Applied Soft Computing 28 (2015) 440–452

Contents lists available at ScienceDirect

Applied Soft Computing

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

Solving the minimum labelling spanning tree problem by intelligent optimization

S. Consoli[a,*], N. Mladenović[b], J.A. Moreno Pérez[c]

[a] National Research Council (CNR), Institute of Cognitive Sciences and Technologies, Via Gaifami 18, 95028 Catania, Italy
[b] School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, Middlesex UB8 3PH, United Kingdom
[c] Department of Computing Engineering, University of La Laguna, 38271 Santa Cruz de Tenerife, Spain

ARTICLE INFO

Article history:
Received 6 December 2013
Received in revised form 4 September 2014
Accepted 15 December 2014
Available online 24 December 2014

Keywords:
Combinatorial optimization
Graphs and networks
Minimum labelling spanning trees
Intelligent optimization
Hybrid methods
Variable neighbourhood search

ABSTRACT

Research on intelligent optimization is concerned with developing algorithms in which the optimization process is guided by an “intelligent agent”, whose role is to deal with algorithmic issues such as parameters tuning, adaptation, and combination of different existing optimization techniques, with the aim of improving the efficiency and robustness of the optimization process. This paper proposes an intelligent optimization approach to solve the minimum labelling spanning tree (MLST) problem. The MLST problem is a combinatorial optimization problem where, given a connected, undirected graph whose edges are labelled (or coloured), the aim is to find a spanning tree whose edges have the smallest number of distinct labels (or colours). In recent work, the MLST problem has been shown to be NP-hard and some effective metaheuristics have been proposed and analysed. The intelligent optimization algorithm proposed here integrates the basic variable neighbourhood search heuristic with other complementary approaches from machine learning, statistics and experimental soft computing, in order to produce high-quality performance and to completely automate the resulting optimization strategy. We present experimental results on randomly generated graphs with different statistical properties, and demonstrate the implementation, the robustness, and the empirical scalability of our intelligent local search. Our computational experiments show that the proposed strategy outperforms heuristics recommended in the literature and is able to obtain high quality solutions quickly.

1. Introduction

Today a wide range of metaheuristic methods for the solution of relevant combinatorial problems have steadily gained success. The practical challenges that the Operations Research community needs to face for the design of heuristic solution strategies are technical and scientific issues regarding the efficient tuning, adaptation, combination and hybridization of the different existing techniques [1,2]. The potential in terms of efficiency or robustness of the obtained metaheuristics is large, but the task is also quite complex. First, the performance of these algorithms depends on a number of components and parameters which need to be tuned by the user through a lengthy trial and error process every time the algorithm has to face different instances of the considered problems [3]. Second, the scientific intent consists also in comprehending the contribution of the different components with respect to the whole algorithm and at discerning the basic principles for achieving successful metaheuristics [1,2]. Consequently, there is a great interest in developing *intelligent optimization algorithms* which make use of mechanisms from machine learning, statistics and experimental soft computing, integrate exact techniques of mathematical programming, hybridize existing metaheuristics, in order to produce effective optimization strategies with high-quality performance and with completely automated parameters tuning processes [4,5,1]. In particular, the present paper considers probability-based components within self-tuned local search to solve the *minimum labelling spanning tree* (MLST) problem with state-of-the-art results. These algorithmic components allows the local search to achieve a proper balance of *diversification (exploration)* and *intensification (exploitation)* during the search process, a fundamental objective for any effective heuristic solution approach. The diversification capability of a metaheuristic refers to its aptitude of exploring thoroughly different zones of the search space in order to identify promising areas. When a promising area is detected, the metaheuristic needs to exploit it intensively to find the

* Corresponding author. Tel.: +39 095 7338 390/391; fax: +39 0332 78 9098.
E-mail addresses: sergio.consoli@istc.cnr.it (S. Consoli), nenad.mladenovic@brunel.ac.uk (N. Mladenović), jamoreno@ull.es (J.A. Moreno Pérez).

http://dx.doi.org/10.1016/j.asoc.2014.12.020
1568-4946/

relative local-optimum, but at the same time without wasting excessive computational resources. This is referred as the intensification capability of the metaheuristic. Finding a good balance between diversification and intensification is indeed an essential task for the proper effectiveness of a metaheuristic [6,7,2].

In the MLST problem we are given an undirected, labelled (or coloured) graph as input, with a label assigned to one or more edges, but with each edge having only one label allocated, and the aim is to find a spanning tree of the graph having the minimum overall number of labels [8]. The MLST problem has many real-world applications in different fields, such as in data compression [9], telecommunications network design [10], and multimodal transportation systems [11]. For example, in multimodal transportation systems there are often circumstances where it is needed to guarantee a complete service between the terminal nodes of the network by using the minimum number of provider companies [12]. This situation can be modelled as a MLST problem, where each edge of the input graph is assigned a label, denoting a different company managing that link, and one wants to obtain a spanning tree of the network using the minimum number of labels. This spanning tree will reduce the construction cost and the overall complexity of the network. A practical example in this context is given by multimodal transportation networks of large territories, from regions to states, or even continents, during humanitarian crisis events like, for example, volcanic eruptions, terrorist threats, floods, tsunamis, etc [13]. In these very delicate crisis management situations, amongst different types of human intervention, it is also necessary to reorganize dynamically the entire transportation network of the damaged area, taking into account the upcoming inaccessible or forbidden zones, and guaranteeing a minimal working transport service among main cities, hospitals, airports, principal way outs, and others, with the minimum number of different transportation carriers and companies.

