Neurocomputing 146 (2014) 210–229

Contents lists available at ScienceDirect

Neurocomputing

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

A cooperative population learning algorithm for vehicle routing problem with time windows

Dariusz Barbucha*

Department of Information Systems, Gdynia Maritime University, Morska 83, 81-225 Gdynia, Poland

ARTICLE INFO

Article history:
Received 31 October 2013
Received in revised form
10 June 2014
Accepted 14 June 2014
Available online 27 June 2014

Keywords:
Computational collective intelligence
Cooperative problem solving
Agent-based optimization
Population learning algorithm
Vehicle routing problems

ABSTRACT

Population-based hybrid metaheuristics, often inspired by biological or social phenomena, belong to a widely used groups of methods suitable for solving complex hard optimization problems. Their effectiveness has been confirmed for providing good quality solutions to many real-life instances of different problems. Recently, an incorporation of the cooperative problem solving paradigm into metaheuristics has become an interesting extension of the population-based hybrid metaheuristics. Cooperation is meant as a problem-solving strategy, consisting of a search performed by different search agents running in parallel. During the search, the agents cooperate by exchanging information about states, solutions or other search space characteristics. This paper proposes an Agent-Based Cooperative Population Learning Algorithm for the Vehicle Routing Problem with Time Windows. In the proposed approach the process of search for the best solution is divided into stages, and different search procedures are used at each stage. These procedures use a set of various heuristics (represented by software agents) which run under the cooperation scheme defined separately for each stage. Computational experiment, which has been carried out, confirmed the effectiveness of the proposed approach.

1. Introduction

In recent years, approaches based on the collective computational intelligence have become an interesting and promising group of efficient methods for solving computationally hard optimization problems. Starting from a general assumption that collective intelligence is an intelligence that emerges from the collaboration and competition of many artificial and/or natural individuals (Levy [27] and Russell [37]), one can say that computational collective intelligence (CCI) is most commonly understood as a field of artificial intelligence including issues of collective problem solving by autonomous entities (agents) operating in the distributed environments using flexible calculation methods (soft computing), enabling the processing of knowledge by those units, and cooperation between them. As a result, increase of intelligence of the whole collection is often observed. Also the effectiveness of performance of the tasks carried-out by the members of the collective may also increase.

An important group of methods within the field of computational collective intelligence are hybrid population-based methods, based on the processes occurring in societies and/or nature. In this context, an interesting approach which extends the population-based metaheuristics group and is inspired by an analogy to the social education systems, has been proposed by Jędrzejowicz [25] under the name Social Learning Algorithm, next renamed to Population Learning Algorithm (PLA). In its classical version, PLA divides the process of solving the problem into stages, in which the considered optimization problem is solved using a set of independent learning/improvement procedures, each procedure at a single stage. Initially, a massive population of individuals is created, either randomly or using a simple constructive heuristic, and it is stored in a sharable memory. Once the initial population has been generated its individuals enter the first learning stage. This stage involves applying a possibly basic and elementary heuristic to each individual. Furthermore, the improved individuals are evaluated. Selected better ones pass to the next stage and the remaining are dropped from the process. At the following stages the whole cycle is repeated. Individuals are subject to improvement and learning by using dedicated procedures, and the selected ones are again promoted to the higher stage. As a rule, the procedures used at higher stages are more complex than ones, used in earlier stages. At the final stage the best and the brightest individuals are evaluated in order to select a final solution to the problem at hand.

The goal of the paper is to propose a Cooperative Population Learning Algorithm (CPLA), which extends the classic PLA and focuses on incorporating a *cooperative problem solving* paradigm

* Tel.: +48 586901287; fax: +48 586206701.
E-mail address: d.barbucha@wpit.am.gdynia.pl

http://dx.doi.org/10.1016/j.neucom.2014.06.033
0925-2312/

into the original PLA. According to Crainic and Toulouse [15], the term *cooperative search* can be defined as "a set of highly *autonomous programs* (APs), each implementing a particular solution method, and a *cooperation scheme* combining these APs into a single problem-solving strategy". A set of autonomous programs may include exact methods, like for example branch and bound, but in most cases different approximate algorithms (local search, variable neighborhood search, evolutionary algorithms, tabu search, simulated annealing, etc.) are engaged in finding the best solution. On the other hand, a cooperation scheme has to provide the mechanism for effective communication between autonomous programs allowing them to dynamically exchange the important pieces of information which next is used by each of them to support the process of search for a solution.

In particular, the proposed Cooperative Population Learning Algorithm extends the classic PLA allowing the use of a few methods (represented by agents) at each learning/improvement stage, which can cooperate during the search. The cooperation is organized in indirect form, which means that results obtained by one heuristic (or team of heuristics) can be shared with other heuristics (or teams of heuristics), engaged in process of solving instances of given problem. The architecture of the proposed CPLA is based on the asynchronous team (A-Team) [44] implementations proposed in [1] and extended in [2], where A-Team is a multi agent architecture, where a collection of autonomous and asynchronous software agents solve a problem by dynamically evolving the population of solutions stored in the common memory. Within A-Team, each agent represents a particular, exact or approximate problem-solving method, whilst memories accumulate results or trial solutions forming populations, processed by agents during their work. The existence of a set of autonomous agents, shared memory, and a mechanism of management of population of solutions, provide a basis for cooperation between agents. Agents working within A-Team cooperate by sharing access to populations of candidate solutions [35]. Solutions obtained by one agent are shared, through the central memory mechanism, with other agents, which can exploit these solutions in order to guide the search through new promising region of the search space, thus increasing chances for reaching the global optimum. In this context A-Teams demonstrate properties of the collective intelligence system since it is expected that the collective of agents can produce better solutions than individual members of such collective, thus achieving a synergetic effect.

