ISSN 2231-346X (Print)

JUSPS-A Vol. 35(2), 12-21 (2023). Periodicity-Monthly

Section A

ISSN 2319-8044 (Online)

JOURNAL OF ULTRA SCIENTIST OF PHYSICAL SCIENCES

An International Open Free Access Peer Reviewed Research Journal of Physical Sciences

website:- www.ultrascientist.org

Estd. 1989

Constraint programming have been identifying as promising technique for efficiently solving discrete optimization problem

AKASH PANDEY[1] and UMESH KUMAR GUPTA[2]

[1]Assistant professor; M.G.P.G. College Gorakhpur, U.P. (India)
[2]Professor; M.G.P.G. College Gorakhpur, U.P. (India)

Corresponding Author E.mail: akashpandey38@gmail.com

http://dx.doi.org/10.22147/jusps-A/350201

Acceptance Date 11th March, 2023, Online Publication Date 4th April, 2023

Abstract

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. To make constraint programming material to practical problems one needs propagation algorithms that are both viable and proficient. The most incredible propagation algorithm for the alldifferent constraint, i.e. the one getting hyper-circular segment consistency, is to be sure extremely productive. The reason is that we can apply matching hypothesis from operations research. Likewise for the symmetric alldifferent constraint and the weighted alldifferent constraint powerful and effective propagation algorithms exist, again dependent on techniques from operations research. From this paper we show that some time alldifferent constraint is more easy way to solve the MIP. For this we will choose an auction strategy by which a company get more revenue. Despite this considerable progress, there remains great potential for further integration, with the concomitant improvement in both modeling and solution method.

Key word : Constraint programming, alldifferent constraint, value of graph.

Introduction

The field of constraint programming is moderately new; the primary global workshop on "Standards and Practice of Constraint Programming" was held in 1993, while it turned into a gathering in 1995. The essential ideas of constraint thinking were developed in the field of man-made brainpower during the 1970s. Further development occurred after the presentation of constraints in logic programming during the 1980s. In spite of the fact that Operations Research (OR) and Constraint Programming (CP) have diverse roots, the connections between the two communities have become more grounded as of late. For tackling combinatorial improvement problems, the techniques of CP as well as will turn out to be interdependent to the point that the two research communities could inevitably blend. Constraint programming and Operation research 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.

Operation research is strongly influenced by its historical roots in linear programming, which formulates problems using inequality constraints. Much of the field today is based on inequality-constrained mathematical programming model, including those of nonlinear programming (NLP), mixed integer/linear programming (MILP), and mixed integer/nonlinear programming.

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. In the context of constraint programming, decision diagrams provide a generic tool of modeling and propagation constraints and conjunction constraints[1,2,3,4] in context of integer programming, recent examples include the use of decision diagram for generating cutting planes and for representing nonlinear objective function[5].

The aim of present work is to give the idea is how to solve the MIP by using *alldifferent* constraint.

Constraint programming and constraint satisfaction :

Constraint programming can be imagined by and large as the implanting of constraints inside a programming dialect. This mix of decisive and procedural modeling gives the client some power over how the problem is comprehended, even while holding the capacity to state constraints definitively.

It a long way from evident how definitive and procedural plans might be consolidated. In a procedural code, for instance, usually to dole out a variable diverse quality at different focuses in the code. This is rubbish in a definitive detailing, since it is conflicting to state constraints that allot a similar variable diverse quality. The developments that offered ascend to constraint programming can in substantial part be viewed as endeavors to address this problem. They started with logic programming and prompted various elective approaches, for example, constraint taking care of tenets, simultaneous constraint programming, constraint logic programming, and constraint programming. Constraint programming "toolboxes" speak to a to some degree more procedural variant of constraint logic programming and are maybe the most broadly utilized option.

