Intelligent Optimization for the Minimum Labelling Spanning Tree Problem

Sergio Consoli[1(✉)], José Andrés Moreno Pérez[2], and Nenad Mladenović[3]

[1] Joint Research Centre, European Commission, Via Enrico Fermi 2749, 21027 Ispra, VA, Italy
[2] DEIOC, IUDR, Facultad de Matemáticas, Universidad de La Laguna, 4a planta Astrofisico F. Sánchez s/n, 38271 Santa Cruz de Tenerife, Spain
[3] School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, Middlesex UB8 3PH, UK
sergio.consoli@istc.cnr.it, jamoreno@ull.es, nenad.mladenovic@brunel.ac.uk

Abstract. Given a connected, undirected graph whose edges are labelled (or coloured), the minimum labelling spanning tree (MLST) problem seeks 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 heuristics have been proposed and analysed. In this paper we present preliminary results of a currently ongoing project regarding the implementation of an intelligent optimization algorithm to solve the MLST problem. This algorithm is obtained by the basic Variable Neighbourhood Search heuristic with the integration of other complements from machine learning, statistics and experimental algorithmics, in order to produce high-quality performance and to completely automate the resulting optimization strategy.

Keywords: Combinatorial optimization · Graphs and networks · Minimum labelling spanning trees · Intelligent optimization · Hybrid local search

1 Preliminary Discussion

In a currently ongoing project, we investigate a new possibility for solving the *minimum labelling spanning tree* (MLST) by an intelligent optimization algorithm. The minimum labelling spanning tree problem is a challenging combinatorial problem [1]. Given an undirected graph with labelled (or coloured) edges as input, with each edge assigned with a single label, and a label assigned to one or more edges, the goal of the MLST problem is to find a spanning tree with the minimum number of labels (or colours).

The MLST problem can be formally formulated as a network or graph problem [2]. We are 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. The purpose is to find a spanning tree T of G such that $|L_T|$ is minimized, where L_T

G. Nicosia and P. Pardalos (Eds.): LION 7, LNCS 7997, pp. 19–23, 2013.
DOI: 10.1007/978-3-642-44973-4_2,

is the set of labels used in T. Although a solution to the MLST problem is a spanning tree, it is easier to work firstly in terms of feasible solutions. A feasible solution is defined as a set of labels $C \subseteq L$, such that all the 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. Moreover, if C is an optimal solution, then any spanning tree of C is a minimum labelling spanning tree. Thus, in order to solve the MLST problem we first seek a feasible solution with the least number of labels [3].

The MLST problem was first introduced in [1]. The authors also proved that it is an NP-hard problem and provided a polynomial time heuristic, the Maximum Vertex Covering Algorithm (MVCA), successively improved in [4]. Other heuristics for the MLST problem have been proposed in the literature [2, 3, 5–9].

The aim of this paper is to present preliminary results concerning the design of a novel heuristic solution approach for the MLST problem, with the goal of obtaining high-quality performance. The proposed optimization strategy is an intelligent hybrid metaheuristic, obtained by combining Variable Neighbourhood Search (VNS) [10] and Simulated Annealing (SA) [11], with the integration of other complements in order to improve the effectiveness and robustness of the optimization process, and to completely automate the resulting solution strategy.

2 Complementary Variable Neighbourhood Search

The first extension that we introduce for the MLST problem is a local search mechanism that is inserted at top of the Variable Neighbourhood Search metaheuristic [10]. The resulting local search method is referred to as *Complementary Variable Neighbourhood Search* (COMPL).

For our implementation, given a labelled graph $G = (V, E, L)$, with n vertices, m edges, ℓ labels, each solution is encoded as a binary string, i.e. $C = (c_1, c_2, \ldots, c_\ell)$ where $c_i = 1$ if label i is in solution C, $c_i = 0$ otherwise, $\forall i = 1, \ldots, \ell$.