The paper includes a study and experimental validation of the proposed Cooperative Population Learning Algorithm designed to solve instances of the Vehicle Routing Problem with Time Windows (VRPTW). Although the number of algorithms for VRPTW is high and many of them are significant, only a few cooperative search approaches are reported in the literature (some of them will be shortly described in Section 2). The author's intent is to increase the pool of this group of methods, proposing a new effective multi-stage cooperative algorithm working within a multi-agent framework. The paper extends the approach proposed in [3].

The rest of the paper is organized as follows. Section 2 presents a formulation of the Vehicle Routing Problem with Time Windows and a review of existing approaches to solve it. Sections 3 and 4 outline the main features of the CPLA, and present an implementation details of the CPLA for VRPTW, respectively. Computational experiment and its result are presented in Section 5. Finally, Section 6 concludes the paper and proposes future work.

2. Vehicle routing problem with time windows

The Vehicle Routing Problem with Time Windows (VRPTW) belongs to the best known and important problems from the logistics and transportation fields. It can be formulated as the problem of determining optimal routes through a given set of locations (customers) and defined on an undirected graph $G=(V,E)$, where $V=\{0,1,\dots,N\}$ is the set of nodes and $E=\{(i,j)|i,j\in V\}$ is a set of edges. Node 0 is a central depot with NV identical vehicles of capacity W. Each other node $i\in V\backslash\{0\}$ denotes customer characterized by coordinates in Euclidean space (x_i,y_i), a non-negative demand d_i, and a service time of the customer s_i. Moreover, with each customer $i\in V$, a time window $[e_i,l_i]$ is associated, wherein the customer has to be supplied. Here e_i is the earliest possible departure (ready time), and l_i – the latest time a service to the customer has to be started. The time window at the depot ($[e_0,l_0]$) is called the scheduling horizon. Each link $(i,j)\in E$ denotes the shortest path from customer i to j and is described by the cost c_{ij} of travel from i to j by shortest path $(i,j\in V)$. It is assumed that $c_{ij}=c_{ji}$ $(i,j\in V)$. It is also often assumed that c_{ij} is equal to travel time t_{ij}.

The goal is to minimize the number of vehicles and the total distance needed to supply all customers (minimization of the fleet size is considered to be the primary objective of the VRPTW), such that each route starts and ends at the depot, each customer $i\in V\backslash\{0\}$ is serviced exactly once by a single vehicle, the total load on any vehicle associated with a given route does not exceed vehicle capacity, each customer $i\in V\backslash\{0\}$ has to be supplied within a time window associated with him (a vehicle arriving before the lower limit of the time window causes additional waiting time on the route), and each route must start and end within the time window associated with the depot.

Due to NP-hardness of VRPTW, most of the proposed approaches to solve it belong to approximate methods group [12,13]. Recently, an attention of researchers from OR and AI community have been mainly focused on metaheuristic approaches and different forms of hybridization of them with other methods. Despite of the fact that metaheuristics require much more computational resources and have to be fine-tuned in order to fit a particular problem, they often provide much better solutions than classical heuristics, especially in case of large-scale instances. Moreover, combining them with different algorithms have provided powerful search algorithms, which outperform methods based on a single optimization procedure [43].

One of the first and the most successful metaheuristics approaches to VRPTW were Tabu Search implementations of Rochat and Taillard [36], Taillard et al. [42], and Cordeau et al. [14]. Rochat and Taillard [36] proposed the tabu search approach based on the so-called *adaptive memory* meant as a pool of routes taken from the best solutions visited during the search. Its main role was to help to diversify and intensify the process of search. In the first phase of their method, tabu search was used to create a number of different solutions stored in the adaptive memory. The routes were selected probabilistically from the memory and the selected tours were improved using tabu search and inserted subsequently back into the adaptive memory. At the end, a post-optimization technique has been used. A set partitioning problem was solved exactly, using the routes stored in the adaptive memory to create the best solution.

Taillard et al. [42] considered a parallel tabu search heuristic for solving VRP with soft Time Windows. The authors proposed a new exchange method called CROSS-exchange, used both to inter- and intra-route operations. They also used adaptive memory of Rochat and Taillard [36], which initially was filled with different types of routes using Solomon's insertion heuristic *I1* [39]. To reduce the complexity of the search, they limited the neighborhood by decomposing solutions into a collection of disjoint subsets of routes, and applying tabu search to each subset separately. A complete solution was reconstructed by merging the new routes found by tabu search. The search was diversified by penalizing