European Journal of Operational Research 232 (2014) 209–220

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

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

Innovative Applications of O.R.

Breakout local search for the Steiner tree problem with revenue, budget and hop constraints

Zhang-Hua Fu, Jin-Kao Hao *

LERIA, Université d'Angers, 2 Boulevard Lavoisier, 49045 Angers Cedex 01, France

ARTICLE INFO

Article history:
Received 1 February 2013
Accepted 26 June 2013
Available online 12 July 2013

Keywords:
Steiner tree problems
Network design
Constrained combinatorial optimization
Heuristic search
Adaptive perturbation

ABSTRACT

The Steiner tree problem (STP) is one of the most popular combinatorial optimization problems with various practical applications. In this paper, we propose a Breakout Local Search (BLS) algorithm for an important generalization of the STP: the Steiner tree problem with revenue, budget and hop constraints (STPRBH), which consists of determining a subtree of a given undirected graph which maximizes the collected revenues, subject to both budget and hop constraints. Starting from a probabilistically constructed initial solution, BLS uses a Neighborhood Search (NS) procedure based on several specifically designed move operators for local optimization, and employs an adaptive diversification strategy to escape from local optima. The diversification mechanism is implemented by adaptive perturbations, guided by dedicated information of discovered high-quality solutions. Computational results based on 240 benchmarks show that BLS produces competitive results with respect to several previous approaches. For the 56 most challenging instances with unknown optimal results, BLS succeeds in improving 49 and matching one best known results within reasonable time. For the 184 instances which have been solved to optimality, BLS can also match 167 optimal results.

1. Introduction

Many problems in network designing, e.g., electricity, telecommunication, heating, transportation, should determine a least cost tree spanning all or some of the vertices of a given graph (Avella, Villacci, & Sforza, 2005; Voß, 2006). These problems usually can be modeled as the Steiner tree problem (STP) or the minimum spanning tree problem (MSTP), which are generally formulated as follows: given a graph $G = (V,E)$ with vertex set $V = \{1,\ldots,n\}$ which is partitioned into two sets: a set of terminal vertices and a set of Steiner vertices, and edge set $E = \{(i,j)\colon i,j \in V, i \neq j\}$ where each edge $(i,j) \in E$ has an associated cost $c_{ij} \geqslant 0$. In some cases, a specified vertex is chosen as the root vertex. The STP consists of determining a subtree spanning all terminal vertices (including the root vertex) and possibly some Steiner vertices, so as to minimize the total cost of the obtained tree. As a special variant of the STP, for the MSTP, all vertices are terminal which should be included in any feasible solution. Unlike the MSTP that can be solved to optimality within polynomial time (Prim, 1957), the STP has proven to be NP-hard (Garey, Graham, & Johnson, 1977).

In this paper, we study an important variant of the STP: the Steiner tree problem with revenue, budget and hop constraints (denoted by STPRBH, as formulated in Costa, Cordeau, & Laporte, 2009). In this problem, in addition to the costs $c_{ij} \geqslant 0$ associated with each edge $(i,j) \in E$, there is also a revenue $r_i \geqslant 0$ associated with each vertex $i \in V$. The problem consists of determining a rooted (without loss of generality, vertex 1 is fixed as the root) subtree of graph G, so as to maximize the collected revenues, while guaranteeing that the total cost of the solution does not exceed a given budget B (budget constraint), and the number of edges from the root to any vertex in the solution subtree does not exceed an upper bound equal to h (hop constraint). As a generalization of both the STPP (STP with profits, see Johnson, Minkoff, & Phillips, 2000; Costa, Cordeau, & Laporte, 2006; Haouari, Layeb, & Sherali, 2013) and the STPH (STP with hop constraints, see Voβ, 1999; Akgün, 2011), the STPRBH is theoretically important and can be used to model many real-life problems, e.g., local access and telecommunication networks, heating or water supply systems, transportation planning, etc., in which the collected revenues should be maximized, while the available budget is limited and the reliability of the system should be guaranteed. For the STPRBH, researchers have developed various solution approaches. Respectively, Costa, Cordeau, and Laporte (2008) proposed several fast heuristics, including a greedy algorithm, a destroy-and-repair algorithm and a tabu search (TS) algorithm. Computational results for instances with up to 500 vertices and 12,500 edges were reported. In addition to the heuristics, several exact algorithms have also been proposed, including branch-and-cut (Costa et al., 2009),

* Corresponding author. Tel.: +33 2 41 73 50 76.
E-mail addresses: fu@info.univ-angers.fr (Z.-H. Fu), hao@info.univ-angers.fr (J.-K. Hao).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.06.048

branch-and-price (Sinnl, 2011). Note that all the existing exact algorithms can only solve instances with up to 500 vertices and 625 edges to optimality, for larger instances, no result has been reported by any exact algorithm.

In this paper, we are interested in the STPRBH and propose a heuristic algorithm based on the Breakout Local Search (BLS) for this problem. BLS follows the general Iterated Local Search scheme (Lourenco, Martin, & Stützle, 2003) and alternates between a neighborhood search phase and a perturbation phase. BLS has recently shown its effectiveness for solving several combinatorial optimization problems, such as sum coloring (Benlic & Hao, 2012), maximum clique (Benlic & Hao, 2013a), quadratic assignment (Benlic & Hao, 2013b), and max-cut (Benlic & Hao, 2013c). For the STPRBH, the proposed BLS algorithm integrates a probabilistic constructive procedure to generate its initial solution, a Neighborhood Search (NS) procedure based on three specifically designed move operators to discover local optima, and an adaptive perturbation strategy to continually move from one local optimum to another one, by varying its perturbations depending on the search status. As a supplementary technique, a number of high-quality solutions are stored in a solutions pool, in order to provide useful information for local optimization and perturbations. Computational results based on a set of 240 STPRBH instances, including 56 the most challenging instances with unknown optimal solutions, demonstrate the effectiveness of the proposed BLS algorithm. In particular, it succeeds in improving 49 and matching one best known results out of these 56 unsolved instances.