Given a solution C, COMPL extracts a solution from the *complementary space* of C, and then replaces the current solution with the solution extracted. The complementary space of a solution C is defined as the set of all the labels that are not contained in C, that is $(L\Delta C)$. To yield the solution, COMPL applies a constructive heuristic, such as the MVCA [1,4], to the subgraph of G with labels in the complementary space of the current solution. Note that COMPL stops if either a feasible solution is obtained (i.e. a single connected component is obtained), or the set of unused labels contained in the complementary space is empty, (i.e. $(L\Delta C) = \emptyset$), producing a final infeasible solution. Then, the basic VNS is applied in order to improve the resulting solution. At the starting point of VNS, it is required to define a suitable neighbourhood structure of size k_{max}. The simplest and most common choice is a structure in which the neighbourhoods have increasing cardinality: $|N_1(\cdot)| < |N_2(\cdot)| < \ldots < |N_{k_{max}}(\cdot)|$. In order to impose a neighbourhood structure on the solution space S, comprising all possible solutions, we define the distance between any two such solutions $C_1, C_2 \in S$, as the Hamming distance: $\rho(C_1, C_2) = |C_1 \Delta C_2| = \sum_{i=1}^{\ell} \lambda_i$, where $\lambda_i = 1$ if label i is included in one of the solutions but not in the other, and 0 otherwise, $\forall i = 1, ..., \ell$. VNS starts from an initial solution C with k increasing from 1 up to the maximum neighborhood size, k_{max}, during the progressive execution.

The basic idea of VNS to change the neighbourhood structure when the search is trapped at a local minimum, is implemented by the shaking phase. It consists of the random selection of another point in the neighbourhood $N_k(C)$ of the current solution C. Given C, we consider its kth neighbourhood, $N_k(C)$, as all the different sets having a Hamming distance from C equal to k labels, where $k \leftarrow 1, 2, \ldots, k_{max}$. In order to construct the neighbourhood of a solution C, the algorithm first proceeds with the deletion of labels from C. In other words, given a solution C, its kth neighbourhood, $N_k(C)$, consists of all the different sets obtained from C by removing k labels, where $k \leftarrow 1, 2, ..., k_{max}$. In a more formal way, given a solution C, its kth neighbourhood is defined as $N_k(C) = \{S \subset L : (|C \Delta S|) = k\}$, where $k \leftarrow 1, 2, ..., k_{max}$.

The iterative process of selection of a new incumbent solution from the complementary space of the current solution if no improvement has occurred, is aimed at increasing the diversification capability of the basic VNS for the MLST problem. When the local search is trapped at a local minimum, COMPL extracts a feasible complementary solution which lies in a very different zone of the search domain, and is set as new incumbent solution for the local search. This new starting point allows the algorithm to escape from the local minimum where it is trapped, producing an immediate peak of diversification.

3 The Intelligent Optimization Algorithm

In order to seek further improvements and to automate on-line the search process, Complementary Variable Neighbourhood Search has been modified by replacing the inner local search based on the deterministic MVCA heuristic with a *probability-based local search* inspired by a "Simulated Annealing cooling schedule" [11], with the view of achieving a proper balance between intensification and diversification capabilities. The strength of this probabilistic local search is tuned by an automated process which allows the intelligent strategy to adapt on-line to the problem instance explored and to react in response to the search algorithm's behavior [12]. The resulting metaheuristic represents the intelligent optimization algorithm that we propose for the MLST problem.

The probability-based local search is another version of the MVCA heuristic, but with a probabilistic choice of the next label to be added. It extends the basic greedy construction criterion of the MVCA by allowing moves to worse solutions. Starting from an initial solution, successively a candidate move is randomly selected; this move is accepted if it leads to a solution with a better objective function value than the current solution, otherwise the move is accepted with a probability that depends on the deterioration, Δ, of the objective function value.

Following the SA criterion, the acceptance probability is computed according to the Boltzmann function as $\exp(-\Delta/T)$, using the temperature (T) as control parameter. The value of T is initially high, which allows many worse moves to be accepted, and is gradually reduced following a specific geometric cooling schedule:

$$T_{k+1} = \alpha \cdot T_k \qquad \text{where} \begin{cases} T_0 = |Best_C|, \\ \alpha = 1/|Best_C| \in [0,1], \end{cases} \tag{1}$$

with $Best_C$ being the current best solution, and $|Best_C|$ its number of labels. This cooling law is very fast for the MLST problem, yielding a good balance between intensification and diversification. Furthermore, thanks to its self-tuning parameters setting, which is guided automatically by the best solution $Best_C$ without requiring any user-intervention, the algorithm is allowed to adapt online to the problem instance explored and to react in response to the search algorithm's behavior [12].

