optimal, or why there is no feasible solution. Explanation is important not only for the sake of interpreting the solution for users, but also for sensitivity analysis and the identification of Benders cuts. Explanation has received some attention in CP [54, 158–160], while duality has been studied in OR for decades. Ideally, these two bodies of thought would join forces, at both a theoretical and practical level, to provide more comprehensive postoptimality analysis.

Another avenue for interaction at the theoretical level is the relationship between *consistency and relaxation*. Interestingly, a concept of consistency, as understood in CP, never developed in the OR literature. Yet both consistency and relaxation influence the amount of backtracking necessary to solve a problem. What has not been adequately studied is the theoretical relationship between them: the extent to which techniques designed to strengthen relaxations, such as cutting planes, also achieve some degree of consistency in a constraint set, and the extent to which consistency maintenance can strengthen a relaxation.

The separate evolution of CP and OR was advantageous for a time, as it allowed the two fields to develop complementary approaches to problem solving. We have reached a point, however, where there is much to be gained—and much has already been gained—by recognizing their underlying unity and combining their insights. Perhaps the future will see CP and optimization taught and practiced as a single field.

References

1. Achterberg, T.: SCIP: Solving constraint integer programs. Mathematical Programming Computation **1**, 1–41 (2008)
2. Achterberg, T., Berthold, T., Koch, T., Wolter, K.: A new approach to integrate CP and MIP. In: L. Perron, M.A. Trick (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 5015, pp. 6–20. Springer (2008)
3. Aggoun, A., Beldiceanu, N.: Extending CHIP in order to solve complex scheduling and placement problems. Mathematical and Computer Modelling **17**, 57–73 (1993)
4. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Linear Programming and Network Flows, 3rd ed. Prentice-Hall, Upper Saddle River, NJ (1993)
5. Akers, S.B.: Binary decision diagrams. IEEE Transactions on Computers **C-27**, 509–516 (1978)
6. Althaus, E., Bockmayr, A., Elf, M., Kasper, T., Jüenger, M., Mehlhorn, K.: SCIL—Symbolic constraints in integer linear programming. In: 10th European symposium on Algorithms (ESA 2002), *Lecture Notes in Computer Science*, vol. 2461, pp. 75–87. Springer (2002)
7. Andersen, H.R.: An introduction to binary decision diagrams. Lecture notes, available online, IT University of Copenhagen (1997)
8. Andersen, H.R., Hadžić, T., Hooker, J.N., Tiedemann, P.: A constraint store based on multivalued decision diagrams. In: C. Bessiere (ed.) Principles and Practice of Constraint Programming (CP 2007), *Lecture Notes in Computer Science*, vol. 4741, pp. 118–132. Springer (2007)
9. Appa, G., Magos, D., Mourtos, I.: A polyhedral approach to the *alldifferent* system. Mathematical Programming **124**, 1–52 (2010)
10. Appa, G., Mourtos, I., Magos, D.: Integrating constraint and integer programming for the orthogonal Latin squares problem. In: P. Van Hentenryck (ed.) Principles and Practice of Constraint Programming (CP 2002), *Lecture Notes in Computer Science*, vol. 2470, pp. 17–32. Springer (2002)
11. Aron, I., Hooker, J.N., Yunes, T.H.: SIMPL: A system for integrating optimization techniques. In: J.C. Régin, M. Rueher (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 3011, pp. 21–36. Springer (2004)
12. Artigues, C., Gendreau, M., Rousseau, L.M., Vergnaud, A.: Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound. Computers & Operations Research **36**, 2330–2340 (2009)
13. Bacchus, F., Dalmao, S., Pitassi, T.: Relaxation search: A simple way of managing optional clauses. In: AAAI Conference on Artificial Intelligence, pp. 835–841 (2014)
14. Bajestani, M.A., Beck, J.C.: Scheduling a dynamic aircraft repair shop with limited repair resources. Journal of Artificial Intelligence Research **47**, 35–70 (2013)
15. Bajgiran, O., Cire, A., Rousseau, L.M.: A first look at picking dual variables for maximizing reduced-cost based fixing. In: M. Lombardi, D. Salvagnin (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 10335, pp. 221–228. Springer (2017)
16. Balas, E., Bockmayr, A., Pisaruk, N., Wolsey, L.: On unions and dominants of polytopes. Mathematical Programming **99**, 223–239 (2004)
17. Baptiste, P., Le Pape, C.: Edge-finding constraint propagation algorithms for disjunctive and cumulative scheduling. In: Proceedings of the Fifteenth Workshop of the U.K. Planning Special Interest Group. Liverpool, U.K. (1996)
18. Baptiste, P., Le Pape, C., Nuijten, W.: Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems. Kluwer, Dordrecht (2001)
19. Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H.: Branch-and-price: Column generation for solving huge integer programs. Operations Research **46**, 316–329 (1998)