Operations research methods in constraint programming :

Various operations research (OR) methods have discovered their way into constraint programming (CP). This development is totally normal, since OR and CP have comparative objectives. Or then again is basically a minor departure from the scientific routine with regards to mathematical modeling. It depicts wonders in a formal dialect that enables one to derive outcomes thoroughly. In contrast to a run of the mill scientific model, be that as it may, anOR model has a prescriptive and in addition a spellbinding reason. It speaks to a human movement with some opportunity of decision, as opposed to a characteristic procedure. The laws of nature move toward becoming constraints that the action must watch, and the objective is to amplify some objective subject to the constraints.

CP's constraint-situated approach to problem tackling represents a prescriptive modeling undertaking fundamentally the same as that of OR. CP truly has been less worried about finding ideal than achievable solutions; however, this is a shallow contrast. It is not out of the ordinary, in this way, that OR methods would discover application in understanding CP models.

There remains a major distinction, be that as it may, in the way that CP as well as comprehend constraints. CP regularly observes a constraint as a strategy, or if nothing else as summoning a methodology, that works on the solution space, typically by diminishing variable domains. Or on the other hand observes a constraint set overall fabric; the solution algorithm works on the whole problem instead of the constraints in it. The two approaches have their points of interest. CP can configuration specific algorithms for individual constraints or subsets of constraints, consequently misusing substructure in the problem that OR methods are probably going to miss. Or then again algorithms, then again, can misuse worldwide properties of the problem that CP can just mostly catch by propagation through factor domains.

Constraint propagation were foreshadowed in the OR literature as early as the 1960s, when Garfinkel and Nemhauser[6] used a technique then known as implicit enumeration to solve integer programming model 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 [7], while true integration began in the 1990s. OR took an early step away from inequality-based modeling in1990, when Beaumont[9] replaced integer variables with logical disjunctions and solved the problem by branching on disjunction. after an early allocation to processing networks in[9], Hooker and Osorio[10] extended this approach to "mixed logical/linear programming". Meanwhile, the integration of propagation and relaxation in branch-and-cut methods was advocated by Hooker in1994 [11] and further explored in a number of publications, such as[12,13]. The CP community also pursued integrated methods during the 1990s, primarily by exchanging information between CP an LP solver, as advocated by Little and Darby- Dowman[14].

Mixed integer/linear modeling :

A mixed integer/linear programming (MILP) problem is a LP problem with the additional confinement that specific factors must take integer esteems. It is an (unadulterated) integer/linear programming (ILP) problem when every one of the factors are integer-esteemed, and a 0-1 linear programming problem when every one of the factors have domain {0,1}.

MILP problems are tackled by a branch-and-bound search mechanism. A LP relaxation of the problem is illuminated at every hub of a search tree. On the off chance that the ideal estimation of the relaxation is more prominent than or equivalent to the estimation of the best applicant solution found up until this point. the search backtracks. Something else, in the event that all factors in the LP solution are indispensable, it becomes a competitor solution. On the off chance that at least one factors are nonintegral, the search branches on one of the nonintegral factors by part its domain. Cutting planes are ordinarily included at the root hub and potentially at different hubs, bringing about a branch-and-cut method.

In spite of the fact that MILP problems are commonly a lot harder to take care of that LP problems, the solution innovation has been the subject of serious development for no less than three decades. Business solvers have accomplished requests of-greatness speedups through the privilege com-bination of cutting planes, branching heuristics, and preprocessing.

The essential job of MILP in CP, be that as it may, is to give a LP relaxation of a constraint or subset of constraints. One formulates a MILP model and drops the integrality condition. MILP is an exceptionally flexible modeling dialect in the event that one is adequately brilliant. Composing a model with a "decent" LP relaxation, be that as it may, is frequently more a craftsmanship than a science. A decent relaxation is for the most part seen as one whose ideal esteem is near that of the MILP problem.

Operation research besed filtering model (Alldifferent constraint) :

Operation Research methods have made major contributions to domain filtering for global constraints in CP. important example is alldifferent constraint. The alldifferent constraint first appeared in 1978 [15]. Filtering algorithem that achieve domain consistency for alldifferent were derived in the early 1990s using results from matching theory in the OR literature[16,17,18,19], which is in turn based on classical network flow theory.

