Technical Communications of the International Conference on Logic Programming, 2010 (Edinburgh), pp. 286–288
http://www.floc-conference.org/ICLP-home.html

LOCAL BRANCHING IN A CONSTRAINT PROGRAMMING FRAMEWORK

FABIO PARISINI

DEIS, University of Bologna,
V.le Risorgimento 2, 40136, Bologna, Italy
E-mail address: fabio.parisini@unibo.it

Local branching is a general purpose heuristic method which searches locally around the best known solution by employing tree search. It has been successfully used in Mixed Integer Programming (MIP) where local branching constraints are used to model the neighborhood of an incumbent solution and improve the bound.

The neighborhoods are obtained by linear inequalities in the MIP model so that MIP searches for the optimal solution within the Hamming distance of the incumbent solution. The linear constraints representing the neighborhood of incumbent solutions are called local branching constraints and are involved in the computation of the problem bound. Local branching is a general framework to effectively explore solution subspaces, making use of the state-of-the-art MIP solvers.

The local branching framework is not specific to MIP: it can be integrated in any tree search strategy. In our research work we propose the integration of the local branching framework in Constraint Programming (CP).

The local branching technique, as introduced in [Fis03], is a complete search method designed for providing solutions of better and better quality in the early stages of search by systematically defining and exploring large neighborhoods. On the other hand, the idea has been used mainly in an incomplete manner since [Fis03]: linear constraints defining large neighborhoods are iteratively added and the neighborhoods are explored, generally in a non-exhaustive way. When this is done within a local search method, the overall algorithm follows the spirit of both *large neighborhood search* [Sha98] and *variable neighborhood search* [Mla97]. The main peculiarity of local branching is that the neighborhoods and their exploration are general purpose.

Integration of local search and CP aided tree search has been long advocated in the literature. See for instance Chapter 9 in [Mil03] and more recently the use of propagation within a large neighborhood search algorithm [Per04], the use of local search to speed up complete search [Sel06] and Beck's *solution-guided multi-point constructive search* [Bec07]. The technique which resembles most to our work is the latter which makes use of the existing solutions to guide the search.

We argue that integrating local branching in CP merges the advantages of the intensification and diversification mechanisms specific to local search methods, with constraint propagation that speeds up the neighborhood exploration by removing infeasible variable value assignments.

Key words and phrases: Local Branching, LDS, Local Search, Tree Search, Constraint Programming.

Technical Communications of the 26th International Conference on Logic Programming, Edinburgh, July, 2010
Editors: Manuel Hermenegildo, Torsten Schaub
LIPIcs - Leibniz International Proceedings in Informatics. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany
Digital Object Identifier: 10.4230/LIPIcs.ICLP.2010.286

Integrating local branching in CP is not simply a matter of implementation but instead requires significant extensions to the original search strategy. The main extensions to the traditional MIP approach we have developed follow:

- First, using a linear programming solver for computing the bound of each neighborhood is not computationally affordable in CP. We have therefore studied a lighter way to compute the bound of the neighborhood which is efficient, effective and incremental, using the additive bounding technique.
- Second, we developed a cost-based filtering algorithm for the local branching constraint by extracting reduced-costs out of additive bounding.
- Third, we have studied a CP-tailored diversification technique that can push the search arbitrarily far from the current incumbent solution. This technique allows us to explore large sections of a big diversification space, using CP-specific modeling elements.

All these aspects have been thoroughly tested on a set of instances of the Asymmetric Traveling Salesman Problem with Time Windows [Asc95], having as first term of comparison a pure CP approach, using the same model and cost based filtering as our CP local branching implementation. Our experimental results demonstrate the practical value of integrating local branching in CP. The results can be summarized as follows: (i) on small-size instances where pure CP proves optimality, we find the optimal solution in a shorter time and prove optimality quicker, (ii) on medium-size instances, we can prove optimality where CP fails, (iii) on large-size instances, where both methods fail to prove optimality, we obtain a better solution quality within the same time limit. Moreover, we obtain even better results when compared with Limited Discrepancy Search (pure and enriched with bound computation) and with pure local search.