20. Beame, P., Kautz, H., Sabharwal, A.: Understanding the power of clause learning. In: International Joint Conference on Artificial Intelligence (IJCAI 2003) (2003)
21. Beaumont, N.: An algorithm for disjunctive programs. European Journal of Operational Research **48**, 362–371 (1990)
22. Beck, J.C.: Checking up on branch-and-check. In: D. Cohen (ed.) Principle and Practice of Constraint Programming (CP), *Lecture Notes in Computer Science*, vol. 6308, pp. 84–98 (2010)
23. Beldiceanu, N., Contejean, E.: Introducing global constraints in CHIP. Mathematical and Computer Modelling **12**, 97–123 (1994)
24. Bellman, R.: Dynamic programming. Princeton University Press, Princeton, NJ (1957)
25. Belov, G., Stuckey, P.J., Tack, G., Wallace, M.: Improved Linearization of Constraint Programming Models. In: Proceedings of CP, *Lecture Notes in Computer Science*, vol. 9892, pp. 49–65. Springer (2016)
26. Benchimol, P., van Hoeve, W.J., Régin, J.C., Rousseau, L.M., Rueher, M.: Improved filtering for weighted circuit constraints. Constraints **17**(3), 205–233 (2012)
27. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik **4**, 238–252 (1962)
28. Benini, L., Bertozzi, D., Guerri, A., Milano, M.: Allocation and scheduling for MPSoCs via decomposition and no-good generation. In: Principles and Practice of Constraint Programming (CP 2005), *Lecture Notes in Computer Science*, vol. 3709, pp. 107–121. Springer (2005)
29. Benini, L., Lombardi, M., Mantovani, M., Milano, M., Ruggiero, M.: Multi-stage Benders decomposition for optimizing multicore architectures. In: L. Perron, M.A. Trick (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 5015, pp. 36–50. Springer (2008)
30. Benoist, T., Laburthe, F., Rottembourg, B.: Lagrange relaxation and constraint programming collaborative schemes for traveling tournament problems. In: C. Gervet, M. Wallace (eds.) CPAIOR Proceedings. Ashford, U.K. (2001)
31. Bergman, D., Cire, A.A.: Discrete Nonlinear Optimization by State-Space Decompositions. Management Science (to appear)
32. Bergman, D., Cire, A.A., van Hoeve, W.J.: MDD Propagation for Sequence Constraints. JAIR **50**, 697–722 (2014)
33. Bergman, D., Cire, A.A., van Hoeve, W.J.: Improved Constraint Propagation via Lagrangian Decomposition. In: Proceedings of CP, *Lecture Notes in Computer Science*, vol. 9255, pp. 30–38. Springer (2015)
34. Bergman, D., Cire, A.A., van Hoeve, W.J.: Lagrangian bounds from decision diagrams. Constraints **20**, 346–361 (2015)
35. Bergman, D., Cire, A.A., van Hoeve, W.J., Hooker, J.N.: Decision Diagrams for Optimization. Springer (2016)
36. Bergman, D., Ciré, A.A., van Hoeve, W.J.: Lagrangian bounds from binary decision diagrams. Constraints **20**, 346–361 (2015)
37. Bergman, D., Ciré, A.A., van Hoeve, W.J., Hooker, J.N.: Optimization bounds from binary decision diagrams. INFORMS Journal on Computing **26**, 253–268 (2013)
38. Bergman, D., Ciré, A.A., van Hoeve, W.J., Hooker, J.N.: Discrete optimization with binary decision diagrams. INFORMS Journal on Computing **28**, 47–66 (2016)
39. Bergman, D., Ciré, A.A., van Hoeve, W.J., Hooker, J.N.: Decision Diagrams for Optimization. Springer (2017)
40. Bergman, D., Hooker, J.N.: Graph coloring facets from all-different systems,. In: N. Jussien, T. Petit (eds.) CPAIOR Proceedings, pp. 50–65. Springer (2012)
41. Bergman, D., Hooker, J.N.: Graph coloring inequalities for all-different systems. Constraints **19**, 404–433 (2014)
42. Bergman, D., van Hoeve, W.J., Hooker, J.N.: Manipulating MDD relaxations for combinatorial optimization. In: T. Achterberg, J.C. Beck (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 6697, pp. 20–35. Springer (2011)
43. Bockmayr, A., Kasper, T.: Branch-and-infer: A unifying framework for integer and finite domain constraint programming. INFORMS Journal on Computing **10**, 287–300 (1998)

44. Bockmayr, A., Pisaruk, N.: Detecting infeasibility and generating cuts for mixed integer programming using constraint progrmaming. In: M. Gendreau, G. Pesant, L.M. Rousseau (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 1234. Springer (2003)
45. Bockmayr, A., Pisaruk, N., Aggoun, A.: Network flow problems in constraint programming. In: T. Walsh (ed.) Principles and Practice of Constraint Programming (CP 2001), *Lecture Notes in Computer Science*, vol. 2239, pp. 196–210. Springer (2001)
46. Bollapragada, S., Ghattas, O., Hooker, J.N.: Optimal design of truss structures by mixed logical and linear programming. Operations Research **49**, 42–51 (2001)
47. Borzabadi, A.H., Sadjadi, M.E.: Optimal control of hybrid systems by logic-based Benders decomposition. In: A. Giua, C. Mahulea, M. Silva, J. Zaytoon (eds.) Analysis and Design of Hybrid Systems, vol. 3, pp. 104–107 (2009)
48. Branch, S., Narodytska, N., Quimper, C.G., Stuckey, P., Walsh, T.: Encodings of the sequence constraint. In: C. Bessiere (ed.) Principles and Practice of Constraint Programming (CP 2007), *Lecture Notes in Computer Science*, vol. 4741, pp. 210–224. Springer (2007)
49. Brown, R.G., Chinneck, J.W., Karam, G.M.: Optimization with constraint programming systems. In: R.S. et al. (ed.) Impact of Recent Computer Advances on Operations Research, *Publications in Operations Research Series*, vol. 9, pp. 463–473. Elsevier, Williamsburg, VA (1989)
50. Brucker, P., Thiele, O.: A branch and bound method for the general-shop problem with sequence-dependent setup times. OR Spektrum **18**, 145–161 (1996)
51. Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers **C-35**, 677–691 (1986)
52. Cambazard, H., Fages, J.G.: New filtering for AtMostNValue and its weighted variant: A Lagrangian approach. Constraints **20**, 362–380 (2015)
53. Cambazard, H., Hladik, P.E., Déplanche, A.M., Jussien, N., Trinquet, Y.: Decomposition and learning for a hard real time task allocation problem. In: M. Wallace (ed.) Principles and Practice of Constraint Programming (CP 2004), *Lecture Notes in Computer Science*, vol. 3258, pp. 153–167. Springer (2004)
54. Cambazard, H., Jussien, N.: Identifying and exploiting problem structures using explanation-based costraint programming. In: R. Barták, M. Milano (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 3524, pp. 94–109. Springer (2005)
55. Cambazard, H., O'Mahony, E., O'Sullivan, B.: Hybrid methods for the multileaf collimator sequencing problem. In: A. Lodi, M. Milano, P. Toth (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 6140, pp. 56–70. Springer (2010)
56. Cambazard, H., Penz, B.: A Constraint Programming Approach for the Traveling Purchaser Problem. In: Proceedings of CP, *Lecture Notes in Computer Science*, vol. 7514, pp. 735–749. Springer (2012)
57. Capone, A., Carello, G., Filippini, I., Gualandi, S., Malucelli, F.: Solving a resource allocation problem in wireless mesh networks: A comparison between a cp-based and a classical column generation. Networks **55**(3), 221–233 (2010)
58. Carlier, J.: One machine problem. European Journal of Operational Research **11**, 42–47 (1982)
59. Carlier, J., Pinson, E.: An algorithm for solving the job-shop problem. Management Science **35**, 164–176 (1989)
60. Carlier, J., Pinson, E.: A practical use of Jackson's preemptive schedule for solving the job shop problem. Annals of Operations Research **26**, 269–287 (1990)
61. Carlier, J., Pinson, E.: Adjustment of heads and tails for the job-shop problem. European Journal of Operational Research **78**, 146–161 (1994)
62. Caseau, Y., Laburthe, F.: Improved CLP scheduling with task intervals. In: Proceedings of the Eleventh International Conference on Logic Programming (ICLP 1994), pp. 369–383. MIT Press (1994)
63. Caseau, Y., Laburthe, F.: Solving small TSPs with constraints. In: L. Naish (ed.) Proceedings, Fourteenth International Conference on Logic Programming (ICLP 1997), vol. 2833, pp. 316–330. The MIT Press (1997)

