



ELSEVIER

Contents lists available at ScienceDirect

Computers & Operations Research

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

A biased random-key genetic algorithm for the capacitated minimum spanning tree problem



Efrain Ruiz^a, Maria Albareda-Sambola^b, Elena Fernández^{c,*}, Mauricio G.C. Resende^d

^a Departament d'Estadística i Investigació Operativa, Universitat Politècnica de Catalunya, BarcelonaTech, Barcelona, Spain

^b Departament d'Estadística i Investigació Operativa, Universitat Politècnica de Catalunya, BarcelonaTech, Carrer Colom, 11, 08222 Terrassa, Spain

^c Departament d'Estadística i Investigació Operativa, Universitat Politècnica de Catalunya, BarcelonaTech, Campus Nord, C5-208, Jordi Girona, 1-3, 08034 Barcelona, Spain

^d Network Evolution Research Department, AT&T Labs Research, 200 S. Laurel Avenue, Room 5A-1F34, Middletown, NJ 07748, USA

ARTICLE INFO

Available online 5 December 2014

Keywords:

Optimization
Combinatorial optimization
Networks
Graphs
Trees
Spanning trees
Capacitated minimum spanning tree
Heuristics
Biased random-key genetic algorithm

ABSTRACT

This paper focuses on the capacitated minimum spanning tree (CMST) problem. Given a central processor and a set of remote terminals with specified demands for traffic that must flow between the central processor and terminals, the goal is to design a minimum cost network to carry this demand. Potential links exist between any pair of terminals and between the central processor and the terminals. Each potential link can be included in the design at a given cost. The CMST problem is to design a minimum-cost network connecting the terminals with the central processor so that the flow on any arc of the network is at most Q . A biased random-key genetic algorithm (BRKGA) is a metaheuristic for combinatorial optimization which evolves a population of random vectors that encode solutions to the combinatorial optimization problem. This paper explores several solution encodings as well as different strategies for some steps of the algorithm and finally proposes a BRKGA heuristic for the CMST problem. Computational experiments are presented showing the effectiveness of the approach: Seven new best-known solutions are presented for the set of benchmark instances used in the experiments.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Minimal spanning trees are among the most studied structures in combinatorial optimization. From a theoretical point of view, the interest of studying minimum-cost spanning trees (MSTs) lies in their particularity as mathematical objects. Due to its matroid structure, a MST can be found in polynomial time with greedy algorithms, like the ones of Prim [40] or Kruskal [32]. From a practical point of view, the connectivity property of MSTs is very useful in multiple applications. For this reason, along with the traveling salesman problem (TSP), MSTs problems are considered to lie in the core of network systems design and of a wide variety of scheduling and routing applications.

Often, TSPs and MST problems take into account additional restrictions, being capacity constraints among the most frequent ones. The problem that consists of finding an MST that satisfies additional capacity constraints is called the *Capacitated Minimum Spanning Tree* (CMST) problem. From now on, we will refer to this problem simply as

the CMST. The study of the CMST is of interest because the mere addition of capacity constraints transforms the MST into an *NP-hard* problem. The CMST often arises in telecommunication network design, but it also has applications in distribution, transportation, and logistics. For example, it is related to vehicle routing problems, due to the influence that MSTs have in constructive heuristics: the TSP heuristic of Christofides [10] is based on spanning trees and similar heuristics were developed for arc routing problems in Frederickson et al. [13] and Frederickson [12]. Because vehicle routing problems usually consider capacity constraints, the design of more effective or efficient algorithms for the CMST can also play a role in the design of efficient methods to obtain feasible solutions for the capacitated vehicle routing problem. Amberg et al. [4] showed that it is possible to transform a multicenter capacitated arc routing problem (M-CARP) into a capacitated minimum spanning tree with additional arc constraints.

Many authors have proposed integer programming formulations for the MST and its extensions [5,23–29,31,49]. For the particular case of the CMST, lower bounds can be found in Gavish [14–16], Gouveia [24], and Uchoa et al. [55]. To date, the most successful exact method is the one of Uchoa et al. [55]. However, due to its *NP-hard* nature (see [39]) the solution of the CMST with exact methods is usually very time consuming and even impossible, already for moderate size instances. This explains why heuristic methods, based

* Corresponding author.

E-mail addresses: efrain.ruiz@upc.edu (E. Ruiz), maria.albareda@upc.edu (M. Albareda-Sambola), e.fernandez@upc.edu (E. Fernández), mgrc@research.att.com (M.G.C. Resende).

on the greedy paradigm [11,9,17], neighborhood exploration [3,50,1,2,51], or dynamic programming [30], have been widely used. More recent heuristics include the second-order algorithm of Martins [34], in which subproblems of the original problem including a set of constraints are solved with the Essau Williams heuristic, the heuristic of Rego et al. [44], which projects dual solutions into the primal feasible space and obtains primal feasible solutions by simple tabu searches and metaheuristics such as the ant colony algorithm by Reimann and Laumanns [45] or the filter-and-fan algorithm by Rego and Mathew [43].

In this paper we propose a biased random-key genetic algorithm (BRKGA) for the CMST, which stems from the Ph.D. Thesis of E. Ruiz [49]. BRKGA is a metaheuristic for combinatorial optimization first proposed in Gonçalves and Resende [19]. A BRKGA evolves a population of random vectors that encode solutions to the combinatorial optimization problem. BRKGA heuristics have been used to tackle a wide range of problems, such as traffic congestion [8], telecommunications [7,38,46,47], container loading problems [20], scheduling [36,21], and arc routing [33]. In many of these applications, BRKGA was shown to produce better solutions than other heuristics. It is important to note that capacity constraints are present in many of these applications, just as they are in the CMST. These two observations were the main motivations for exploring the suitability of the BRKGA for the CMST.