The aim of the probabilistic local search is to allow, with a specified probability, worse components with a higher number of connected components to be added to incomplete solutions. Probability values assigned to each label are inversely proportional to the number of components they give. So the labels with a lower number of connected components will have a higher probability of being chosen. Conversely, labels with a higher number of connected components will have a lower probability of being chosen. Thus, the possibility of choosing less promising labels is allowed. Summarizing, at each step the probabilities of selecting labels giving a smaller number of components will be higher than the probabilities of selecting labels with a higher number of components. Moreover, these differences in probabilities increase step by step as a result of the reduction of the temperature for the adaptive cooling schedule. It means that the difference between the probabilities of two labels giving different numbers of components is higher as the algorithm proceeds. The probability of a label with a high number of components will decrease as the algorithm runs and will tend to zero. In this sense, the search becomes MVCA-like.

A simple VNS implementation which uses the probabilistic local search as constructive heuristic has been tested. However, the best results were obtained by combining Complementary Variable Neighbourhood Search with the probabilistic local search, resulting in the hybrid intelligent algorithm that we propose. Note that the probabilistic local search is applied both in COMPL, to obtain a solution from the complementary space of the current solution, and in the inner local search phase, to restore feasibility by adding labels to incomplete solutions.

4 Summary and Outlook

Concerning the achieved optimization strategy, the whole approach seems to be highly promising for the MLST problem. Ongoing investigation will consist in a statistical comparison of the resulting strategy against the best MLST algorithms in the literature, in order to quantify and qualify the improvements obtained by the proposed intelligent optimization algorithm.

References

1. Chang, R.S., Leu, S.J.: The minimum labelling spanning trees. Inf. Process. Lett. **63**(5), 277–282 (1997)
2. Consoli, S., Darby-Dowman, K., Mladenović, N., Moreno-Pérez, J.A.: Greedy randomized adaptive search and variable neighbourhood search for the minimum labelling spanning tree problem. Eur. J. Oper. Res. **196**(2), 440–449 (2009)
3. Xiong, Y., Golden, B., Wasil, E.: A one-parameter genetic algorithm for the minimum labelling spanning tree problem. IEEE Trans. Evol. Comput. **9**(1), 55–60 (2005)
4. Krumke, S.O., Wirth, H.C.: On the minimum label spanning tree problem. Inf. Process. Lett. **66**(2), 81–85 (1998)
5. Xiong, Y., Golden, B., Wasil, E.: Improved heuristics for the minimum labelling spanning tree problem. IEEE Trans. Evol. Comput. **10**(6), 700–703 (2006)
6. Cerulli, R., Fink, A., Gentili, M., Voß, S.: Metaheuristics comparison for the minimum labelling spanning tree problem. In: Golden, B.L., Raghavan, S., Wasil, E.A. (eds.) The Next Wave on Computing, Optimization, and Decision Technologies, pp. 93–106. Springer-Verlag, New York (2005)
7. Brüggemann, T., Monnot, J., Woeginger, G.J.: Local search for the minimum label spanning tree problem with bounded colour classes. Oper. Res. Lett. **31**, 195–201 (2003)
8. Chwatal, A.M., Raidl, G.R.: Solving the minimum label spanning tree problem by ant colony optimization. In: Proceedings of the 7th International Conference on Genetic and Evolutionary Methods (GEM 2010), Las Vegas, Nevada (2010)
9. Consoli, S., Moreno-Pérez, J.A.: Solving the minimum labelling spanning tree problem using hybrid local search. In: Proceedings of the mini EURO Conference XXVIII on Variable Neighbourhood Search (EUROmC-XXVIII-VNS), Electronic Notes in Discrete Mathematics, vol. 39, pp. 75–82 (2012)
10. Hansen, P., Mladenović, N.: Variable neighbourhood search: principles and applications. Eur. J. Oper. Res. **130**, 449–467 (2001)
11. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
12. Osman, I.H.: Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Ann. Oper. Res. **41**, 421–451 (1993)

Springer

http://www.springer.com/978-3-642-44972-7

Learning and Intelligent Optimization
7th International Conference, LION 7, Catania, Italy,
January 7-11, 2013, Revised Selected Papers
Nicosia, G.; Pardalos, P. (Eds.)
2013, XV, 470 p. 120 illus., Softcover
ISBN: 978-3-642-44972-7