Computers & Operations Research 44 (2014) 33–41

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

# A cooperative parallel metaheuristic for the capacitated vehicle routing problem

Jianyong Jin [a], Teodor Gabriel Crainic [b,*], Arne Løkketangen [a]

[a] Molde University College, Specialized University in Logistics, N-6411, Molde, Norway
[b] School of Management, UQAM & Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation, Montréal, QC, Canada

ARTICLE INFO

Available online 25 October 2013

Keywords:
Vehicle routing
Parallel metaheuristic
Cooperative search
Solution clustering

ABSTRACT

This paper introduces a cooperative parallel metaheuristic for the capacitated vehicle routing problem. The proposed metaheuristic consists of multiple parallel tabu search threads that cooperate by asynchronously exchanging best-found solutions through a common solution pool. The solutions sent to the pool are clustered according to their similarities. The search history information identified from the solution clusters is applied to guide the intensification or diversification of the tabu search threads. Computational experiments on two sets of large-scale benchmark instance sets from the literature demonstrate that the suggested metaheuristic is highly competitive, providing new best solutions to ten of those well-studied instances.



## 1. Introduction

In recent years, cooperative parallel metaheuristics have increasingly been used for solving a variety of difficult combinatorial problems [1]. Such parallel metaheuristics usually use multiple processes (threads) working simultaneously on available processors, with varying degrees of cooperation, to solve a given problem instance. The rationale behind this phenomenon may be twofold. First, it has been demonstrated that such parallel algorithms are capable of both speeding up the search and improving the robustness (ability of providing equally good solutions to a large and varied set of problem instances) and the quality of the solutions obtained [2]. Second, parallel computing resources have become increasingly available with the advent of computer clusters and multi-core processors. The computer clusters usually consist of a set of identical computers that run standard operating systems and are connected to each other through high speed networks. Many universities nowadays possess such computer clusters. In addition, many laptops and desktop computers today use dual- or quad-core processors. Thus, using parallelism has become an advantageous and practical option. For a detailed introduction to parallel metaheuristics, we refer to the book of Alba [3] and the survey papers of Crainic [2] and Crainic and Toulouse [4].

The capacitated vehicle routing problem (CVRP), the classical version of the vehicle routing problem (VRP), aims to determine the minimum total cost routes for a fleet of homogeneous vehicles to serve a set of customers. The CVRP can be defined on a graph $G=(N,E)$, where $N=\{0,\dots,n\}$ is a vertex or node set and $E=\{(i,j): i,j\in N\}$ is an edge set. Vertex 0 is the depot where the vehicles depart from and return to. The other vertices are the customers which have a certain demand $d$ to be delivered (or picked up). The travel cost between node $i$ and $j$ is defined by $c_{ij}>0$. The vehicles are identical. Each vehicle has a capacity of $Q$. The objective is to design a least cost set of routes, all starting and ending at the depot. Each customer is visited exactly once. The total demand of all customers on any route must not exceed the vehicle capacity $Q$. Some CVRP instances may have an additional route duration limit constraint, restricting the duration (or length) of any route to a preset bound $D$. A detailed introduction to the CVRP and its solution methods can be found in the book of Toth and Vigo [5], and the survey paper of Laporte [6]. Even though a large number of solution methods have been proposed in the literature during the last 50 years, it still remains computationally challenging to quickly produce high-quality solutions to large scale CVRP instances.

The purpose of this paper is to present a cooperative parallel metaheuristic that takes advantage of modern parallel computing resources to address large scale CVRP instances. The proposed algorithm incorporates multiple tabu search threads which cooperate by asynchronously exchanging the best-found solutions through a common solution pool, and includes several novel features. Intensification and diversification of the tabu searches are based on solution clustering. Four variants of the reinsertion neighborhood are applied and unfeasible solutions may also be sent to the solution pool. These features are clearly different from previous work (e.g., [7,8]) and largely contribute to the high performance of the proposed metaheuristic. The computational experiments on two sets of

* Corresponding author. Tel.: +1 514 343 7143; fax: +1 514 343 7121.
E-mail address: TeodorGabriel.Crainic@cirrelt.ca (T.G. Crainic).

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.10.004

large scale CVRP benchmark instances demonstrate that the suggested metaheuristic can quickly produce solutions to benchmark problems that are highly competitive with the best solutions reported in the literature. New best solutions to 10 out of the 32 instances have been identified.

The remainder of this paper is organized as follows. In the next section the description of the proposed metaheuristic is presented. Then Section 3 reports the computational results. Concluding remarks are given in the last section.

## 2. Description of the cooperative parallel metaheuristic

In the proposed cooperative parallel metaheuristic (CPM), illustrated in Fig. 1, multiple tabu search (TS) threads are run in parallel to address a given CVRP instance. Some of the TS threads are designated to concentrate on intensification while the others are assigned to pursue diversification. These threads communicate asynchronously through a common solution pool.