Definition (Alldifferent constraint). Let $\boldsymbol{x_1, x_2, \ldots, x_n}$ *be variables* with respective finite domains $D_1, D_2, \ldots\ldots, D_n$ Then

$$alldifferent(x_1, \ldots\ldots, x_n) = \{(d_1, \ldots\ldots d_n) \mid d_1 \in D_i, d_i \neq d_j \ for \ i \neq j\}$$

Definition (Value graph).: Let $\boldsymbol{X = x_1, x_2, \ldots\ldots, x_n}$ *be a sequence of variables with respective finite domains* $\boldsymbol{D_1, D_2, \ldots\ldots, D_n}$. *The bipartite graph* $\boldsymbol{G = (X \cup D_X, E)}$ *with* $\boldsymbol{E = \{x_i d \mid d \in D_i\}}$ *is called the value graph of X.*

Theorem.1 Let $\boldsymbol{X = x_1, x_2, \ldots\ldots, x_n}$ *be a sequence of variables with re-spective finite domains* $\boldsymbol{D_1, D_2, \ldots\ldots, D_n}$ *Let G be the value graph of X. Then*
$(\boldsymbol{d_1, d_2, \ldots\ldots, d_n}) \in alldifferent\ (\boldsymbol{x_1, \ldots\ldots, x_n})$ if and only if $M = \{x_1 d_1, \ldots\ldots, x_n d_n\}$ is a matching in G.

Proof. An edge x_i, d_i (for some $i \in \{1, \ldots\ldots, n\}$) in *M* corresponds to the assignment $x_i = d_i$ As no edges in *M* share a vertex, $x_i \neq x_j$ for all $i \neq j$

Note that the matching *M* in Theorem 1 covers *X*, and is therefore a maximum-size matching.

Example : We want to assign four tasks (1, 2, 3 and 4) to five machines (A, B, C, D and E). To each machine at most one task can be assigned. However, not every task can be assigned to every machine. Table 1 below presents the possible combinations. For example, task 2 can be assigned to machines B and C.

Table 1. Possible task - machine combinations.

Task	Machines
1	B, C, D, E
2	B, C
3	A, B, C, D
4	B, C

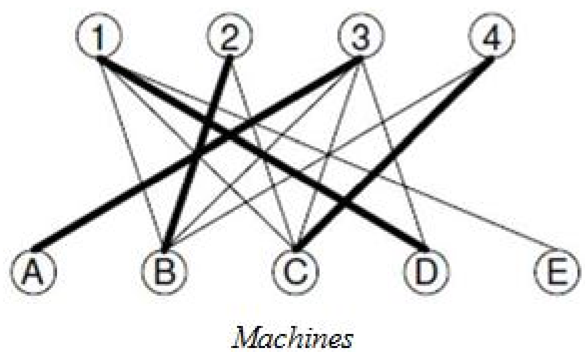

Fig. 1. The value graph for the task assignment problem of Example1.
Bold edges form a matching covering all tasks.

This problem is modelled as follows. We introduce a variable for task . whose value represents the machine to which task is assigned. The initial domains of the variables are defined by the possible combinations in Table 1. Since the tasks have to be assigned to different machines, we introduce an alldifferent constraint. The problem is thus modelled as the CSP.

$$x_1 \in \{B, C, D, E\}, x_2 \in \{B, C\}, x_3 \in \{A, B, C, D\}, x_4 \in \{B, C\}$$
$$\boldsymbol{alldifferent}(x_1, x_2, x_3, x_4).$$

Application of LPP model and solution by alldifferent constraint :

There are five parties X_1, X_2, X_3, X_4, X_5 they want to bid on 4 item A, B, C, D. The party X_1 have 10 crore and it is interested to bid on A, B. Party X_2 have 20 cr. And it is interested to bid on A, C. Party X_3 have 30 cr. And it is interested to bid on BD. Party X_4 have 40 crores to bid on BCD. Party X_5 have 14 crores to bid on A. We wish to sell 4 items (maximize revenue), given these 4 bids. One thing should remember that if any party bids on more than one item then bid amount must be divided equally for each biding item.