The first results are encouraging, but much research work still has to be done. In particular, the diversification technique we have developed for escaping the local minima of Local Branching is very promising and its study is just at an initial phase. This technique works on the best solution found during Local Branching iterations, named as reference solution, and performs a diversification search by explicitely setting difference constraints on a subset of the problem variables. For example, if we have the reference solution $\bar{x}$ having values $\bar{x_1} = 4, \bar{x_2} = 2, \bar{x_3} = 5, \bar{x_4} = 1, \bar{x_5} = 2$ we can perform diversification by simply setting constraints like $x_2 \neq 2, x_3 \neq 5, x_5 \neq 2$ and executing a normal CP search using the reduced variables domains.

This diversification approach has a very strong potentiality for CP Based Local Branching, as the capability of effectively setting difference constraints is something natural in Constraint Programming framework, while it is not in MIP; using this kind of diversification approach exploits CP specific peculiarities, still mantaining the Local Branching framework completely general. We are also working on the definition of problem independent criteria to help the selection process of the variables to set difference constraints for (currently the best results are given by random choices); tests needs to be done to understand how many constraints is better to set on different kinds of problem instances.

Moreover a similar approach can be used for intensification processes, by using a reference solution $\bar{x}$ as guide and setting equality constraints of the kind $x_i = \bar{x_i}$ on a subset of variables to explore the neighborhood of $\bar{x}$; we can easily explore portions of the search space which are quite far in terms of discrepancy from the reference solution $\bar{x}$, reaching discrepancy values that we could never reach with Limited Discrepancy Search (LDS) [Har95].

In practice, when dealing with a problem instance modeled by n domain variables, we can explore portions of the search space up to a discrepancy value of k by setting $n-k$ equality constraints and performing a normal CP search on the remaining k variables, having a much reduced search space.

It is immediate to observe that both the intensification and the diversification techniques that we are studying are based on strong randomization elements, i.e. the choice of the variables to set constraints for; this is absolutely normal in a setting where we give up on performing a complete search to obtain the optimal solution because of the extreme complexity of the problem instances.

The intensification and diversification techniques that we have briefly outlined above must be carefully studied and tested over real problem instances. Many parameters have to be tuned, like the number of equality or difference constraints to set, the relation of this number with the problem size, whether it is opportune to set both equality and difference constraints together, but above all if there is an effective and general way to select the best variables to set constraints for. This kind of work is very interesting as intensification and diversification techniques are basic elements of many search techniques, first of all in the neighborhood and diversification searches within Local Branching, but in general any time we have to perform a tree search in CP.

References

[Asc95] N. Ascheuer. *Hamiltonian path problems in the on-line optimization of flexible manufacturing systems.* Ph.D. thesis, Technische Universität Berlin, 1995.

[Bec07] J. C. Beck. Solution-guided multi-point constructive search for job shop scheduling. *J. Artif. Intell. Res. (JAIR)*, 29:49–77, 2007.

[Fis03] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98:23–47, 2003.

[Har95] W. Harvey and M.L. Ginsberg. Limited discrepancy search. In *Proc. of IJCAI-95*, pp. 607–615. Morgan Kaufmann, 1995.

[Mil03] M. Milano. *Constraint and Integer Programming: Toward a Unified Methodology.* Kluwer Academic Publishers, 2003.

[Mla97] N. Mladenovic and P. Hansen. Variable neighbourhood search. *Computers and Operations Research*, 24:1097–1100, 1997.

[Per04] L. Perron, P. Shaw, and V. Furnon. Propagation guided large neighbourhood search. *Proc. of CP-04, LNCS*, 3258:468–481, 2004.

[Sel06] M. Sellmann and C. Ansotegui. Disco - novo - gogo: integrating local search and complete search with restarts. In *Proc. of AAAI-06*, pp. 1051–1056. AAAI Press, 2006.

[Sha98] P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. *Proc. of CP-98, LNCS*, 1520:417–431, 1998.