Michel Gendreau *Gilles Pesant and Louis-Martin Rousseau*

Centre for Research on Transportation, Université de Montréal

Using constraint-based operators with Variable Neighborhood Search to solve the Vehicle Routing Problem with Time Windows

1 Introduction

This paper presents two operators for vehicle routing problems that make use of constraint programming and local search to explore their neighborhood. These operators are used to optimize restricted version of the original problem and are combined in a Variable Neighborhood Search (VNS) framework to solve the Vehicule Routing Problem with Time Windows (VRPTW).

2 Method

2.1 VNS framework

The two operators presented below are used in a local descent strategy, as they never allow the objective function to increase; it is known that such strategies get trapped in local minimum when no better solution can be found in a neighborhood. To circumvent such a problem, let us consider the Variable Neighborhood Search (VNS) scheme introduced by Mladenovic and Hansen (1997). The VNS strategy is to use one operator until we are sufficiently sure that we are trapped in a local minimum, then to start applying the second one until itself can no longer improve the solution. A VNS will oscillate in this way between two or more operators, hoping that changes in neighborhood structure will permit an escape from most local minima. For this principle to work well, we think that the operators used must should be very different with respect to the neighborhood structure they generate. This is the case with the operators we propose, one being centered on customer insertion and the other on arc exchange.

2.2 Ideas behind operators

In a traditional metaheuristic context, an operator is defined as a recipe to modify a solution and obtain a potentially better one. An operator defines a neighborhood, that is the set of solutions that can be produced by applying that operator on one solution. A move is a transition from one solution to the one in its neighborhood. It is easy to understand that larger neighborhoods will tend to include better solutions, but the time needed to explore those neighborhoods (as we have to look at every solution) grows rapidly and substantially limits the scope of an operator. The idea, originally introduced by Pesant and Gendreau

(1999), is to explore the set of neighboring solutions with a branch and bound search in a constraint programming framework, thus implicitly using propagation and pruning to eliminate subsets of neighbors and furthermore limiting the number of solutions that are actually visited. The two operators presented here reflect that line of thought.

2.3 Operator : LNS-GENI

The first operator introduced is inspired by ideas of the Large Neighborhood Search presented by Shaw (1998). The algorithm first removes a subset of customers from the solution and then looks for a better way to reinsert them in the partial solution. Instead of performing simple insertions, we use a constraint programming version of the GENI algorithm. The GENeralized Insertion algorithm with its time window variant (see Gendreau *et al.* (1995)) try to insert a customer in a given route between any two customers of that route: if the chosen customers are not consecutive a local optimization is performed to make the insertion possible. The constraint programming version developed by Pesant, Gendreau and Rousseau (1997) differs from the original by the fact that instead of looking at all possible local optimizations, it looks for the best one using branch and bound with constraint propagation.

2.4 Operator SMART

The SMAll RouTing operator bears its name because it actually solves a smaller VRP that is a restriction of the original one. The general concept is that, instead of removing customers, we remove arcs from the problem thus creating an incomplete solution to the original problem. This incomplete solution could be interpreted as a smaller VRP in two different ways. The first one is to suppose that the freed arcs are all consecutive, then the last customer before and the first customer after the freed sequences of all routes would become the new depots, and all we need to do is solve (exactly or not) the smaller problem. A second way of interpreting the relaxation is to consider the general case where the removed arcs are randomly distributed across the solution: we could then replace each of the remaining sub-sequences by a single customer. After adjusting the distance table we would then have a smaller asymmetric VRP.

If we choose the size of a SMART problem correctly, we can manage to solve it exactly. To do so we use a modified version the TSPTW model developed by Pesant *and al.* (1997).

3 Results

The solutions produced by our hybrid method seem to be competitive with those published in the literature: we note that both the number of vehicles used and the total traveled distance are around 0.5% of the other good methods, if not better. We also report new best solutions on 12 of the 56 Solomon's problems.