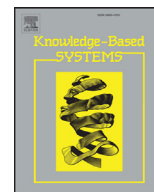




Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Knowledge-guided local search for the prize-collecting Steiner tree problem in graphs

Zhang-Hua Fu^{a,b}, Jin-Kao Hao^{b,c,*}

^aRobotics Laboratory for Logistics Service, Institute of Robotics and Intelligent Manufacturing, The Chinese University of Hong Kong, Shenzhen, 518172, Shenzhen, China

^bLERIA, Université d'Angers, 2 Bd Lavoisier, 49045 Angers Cedex 01, France

^cInstitut Universitaire de France, 1 rue Descartes, 75231 Paris Cedex 05, France

ARTICLE INFO

Article history:

Received 23 December 2016

Revised 21 April 2017

Accepted 22 April 2017

Available online xxx

Keywords:

Prize-collecting Steiner tree problem

Network design and optimization

Knowledge-guided local search

Tree transformation operators

ABSTRACT

The prize-collecting Steiner tree problem in graphs (PCSPG), as well as its rooted variant (RPCST), are target problems of the 11th DIMACS (the Center for Discrete Mathematics and Theoretical Computer Science) Implementation Challenge held in collaboration with ICERM (the Institute for Computational and Experimental Research in Mathematics). To solve these two problems, this paper proposes a knowledge-guided local search algorithm (K-ILS),¹ which integrates dedicated search strategies and explores structure information of problem instances. K-ILS uses an effective swap-vertex operator for tree transformation associated with a discriminating auxiliary evaluation function as well as several knowledge-guided perturbation strategies. K-ILS additionally employs two new path-based move operators to generate neighboring solutions. The computational results achieved on the benchmark instances of the 11th DIMACS Implementation Challenge using the same computing platform and competition rules demonstrate that K-ILS performs very well compared to the leading algorithms of the challenge. We report additional experiments to analyze the impact of the key components to the performance of the proposed algorithm.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Given an undirected graph $G(V, E)$ with a set V ($|V| = n$) of vertices and a set E ($|E| = m$) of edges, each vertex $i \in V$ is associated with a real-valued prize $p_i \geq 0$ (vertex i is a customer vertex if $p_i > 0$, and a non-customer or Steiner vertex otherwise), and each edge $e \in E$ is associated with a real-valued cost $c_e \geq 0$, then the Prize-collecting Steiner Tree Problem in graphs (PCSPG) involves finding a subtree T (with vertex set V_T and edge set E_T respectively) of G , so as to minimize the sum of the costs of its edges plus the prizes of the vertices not spanned by T , i.e., [28]:

$$\text{Minimize } f(T) = \sum_{e \in E_T} c_e + \sum_{i \notin V_T} p_i. \quad (1)$$

In addition to this basic problem, the PCSPG has some other closely related variants, such as the rooted version of the PCSPG (named RPCST for short), which considers an additional source vertex as the root which must be part of any feasible solution. More-

over, the classical Steiner tree problem in graphs (SPG) [25] is a particular case of the PCSPG, if each terminal of the graph is associated with a high enough prize, and each Steiner vertex is associated with a prize equals to zero.

The PCSPG, as well as its related variants, are known to be relevant models to formulate many important network design problems. Meanwhile, given that the classical SPG is NP-hard [30], the PCSPG is at least as difficult and computationally challenging as the SPG in the general case. Recognizing their theoretical importance and wide practical applications, the 11th DIMACS Implementation Challenge in collaboration with ICERM (from June 2013 to December 2014) was dedicated to the class of broadly defined Steiner tree problems including both the PCSPG and the RPCST [29].

The PCSPG (along with the RPCST) has been extensively investigated since it was proposed [11,37] (where it firstly appeared as the so-called *node weighted Steiner tree problem*). Many approaches have been proposed for this problem, which could be mainly classified into three categories: approximation algorithms which aim to find solutions with provable quality, exact algorithms which warrant optimality of the obtained solutions, and heuristics which seek high-quality solutions within a reasonable time frame. Moreover, effective reduction tests like those introduced in [38] were

* Corresponding author.

E-mail addresses: fu@info.univ-angers.fr (Z.-H. Fu), jin-kao.hao@univ-angers.fr (J.-K. Hao).

¹ This paper extends [15], which was presented at the workshop of the 11th DIMACS Implementation Challenge, but has not been formally published.

usually used as a pre-processing technique to transform the original graph to an equivalent reduced graph.

Among the approximation approaches, Bienstock et al. proposed a first 3-approximation algorithm [6]. Later, Goemans and Williamson [20] used a primal-dual scheme to derive a $(2 - \frac{1}{n-1})$ -approximation for the RPCST in $O(n^2 \log n)$. By trying all possible choices for the root, they obtained a $(2 - \frac{1}{n-1})$ -approximation for the PCSPG with a complexity of $O(n^3 \log n)$ [21]. Furthermore, Johnson et al. showed a $(2 - \frac{1}{n-1})$ -approximation algorithm for the PCSPG with $O(n^2 \log n)$ running time [28], which was subsequently extended to the RPCST in [35]. The new algorithm by Feofiloff et al. [12] achieved an approximation ratio of $(2 - \frac{2}{n})$ within $O(n^2 \log n)$ time. Finally, Archer et al. [3] reported the best approximation ratio with an upper bound of 1.9672.