The rest of this paper is organized as follows: After giving some preliminary definitions in Section 2, Section 3 describes the details of the proposed BLS approach. Computational results are provided in Section 4, and Section 5 concludes this paper.

2. Preliminary definitions

In this section, we provide some preliminary definitions which are useful for a precise description of the proposed algorithm.

Definition 1. A *budget and hop constrained Steiner tree (BHS-tree)* is a rooted subtree of graph G meeting both the budget and hop constraints. A BHS-tree is also called a *feasible solution* of the problem.

Definition 2. Given a BHS-tree T, a *feasible candidate path* with respect to T is a path originating at a vertex $i \in \mathcal{V}(T)$ ($\mathcal{V}(T)$ denotes the set containing all the vertices belonging to solution T) and connecting to an uncollected profitable vertex j ($j \notin \mathcal{V}(T)$, $r_j > 0$), such that even after inserting this path to T, the obtained solution is still a BHS-tree, i.e., satisfying both the budget and hop constraints.

Definition 3. A *saturated BHS-tree* is a BHS-tree for which no feasible candidate path exists. Otherwise, the BHS-tree is an *unsaturated* (*or partial*) *BHS-tree*. Contrary to a saturated BHS-tree, an unsaturated (or partial) BHS-tree can be further extended by adding some feasible candidate path without violating the budget and hop constraints.

Definition 4. The *constrained search space* Ω is composed of all possible BHS-trees (including saturated ones or unsaturated ones). The *saturated constrained search space* $\overline{\Omega}$ is composed of all possible saturated BHS-trees which is clearly a subspace of Ω.

As detailed below, our BLS algorithm restricts its search within the saturated constrained search space $\overline{\Omega}$. By doing so, the search process focuses always on the reduced zones composed of the most promising candidate solutions.

3. The proposed BLS algorithm

In this paper, we present for the first time a Breakout Local Search (BLS) approach for solving the STPRBH, just as outlined in Algorithm 1, whose key components are presented in the following subsections.

Algorithm 1. Breakout Local Search $BLS(G,B,h)$ for the STPRBH

Require: Graph $G(V,E)$, budget limit B, hops limit h, jump magnitude $L \in [L_{min}, L_{max}]$, high-quality (elite) solution pool HSP
Ensure: The best solution found meeting both the budget and hop constraints
1: /∗ Initialization phase ∗/
2: $HSP \leftarrow InitHSP()$ /∗ Initialize HSP, see Section 3.3.2 ∗/
3: $T \leftarrow InitSolution(G,B,h)$ /∗ Construct an initial solution, see Section 3.2 ∗/
4: $T \leftarrow NS(T)$ /∗ Optimize T by neighborhood search, see Section 3.3 ∗/
5: $T^{best} \leftarrow T$
6: $L \leftarrow L_{min}$
7: /∗ Main search procedure which is iterated until the stop condition is met ∗/
8: **while** The stop condition is not met **do**
9: /∗ Perturb T with L and HSP (Section 3.4) and then improve it (Section 3.3) ∗/
10: $T' \leftarrow Perturb(T, HSP, L)$
11: $T^{*} \leftarrow NS(T')$
12: /∗ Update the best solution T^{best} found so far if needed ∗/
13: **if** T^{*} is better than T^{best} (see Section 3.3.1) **then**
14: $T^{best} \leftarrow T^{*}$
15: **end if**
16: /∗ Determine the jump magnitude L adaptively, detailed in Section 3.4 ∗/
17: **if** T^{*} is too close to T (defined in Section 3.4) **then**
18: $L \leftarrow Min(L+1, L_{max})$
19: **else**
20: $L \leftarrow Max(L-1, L_{min})$
21: **end if**
22: /∗ Update T, which serves as the starting point of a new round of search ∗/
23: $T \leftarrow T^{*}$
24: **end while**
25: **return** T^{best}

Our BLS algorithm operates within the saturated constrained search space $\overline{\Omega}$ (Section 2). The main idea of the approach for the STPRBH can be described as follows: starting from a saturated BHS-tree probabilistically constructed by the dedicated probabilistic constructive procedure (see Algorithm 1, line 3 and Section 3.2), BLS applies a Neighborhood Search (NS) procedure to reach a local optimum at first (line 4, see Section 3.3). After local optimization, BLS then attempts to continually move from one local optimum to another by employing varying perturbations, depending on the state of the search. For this purpose, an adaptive perturbation mechanism is developed, which is guided by some dedicated information of a number of recorded high quality solutions stored in the HSP (line 2 and line 10, see Sections 3.3.2 and 3.4). Each time the incumbent solution is perturbed, the NS procedure is called again to improve it to a new local optimum (line 11). If the NS procedure reaches a local optimum not far enough from the original one, BLS then perturbs it more strongly, otherwise, BLS switches to weaker perturbations subsequently (lines 16–21). This process is repeated until (1) the upper bound of the collected revenues in Eq. (2) (see