A GA-VNS based algorithm for the multi-objective spanning tree problem

1st BOUMESBAH Asma
dept. Operational Research.
University of Sciences and Technology Houari Boumediene.
Algiers, Algeria
aboumesbah@usthb.dz

2nd CHERGUI Mohamed El-Amine
dept. Operational Research.
University of Sciences and Technology Houari Boumediene.
Algiers, Algeria
mchergui@usthb.dz

***Abstract*—The Multi-Objective Minimum Spanning Tree problem ($MOST$) has been shown to be NP-hard even with two criteria. In this study we propose a hybrid GA-VNS algorithm that exploits the advantages of both "Non-dominated Sorting Genetic Algorithm" (NSGA-II) and "Variable Neighborhood Search" (VNS) metaheuristics to find as good an approximation as possible to the Pareto front of $MOST$ problem. Experimental studies provide the efficiency of the method which produces solutions as close as possible to the Pareto optimal front.**

***Index Terms*—Minimum spanning tree, Multiple objective linear optimization, Combinatorial optimization, Non-Sorting Genetic Algorithm, Variable Neighborhood Search.**

INTRODUCTION

The purpose of multi-objective optimization problems is to find the best solutions with regard to multiple and often conflicting objectives. The Multi-Objective Minimum Spanning Tree ($MOST$) is an important example of such problems and arises in many systems such as communication networks, electric power systems, and drain systems. This problem is well-known to be NP-hard [2], [6], therefore when the number of edges of the given graph is large, exact algorithms to solve $MOST$ problem are slow to converge and become impractical that is why approximate methods have been successfully used to produce solutions as close as possible to the Pareto optimal front.

The main objective of this paper is to provide a method to deal with $MOST$ problem able to generate a good approximation of the Pareto front. Based on an hybridization between NSGA-II [4] and VNS [7], the GA-VNS algorithm (Genetic Algorithm-Variable Neighborhood Search) adopts novel two point crossover operator and a local search heuristic as a mutation operator. Then, the based VNS algorithm is called to improve the best front obtained by NSGA-II. Through the computational studies, it seems that the proposed GA-VNS algorithm does not have the limitations of previous algorithms because of its speed, its scalability to solve $MOST$ problem considering complete graphs with more than 200 nodes, and its ability to find both the supported and non-supported solutions.

I. PRELIMINARIES

Given a connected and undirected simple graph $G = (V, E)$ of order n, where $V = \{v_1, v_2, \dots, v_n\}$ is a set of vertices and $E = \{e_1, e_2, \dots, e_m\}$ is a set of edges and each edge $e_i \in E$, $i = \overline{1,m}$, is valued by a cost-vector $C_{e_i} = (C_{ik})$, $k = \overline{1,r}$, $r \geq 2$. Let $T \subset E$ a spanning tree of G, the cost-vector of T is given by $C_k(T) = \sum_{e_i \in T} C_{ik}$, $k = \overline{1,r}$. We note $C(T) = (C_k(T))_{k=\overline{1,r}}$.

We say that vector $C(T)$ dominates another vector $C(T')$ if $C_k(T) \leq C_k(T')$ $\forall k = \overline{1,r}$ and $C_k(T) < C_k(T')$ for at least one index $k \in \{1, ..., r\}$. A spanning tree T of G is efficient if there is no spanning tree T' of G such that $C(T')$ dominates $C(T)$.

The set of all Pareto optimal spanning trees is called the efficient set and the set of non-dominated trees is called the Pareto front.

II. GA-VNS ALGORITHM DESCRIPTION

One of the main reasons why population based metaheuristics (Genetic Algorithms, Swarm Optimozation, etc.) are more used than trajectory based metaheuristics (VNS, Tabu Search, etc.) in the multi-objective optimization is because the former uses a set of solutions while the later traditionally uses only one solution. We propose to overcome this situation by considering the approximate Pareto front found during the search process by NSGA-II as the incumbent solution to a multi-objective minimum spanning tree problem for the VNS metaheuristic.

A. Procedure: Encoding

Initially the edges of the graph $G = (V, E)$ having n vertices and m edges, are numbered. The direct encoding of a chromosome (corresponding to a spanning tree) is a set of $n-1$ dimension, wherein each element represents the associated number of an edge. In the article [8] the authors point out that the direct encoding is better and more efficient than Purfer-based encoding, contrary to what Zhou et *al*. showed in their paper [9]. We note $E(T)$ the set of the edges representing a spanning tree T.

B. Procedure: Generation of the initial population

The first population of size p of spanning trees is generated using:
1- the optimal spanning trees corresponding to each criterion,
2- random generation with aggregation of criteria,
3- applying Kruskal's algorithm by randomly selecting edges of graph G.

