Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 35 (2008) 198–211
computers & operations research
www.elsevier.com/locate/cor

Computing all efficient solutions of the biobjective minimum spanning tree problem

Sarah Steiner*, [1], Tomasz Radzik

Department of Computer Science, King's College London, Strand, London WC2R 2LS, UK

Available online 19 April 2006

Abstract

A common way of computing all efficient (Pareto optimal) solutions for a biobjective combinatorial optimisation problem is to compute first the extreme efficient solutions and then the remaining, non-extreme solutions. The second phase, the computation of non-extreme solutions, can be based on a "*k*-best" algorithm for the single-objective version of the problem or on the branch-and-bound method. A *k*-best algorithm computes the *k*-best solutions in order of their objective values. We compare the performance of these two approaches applied to the biobjective minimum spanning tree problem. Our extensive computational experiments indicate the overwhelming superiority of the *k*-best approach. We propose heuristic enhancements to this approach which further improve its performance.

Keywords: Multiple objective programming; Combinatorial optimisation; Minimum spanning tree; *k*-best algorithm

1. Introduction

Combinatorial optimisation problems with multiple objectives are natural extensions, practical as well as theoretical, of single objective problems. Single objective problems that model real-world situations are often simplistic as there can be a number of additional factors influencing the choice of a solution. However, generalising a single objective to multiple objectives often dramatically increases computational complexity of the problem; a polynomial-time single objective problem often turns into an NP-hard problem when we add more objectives [1].

There are different definitions of the notion of optimal solutions of a multiobjective combinatorial optimisation (MOCO) problem [2]. *Efficiency*, also called *Pareto optimality*, is the most common one. An *efficient solution* is one such that there is no other solution which is better on all objectives. We consider in this paper the problem of finding all efficient solutions. In other words, without having any additional information about the relative importance of different objectives, our task is to present the decision makers with all possible solutions and let them make the final selection. For a recent comprehensive review of the MOCO literature we refer the reader to [2].

* Corresponding author. Fax: +44 20 7848 1518.
E-mail address: sarah.steiner@kcl.ac.uk (S. Steiner).
[1] This research is supported by the UK Engineering and Physical Sciences Research Council (EPSRC), award number 0230298X.

0305-0548/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2006.02.023

The underlying aim of the work which we present in this paper is to identify a fast way of computing all efficient solutions of bicriteria instances of the *minimum spanning tree* (MST) problem. The MST problem is a classic combinatorial optimisation problem, which has many direct and indirect applications and has been studied in great depth [3,4].

The idea of using a k-best algorithm within a MOCO algorithm was proposed in [5] for use in solving the bicriteria shortest path problem. It has subsequently been used in different ways for the multiobjective MST problem [6] and to MOCO problems with the max-ordering objective [7].

An alternative way of computing the non-extreme efficient solutions for the biobjective MST problem, based on the branch-and-bound method, was described by Ramos et al. [8]. The computational results presented in [8] would suggest, however that this approach might not be practical for instances other than the small ones. This partially motivates our focus on the k-best method. An additional motivation is provided by the fact that there are efficient k-best MST algorithms [9,10]. The k-best method and the branch-and-bound method seem to be the only general frameworks proposed for computing *all* non-extreme efficient solutions for the biobjective MST problem.

We have implemented both the k-best method and the branch-and-bound method and present in this paper a comparison of the actual performances of our implementations on different types of generated input instances. The results show the clear superiority of the k-best method: in all our tests the implementation of this method is significantly faster than the implementation of the branch-and-bound method, and the relative difference between the running times of these implementations sharply increases with the increase of the size of the input graphs. We propose heuristic enhancements to the k-best method which generally lead to even faster running times. An inherent aspect of the k-best method is that it may, and normally does, recompute the same solutions many times. Our enhancements aim to reduce the number of such recomputations. We also try to enhance the branch-and-bound method presented in [8] by using a stronger bounding procedure. This leads to a significant decrease in the number of partial solutions considered during the computation, but in our experiments the overall running time decreases only for input instances when the two cost functions are strongly positively correlated.

Countinho-Rodrigues et al. [11] use the basic form of the "k-best method" as part of an interactive method to solve the bicriteria shortest path problem. They focus on comparing the contribution to performance of using different k-best algorithms rather than improving the general method as we propose here.

In Section 2 we formally define the biobjective MST problem and explore the relationship between efficient solutions. In Sections 3, 4 and 6 we describe the computation of the extreme efficient solutions using the geometric method, the computation of the non-extreme efficient solutions using a k-best MST algorithm, and the way of computing the non-extreme efficient solutions based on the branch-and-bound method, respectively. In Section 5 we show a detailed example of the computation of the k-best method. In Section 7 we discuss some details of our implementations, and we present our computational results. Section 8 contains conclusions and suggestions for further work.

2. Background and definitions

2.1. The Biobjective Minimum Spanning Tree problem

The *Biobjective Minimum Spanning Tree* (BMST) problem is an extension of the classic single objective MST problem. It considers the more realistic case of more than one cost function influencing whether an edge is included in the tree. Given an undirected, connected graph $G = (V, E)$, with two cost functions $f^i: E \to \mathbb{Z}^+,\ i \in \{1, 2\}$, a pair of numbers $(f^1(e), f^2(e))$ is the pair of costs associated with an edge $e \in E$. We denote the set of all spanning trees of G by $ST(G)$. Each tree $T \in ST(G)$ has a pair of costs $F^1(T) = \sum_{e \in T} f^1(e)$ and $F^2(T) = \sum_{e \in T} f^2(e)$ associated with it. We denote by n and m, respectively, the number of vertices and the number of edges in G. In this paper, without loss of generality, we consider only integer costs.

We say that a pair of numbers (x_1, y_1) *dominates* a pair of numbers (x_2, y_2) if $x_1 \leqslant x_2$, $y_1 \leqslant y_2$ and at least one of these inequality is strict. A solution $T \in ST(G)$ is *efficient* if $(F^1(T), F^2(T))$ is not dominated by $(F^1(T'), F^2(T'))$ for any $T' \in ST(G)$. The objective of the BMST problem is to find the set of efficient solutions—the *efficient set*.

A solution can be viewed as a tree T or as the cost pair $(F^1(T), F^2(T))$ associated with it. In this paper we often refer to a solution as just the cost pair $(F^1(T), F^2(T))$. Thus, by saying that a pair of numbers (x, y) is an efficient solution, we mean that there exists $T \in ST(G)$ with $(F^1(T), F^2(T)) = (x, y)$, and there is no $T' \in ST(G)$ such that