Chapter 9
A Multi-thread GRASPxELS for the Heterogeneous Capacitated Vehicle Routing Problem

Christophe Duhamel, Christophe Gouinaud, Philippe Lacomme, and Caroline Prodhon

Abstract. This chapter focuses on the definition of an efficient parallel metaheuristic which takes advantage of the multi-core design of recent processors. The approach is designed as a Greedy Randomized Adaptive Search Procedure (GRASP) hybridized with a multi-threaded version of an Evolutionary Local Search (ELS) metaheuristic scheme. Our approach is evaluated on an extension of the Vehicle Routing Problem where a heterogeneous fleet of vehicles is available to service a set of customers. The objective consists in designing a set of trips for a limited heterogeneous fleet of vehicles located at a depot node which minimizes the total transportation cost. Each type of vehicles is defined by a capacity and by the number of available vehicles. The efficiency of the parallel approach is evaluated on a new set of real-life instances built out of data from the French districts. A fair comparative study, using a same implementation, is done to evaluate the impact of the number of threads on the convergence rate. Thus, a better trade-off between solution quality and computational time can be reached. The numerical experiments show that the hybrid GRASPxparallel ELS outperforms the classical iterative version and provides numerous new best solutions.

9.1 Introduction

The design of parallel implementations of metaheuristics has received a considerable amount of attention in the last two decades with the development of new hardware
technologies which provide several efficient and cheap opportunities. This chapter focuses on the definition of a Greedy Randomized Adaptive Search Procedure (GRASP) hybridized with a multi-threaded version of an Evolutionary Local Search (ELS) metaheuristic scheme which takes advantage of the multi-core design of the recent processors. Our approach is tested on the Heterogeneous Vehicle Routing Problem (HVRP), an extension of the classical Vehicle Routing Problem (VRP) where a heterogeneous fleet of vehicles is available to service a set of customers.

A brief reminder of the recent technological advances in parallel implementations of algorithms is presented in the next Section, with a state of the art on parallel metaheuristics. Then, Section 9.3 introduces the problem under consideration in this chapter, the HVRP. The parallel hybrid framework is stated in Section 9.4. Its implementation for the HVRP is explained in Section 9.5. Then, the details of the components of the method are exposed in Section 9.6. Finally, a computational evaluation of the method is proposed through Section 9.7 before concluding remarks.

9.2 Parallel Metaheuristics

Before presenting our hybrid metaheuristic based on a GRASP and a parallel implementation of an ELS scheme, this first section gives an overview of the available technologies and of the state of the art to justify our choice of a multi-thread implementation to solve the HVRP.

9.2.1 Technologies

Parallel computing environment offers various ways to implement parallelization. More specifically, three main trends can be identified.

Historically, the first trend is based on computer cluster architectures and software environment. It relies on communication frameworks as MPI (Message Passing Interface) and it usually requires a large communication bandwidth in order to stay efficient. Such an approach provides a master-slave message passing paradigm but it also requires a huge financial investment in order to build the cluster and the high-speed communication network.

The second trend is more recent. It consists in taking advantage of the capabilities of the graphics processors included in modern computers (Graphics Processing Unit - GPU). Even if they were first developed to afford the computing requirements of the visual effects in modern 3D games, it has been lately evidenced they could be used for other purposes. For instance, NVIDIA provides a CUDA (Compute Unified Device Architecture) library with a user-friendly C++ interface in order to simplify the development of parallel algorithms. Their GPU can handle a huge number of threads to be used but they are unfortunately gathered in blocks with a slow shared memory. Depending on the GPU capability, the dedicated memory is