64. Castro, P.M., Grossmann, I.E.: An efficient MILP model for the short-term scheduling of single stage batch plants. technical report, Departamento de Modela cão e Simula cão de Processos, INETI, Lisbon (2006)
65. Çoban, E., Hooker, J.N.: Single-facility scheduling by logic-based benders decomposition. Annals of Operations Research **210**, 245–272 (2013)
66. Chabrier, A.: A cooperative CP and LP optimizer approach for the pairing generation problem. In: CPAIOR Proceedings. Ferrara, Italy (2000)
67. Chabrier, A.: Heuristic branch-and-price-and-cut to solve a network design problem. In: M. Gendreau, G. Pesant, L.M. Rousseau (eds.) CPAIOR Proceedings. Montréal (2003)
68. Christofides, N., Mingozzi, A., Toth, P.: State-space relaxation procedures for the computation of bounds to routing problems. Networks **11**(2), 145–164 (1981)
69. Chu, G., Gange, G., Stuckey, P.J.: Lagrangian Decomposition via Sub-problem Search. In: Proceedings of CPAIOR, *Lecture Notes in Computer Science*, vol. 9676, pp. 65–80. Springer (2016)
70. Chu, Y., Xia, Q.: A hybrid algorithm for a class of resource-constrained scheduling problems. In: R. Barták, M. Milano (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 3524, pp. 110–124. Springer (2005)
71. Chvátal, V.: Linear Programming. W. H. Freeman, New York (1983)
72. Ciré, A.A., Çoban, E., Hooker, J.N.: Mixed integer programming vs logic-based Benders decomposition for planning and scheduling. In: C. Gomes, M. Sellmann (eds.) CPAIOR Proceedings, pp. 325–331 (2013)
73. Corréa, A.I., Langevin, A., Rousseau, L.M.: Dispatching and conflict-free routing of automated guided vehicles: A hybrid approach combining constraint programming and mixed integer programming. In: J.C. Régin, M. Rueher (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 3011, pp. 370–378. Springer (2004)
74. Cortés, C.E., Gendreau, M., Rousseau, L.M., Souyris, S., Weintraub, A.: Branch-and-price and constraint programming for solving a real-life technician dispatching problem. European Journal of Operational Research **238**, 300–312 (2014)
75. Costa, M.C.: Persistency in maximum cardinality bipartite matchings. Operations Research Letters **15**, 143–149 (1994)
76. Côté, J.F., Dell'Amico, M., Iori, M.: Combinatorial Benders cuts for the strip packing problem. Operations Research **62**, 643–661 (2014)
77. Cronholm, W., Ajili, F.: Strong cost-based filtering for Lagrange decomposition applied to network design. In: M. Wallace (ed.) Principles and Practice of Constraint Programming (CP 2004), *Lecture Notes in Computer Science*, vol. 3258, pp. 726–730. Springer (2004)
78. Davies, T.O., Gange, G., Stuckey, P.J.: Automatic logic-based Benders decomposition with mini-zinc. In: 31st AAAI Conference on Artificial Intelligence (AAAI 2017), pp. 787–793 (2017)
79. Davies, T.O., Pearce, A.R., Stuckey, P.J., Lipovetzky, N.: Sequencing operator counts. In: International Conference on Automated Planning and Scheduling (ICAPS), pp. 61–69 (2015)
80. Delorme, M., Iori, M., Martello, S.: Logic based Benders' decomposition for orthogonal stock cutting problems=. Computers and Operations Research **78**, 290–298 (2017)
81. Demassey, S., Artiques, C., Michelon, P.: A hybrid constraint propagation-cutting plane prtocedure for the RCPSP. In: N. Jussien, F. Laburthe (eds.) Proceedings of the International Workshop on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2002). Le Croisic, France (2002)
82. Demassey, S., Pesant, G., Rousseau, L.M.: Constraint-programming based column generation for employee timetabling. In: R. Barták, M. Milano (eds.) Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2005), *Lecture Notes in Computer Science*, vol. 3524, pp. 140–154. Springer (2005)
83. Demassey, S., Pesant, G., Rousseau, L.M.: A Cost-Regular Based Hybrid Column Generation Approach. Constraints **11**(4), 315–333 (2006)
84. Dorndorf, U., Pesch, E., Phan-Huy, T.: Solving the open shop scheduling problem. Journal of Scheduling **4**, 157–174 (2001)

85. Doulabi, S.H.H., Rousseau, L.M., Pesant, G.: A Constraint Programming-Based Column Generation Approach for Operating Room Planning and Scheduling. In: Proceedings of CPAIOR, *Lecture Notes in Computer Science*, vol. 8451, pp. 455–463. Springer (2014)
86. Doulabi, S.H.H., Rousseau, L.M., Pesant, G.: A Constraint-Programming-Based Branch-and-Price-and-Cut Approach for Operating Room Planning and Scheduling. INFORMS Journal on Computing **28**, 432–448 (2016)
87. Downing, N., Feydy, T., Stuckey, P.J.: Explaining Flow-Based Propagation. In: Proceedings of CPAIOR, *Lecture Notes in Computer Science*, vol. 7298, pp. 146–162. Springer (2012)
88. Easton, K., Nemhauser, G., Trick, M.: The traveling tournament problem description and benchmarks. In: T. Walsh (ed.) Principles and Practice of Constraint Programming (CP 2001), *Lecture Notes in Computer Science*, vol. 2239, pp. 580–584. Springer (2001)
89. Easton, K., Nemhauser, G., Trick, M.: Solving the traveling tournament problem: A combined integer programming and constraint programming approach. In: Proceedings of the International Conference on the Practice and Theory of Automated Timetabling (PATAT 2002) (2002)
90. Edis, E.B., Oguz, C.: Parallel Machine Scheduling with Additional Resources: A Lagrangian-Based Constraint Programming Approach. In: Proceedings of CPAIOR, *Lecture Notes in Computer Science*, vol. 6697, pp. 92–98. Springer (2011)
91. Emeretlis, A., Theodoridis, G., Alefragis, P., Voros, N.: Mapping DAGs on heterogeneous platforms using logic-based Benders decompostion. In: IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 119–124. IEEE (2015)
92. Erschler, J., Lopez, P., Esquirol, P.: Ordonnancement de tâches sous contraintes: Une approche énergétique. RAIRO Automatique, Productique, Informatique Industrielle **26**, 453–481 (1992)
93. Erschler, J., Lopez, P., Thuriot, C.: Raisonnement temporel sous contraintes de ressource et problèmes d'ordonnancement. Revue d'Intelligence Artificielle **5**, 7–32 (1991)
94. Fahle, T., Junker, U., Karish, S.E., Kohn, N., Sellmann, M., Vaaben, B.: Constraint programming based column generation for crew assignment. Journal of Heuristics **8**, 59–81 (2002)
95. Fazel-Zarandi, M.M.: Using logic-based Benders decomposition to solve the capacity- and distance-constrained plant location problem. INFORMS Journal on Computing **24**, 387–398 (2012)
96. Fazel-Zarandi, M.M., Beck, J.C.: Solving a location-allocation problem with logic-based Benders decomposition. In: I.P. Gent (ed.) Principles and Practice of Constraint Programming (CP 2009), *Lecture Notes in Computer Science*, vol. 5732, pp. 344–351. Springer, New York (2009)
97. Focacci, F., Lodi, A., Milano, M.: Cost-based domain filtering. In: J. Jaffar (ed.) Principles and Practice of Constraint Programming (CP 1999), *Lecture Notes in Computer Science*, vol. 1713, pp. 189–203. Springer (1999)
98. Focacci, F., Lodi, A., Milano, M.: Solving TSP with time windows with constraints. In: International Conference on Logic Programming (ICLP 1999), pp. 515–529. MIT Press (1999)
99. Focacci, F., Lodi, A., Milano, M.: Cutting planes in constraint programming: An hybrid approach. In: R. Dechter (ed.) Principles and Practice of Constraint Programming (CP 2000), *Lecture Notes in Computer Science*, vol. 1894, pp. 187–201. Springer (2000)
100. Focacci, F., Lodi, A., Milano, M.: A Hybrid Exact Algorithm for the TSPTW. INFORMS Journal on Computing **14**, 403–417 (2002)
101. Focacci, F., Lodi, A., Milano, M.: Embedding Relaxations in Global Constraints for Solving TSP and TSPTW. Annals of Mathematics and Artificial Intelligence **34**, 291–311 (2002)
102. Focacci, F., Lodi, A., Milano, M.: Optimization-Oriented Global Constraints. Constraints **7**, 351–365 (2002)
103. Focacci, F., Nuijten, W.P.M.: A constraint propagation algorithm for scheduling with sequence dependent setup times. In: U. Junker, S.E. Karisch, S. Tschöke (eds.) Proceedings of the International Workshop on Integration of Artificial Intelligence