C. Procedure: Crossover

1) New two point crossover: it is very similar to single point crossover except that two cut-points are randomly generated instead of one. The obtained offspring chromosomes do not correspond necessarily to spanning trees. To overcome this infeasibility, we have introduced procedures of rearrangement of edges for each chromosome offspring which is not a spanning tree.
The two point crossover is described as follows:
Choose edges of $A4$ in decreasing order of the sums of costs

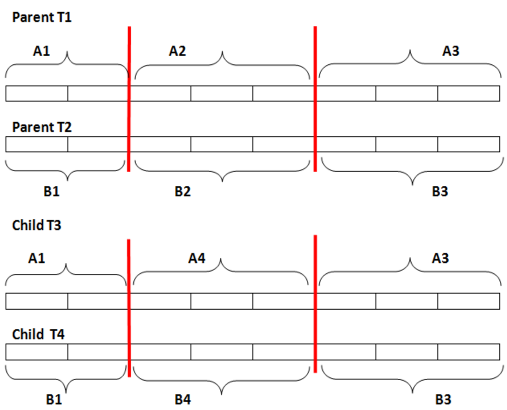

Fig. 1. Two point crossover

in the set: $E(T2) \cup E \setminus (A1 \cup A3)$. Those of $B4$ in decreasing order of the sums of costs in the set: $E(T1) \cup E \setminus (B1 \cup B3)$ Every offsprings obtained by the crossover operator are spanning tree therefore no correction has to be made. Furthermore, the experimentation provides better results keeping the crossover operator unlike expressed by the authors in [3].

D. Mutation operator

This consists to choose randomly a gene which is not in the offspring chromosome $T1$, i.e., an edge $e = \{x, y\}$ of $G \setminus T1$, and replace it by an another edge f whose cost-vector dominates the one of e. The edge f is in the current spanning tree and belongs to the elementary chain Γ in $T1$ connecting the vertices x and y, $\Gamma \cup e$ is then a cycle of G. This allows finding another chromosome (i.e., a spanning tree) whose cost-vector dominates the offspring cost-vector. If there is no edge in the chain Γ which cost-vector dominates the one of the edge e, this last one is replaced by any edge chosen randomly from the chain Γ provided that his cost-vector is not dominated by the one of the edge e. Doing this, a new spanning tree $T2$ is generated.

E. Selection operator

The selection operator consists in proceed first with the partition of the population of size $2p$ at least, composed of parents and offspring in classes, the first containing the non-dominated individuals, the second still contains non-dominated individuals after removing elements of the first class, and so on. The new population consists in choosing individuals respecting to the order of dominance classes. Individuals of the class k have to complete the p size of the new population are selected according to a distance of "crowding".

F. Improvement of the first class of NSGA-II

Starting from the set SND of non-dominated solutions of the last population of NSGA-II, neighboring solutions are generated through a VNS based algorithm to obtain a novel SND set. The update of the latter is done according to the Pareto dominance relation and at each iteration, the obtained populations have not necessarily the same size. Hence, an improved approximation with a higher number of non-dominated solutions is maintained applying a VNS based algorithm as follows:
SND: set of non-dominated solutions constructed by NSGA-II.
$N_k, k = 1, 2, 3$: set of neighborhood structures which is called $k - opt$ of current solution T as well. In the $1 - opt$ neighborhood a local search is applied, in the $k - opt$ each of neighbors $N_k(T)$ can be reached changing exactly k edges which belong to different cycles of G.
The choice of the edges to be changed in a given spanning tree T, is done in the following way:
1- Choose randomly k edges of $G \setminus T$.
2- For each edge $e \in G \setminus T$ (e is one of the k edges already chosen) edges , find the unique chain that connects its extremities in T.
3- If there exists an edge f such that $C(f)$ dominates $C(e)$, then $T := T \setminus \{e\} \cup \{f\}$.
Using these three structures, we ensure the feasibility of the neighboring solutions (i.e., spaniing trees) obtained. The steps of VNS based are described in the following:
Step 1: While stopping criterion is not satisfied repeat
Step 2: Select randomly an unvisited solution T of SND and mark T as visited solution;
Step 3: Select at random a neighborhood structure $N_k, k = 1, 2, 3$;
Step 4: Determine randomly a solution T' from $Nk(T)$;
Step 5: For each neighbor $T'' \in N_k(T')$ do
Step 5.1: Evaluate the solution T'';
Step 5.2: If the cost-vector $C(T'')$ of the resultant spanning tree T'' is not dominated by the elements of the SND set, then $SND := SND \cup \{T''\}$;
Step 6: If all the solutions of SND are marked as visited

then, Return SND.