It is possible to express the MLST problem in a more formal way as a network or graph problem as follows [14]:

Definition 1.1. Minimum labelling spanning tree problem:

GIVEN: A labelled connected undirected graph $G = (V, E, L)$, where V is the set of nodes, E is the set of edges, and L is the set of labels.
GOAL: Find a spanning tree T of G such that $\min|L_T|$, where L_T is the set of labels used in T.

The left graph of Fig. 1 is an example of an input graph, whose MLST solution is shown on the right.

It has been demonstrated by Xiong et al. [15] that any spanning tree of a feasible optimal solution for the MLST problem is a minimum labelling spanning tree. A feasible solution is defined as a set of labels, $C \subseteq L$, such that all edges with labels in C represent a connected subgraph of G and span all the nodes in G. If C is a feasible solution, then any spanning tree of C has at most $|C|$ labels. Therefore to solve the MLST problem, it is easier to get firstly a feasible solution with the least number of labels, and then to use any polynomial time algorithm already known in the literature to extract from the obtained feasible solution a spanning tree with the minimum number of labels [15].

The structure of the paper is as follows. In Section 2 the MLST algorithms in the literature are reviewed. In particular this section will give the details of an exact method [14], and those of the heuristics recommended in the literature [14]: greedy randomized adaptive search procedure (GRASP) and variable neighbourhood search (VNS). Section 3 describes the intelligent algorithm that we propose, which derives from the basic VNS heuristic and is extended by other complementary approaches in order to improve the effectiveness and robustness of the optimization process. Section 4 contains a computational analysis and statistical evaluation of the results. Finally, our conclusions are described in Section 5. For a survey on the basic concepts of metaheuristics and combinatorial optimization, the reader is referred to [6,16,7,1].

2. MLST algorithms in the literature

Chang and Leu [8] first introduced the MLST problem, along with the proof of its NP-hard complexity. They also presented the Maximum Vertex Covering Algorithm (MVCA), a polynomial time heuristic for the problem successively refined by Krumke and Wirth [17]. Starting from an empty graph, MVCA iteratively adds at random unused labels to the partial solution, by greedily minimizing the number of connected components at each step. The procedure continues until only one connected component is left, i.e. when only a connected subgraph is obtained.

Krumke and Wirth [17] also proved that MVCA yields a solution with a value no greater than $(1 + 2\log n)$ times optimal, where n is the total number of nodes. Later, Wan et al. [18] obtained a better bound for the greedy algorithm introduced by Krumke and Wirth [17]. The algorithm was shown to be a $(1 + \log(n-1))$-approximation for any graph with n nodes ($n > 1$).

Brüggemann et al. [19] used a different approach; they applied local search techniques based on the concept of j-switch neighbourhoods to a restricted version of the MLST problem. In addition, they proved a number of complexity results and showed that if each label appears at most twice in the input graph, the MLST problem is solvable in polynomial time.

Xiong et al. [20] derived tighter bounds than those proposed by Wan et al. [18]. For any graph with label frequency bounded by b, they showed that the worst-case bound of MVCA is the b^{th}-harmonic number H_b that is: $H_b = \sum_{i=1}^{b}(1/i) = 1 + (1/2) + (1/3) + \cdots + (1/b)$.

Subsequently, they constructed a worst-case family of graphs such that the MVCA solution is H_b times the optimal solution. Since $H_b < (1 + \log(n-1))$ and $b \leq (n-1)$ (since otherwise the subgraph induced by the labels of maximum frequency contains a cycle and one can safely remove edges from the cycle), the tight bound H_b obtained is, therefore, an improvement on the previously known performance bound of $(1 + \log(n-1))$ given by Wan et al. [18].

The usual rule of Krumke and Wirth [17] to select the label that minimizes the total number of connected components at each step, results in fast and good quality solutions. However, a difficulty arises when more than one label with same resulting minimum number of connected components is detected in a specific step. Since there may be many labels with this minimum number of connected components, the results mainly depend on the rule chosen to select a candidate from this set of ties. If the initial label from this set is chosen, the results are affected by the sorting of the labels. Therefore, different executions of the algorithm may result in different solutions, with a slightly different number of labels.

Several other heuristic approaches to the MLST problem have been proposed in the literature. For example, Xiong et al.[15] presented a Genetic Algorithm outperforming MVCA in most cases. Subsequently, Cerulli et al. [21] applied to the MLST problem the Pilot Method, which is a greedy heuristic developed by Voß et al. [22]. Considering different sets of instances of the MLST problem, Cerulli et al. [21] compared this method with other metaheuristics: Tabu Search, Simulated Annealing, and a variable neighbourhood search attempt. Their Pilot Method obtained the best results in most of the cases. It generated high-quality solutions to the MLST problem, but running times were quite large.

Xiong et al. [23] implemented modified versions of MVCA focusing on the initial label added. For example, after the labels were