Neighborhood Interchangeability and Dynamic Bundling for Non-binary CSPs

Anagh Lal
University of Nebraska - Lincoln, alal@cse.unl.edu

Berthe Y. Choueiry
University of Nebraska - Lincoln, choueiry@cse.unl.edu

Eugene C. Freuder
University College Cork, e.freuder@cs.ucc.ie

Follow this and additional works at: http://digitalcommons.unl.edu/cseconfwork

Part of the Computer Sciences Commons

http://digitalcommons.unl.edu/cseconfwork/168

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Conference and Workshop Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.
Neighborhood Interchangeability and Dynamic Bundling for Non-binary CSPs
Anagh Lal and Berthe Y. Choueiry
Constraint Systems Laboratory • University of Nebraska-Lincoln

1. Interchangeability: An algorithm for computing interchangeability in non-binary CSPs.
2. Dynamic bundling: Integration of the above with backtrack search for solving non-binary CSPs.

Contributions

1. Interchangeability
2. Dynamic Bundling

Dynamic Bundling for Non-Binary CSPs
Dynamic bundling (DynBndl) was thought to be an overkill. We show DynBndl is worthwhile for:
- Finding all solutions: theoretically best
- Finding first solutions: empirical evidence

Because DynBndl:
- Bundles solutions
- Bundles no-goods (i.e., bundles of inconsistent partial solutions).

Experiments:
- FC: First Bundle Size
- CPU time, number of nodes visited

Varying tightness

- Large FBS: 33 at t = 0.35
- Small bundling overhead.

Future Research Directions

- Use new join algorithm when materializing join queries.
- Exploit bundled results in data-analysis/data-mining packages.
- Assist in query size estimation
- Improve accuracy of sampling operators.

Constraint Satisfaction Problems

A Constraint Satisfaction Problem (CSP) is a combinatorial decision problem defined by a set of variables, a set of domain values for these variables, and a set of constraints restricting the combinations of values for variables, where the task is to find a solution (i.e., an assignment of a value to each variable satisfying all constraints), or to find all such solutions.

Interchangeability & Bundling

Interchangeability

- Identifies values equivalent in all solutions of a CSP [2].
- Full Interchangeability (FI): d, e, and f can be swapped for V in any solution.
- Neighborhood Interchangeability (NI): finds e and f but misses d. It efficiently approximates FI.

Bundling = Search + neighborhood interchangeability [3]
Dynamic bundling = Search + dynamic NI [1]

Interchangeability in non-binary CSPs

We show how to compute NI for non-binary constraints by:
1. Building the non-binary discrimination tree, nb-DT(V, C), a data-structure that determines the NI sets of a variable V given a constraint C defined on V.
2. Intersecting the NI sets from the nb-DTs of a set of constraints, which yields the domain partition of the variable given the constraints.

Experiments

- Tested random CSPs, Model B, 1000 instances per sample
- Criteria: FBS (First Bundle Size), CPU time, number of nodes visited (NV), and number of constraint checks.
- Statistical tools: ANOVA and t-distribution for confidence intervals.

Sorting-based bundling algorithm
- Partitions domains in a memory efficient manner.
- Fits into the iterator model of databases and produces one bundle at a time.

Sort-merge join algorithm based on dynamic bundling
- Reduces number of tuples compared in the main memory.
- Is memory efficient and produces compacted results, saving I/O for the next operator and disk space (and network bandwidth in distributed databases).

References