and Operations Research Techniques in Constraint Programming for Combintaorial Optimization Problems (CPAIOR 2000), pp. 53–55. Paderborn, Germany (2000)
104. Fontaine, D., Michel, L.D., Hentenryck, P.V.: Constraint-Based Lagrangian Relaxation. In: Proceedings of CP, *Lecture Notes in Computer Science*, vol. 8656, pp. 324–339. Springer (2014)
105. Gabteni, S., Grönkvist, M.: A Hybrid Column Generation and Constraint Programming Optimizer for the Tail Assignment Problem. In: Proceedings of CPAIOR, *Lecture Notes in Computer Science*, vol. 3990, pp. 89–103. Springer (2006)
106. Gabteni, S., Grönkvist, M.: Combining column generation and constraint programming to solve the tail assignment problem. Annals of Operations Research **171**(1), 61–76 (2009)
107. Garfinkel, R., Nemhauser, G.L.: Optimal political districting by implicit enumeration techniques. Management Science **16**, B495–B508 (1970)
108. Gaudin, E., Jussien, N., Rochart, G.: Implementing explained global constraints. In: In Proceedings of the CP04 Workshop on Constraint Propagation and Implementation, pp. 61–76 (2004)
109. Gellermann, T., Sellmann, M., Wright, R.: Shorter-path constraints for the resource constrained shortest path problem. In: R. Barták, M. Milano (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 3524, pp. 201–216. Springer (2005)
110. Gendron, B., Lebbah, H., Pesant, G.: Improving the cooperation between the master problem and the subproblem in constraint programming based column generation. In: R. Barták, M. Milano (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 3524, pp. 217–227. Springer (2005)
111. German, G., Briant, O., Cambazard, H., Jost, V.: Arc consistency via linear programming. In: Proceedings of CP, *LNCS*, vol. 10416, pp. 114–128. Springer (2017)
112. Glover, F.: Maximum matching in a convex bipartite graph. Naval Research Logistics Quarterly **316**, 313–316 (1967)
113. Gomes, C.P., Sellmann, M. (eds.): Decision diagrams and dynamic programming, *Lecture Notes in Computer Science*, vol. 7874. Springer (2013)
114. Gong, J., Lee, E.E., Mitchell, J.E., Wallace, W.A.: Logic-based multiobjective optimization for restoration planning. In: W. Chaovalitwongse, K.C. Furman, P.M. Pardalos (eds.) Optimization and Logistics Challenges in the Enterprise, pp. 305–324. Springer (2009)
115. Grönkvist, M.: A constraint programming model for tail assignment. In: J.C. Régin, M. Rueher (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 3011, pp. 142–156. Springer (2004)
116. Grönkvist, M.: Accelerating column generation for aircraft scheduling using constraint propagation. Computers & Operations Research **33**(10), 2918–2934 (2006)
117. Grossmann, I.E., Hooker, J.N., Raman, R., Yan, H.: Logic cuts for processing networks with fixed charges. Computers and Operations Research **21**, 265–279 (1994)
118. Gu, H., Schutt, A., Stuckey, P.J.: A Lagrangian Relaxation Based Forward-Backward Improvement Heuristic for Maximising the Net Present Value of Resource-Constrained Projects. In: Proceedings of CPAIOR, *Lecture Notes in Computer Science*, vol. 7874, pp. 340–346. Springer (2013)
119. Gualandi, M.: k-clustering minimum biclique completion via a hybrid CP and SDP approach. In: W.J. van Hoeve, J.N. Hooker (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 5547, pp. 87–101. Springer, New York (2009)
120. Gualandi, S., Malucelli, F.: Exact Solution of Graph Coloring Problems via Constraint Programming and Column Generation. INFORMS Journal on Computing **24**, 81–100 (2012)
121. Gualandi, S., Malucelli, F.: Resource Constrained Shortest Paths with a Super Additive Objective Function. In: Proceedings of CP, *Lecture Notes in Computer Science*, vol. 7514, pp. 299–315. Springer (2012)
122. Guignard, M., Kim, S.: Lagrangian Decomposition: A Model Yielding Stronger Lagrangian Bounds. Mathematical Programming **39**, 215–228 (1987)
123. Ha, M.H., Quimper, C.G., Rousseau, L.M.: General Bounding Mechanism for Constraint Programs. In: Proceedings of CP, *Lecture Notes in Computer Science*, vol. 9255, pp. 158–172. Springer (2015)

