

Intelligent Domain Splitting for CSPs with Ordered Domains

Marius-Călin Silaghi, Djamila Sam-Haroud, and Boi Faltings

Swiss Federal Institute of Technology, Artificial Intelligence Lab, DI-LIA,
CH-1015 Lausanne, Switzerland
{silaghi, haroud, faltings}@lia.di.epfl.ch

This paper presents intelligent domain splitting, an approach for searching in CSPs with ordered domains. The technique has a particular strength for finding all solutions. It represents the search space by aggregations obtained through the cooperation of two known clustering concepts, namely intervals and cross-products. Intelligent domain splitting searches for solutions by iteratively breaking up the space into smaller subspaces in a meaningful way. The proposed backtracking technique benefits from the strengths of Hull-consistency (i.e. 2B-consistency [4]) and of the Cartesian product representation [2]. The algorithm can be applied to general systems of constraints with explicit representations. Even though designed for generating all solutions, it also proves useful for finding the first solution of hard problems, as shown by preliminary experiments.

Albeit the technique we present is general, we mainly focus on the applications where the order of the domains is given and induces a natural grouping of the feasible tuples within the explicit constraint representations (the legal entries tend to be consecutive in the matrix representation of the constraints). This is typical in integer programming and discrete numerical applications.

Informally, the idea is then to cover each constraint by entirely feasible boxes, this means that each box contains only legal tuples. The boxes are represented concisely as sets of intervals and are used as meta-values during search. In ordered domains, the simplest way to construct the covering is by hierarchical binary splitting, as usually done for numerical continuous applications. The domains are recursively cut in the middle until completely feasible/infeasible boxes (sub-matrices) are found. Figure 1 illustrates the aforementioned notions: a) an integer numerical CSP with two linear constraint (dashed lines), the shadowed regions represent legal entries, b) the box covering obtained by binary split.

Although this procedure constructs entirely feasible boxes, the aggregation mechanism provided remains uninformed since it does not take into account the characteristics of the constraints. This mechanism has therefore great chance to split just in the middle of contiguous feasible regions, requiring further additional splits to take place. In this work, we propose an intelligent splitting mechanism that tries to reduce the search space by choosing more appropriate splitting points (discontinuities/limits of the constraint space), see Figure 1.c).

The search space is generated as Cartesian products of intervals. From this hybrid continuous/discrete representation stems the idea of interleaving search with Hull-consistency(HC), a powerful local consistency technique commonly used for numerical constraints. Although weaker than AC in general, HC brings

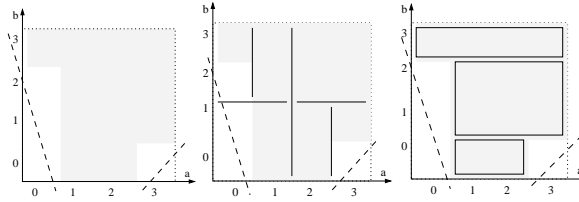


Fig. 1. a) the constraints between a and b are $b \geq 3 - 3a \wedge b > a - 3$; b) binary split; c) intelligent split

similar pruning when the feasible entries tend to be consecutive. The space needed reduces however from $O(nd^2)$ for AC to $O(n)$ for HC, where n is the number of variables and d the size of domains.

A central difference with the previous works on Cartesian products representations is that we decide the dynamical aggregation and domain splitting after each future variable is pruned. At each step, we apply incremental MHC (Maintaining HC). This technique leads to earlier detection of domain wipe-out.

The backtracking algorithm implementing intelligent splitting is called DCPR and its enhancement with hull-consistency, MHC-DCPR. Preliminary experiments were run to compare MHC-DCPR and DCPR against MHC and conventional backtracking on 400'000 instances of hard 3- and 4-coloring problems. Exponential improvements of the speed, as function of the number of variables, was noticed for both obtaining the first solution and all solutions with MHC-DCPR. Similar results were obtained for the compression of the spatial representation when all the solutions were generated.

As to related work, an idea similar to our box-covering was presented in [1] for the general case. However, the decomposition proposed was static, being done as a preprocessing. Our algorithm performs dynamic covering, similarly to what is done in [2] with a worst case complexity of $O(ad^a)$, where a is the maximal arity (vs. $O(d^a)$ for [1]). It is however completely depth first with less structure management, promising better average costs for ordered domains. An idea close to our intelligent splitting was introduced in [3]. In that work, partitioning involves only future variables and instantiation is performed value by value during search.

References

1. A. Haselböck. Exploiting interchangeabilities in constraint satisfaction problems. In *Proceedings of IJCAI'93*, pages 282–287, 93.
2. P. D. Hubbe and E. C. Freuder. An efficient cross product representation of the constraint satisfaction problem search space. In *AAAI-92, Proceedings*, pages 421–427. AAAI, July 92.
3. J. Larrosa. Merging constraint satisfaction subproblems to avoid redundant search. In *Proceedings of IJCAI'97*, pages 424–429, 97.
4. O. Lhomme. Consistency techniques for numeric CSPs. In *Proceedings of IJCAI'93*, pages 232–238, 93.