Several exact approaches (mainly based on different integer programming formulations) have been proposed for the PCSPG. Lucena and Resende [34] presented a polyhedral cutting plane based algorithm, which solves optimally 96 of the 114 classical test instances (with up to 1000 vertices and 25,000 edges). Ljubić et al. [33] re-formulated the PCSPG in a directed formulation and implemented a branch-and-cut algorithm which yields outstanding results (with all the 189 test instances solved to optimality, including 35 real-world instances with up to 1825 vertices and 214,095 edges). Salles Da Cunha et al. [36] used a Lagrangian non delayed relax and cut (NDRC) algorithm to generate primal and dual bounds to the problem. Experimental results on a new group of difficult instances showed its competitiveness for solving the PCSPG. Meanwhile, no exact algorithm in the literature is able to optimally solve all the existing PCSPG benchmark instances. This is especially the case for a number of particularly difficult instances newly introduced for the 11th DIMACS Challenge.

Given the NP-hard nature of the PCSPG, heuristics naturally constitute another important solution approach. For instance, Canuto et al. [7] described a multi-start local search algorithm for the PCSPG, which uses a primal-dual algorithm to generate initial solutions, two basic move operators (add or remove a vertex) to obtain neighboring solutions, a path-relinking procedure to create intermediate solutions between two elite solutions, followed by a post-optimization procedure using variable neighborhood search. Klau et al. [31] developed a complex algorithmic framework, which first applies an extensive pre-processing procedure to reduce the given graph without changing the optimal solution. The reduced problem is then solved by a memetic search algorithm which combines a steady-state evolutionary algorithm and an exact subroutine for the problem on trees. The solution is finally improved by an integer linear programming (ILP) based post-optimization subroutine. Goldberg et al. [23] presented another hybrid algorithm, which uses a transgenetic algorithm (based on several coding, evolution and updating strategies) to search good solutions, and then applies a path-relinking procedure for further improvement. Biazzo et al. [5] introduced an approach derived from the cavity method, based on the zero temperature limit of the cavity equations, to form a simple (a fixed point equation solved iteratively) and parallelizable algorithm. Very recently, Akhmedov et al. [1] presented a divide-and-conquer matheuristic method, which is composed of a pre-processing procedure, a heuristic clustering algorithm and an exact solver. This algorithm was tested on a number of huge PCSPG instances transformed from biological graphs with special structures, and showed good performances for these specific instances.

The 11th DIMACS Implementation Challenge has attracted a number of new algorithms, including approximation algorithms [26], exact algorithms [2,10,18], heuristics [5,15] and hybrid algorithms [13]. These algorithms definitively advanced the state of the art on the RPCST and PCSPG. For each existing problem instance,

the previous best-known result (upper bound) was matched or improved by the best competing algorithms.

In this work, we are interested in knowledge-based methods which have been successfully applied to solve many problems, including several tree-related problems [8,24]. Specifically, we introduce the knowledge-guided iterated local search algorithm (K-ILS) for solving the PCSPG (and the RPCST). K-ILS extends our previous KTS algorithm [15] which took part in the 11th DIMACS Implementation Challenge and won several competing subcategories during the challenge. The main goal of this work aims to bring further improvements.

From an algorithmic perspective, our K-ILS algorithm belongs to the class of stochastic local search methods [27]. To be effective, K-ILS integrates dedicated and innovative ingredients for both its local optimization and perturbation phases. We identify the main contributions of the work as follows.

- First, for local optimization, K-ILS relies on an effective swap-vertex operator for tree transformations associated with an informative auxiliary evaluation function, which is shown to be extremely useful to solve a number of difficult instances with uniform or nearly uniform edge costs. For the purpose of an effective search diversification, K-ILS calls for several knowledge-guided perturbation strategies which take advantage of structure information of the given instance to direct the search towards promising areas. K-ILS additionally introduces two path-based tree transformation operators (*Connect_Customer* and *Disconnect_Customer*) to generate neighboring solutions. As such, K-ILS distinguishes itself from the existing heuristics designed for the PCSPG and RPCST problems like those presented in [1,5,7,23,31].
- Second, in terms of computational performance, K-ILS outperforms our previous KTS algorithm, which achieved an excellent performance on the RPCST and the PCSPG during the 11th DIMACS Implementation Challenge. Referring to the challenge, there are four main challenge subcategories for each problem, i.e., two *Quality Challenge* (primal bounds) subcategories regarding only the solution quality, and two *Pareto Challenge* (primal integrals) subcategories regarding both the solution quality and running time. Experimental results (based on the same computing platform and competition rules) show that, if we have used K-ILS instead of KTS to participate in the challenge, K-ILS would have won three subcategories, while being ranked tied the first place on two subcategories, and the third place on the remaining three subcategories, with respect to other competing algorithms. Although K-ILS does not change the ranks achieved by KTS during the challenge, K-ILS does obtain better solutions than KTS on a number of difficult instances, and statistically outperforms KTS.

The reminder of the paper is organized as follows. Section 2 provides a detailed description of the proposed K-ILS algorithm. Section 3 shows computational results obtained by our algorithm. Section 4 is dedicated to an analysis of some key ingredients of the K-ILS algorithm, followed by concluding comments in the last section. A study on parameter tunings is provided in the Appendix.

2. Proposed approach

2.1. General framework

Hundreds of PCSPG and RPCST instances were collected by the 11th DIMACS Implementation Challenge, which belong to different types with very different structures. According to the No Free Lunch Theorem [40], there is no algorithm which performs globally the best on all instances. Therefore, the proposed K-ILS approach (Algorithm 1) adopts different search strategies to tackle different

Download English Version:

<https://daneshyari.com/en/article/4946217>

Download Persian Version:

<https://daneshyari.com/article/4946217>

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