Table 2.

ITEM	BID AMOUNT	MIP VARIABLE
A, B	10	X_1
A, C	20	X_2
B, D	30	X_3
B, C, D	40	X_4
A	14	X_5

Then lpp of following bid

$$Max\ Z = 10X_1 + 20X_2 + 30X_3 + 40X_4 + 14X_5$$

$$\text{St.} \quad X_1 + X_2 + X_5 \leq 1$$
$$X_1 + X_3 + X_4 \leq 1$$
$$X_2 + X_4 \leq 1$$
$$X_3 + X_4 \leq 1$$
$$X_1, X_2, X_3, X_4, X_5 \text{ are binary}$$

To apply branch and bound method, the following 5 constraints have to be added to the LP model,

$$X_1 \leq 1$$
$$X_2 \leq 1$$
$$X_3 \leq 1$$
$$X_4 \leq 1$$
$$X_5 \leq 1$$

We can write the solution step by Big M method

$$Max\ Z = 10X_1 + 20X_2 + 30X_3 + 40X_4 + 14X_5$$
$$X_1 + X_2 + X_5 \le 1$$
$$X_1 + X_3 + X_4 \le 1$$
$$X_2 + X_4 \le 1$$
$$X_3 + X_4 \le 1$$
$$X_1 \le 1$$
$$X_2 \le 1$$
$$X_3 \le 1$$
$$X_4 \le 1$$
$$X_5 \le 1$$

And $X_1, X_2, X_3, X_4, X_5 \ge 0$

The solution of above is $MaxZ_A = 54(X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 1, X_5 = 1)$ and $Z_L = 54(X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 1, X_5 = 1)$ obtain by the rounded off solution values. This problem has integer solution so no further branching is required. The branch and bound diagram

$$X_1 = 0 \quad X_2 = 0 \quad X_3 = 0 \quad X_4 = 0 \quad X_5 = 0$$
$$Z_A = 54$$
$$Z_L = 54$$

Solution step by Big M method

The following lpp can be written in following constrain programming

Data Structures

- Domain types (Integer, Real)
- Domain implementation Constraint Propagation

AllDifferent Variables must be assigned distinct value

$$X_1 \in \{A, B\}$$
$$X_2 \in \{A, C\}$$
$$X_3 \in \{B, D\}$$
$$X_4 \in \{B, C, D\}$$
$$X_5 \in \{A\}$$

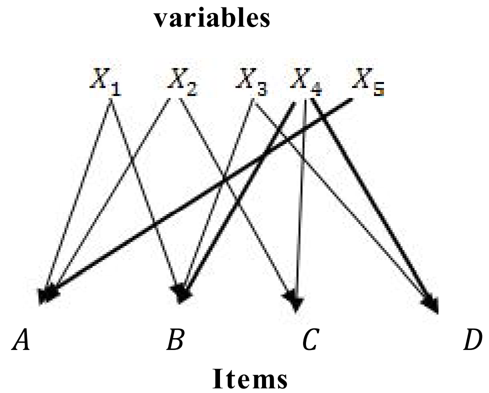

Fig 2. The value graph for the task assignment of item to the bidder (variable). Bold edges form a matching covering

Variable	X_1 ,	X_2,	X_3,	X_4	X_5
Domain	D_1	D_2	D_3	D_4	D_5
	(A, B)	(A, C)	(B, D)	(B, D, C)	(A)

The constraint satisfaction can be written as:

$$C_1((X_1, X_5) \qquad (X_5 = A \neq X_1))$$

$$C_2((X_2, X_3, X_4) \qquad (X_4 = B, C, D \neq X_2 \neq X_3))$$

It is clear that the value graph of X= X= X_1 , X_2, X_3, X_4, X_5 in presented in fig. The bold edge in the value graph denoted a matching covering X. It corresponds to a solution to the CSP

i.e., $X_4 = B, C, D \quad X_5 = A$

this shows that party X_4 will get item B, C, D at the bid amount 40cr and party X_5 will get item A at the bid amount 14 cr.
Thus, we saw that by selling these 4 items by biding we get total revenue 54 crore.

