Toward a Generic Comet Implementation of Very Large-Scale Neighborhoods

Sébastien Mouthuy and Yves Deville
Department of Computing Science and Engineering
Université catholique de Louvain
2, place Sainte Barbe
B-1348 Louvain-la-Neuve, Belgium
e-mail: {sebastien.mouthuy,yves.deville}@uclouvain.be

Pascal Van Hentenryck
Department of Computer Science
Brown University
Box 1910
Providence, RI 02912, USA
email: pvh@cs.brown.edu

January 14, 2008

Abstract

We describe here preliminary results of our Comet implementation of generic abstractions for Very Large-Scale Neighborhoods. These neighborhoods are used in local search. Their size is exponential but the best neighbor can be computed in polynomial time. They have proven to be very efficient on a large set of problems. However designing such neighborhoods requires high development efforts. Our objective is to design VLSN abstractions and to implement them in the high-level language Comet in order to provide an experimental platform for studying Very Large-Scale Neighborhoods. This will allow a user to easily define such neighborhood in a high-level paradigm, without the requirement of implementing a dedicated algorithm from scratch.

Keywords: Local Search, Very Large-Scale Neighborhood, TSP

Introduction

Neighborhood search (or local search) is a wide-recognized technique for solving hard combinatorial optimization problems. It has proven to be very efficient for many hard problems and is appealing by the simplicity of its concepts. A neighborhood search algorithm begins with an initial solution of the problem to solve and tries iteratively to obtain a better solution by considering, for each solution, a *neighborhood*. The Neighborhood of a solution is usually defined as the set of solutions that one can obtain by slightly modifying the current solution.

Comet [1] is an object-oriented programming language for constraint-based local search. It features advanced modeling and control abstractions to simplify the design and implementation of neighborhood search algorithms, including invariants, differentiable events, events, and non-determinism. Comet's underlying computational paradigm is Constraint-Based Local Search, the idea of specifying local search algorithms as two components: a high-level model describing the applications in terms of constraints, constraint combinators, and objective functions; a search procedure expressed in terms of the model at a high abstraction level.

Very Large-Scale Neighborhood [2, 3] is a promising direction of research in local search. In neigborhood search, it is well admitted that the bigger the size of the neighborhood considered, the better the quality of the best solution found. However, considering a bigger neighborhood does require time to find the best neighbor, so one needs to find a good trade-off between the quality of the best solution in the neighborhood and the time required to find this best solution. Very large-scale neighborhoods (VLSN) are neighborhoods of large size (usually exponential in the size of the problem) and whose structure is such that the best neighbor can be found very efficiently. One example of such VLSN for the Traveling Salesman Problem (TSP) is given in

[4], where the neighborhood of a permutation π is given by $\{\pi \circ \sigma : \sigma \in S^n$ and σ is pyramidal$\}$. A permutation $\sigma = \langle a_1, \ldots, a_n \rangle$ is pyramidal iff $\exists i \in [1 : n] : a_1 < \ldots < a_i$ and $a_i > \ldots > a_n$. There exists 2^{n-1} such permutations. In [4], a dynamic program is presented to find the best pyramidal permutation. Thus this neighborhood can be categorized as a VLSN.

Toward VLSN abstractions

Some abstractions of VLSN are presented in [2, 5, 6, 7]. In [7], they focus on the Traveling Salesman Problem (TSP) and it is shown that some VLSN can be specified by a context-free grammar. A generic dynamic parsing program is given; it computes the best neighbor in polynomial time. Grammar for many VLSN presented in [4, 2, 8, 9] are also presented. However no implementation nor experimental results are provided.

For example, the above pyramidal neighborhood can be defined by the following grammar

$$\begin{array}{rcll} S & \rightarrow & A_{1,n} & \\ A_{j,n} & \rightarrow & j, A_{j+1,n} & \forall 1 \leq j \leq n-1 \\ A_{j,n} & \rightarrow & A_{j+1,n}, j & \forall 1 \leq j \leq n-1 \\ A_{n,n} & \rightarrow & n & \end{array}$$

Implementation in Comet. We implemented the algorithm of [7] in Comet. A straighforward implementation of this algorithm is not efficient as it does not take into account the state-space graph of the dynamic program. To overcome this problem, our implementation maintain a specific data-structure, similarly as in [9]. With this data-structure, the parsing of the grammar is much more efficient as it takes the structure of the grammar into account.

Our implementation of the generic algorithm allows a user to define a pyramidal VLSN for the TSP as follows :

```
symbol sT[i in 1..N] =
    addSymbol(IntToString(i), true, {i});
symbol A[j in 1..N] =
    addSymbol("A_"+IntToString(j), false,
              collect(e in j..N) e        );
setS(A[1]);
forall (j in 1..N-1) {
  addClause( A[j], sT[j], A[j+1]);
    addClause( A[j], A[j+1], sT[j]);      }
  addClause( A[N], sT[N]);
```

This VLSN model is close to its declarative definition.

Future Work: In order to provide efficient abstractions of VLSN in Comet we will extend the results of [7] to different problems such as sequencing problems. We will also provide abstractions for VLSN defined by cycles in *improvement graphs* as introduced in [10]. These are very efficient when problems have a partitionning substructure. We believe this direction of research in interesting as the improvement graph could be defined and updated by mean of Comet invariants.

References

[1] P. Van Hentenryck and L. Michel, *Constraint-Based Local Search*. The MIT Press, 2005.

[2] V. Deineko and G. Woeginger, "A study of exponential neighborhoods for the travelling salesman problem and for the quadratic assignment problem," *Mathematical Programming*, vol. 87, pp. 519–542, 2000.

[3] R. K. Ahuja, Özlem Ergun, J. B. Orlin, and A. P. Punnen, "A survey of very large-scale neighborhood search techniques," *Discrete Appl. Math.*, vol. 123, no. 1-3, pp. 75–102, 2002.

[4] J. Carlier and P. Villon, "A new heuristic for the traveling salesman problem," *RAIRO Operational Research*, vol. 24, pp. 245–253, 1990.

[5] G. Gutin, A. Yeo, and A. Zverovitch, *Exponential Neighborhoods and Domination Analysis for the TSP*, ch. The Traveling Salesman Problem and Its Variations. Springer US, 2004.

[6] O. Ergun and J. B. Orlin, "A dynamic programming methodology in very large scale neighborhood search applied to the traveling salesman problem," *Discrete Optimization*, vol. 3, pp. 78–85, Mar. 2006.

[7] Bompadre and Orlin, "Using grammars to generate very large scale neighborhoods for the traveling salesman problem and other sequencing problems," *Integer Programming and Combinatorial Optimization*, vol. 3509/2005, pp. 437–451, 2005.

[8] C. Potts and S. van de Velde, "Dynasearch-iterative local improvement by dynamic programming. part i. the traveling salesman problem," tech. rep., University of Twente, The Netherlands, 1995.

[9] E. Balas and N. Simonetti, "Linear time dynamic programming algorithms for new classes of restricted tsps: A computational study," 2001.

[10] P. M. Thompson and J. B. Orlin, "The theory of cyclic transfers," Tech. Rep. OR 200-89, Massachusetts Institute of Technology, Operations Research Center, 1989.