Synchronized Arc Routing for Snow Plowing Operations

M. Angélica Salazar-Aguilar
CIRRELT, HEC Montréal, e.mail: angelica.salazar@cirrelt.ca
André Langevin
CIRRELT and Département de mathématiques et de génie industriel
École Polytechnique de Montréal, e.mail: andre.langevin@polymtl.ca
Gilbert Laporte
CIRRELT and Canada Research Chair in Distribution Management,
HEC Montréal, e.mail: gilbert.laporte@cirrelt.ca

Abstract. This work introduces a synchronized arc routing problem arising from a real-world application in snow plowing. The problem consists of determining a set of routes such that all street segments are serviced within the least possible time. Each segment is cleaned by snow plow vehicles that start and end their journey at the depot. An adaptive large neighborhood search heuristic is proposed and evaluated over a large instance set, including artificial and real data.

*Keywords:*Arc routing, Snow plowing, Road maintenance, Synchronized Routes, Adaptive large neighborhood search.

Introduction

Problems arising in winter road maintenance are complex, costly and site-specific because of the variation of climatic conditions, demographics, economics, and technology. The importance of the winter road maintenance is due to the magnitude of the expenditures associated to these operations, and to the indirect costs resulting from the loss of productivity and decreased mobility. For instance, in 2010 the average cost of a 20 cm snow storm in Montreal city was $17 million (canadian dollars). On average, there are 65 weather events calling for response every winter. Plowing operations begin as soon as there is an accumulation of 2.5 cm of snow on the ground and continue as long as the storm lasts, ending about eight hours after the snow stops falling. An important practical consideration absent from the literature on the planning of snow plowing operations is the need to synchronize the vehicles required at the same time on the same street segment. When cleaning a two-lane or three-lane street, several snow ploughs often follow one another to build a snow mound in the central part of the street which will later be cleaned by a snow blower. In this work we introduce and solve the synchronized arc routing problem (SyARP) for snow plowing operations.

Problem description

Given a network of streets and a fleet of snow plowing vehicles based at a depot, the SyARP consists of finding a set of routes such that all streets, some of which have multiple lanes, are plowed by using synchronized vehicles, and the ending time of the longest route is minimized. The street segments have different number of lanes in one or two directions and all lanes in the same direction should be plowed by the required vehicles at the same time in order to reduce the length of time during which they can not be used by motorists because of the accumulation of snow created by the snow plows. The number of vehicles that service a given segment is equal to its number of lanes. Deadheading is allowed, so that any vehicle can traverse any arc several times without providing service to it.

Solution procedure

We have developed a heuristic solution technique based on the *Adaptive large neighborhood search (ALNS) metaheuristic*. Our heuristic generates a first set of routes, and then attempts to improve them by mean of different destroy/repair operators. The choice of operators is controlled dynamically according to their past performance. Each operator has a weight which controls how often it is used during the search. The weights are adjusted dynamically as the search progresses so that the heuristic adapts to the instance at hand and to the state of the search. Given an initial solution P, the ALNS heuristic is applied until a stopping criterion is reached. Several initial solutions are successively used as input. At the end of the procedure we report the best solution encountered during the search. The selection of an operator ω is done by following a *roulette wheel principle*. Once an operator has been chosen, it is applied until three consecutive non-improving solutions are found. A non-improving solution can be randomly accepted if its value does not exceed that of the incumbent solution by more than 10%. For each operator ω, at most 50 non-improving solutions can be accepted. Let Φ be the set of non-improvement solutions accepted in the previous iterations and let $P' \in \Phi$ be a non-improving solution that does not exceed the incumbent solution by more that 10%. An uniform random number $r \in [0, 50]$ is generated, if $|\Phi| < r < 50$ then P' is accepted, otherwise P' is rejected.

We have developed five destroy/repair operators which basically consist of removal and insertion of sequences of arcs, interchanging of arc statuses and reoptimization of routes. The stopping criterion used in our procedure is the number of iterations.

Computational results

The algorithm just described, was coded in C++ and compiled on a 2592.574 MHz AMD Opteron(tm) processor 285 with 1GB of RAM under the Linux operating system. We have first tested our algorithm on several set of randomly generated instances. To this end, we have generated instance sets with 42, 180, and 300 vertices on the same grid shape. The number of arcs for each set was 113, 499, and 795, respectively. Fifteen instances were generated on each instance set and all of them have arcs with one, two or three lanes. The number of available vehicles for each set was 12, 18, and 25, respectively. The maximum number of iterations was set to 1500. Computational results reveal the good performance of our algorithm. In addition, a real-world case from Dieppe (a suburb of Moncton, in New Brunswick, Canada) was solved efficiently by our algorithm. The Dieppe instance has 430 vertices and 1056 arcs which have one or two lanes.