Conclusion

An imperative perception is the accompanying. To make constraint programming material to practical problems one needs propagation algorithms that are both viable and proficient. The most incredible propagation algorithm for the alldifferent constraint, *i.e.* the one getting hyper-circular segment consistency, is to be sure extremely productive. The reason is that we can apply matching hypothesis from operations research. Likewise for the symmetric alldifferent constraint and the weighted alldifferent constraint powerful and effective propagation algorithms exist, again dependent on techniques from operations research.

In this paper, it is presented that determines optimum bids in a competitive-bidding situation where each competitor submits one closed bid. the number of bidders may be large or may be unknown.

Competitive bidding is a common procurement practice that involves inviting multiple vendors or service provider to submit offer for any particular material or services. Competitive bidding allows transparency, equality of opportunity and the ability to demonstrate that the outcomes represent the best value hence, high value acquisitions usually undergo the competitive bidding process.

From this paper we show that some time alldifferent constraint is more easy way to solve the MIP. For this we will choose an auction strategy by which a company get more revenue. Despite this considerable progress, there remains great potential for further integration, with the concomitant improvement in both modeling and solution method.

scope of future work:

This research can be extended in the future by considering additional parameters and constraints related to the real world. This will lead to developing a mathematical model for nonlinear integrated production planning and scheduling problem, imple- menting the developed model on a real-world problem, and then solving the actual industrial problem with the suitable multi- objective optimization method.

References

1. David Bergman, Andre A. Cire (2018) Discrete Nonlinear Optimization by State-Space Decompositions. Management Science *64(10):* 4700-4720.
2. 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).
3. Perez, G., Regin, J.C.; Efficient Operations on MDDs for Building Constraint Programming Models In: Proceedings of IJCAI, pp. 374-380 (2015).
4. Perez, G., Regin, J.C.; MDDs are Efficient Modeling Tool: An Application to Some Statistical Constraints. CPAIOR, LNCS Vol 10335, pp 30-40. Springer (2017).
5. Begman , D., Cire, A.A., van Hoeve,W.J.: Lagrangian bounds from decision diagrams. Constraints *20,* 346-361 (2015).
6. Garfinkel, R., Nemhauser, G.L.: Optimal political districting by implicit enumeration techniques. Management Science *16,* B495-B508 (1970).
7. 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).
8. Beaumont, N.: An algorithm for disjunctive programs. European Journal of Operational Research *48,* 362-371 (1990).
9. 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).

10. Hooker, J.N., Osorio, M.A.: Mixed logical/linear programming. Discrete Applied Mathematics 96-97, 395-442 (1999).
11. 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,* pp336-349. Springer (1994).
12. 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).
13. Hooker J.N.: Logic -based methods for optimization: Combining Optimization and Constraint Satisfaction. Wiley, New York (2000).
14. 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).
15. Lauriere, J.L.: A language and a program for stating and solving combinatorial problem. Artificial Intelligence *10,* 29-127 (1978).
16. Costa, M.C.: Persistency in maximum cardinality bipartite matchings. Operations Research Letters *15,* 143-149 (1994).
17. Lovasz, L., Plummer, M.: Matching Theory, Annals of discrete Mathematics, vol. *29.* North-Holland (1986).
18. Hopcroft, J.E., Karp, R.M,: A $n^{5/2}$ algorithm for maximum matching in bipartite graph. SIAM Journal on Computing *2,* 225-231 (1973).
19. Pandey A, Sharma S.S, Gupta U.K,: An approach to operation research techniques and constraint programming, IJIRT Vol. *8,* pp. 525-535 (2021).