The main contributions of this paper are:

- We propose a new BRKGA for the CMST. We test our heuristic on 126 well-known benchmark instances. We are not aware of any other exact or heuristic algorithm for the CMST, tested on such an extensive set of instances. Using a fixed set of parameters values, our BRKGA consistently produces good results with quite modest computing requirements, independently of the type of test instance. The numerical results of our extensive computational experiments indicate that our BRKGA outperforms other heuristics for the CMST both in terms of average deviations from best-known solutions and number of best-known solutions found.
- We study various potential ingredients for the BRKGA and analyze their individual contribution to the overall algorithm. Two alternative decoders are proposed to identify the most effective way of transmitting genetic information from parents to offsprings for the CMST. We are not aware of any code in the literature that represents spanning trees and takes into account capacity constraints. Both decoders are enhanced with an improvement phase which incorporates a local search involving four different neighborhoods. Strategic oscillation is applied as well. In all cases, alternative strategies are considered and compared.
- Our BRKGA is able to improve the best-known solution for seven out of the 25 instances in our benchmark set with unknown optimal solution. This is remarkable taking into account that these are very well-known benchmark instances, which have been much-used by different authors.

The paper is organized as follows. In Section 2, we describe the CMST. In Section 3, we briefly recall biased random-key genetic algorithms. In Section 4, we describe the two decoders as well as the improvement phase that we have incorporated into our BRKGA for the CMST. Different implementation alternatives and reinforcements for the decoders and search strategies are successively presented. The individual impact of each of the proposed ingredients is analyzed in the first part of Section 5. The section concludes with the computational results of the overall proposed BRKGA. The paper ends in Section 6 with some concluding remarks.

2. Notation and problem description

Let $G=(V,E)$ be a given simple graph, with $V=\{0,1,\dots,n\}$, where 0 is a central processor and $V^+=\{1,\dots,n\}\subset V$ is a set of n

terminals. Associated with each edge $e=(i,j)\in E$ there is a cost $c_{ij}>0$. Each terminal $i, i=1,\dots,n$, has an associated demand $w_i\geq 0$.

Given a spanning tree $T\subset E$ of G , rooted at 0, the cost of T is naturally defined as $c(T)=\sum_{e\in T}c_e$. We denote by *subroot* of T any vertex directly connected to the root vertex 0. A *subtree* of T rooted at vertex $i\in V$ is denoted by T_i . A subtree $T_i\subseteq T$ where i is a subroot of T is called *s-tree*. We use the notation $V(T_i)\subset V$ to denote the set of terminals that are part of subtree T_i and $w(T_i)=\sum_{j\in V(T_i)}w_j$ to denote the *demand* of subtree T_i .

Definition 1. Given a graph $G=(V,E)$ with a distinguished vertex $0\in V$, a demand w_i associated with each terminal vertex $i\in V^+$, a nonnegative cost c_{ij} associated with each edge $(i,j)\in E$, and a capacity $Q>0$, the CMST is to find a minimum-cost spanning tree of G , rooted at 0, such that the demand of no *s-tree* exceeds Q .

3. Biased random-key genetic algorithms

Genetic algorithms with random keys, or *random-key genetic algorithms* (RKGA), were introduced by Bean [6] for solving combinatorial optimization problems involving sequencing and other optimization problems where the solution can be represented as a permutation vector. In a genetic algorithm, solutions are often referred to as *individuals* or *chromosomes*. In a RKGA individuals are represented as vectors of n random keys, i.e. n real numbers independently generated at random in the uniform interval $[0,1)$. Parameter n is problem dependent. A *decoder* is a deterministic algorithm that takes as input a vector of random keys and produces from it a feasible solution for which an objective value or fitness can be computed.

A RKGA evolves a population of random-key vectors over a number of iterations, called *generations*. The initial population is made up of p vectors of n random keys. In generation k the fitness of each individual is computed by the decoder. The population is then partitioned into two groups of individuals: a small group of $p_e<p/2$ *elite* individuals, i.e. those with the best fitness values, and the remaining set of $p-p_e$ *non-elite* individuals. To evolve the population, a new generation of individuals must be produced. This is done in three steps.

In step 1, all elite individuals of the population of generation k are copied without modification to the population of generation $k+1$. RKGAs implement mutation by introducing *mutants* into the population. A mutant is simply a vector of n random keys generated as the individuals of the initial population. The role of mutants is to inject noise into the population with the goal of avoiding getting stuck at a locally optimal solution. In step 2, p_m mutants are introduced into the population of generation $k+1$. With the p_e elite individuals and the p_m mutants accounted for in population $k+1$, $p-p_e-p_m$ additional individuals need to be produced to complete the p individuals that make up the new population. This is done in step 3 by producing $p-p_e-p_m$ offspring through the process of mating or crossover. Bean [6] selects two parents at random from the entire population and combines them using *parameterized uniform crossover* [52].

A *biased random-key genetic algorithm*, or BRKGA [19,22], differs from a RKGA in the way parents are selected for mating and what role each parent plays in crossover. Unlike in a RKGA, where parents are selected at random from the entire population, in a BRKGA each offspring is generated combining one individual selected at random from the elite partition of the population and another from the non-elite partition. As in a RKGA, repetition in the selection of mates is allowed and therefore an individual can produce more than one offspring in the same generation. Let ρ_e be the probability that an offspring inherits the random key of its elite parent. In order to try to keep its good quality, this probability is typically taken as $\rho_e>0.5$. If n is the number of random keys in an individual, then for $i=1,\dots,n$, the i -th component $o[i]$ of the offspring vector o takes

Download English Version:

<https://daneshyari.com/en/article/475446>

Download Persian Version:

<https://daneshyari.com/article/475446>

[Daneshyari.com](https://daneshyari.com)