124. Hadžić, T., Hooker, J.N.: Discrete global optimization with binary decision diagrams. In: Workshop on Global Optimization: Integrating Convexity, Optimization, Logic Programming, and Computational Algebraic Geometry (GICOLAG). Vienna (2006)
125. Hadžić, T., Hooker, J.N.: Postoptimality analysis for integer programming using binary decision diagrams, presented at GICOLAG workshop (Global Optimization: Integrating Convexity, Optimization, Logic Programming, and Computational Algebraic Geometry), Vienna. Tech. rep., Carnegie Mellon University (2006)
126. Hadžić, T., Hooker, J.N.: Cost-bounded binary decision diagrams for 0–1 programming. In: P. van Hentemryck, L. Wolsey (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 4510, pp. 332–345. Springer (2007)
127. Hadžić, T., Hooker, J.N., Tiedemann, P.: Propagating separable inequalities in an MD store. In: L. Perron, M.A. Trick (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 5015, pp. 318–322. Springer (2008)
128. Hamdi, I., Loukil, T.: Logic-based Benders decomposition to solve the permutation flowshop scheduling problem with time lags. In: International Conference on Modeling, Simulation and Applied Optimization (ICMSAO), pp. 1–7. IEEE (2013)
129. Harjunkoski, I., Grossmann, I.E.: Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. Computers and Chemical Engineering **26**, 1533–1552 (2002)
130. Heching, A., Hooker, J.N.: Scheduling home hospice care with logic-based Benders decomposition. In: C.G. Quimper (ed.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 9676, pp. 187–197. Springer (2016)
131. Hellsten, L., Pesant, G., van Beek, P.: A domain consistency algorithm for the stretch constraint. In: M. Wallace (ed.) Principles and Practice of Constraint Programming (CP 2004), *Lecture Notes in Computer Science*, vol. 3258, pp. 290–304. Springer (2004)
132. Herbrard, E., O'Mahony, E., O'Sullivan, B.: A constraint integer programming approach for resource-constrained project scheduling. In: A. Lodi, M. Milano, P. Toth (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 6140, pp. 313–317. Springer (2010)
133. Hladik, P.E., Cambazard, H., Déplanche, A.M., Jussien, N.: Solving a real-time allocation problem with constraint programming. Journal of Systems and Software **81**, 132–149 (2008)
134. Hoda, S., van Hoeve, W.J., Hooker, J.N.: A systematic approach to MDD-based constraint programming. In: Proceedings of the 16th International Conference on Principles and Practices of Constraint Programming, Lecture Notes in Computer Science. Springer (2010)
135. van Hoeve, W.J.: Over-Constrained Problems. In: P.V. Hentenryck, M. Milano (eds.) Hybrid Optimization: The Ten Years of CPAIOR, pp. 191–225. Springer (2011)
136. van Hoeve, W.J., Gomes, C.P., Selman, B., Lombardi, M.: Optimal Multi-Agent Scheduling with Constraint Programming. In: Proceedings of AAAI, pp. 1813–1818. AAAI Press (2007)
137. van Hoeve, W.J., Pesant, G., Rousseau, L.M.: On Global Warming: Flow-Based Soft Global Constraints. Journal of Heuristics **12**, 347–373 (2006)
138. van Hoeve, W.J., Pesant, G., Rousseau, L.M., Sabharwal, A.: New Filtering Algorithms for Combinations of Among Constraints. Constraints **14**, 273–292 (2009)
139. Hooker, J.N.: Logic-based methods for optimization. In: A. Borning (ed.) Principles and Practice of Constraint Programming (CP 2002), *Lecture Notes in Computer Science*, vol. 874, pp. 336–349. Springer (1994)
140. Hooker, J.N.: Logic-based Benders decomposition. In: INFORMS National Meeting (INFORMS 1995) (1995)
141. Hooker, J.N.: Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction. Wiley, New York (2000)
142. Hooker, J.N.: Logic, optimization and constraint programming. INFORMS Journal on Computing **14**, 295–321 (2002)
143. Hooker, J.N.: A framework for integrating solution methods. In: H.K. Bhargava, M. Ye (eds.) Computational Modeling and Problem Solving in the Networked World (Proceedings of ICS2003), pp. 3–30. Kluwer (2003)
144. Hooker, J.N.: A hybrid method for planning and scheduling. Constraints **10**, 385–401 (2005)

145. Hooker, J.N.: Planning and scheduling to minimize tardiness. In: Principles and Practice of Constraint Programming (CP 2005), *Lecture Notes in Computer Science*, vol. 3709, pp. 314–327. Springer (2005)
146. Hooker, J.N.: An integrated method for planning and scheduling to minimize tardiness. Constraints **11**, 139–157 (2006)
147. Hooker, J.N.: Integrated Methods for Optimization. Springer (2007)
148. Hooker, J.N.: Planning and scheduling by logic-based Benders decomposition. Operations Research **55**, 588–602 (2007)
149. Hooker, J.N.: Integrated Methods for Optimization, 2nd ed. Springer (2012)
150. Hooker, J.N.: Projection, consistency, and George Boole. Constraints **21**, 59–76 (2016)
151. Hooker, J.N.: Projection, inference and consistency. In: IJCAI 2016 Proceedings, pp. 4175–4179 (2016)
152. Hooker, J.N., Osorio, M.A.: Mixed logical/linear programming. Discrete Applied Mathematics **96–97**, 395–442 (1999)
153. Hooker, J.N., Ottosson, G.: Logic-based Benders decomposition. Mathematical Programming **96**, 33–60 (2003)
154. Hooker, J.N., Ottosson, G., Thorsteinsson, E.S., Kim, H.J.: A scheme for unifying optimization and constraint satisfaction methods. Knowledge Engineering Review **15**, 11–30 (2000)
155. Hopcroft, J.E., Karp, R.M.: A $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM Journal on Computing **2**, 225–231 (1973)
156. Jain, V., Grossmann, I.E.: Algorithms for hybrid MILP/CP models for a class of optimization problems. INFORMS Journal on Computing **13**, 258–276 (2001)
157. Junker, U., Karish, S.E., Kohl, N., Vaaben, B., Fahle, T., Sellmann, M.: A framework for constraint programming based column generation. In: J. Jaffar (ed.) Principles and Practice of Constraint Programming (CP 1999), *Lecture Notes in Computer Science*, vol. 1713, pp. 261–275. Springer (1999)
158. Jussien, N.: The versatility of using explanations within constraint programming. Research report, École des Mines de Nantes, France (2003)
159. Jussien, N., Barichard, V.: The PaLM system: Explanation-based constraint programming. In: Proceedings of TRICS: Techniques foR Implementing Constraint programming Systems, a post-conference workshop of CP 2000, pp. 118–133 (2000)
160. Jussien, N., Ouis, S.: User-friendly explanations for constraint programming. In: Eleventh Workshop on Logic Programming environments (WLPE 2001). Paphos, Cyprus (2001)
161. Katriel, I., Thiel, S.: Fast bound consistency for the global cardinality constraint. In: F. Rossi (ed.) Principles and Practice of Constraint Programming (CP 2003), *Lecture Notes in Computer Science*, vol. 2833, pp. 437–451. Springer (2003)
162. Khemmoudj, M.O., Bennaceur, H., Nagih, A.: Combining arc consistency and dual Lagrangean relaxation for filtering CSPs. In: R. Barták, M. Milano (eds.) Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2005), *Lecture Notes in Computer Science*, vol. 3524, pp. 258–272. Springer (2005)
163. Kim, H.J., Hooker, J.N.: Solving fixed-charge network flow problems with a hybrid optimization and constraint programming approach. Annals of Operations Research **115**, 95–124 (2002)
164. Kinable, J., Cire, A.A., van Hoeve, W.J.: Hybrid optimization methods for time-dependent sequencing problems. European Journal of Operational Research **259**, 887–897 (2017)
165. Kinable, J., Trick, M.: A logic-based Benders approach to the concrete delivery problem. In: H. Simonis (ed.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 8451, pp. 176–192. Springer (2014)
166. Kohl, N.: Application of or and cp techniques in a real world crew scheduling system. In: Proceedings of the International Workshop on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2000). Paderborn, Germany (2000)
167. Laborie, P., Rogerie, J.: Temporal linear relaxation in IBM ILOG CP Optimizer. J. Scheduling **19**(4), 391–400 (2016)