The general scheme of CPM is displayed in Algorithm 1. During the search process, the solution pool receives solutions sent from the search threads. Whenever a solution is received from a search thread, the pool performs the clustering, selects a solution, and sends it back to the same thread. Each of the TS threads carries out its search independently and periodically the search halts and exports its best-found solution. It then receives a solution from the pool and resumes its search from this solution. The detailed description of the solution pool and the TS threads is provided in Sections 2.1 and 2.2 respectively.

The termination of CPM can be controlled in two ways. In the first setting (identified as TC1), termination is triggered by the first TS thread. The metaheuristic terminates after the thread runs for a certain number of iterations. In the other setting (called TC2), the metaheuristic terminates once the solution pool receives a certain number of non-improving solutions consecutively. A solution is regarded as non-improving when it is unfeasible or its value is not better than that of the current best feasible solution in the pool.

**Algorithm 1.** CPM

Initialize TS threads and the solution pool;
**while** termination condition not met
  Solution pool
    Receives solutions;
    Clusters solutions;
    Selects and sends solutions back;
  Each TS thread asynchronously
    Performs the search;
    Sends best found solution to the solution pool;
    Receives new solution to start from;
**end while**
Return best feasible solution.

In terms of the taxonomy of Crainic and Hail [9] for parallel metaheuristics, CPM fits into the *pC/KC/MPDS* classification. The first dimension *pC* indicates that the global search is controlled by multiple cooperative threads. The second dimension *KC* stands for knowledge collegial information exchange and refers to the fact that multiple threads share information asynchronously and knowledge is created from the exchanged information to guide the cooperating threads. The last dimension *MPDS* indicates that multiple search threads start from different points in the solution space and follow different search strategies.

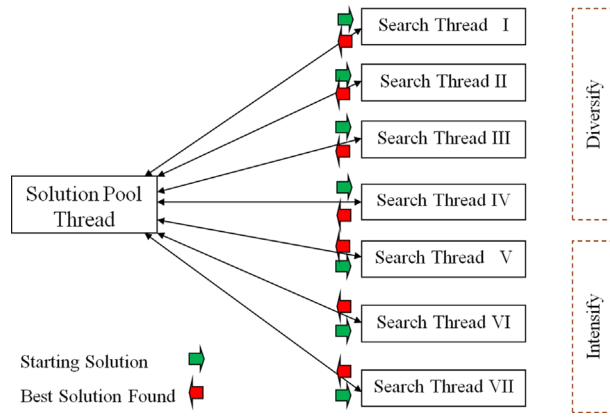


**Fig. 1.** Framework of CPM.

### 2.1. Solution pool

To explore a search space effectively and efficiently, a metaheuristic approach should be able to both intensively investigate the areas of the search space displaying high quality solutions, and to move to unexplored areas of the search space when necessary. These goals are usually achieved by intensification and diversification mechanisms of the metaheuristic [10]. Glover and Laguna [11] highlight that intensification is to carefully search the neighborhood of elite solutions while diversification encourages the search process to generate solutions that differ from those seen before. A solution clustering approach is used in CPM to implicitly identify common features of solutions and collect search history information, which then provides a good basis for selecting promising search areas for intensification and less explored areas for diversification.

During the whole search process, the solutions sent to the solution pool from the search threads are dynamically clustered into groups according to their similarity. For the CVRP, similarity can be measured in terms of the number of edges solutions have in common. Solutions kept in the solution pool can then be grouped into clusters where all solutions belonging to the same cluster have a given number of edges in common. Each cluster can thus approximately represent a region of the search space that CPM has explored. The features of the solutions in a cluster, such as the number of solutions and the quality of the solutions, can indicate how thoroughly a search region has been explored and how promising it may be. Such search history information is used to guide the starting solution selection for the TS threads so that they can pursue intensification or diversification effectively.

The solution clustering approach has been applied by Voß [12] for the quadratic assignment problem. In his algorithm, a small number of elite solutions previously found are stored by a clustering approach and are used as the starting solutions for the intensification phases. In CPM, the solution clustering approach is applied differently in three aspects. First, all solutions sent to the pool are clustered, regardless of their quality. Second, the solutions are clustered for both intensification and diversification purposes. Third, the actual clustering mechanism is different.

#### 2.1.1. Solution clustering

Clustering is often defined as the process of grouping of a collection of patterns into dissimilar segments or clusters based on a suitable notion of closeness or similarity among these patterns. In CPM, solutions are grouped into clusters based on their similarity. A cluster, in this context, refers to a collection of solutions that are similar. All solutions sent to the solution pool are clustered.