Expert Systems With Applications 50 (2016) 1–8

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A multi-operator genetic algorithm for the generalized minimum spanning tree problem

Carlos Contreras-Bolton[a,*], Gustavo Gatica[b], Carlos Rey Barra[a], Víctor Parada[a]

[a] Departamento de Ingeniería Informática, Universidad de Santiago de Chile, Av. Ecuador, 3659 Santiago, Chile
[b] Facultad de Ingeniería, Universidad Andres Bello, Universidad de Santiago de Chile, Chile

ARTICLE INFO

Keywords:
Generalized minimum spanning tree
Genetic algorithms
Multi-operator
Meta-heuristics

ABSTRACT

The generalized minimum spanning tree problem, with applications in the field of communication networks, is a computational challenge due essentially to its NP-hardness. The problem consists of finding a minimum cost spanning tree in an undirected graph whose vertices are grouped in clusters, such that the spanning tree contains only one vertex of each cluster. The algorithms that have provided the best results still do not optimally solve all instances in the literature. One of the most widely studied approaches to the problem is the use of genetic algorithms that, in all cases, use only single operators for crossover and mutation, disregarding the potential synergy of multi-operators. We present a multi-operator genetic algorithm of the genotype-phenotype class, in which the genotype is a chain of integers that represents a cluster's selected vertex. Therefore, the phenotype is a minimum cost spanning tree that is generated by means of Kruskal's algorithm and joins the vertices selected from each cluster. Two operators are used for crossover and five for mutation, three of which are local search operators. The performance of the resultant algorithm is evaluated using the most challenging instances in the literature, the results of which are compared with those of other mono-operator genetic algorithms and with the best existing results. With the 101 instances that are considered, an average error of 0.0142% is achieved, and in 83 instances, the best solution cost is obtained. Such performance is due both to the synergistic effect produced among the operators and the mutation operators working as local searches. Additionally, the results suggest that for many other combinatorial optimization problems, which have been addressed with a genetic algorithm, better results could possibly be obtained simply by using a greater number of variation operators.

1. Introduction

The generalized minimum spanning tree problem (GMSTP) is defined by means of an undirected graph whose nodes are divided into k clusters with the purpose of determining a minimum spanning cost tree that includes only one vertex of each cluster. Let there be an undirected graph $G = (V, E)$ of n vertices and m edges, and let $V_1,\ldots, V_k$ be a division of V into k clusters, such that $V = V_1 \cup V_2 \cup \ldots \cup V_k$ with $V_l \cap V_s = \emptyset, \forall\, l, s \in \{1,\ldots, k\}$ and $l \neq s$. The edges are defined only between vertices that belong to different clusters, and the cost of an edge $e = (i, j) \in E$ is given by c_e. Fig. 1 presents a GMSTP example with 5 clusters and 17 vertices and shows that, for each cluster, only one node is chosen and that the cost of the objective function of the GMSTP corresponds to the sum of the costs. The GMSTP has several applications, which include the definition of automated watering systems (Dror, Haouari, & Chaouachi, 2000), the design of wireless networks and computer networks (Myung, Lee, & Tcha, 1995), and applications in physics (Kansal & Torquato, 2001). Although the problem is very similar to the minimum spanning tree problem (MSTP), which is solved by means of the well-known polynomial algorithms of Prim (1957) and Kruskal (1956), the inclusion of the clusters concept turns it into an NP-hard problem. To show this property, Myung et al. (1995) used polynomial transformation of the node cover problem.

In the search for optimum solutions for GMSTP, exact and heuristic combination methods have been used. The exact methods consider various formulations of integer linear programming, which are approached using branch and bound, Lagrangian relaxation, network flow approach, etc. (Pop, 2009). These procedures allow the finding of good lower bounds but require excessive computing time; in some cases, it has not been possible to find a bound for large instances due to the required computing capacity (Öncan, Cordeau, & Laporte, 2008).

To achieve good solutions for most of the GMSTP instances, particular emphasis has been placed on genetic algorithms (GAs).

* Corresponding author. Tel: +56 227180927.
E-mail addresses: carlos.contrerasb@usach.cl, ccontreras.bolton@gmail.com (C. Contreras-Bolton), ggatica@unab.cl, gustavo.gatica@usach.cl (G. Gatica), carlos.reyb@usach.cl (C.R. Barra), victor.parada@usach.cl (V. Parada).

http://dx.doi.org/10.1016/j.eswa.2015.12.014
0957-4174/

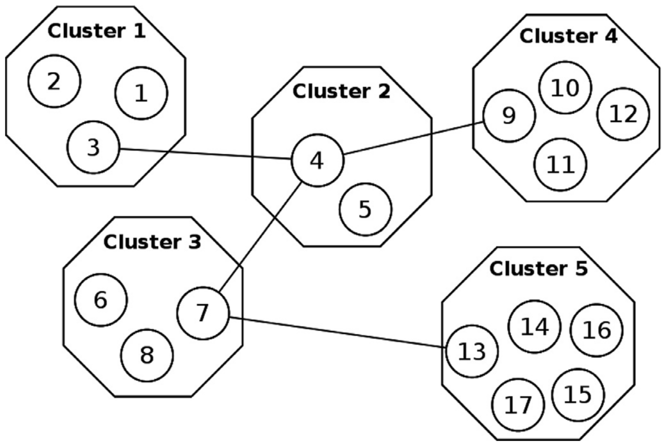

Fig. 1. Example of GMSTP with 5 clusters and 17 vertices.