168. Lam, E., Van Hentenryck, P.: A branch-and-price-and-check model for the vehicle routing problem with location congestion. Constraints **21**(3), 394–412 (2016)
169. Lam, E., Van Hentenryck, P.: Branch-and-check with explanations for the vehicle routing problem with time windows. In: J.C. Beck (ed.) Proceedings of CP, *Lecture Notes in Computer Science*, vol. 10416, pp. 579–595. Springer (2017)
170. Laurière, J.L.: A language and a program for stating and solving combinatorial problems. Artificial Intelligence **10**, 29–127 (1978)
171. Lee, J., Leung, K.: Towards Efficient Consistency Enforcement for Global Constraints in Weighted Constraint Satisfaction. In: Proceedings of IJCAI, pp. 559–565 (2009)
172. Little, J., Darby-Dowman, K.: The significance of constraint logic programming to operational research. In: M. Lawrence, C. Wilsden (eds.) Operational Research Tutorial Papers (Invited tutorial paper to the Operational Research Society Conference, 1995), pp. 20–45 (1995)
173. Liu, W., Yuan, M., He, X., Gu, Z., Liu, X.: Efficient SAT-based mapping and scheduling of homogeneous synchronous dataflow graphs for throughput optimization. In: Real-Time Systems Symposium, pp. 492–504. IEEE (2008)
174. Lombardi, M., Gualandi, S.: A New Propagator for Two-Layer Neural Networks in Empirical Model Learning. In: Proceedings of CP, *Lecture Notes in Computer Science*, vol. 8124, pp. 448–463. Springer (2013)
175. Lombardi, M., Gualandi, S.: A lagrangian propagator for artificial neural networks in constraint programming. Constraints **21**, 435–462 (2016)
176. Lombardi, M., Milano, M., Ruggiero, M., Benini, L.: Stochastic allocation and scheduling for conditional task graphs in multi-processor systems-on-chip. Journal of Scheduling **13**, 315–345 (2010)
177. Lovász, L., Plummer, M.: Matching Theory, *Annals of discrete Mathematics*, vol. 29. North-Holland (1986)
178. Maher, M.J.: SOGgy Constraints: Soft Open Global Constraints. In: Proceedings of CP, *Lecture Notes in Computer Science*, vol. 5732, pp. 584–591. Springer (2009)
179. Maher, M.J., Narodytska, N., Quimper, C.G., Walsh, T.: Flow-based propagators for the SEQUENCE and related global constraints. In: P.J. Stuckey (ed.) Principles and Practice of Constraint Programming (CP 2008), *Lecture Notes in Computer Science*, vol. 5202, pp. 159–174. Springer (2008)
180. Maravelias, C.T., Grossmann, I.E.: Using MILP and CP for the scheduling of batch chemical processes. In: J.C. Régin, M. Rueher (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 3011, pp. 1–20. Springer (2004)
181. Mehlhorn, K., Thiel, S.: Faster algorithms for bound-consistency of the sortedness and the alldifferent constraint. In: R. Dechter (ed.) Principles and Practice of Constraint Programming (CP 2000), *Lecture Notes in Computer Science*, vol. 1894, pp. 306–319. Springer (2000)
182. Menana, J., Demassey, S.: Sequencing and Counting with the multicost-regular Constraint. In: Proceedings of CPAIOR, *Lecture Notes in Computer Science*, vol. 5547, pp. 178–192. Springer (2009)
183. Métivier, J.P., Boizumault, P., Loudni, S.: Σ-AllDifferent: Softening AllDifferent in Weighted CSPs. In: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 223–230. IEEE (2007)
184. Métivier, J.P., Boizumault, P., Loudni, S.: Softening Gcc and Regular with preferences. In: Proceedings of the 2009 ACM symposium on Applied Computing (SAC), pp. 1392–1396. ACM (2009)
185. Métivier, J.P., Boizumault, P., Loudni, S.: Solving Nurse Rostering Problems Using Soft Global Constraints. In: Proceedings of CP, *Lecture Notes in Computer Science*, vol. 5732, pp. 73–87. Springer (2009)
186. Milano, M., van Hoeve, W.J.: Reduced cost-based ranking for generating promising subproblems. In: P. Van Hentenryck (ed.) Principles and Practice of Constraint Programming (CP 2002), *Lecture Notes in Computer Science*, vol. 2470, pp. 1–16. Springer (2002)
187. Mingozzi, A.: State space relaxation and search strategies in dynamic programming. In: Proceedings of Abstraction, Reformulation, and Approximation, *Lecture Notes in Computer Science*, vol. 2371, pp. 51–51. Springer (2002)

188. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.: Minizinc: Towards a standard CP modelling language. In: C. Bessiere (ed.) Principles and Practice of Constraint Programming (CP 2007), *Lecture Notes in Computer Science*, vol. 4741, pp. 529–543. Springer (2007)
189. Nightingale, P., Akgün, Ö., Gent, I.P., Jefferson, C., Miguel, I., Spracklen, P.: Automatically improving constraint models in Savile Row. Artificial Intelligence **251**, 35–61 (2017)
190. Nuijten, W.P.M.: Time and resource constrained scheduling. Ph.D. thesis, Eindhoven University of Technology (1994)
191. Nuijten, W.P.M., Aarts, E.H.L.: Constraint satisfaction for multiple capacitated job shop scheduling. In: A. Cohn (ed.) Proceedings of the 11th European Conference on Artificial Intelligence (ECAI 1994), pp. 635–639. Wiley (1994)
192. Nuijten, W.P.M., Aarts, E.H.L.: A computational study of constraint satisfaction for multiple capacitated job shop scheduling. European Journal of Operational Research **90**, 269–284 (1996)
193. Nuijten, W.P.M., Aarts, E.H.L., van Erp Taalman Kip, D.A.A., van Hee, K.M.: Randomized constraint satisfaction for job-shop scheduling. In: AAAI-SIGMAN Workshop on Knowledge-Based Production Planning, Scheduling and Control (1993)
194. Osorio, M., Glover, F.: Logic cuts using surrogate constraint analysis in the multidimensional knapsack problem. In: C. Gervet, M. Wallace (eds.) Proceedings of the International Workshop on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming for Combintaorial Optimization Problems (CPAIOR 2001). Ashford, U.K. (2001)
195. Ottosson, G., Thorsteinsson, E., Hooker, J.N.: Mixed global constraints and inference in hybrid IP-CLP solvers. In: Proceedings of CP99 Post-Conference Workshop on Large-Scale Combinatorial Optimization and Constraints, http://www.dash.co.uk/wscp99, pp. 57–78 (1999)
196. Ottosson, G., Thorsteinsson, E., Hooker, J.N.: Mixed global constraints and inference in hybrid CLP-IP solvers. Annals of Mathematics and Artificial Intelligence **34**, 271–290 (2002)
197. Perez, G., Régin, J.C.: Efficient Operations On MDDs for Building Constraint Programming Models. In: Proceedings of IJCAI, pp. 374–380 (2015)
198. Perez, G., Régin, J.C.: Constructions and In-Place Operations for MDDs Based Constraints. In: Proceedings of CPAIOR, *LNCS*, vol. 9676, pp. 279–293. Springer (2016)
199. Perez, G., Régin, J.C.: MDDs are Efficient Modeling Tools: An Application to Some Statistical Constraints. In: Proceedings of CPAIOR, *LNCS*, vol. 10335, pp. 30–40. Springer (2017)
200. Pesant, G.: A filtering algorithm for the stretch constraint. In: T. Walsh (ed.) Principles and Practice of Constraint Programming (CP 2001), *Lecture Notes in Computer Science*, vol. 2239, pp. 183–195. Springer (2001)
201. Pesant, G.: A regular language membership constraint for finite sequences of variables. In: M. Wallace (ed.) Principles and Practice of Constraint Programming (CP 2004), *Lecture Notes in Computer Science*, vol. 3258, pp. 482–495. Springer (2004)
202. Pisinger, D., Sigurd, M.: Using Decomposition Techniques and Constraint Programming for Solving the Two-Dimensional Bin-Packing Problem. INFORMS Journal on Computing **19**, 36–51 (2007)
203. Powell, W.B.: Approximate Dynamic Programming: Solving the Curses of Dimensionality (Wiley Series in Probability and Statistics). Wiley-Interscience (2007)
204. Quadrifoglio, L., Dessouky, M.M., nez, F.O.: Mobility allowance shuttle transit (MAST) services: MIP formulation and strengthenining with logic constraints. In: L. Perron, M.A. Trick (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 5015, pp. 387–391. Springer (2008)
205. Rasmussen, R., Trick, M.A.: A Benders approach to the constrained minimum break problem. European Journal of Operational Research **177**, 198–213 (2007)
206. Refalo, P.: Tight cooperation and its application in piecewise linear optimization. In: J. Jaffar (ed.) Principles and Practice of Constraint Programming (CP 1999), *Lecture Notes in Computer Science*, vol. 1713, pp. 375–389. Springer (1999)

207. Refalo, P.: Linear formulation of constraint programming models and hybrid solvers. In: R. Dechter (ed.) Principles and Practice of Constraint Programming (CP 2000), *Lecture Notes in Computer Science*, vol. 1894, pp. 369–383. Springer (2000)
208. Régin, J.C.: A filtering algorithm for constraints of difference in CSP. In: National Conference on Artificial Intelligence (AAAI 1994), pp. 362–367. AAAI Press (1994)
209. Régin, J.C.: Generalized arc consistency for *global cardinality* constraint. In: National Conference on Artificial Intelligence (AAAI 1996), pp. 209–215. AAAI Press (1996)
210. Régin, J.C.: Arc Consistency for Global Cardinality Constraints with Costs. In: Proceedings of CP, *Lecture Notes in Computer Science*, vol. 1713, pp. 390–404. Springer (1999)
211. Régin, J.C.: Cost-Based Arc Consistency for Global Cardinality Constraints. Constraints **7**, 387–405 (2002)
212. Riedler, M., Raidl, G.: Solving a selective dial-a-ride problem with logic-based Benders decomposition. Technical report AC-TR-16-007, Institute of Computer Graphics and Algorithms, TU Wien (2016)
213. Rodošek, R., Wallace, M., Hajian, M.: A new approach to integrating mixed integer programming and constraint logic programming. Annals of Operations Research **86**, 63–87 (1999)
214. Rousseau, L.M.: Stabilization issues for constraint programming based column generation. In: J.C. Régin, M. Rueher (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 3011, pp. 402–408. Springer (2004)
215. Rousseau, L.M., Gendreau, M., Pesant, G.: Solving small VRPTWs with constraint programming based column generation. In: N. Jussien, F. Laburthe (eds.) Proceedings of the International Workshop on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming for Combintaorial Optimization Problems (CPAIOR 2002). Le Croisic, France (2002)
216. Rousseau, L.M., Gendreau, M., Pesant, G., Focacci, F.: Solving VRPTWs with Constraint Programming Based Column Generation. Annals of Operations Research **130**(1–4), 199–216 (2004)
217. Ruggiero, M., Guerri, A., Bertozzi, D., Poletti, F., Milano, M.: Communication-aware allocation and scheduling framework for stream-oriented multi-processor systems-on-chip. In: Proceedings of the Conference on Design, Automation and Test in Europe, pp. 3–8. European Design and Automation Association (2006)
218. Sadykov, R.: A hybrid branch-and-cut algorithm for the one-machine scheduling problem. In: J.C. Régin, M. Rueher (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 3011, pp. 409–415. Springer (2004)
219. Sadykov, R., Wolsey, L.A.: Integer Programming and Constraint Programming in Solving a Multimachine Assignment Scheduling Problem with Deadlines and Release Dates. INFORMS Journal on Computing **18**, 209–217 (2006)
220. Salvagnin, D., Walsh, T.: A hybrid MIP/CP approach for multi-activity shift scheduling. In: M. Milano (ed.) Principles and Practice of Constraint Programming, *Lecture Notes in Computer Science*, vol. 7514, pp. 633–646. Springer (2012)
221. Satish, N., Ravindran, K., Keutzer, K.: A decomposition-based constraint optimization approach for statically scheduling task graphs with communication delays to multiprocessors. In: Proceedings of the Conference on Design, Automation and Test in Europe, pp. 57–62. EDA Consortium (2007)
222. Sellmann, M.: Theoretical foundations of constraint programming-based Lagrangean relaxation. In: M. Wallace (ed.) Principles and Practice of Constraint Programming (CP 2004), *Lecture Notes in Computer Science*, vol. 3258, pp. 634–647. Springer (2004)
223. Sellmann, M., Fahle, T.: Constraint programming based Lagrangian relaxation for a multimedia application. In: C. Gervet, M. Wallace (eds.) Proceedings of the International Workshop on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming for Combintaorial Optimization Problems (CPAIOR 2001). Ashford, U.K. (2001)
224. Sellmann, M., Fahle, T.: Constraint Programming Based Lagrangian Relaxation for the Automatic Recording Problem. Annals of Operations Research **118**(1–4), 17–33 (2003)