Initially, a binary GA outperformed heuristics adapted from the MSTP (Dror et al., 2000); subsequently, this same GA was outperformed by, an effective random greedy algorithm that incorporates three diversification techniques: randomness, perturbation, and penalizing (Haouari & Chaouachi, 2006). A second effective GA, using an integer representation, was able to obtain better performance than PROGRES. This performance was achieved using a procedure of pre-processing the data, which reduced the number of edges and consequently the search space. This strategy is effective in reducing computing time without losing the quality of solutions. Golden, Raghavan, and Stanojević (2005) proposed a new GA with integer representation that combines a local search as the mutation operator and a heuristics procedure as the crossover operator. The best performance was achieved by this GA in a set of 211 instances. From these studies, the frequent use of a single mutation operator to complement the search of a single crossover operator can be observed.

Two of the most effective metaheuristic algorithms for the GMSTP are based on the tabu search (TS) and GRASP methods. The first approach considers three neighborhoods that allow the visiting of all the clusters the same number of times, responding to the defined aspiration level (Öncan et al., 2008). A method that has a similar computational performance to the GA of Golden et al. (2005) was numerically tested with 101 problem instances, in 72 of which the optimal solution was found. The other approach considers six different versions of GRASP (Ferreira, Ochi, Parada, & Uchoa, 2012). In the first stage of GRASP, a feasible solution is constructed with random variants based on the algorithms that solve the MSTP. In the second stage, the constructed solution is improved through iterated local search (Talbi, 2009). The solution is refined with path-relinking, which significantly improves the performance. Based on the concept of the bottleneck distance, a preprocessing of the instances was also proposed, which reduced the number of edges of the instances by an average of 14%. The results delivered new and better costs for 22 of the 101 difficult instances. In spite of producing the best results, these two methods do not provide values of the optimum solution for various instances reported in the literature.

The GAs proposed for the GMSTP have typically considered single operators for crossover and mutation and disregarded the potential synergy that typically arises when multi-operators are used. This issue gains importance because it is known that the algorithms that have reported the best results for the GMSTP still do not solve all the available instances. Both the crossover operator that generates new solutions by combining the characteristics of two selected solutions and the mutation operator that maintains the diversity of the solution in a population are responsible for defining the size of the solution space to be searched. Both operators can be complemented by generating a synergy due to their different styles of covering the space (Yoon & Moon, 2002). Making use of this potential, various multi-operator procedures have been proposed for other problems (Esquivel, Leiva, & Gallard, 1997; Hong, 1998; Hong, Kahng, & Moon, 1995; Hong, Wang, & Chen, 2000; Hong, Wang, Lin, & Lee, 2002). However, exhaustive numerical tests have been carried out mainly with non-combinatorial optimization problems such as GMSTP. Research on the use of GA multi-operators for these types of problems is just emerging (Acan, Altincay, Tekol, & Unveren, 2003; Elaoud, Teghem, & Loukil, 2010; Elsayed, Sarker, & Essam, 2011; Zhang, Wang, & Zheng, 2006).

This paper presents an approach based on a multi-operator GA for the GMSTP. The GA considers a genotype-phenotype representation (Rothlauf, 2006), in which the genotype is a string of integers that corresponds to the set of selected vertices and the phenotype corresponds to a minimum cost spanning tree that is constructed by means of Kruskal's algorithm. The multi-operators are considered in the crossover and in the mutation by specifically considering two crossover and five mutation operators, three of which are local searches. This allows the algorithm to be competitive with respect to the best algorithms in the literature, both in terms of quality of the solution and in computing time.

The following section presents the proposed multi-operator GA, Section 3 presents and discusses the results of the experiment, and the last section presents the conclusions of the work.

2. Proposed genetic algorithm

In this section, the evolutionary process, the genotype/phenotype representation, the generation of the initial population, the evaluation function, and the crossover and mutation genetic operators are described.

2.1. Evolutionary process

The evolutionary process consists of the creation of new populations for a fixed number of generations. Each element of the integer representation of an individual references the label of the selected vertex in a cluster that, together with the other selected vertices, compose the input to Kruskal's algorithm to obtain the minimum spanning tree covering those vertices. Initially, a population with feasible solutions is constructed and, in each new generation, new solutions are constructed using the selection, crossover and mutation operators. The selection is made in a tournament of five individuals (Eiben & Smith, 2007), the crossover considers two operators, and five operators participate in the mutation. Elitism is also implemented, in which the best current parents replace 10% of the worst individuals generated in each generation. The generational process is repeated until a predefined maximum number of generations is reached or until either the optimum value or the lower bound is obtained. If the maximum number of generations is reached, the GA delivers the best solution. This evolutionary process is described in Algorithm 1. In the initial stage, a population of individuals is generated and evaluated using the instructions in lines 2–6. Subsequently, the main loop of the algorithm, presented in lines 7–22, is responsible for generating a new population from the current one via a probabilistic selection of variation operators, as described in lines 10 and 11.

2.2. Representation of the problem

A representation based on that of Haouari and Chaouachi (2006) is proposed as the genotype, using a string of integers of length k, where the kth value corresponds to the selected vertex. The kth position can take a value in the [1, t] interval, where t is