225. Sellmann, M., Kliewer, G., Koberstein, A.: Lagrangian Cardinality Cuts and Variable Fixing for Capacitated Network Design. In: Proceedings of ESA, *Lecture Notes in Computer Science*, vol. 2461, pp. 845–858. Springer (2002)
226. Sellmann, M., Zervoudakis, K., Stamatopoulos, P., Fahle, T.: Crew assignment via constraint programming: Integrating column generation and heuristic tree search. Annals of Operations Research **115**, 207–225 (2002)
227. Shen, K., Schimpf, J.: Eplex: Harnessing Mathematical Programming Solvers for Constraint Logic Programming. In: Proceedings of CP, *Lecture Notes in Computer Science*, vol. 3709, pp. 622–636. Springer (2005)
228. Smith, B.M., Brailsford, S.C., Hubbard, P.M., Williams, H.P.: The progressive party problem: Integer linear programming and constraint programming compared. In: U. Montanari, F. Rossi (eds.) Principles and Practice of Constraint Programming (CP 1995), *Lecture Notes in Computer Science*, vol. 976, pp. 36–52. Springer (1995)
229. Steiger, R., van Hoeve, W.J., Szymanek, R.: An efficient generic network flow constraint. In: Proceedings of the ACM Symposium on Applied Computing (SAC), pp. 893–900 (2011)
230. Stuckey, P.J., de la Banda, M.G., Maher, M., Marriott, K., Slaney, J., Somogyi, Z., Wallace, M., Walsh, T.: The G12 project: Mapping solver independent models to efficient solutions. In: P. van Beek (ed.) Principles and Practice of Constraint Programming (CP 2005), *Lecture Notes in Computer Science*, vol. 3668, pp. 314–327. Springer (2005)
231. Taşkın, Z.C., Smith, J.C., Ahmed, S., Schaefer, A.J.: Cutting plane algorithms for solving a stochastic edge-partition problem. Discrete Optimization **6**, 420–435 (2009)
232. Tarim, S., Armagan, S., Miguel, I.: A hybrid Benders decomposition method for solving stochastic constraint programs with linear recourse. In: B. Hnich, M. Carlsson, F. Fages, F. Rossi (eds.) International Workshop on Constraint Solving and Constraint Logic Programming (CSCLP), pp. 133–148. Springer (2006)
233. Terekhov, D., Beck, J.C., Brown, K.N.: Solving a stochastic queueing design and control problem with constraint programming. In: Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI 2007), vol. 1, pp. 261–266. AAAI Press (2007)
234. Thorsteinsson, E.: Branch and check: A hybrid framework integrating mixed integer programming and constraint logic programming. In: T. Walsh (ed.) Principles and Practice of Constraint Programming (CP 2001), *Lecture Notes in Computer Science*, vol. 2239, pp. 16–30. Springer (2001)
235. Timpe, C.: Solving planning and scheduling problems with combined integer and constraint programming. OR Spectrum **24**, 431–448 (2002)
236. Tjandraatmadja, C., van Hoeve, W.J.: Target cuts from relaxed decision diagrams. INFORMS Journal on Computing (to appear)
237. Torres, P., Lopez, P.: On not-first/not-last conditions in disjunctive scheduling. European Journal of Operational Research **127**, 332–343 (2000)
238. Tran, T.T., Beck, J.C.: Logic-based Benders decomposition for alternative resource scheduling with sequence dependent setups. In: European Conference on Artificial Intelligence (ECAI), *Frontiers in Artificial Intelligence and Applications*, vol. 242, pp. 774–779. IOS Press (2012)
239. Trick, M.: A dynamic programming approach for consistency and propagation for knapsack constraints. In: C. Gervet, M. Wallace (eds.) Proceedings, Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2001), pp. 113–124. Ashford, U.K. (2001)
240. Trick, M.A.: A Dynamic Programming Approach for Consistency and Propagation for Knapsack Constraints. Annals of Operations Research **118**(1–4), 73–84 (2003)
241. Trick, M.A., Yildiz, H.: Benders cuts guided search for the traveling umpire problem. In: P. van Hentemryck, L. Wolsey (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 4510, pp. 332–345. Springer (2007)
242. Türkay, M., Grossmann, I.E.: Logic-based MINLP algorithms for the optimal synthesis of process networks. Computers and Chemical Engineering **20**, 959–978 (1996)
243. Van Hentenryck, P., Lustig, I., Michel, L., Puget, J.F.: The OPL Optimization Programming Language. MIT Press, Cambridge, MA (1999)

244. van Hoeve, W.J.: A hybrid constraint programming and semidefinite programming approach for the stable set problem. In: F. Rossi (ed.) Principles and Practice of Constraint Programming (CP 2003), *Lecture Notes in Computer Science*, vol. 2833, pp. 407–421. Springer (2003)
245. van Hoeve, W.J.: A hyper-arc consistency algorithm for the soft alldifferent constraint. In: M. Wallace (ed.) Principles and Practice of Constraint Programming (CP 2004), *Lecture Notes in Computer Science*, vol. 3258, pp. 679–689. Springer (2004)
246. van Hoeve, W.J., Pesant, G., Rousseau, L.M., Sabharwal, A.: Revisiting the sequence constraint. In: F. Benhamou (ed.) Principles and Practice of Constraint Programming (CP 2006), *Lecture Notes in Computer Science*, vol. 4204, pp. 620–634. Springer (2006)
247. Veinott, A.F., Wagner, H.: Optimal capacity scheduling I. Operations Research **10**, 518–532 (1962)
248. Wallace, M., Novello, M.S., Schimpf, J.: ECLiPSe: A platform for constraint logic programming. ICL Systems Journal **12**, 159–200 (1997)
249. Williams, H.P., Yan, H.: Representations of the all_different predicate of constraint satisfaction in integer programming. INFORMS Journal on Computing **13**, 96–103 (2001)
250. Xia, Q., Eremin, A., Wallace, M.: Problem decomposition for traffic diversions. In: J.C. Régin, M. Rueher (eds.) CPAIOR Proceedings, *Lecture Notes in Computer Science*, vol. 3011, pp. 348–363. Springer (2004)
251. Yan, H., Hooker, J.N.: Tight representations of logical constraints as cardinality rules. Mathematical Programming **85**, 363–377 (1995)
252. Yunes, T.H., Aron, I., Hooker, J.N.: An integrated solver for optimization problems. Operations Research **58**, 342–356 (2010)
253. Yunes, T.H., Aron, I., Hooker, J.N.: An integrated solver for optimization problems. Operations Research (to appear)
254. Yunes, T.H., Moura, A.V., de Souza, C.C.: Exact solutions for real world crew scheduling problems (1999). Presentation at INFORMS national meeting, Philadelphia
255. Yunes, T.H., Moura, A.V., de Souza, C.C.: Hybrid column generation approaches for urban transit crew management problems. Transportation Science **39**, 273–388 (2005)