III. Comutational Experiments

The goal of this work is to demonstrate that combining the two metaheuristics NSGA-II and VNS, it can be obtained a good approximation of the Pareto front of $MOST$ problem. To do this the algorithm is compared with the $MOST$-algorithm described in [1] which generates the set of all non-dominated spanning trees. The experimental study is performed under Matlab 2014a on an hp laptop, Intel core i5, 8GB of Ram. The graphs are randomly generated as described by Erdos-Renyi in [5] who proposed to the user to fix the number of vertices of the graph and the probability that an edge exists between two vertices, and the cost-vectors of dimensions 3 and 5, which are uniformly distributed in the interval $[-50, 100]$. We run GA-VNS algorithm and $MOST$-algorithm for ten instances of same dimension and then, the results are compared on average using the proportional measure which is defined as follows. We denote $SND/ASND$ the proportion of solutions of the set $ASND$ belonging to the exact set of non-dominated solutions found by $MOST$-algorithm. From this comparative table I we can see that over $70,47\%$ of non-dominated solutions have been found by the GA-VNS algorithm.

n	m	r	$SND/ASND$		
			avg	min	max
20	[46,55]	3	67,14%	49,36%	77,81%
30	[56,65]	3	70,89%	50,79%	81%
50	[65,75]	3	68,60%	58,12%	79,06%
20	[46,55]	5	76,27%	56,26%	84,16%
30	[56,65]	5	71,28%	47,70%	88,12%
50	[65,75]	5	68,64%	42,89%	86,24%

TABLE I
Comparaison between $MOST$-Algorithm and GA-VNS algorithm

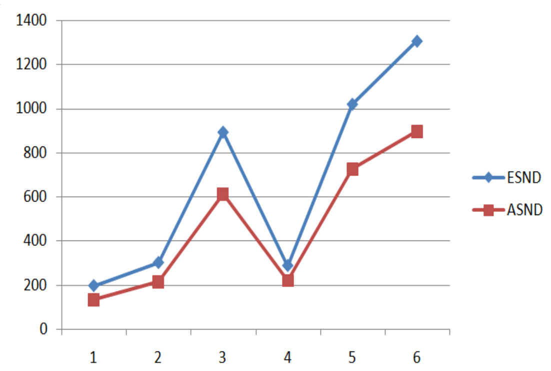

Fig. 2. Comparaison between $MOST$-Algorithm and GA-VNS algorithm

Another comparative study between our algorithm and the algorithm presented KEA in [3] is done with the same graph generator and the results are presented in table II.

n	r	$\lvert R\rvert$	$\lvert ND1\rvert$	$\lvert R\cap ND1\rvert$	$\lvert ND2\rvert$	$\lvert R\cap ND2\rvert$
$K80$	3	1100	1053	968	989	450
$K100$	3	1428	1324	1041	1043	472
$K200$	3	1238	1277	1189	686	163
$K80$	5	1320	1288	1136	600	158
$K100$	5	1476	1295	1210	870	503
$K200$	5	2027	1783	1392	960	370

TABLE II
Comparaison between GA-VNS and KEA algorithms

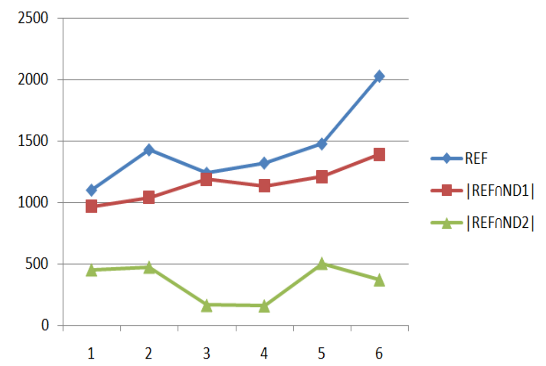

Fig. 3. Comparaison between GA-VNS and KEA algorithms

For each problem instance, we compare the non-dominated soltions obtained by the two algorithms. We denoted by $ND1$ and $ND2$ the sets of non-dominated solutions obtained by the algorithms GA-VNS and KEA respectively, and by $|ND1|$ and $|ND2|$ cardinals of these two sets. The set constructed by gathering all non-dominated sotutions obtained by the two algorithms, is named "reference set" denoted by (R). The performance of an algorithm is then measured in terms of quality of the solutions obtained by GA-VNS with respect to the solutions in R. In this paper we used the cardinal measure which is defined as follow:
Cardinal measure: For the two algorihms GA-VNS and KEA, we compute the number of obtained non-dominated solutions that belong to the reference set, i.e. $|R \cap ND1|$ and $|R \cap ND2|$.
Table II presents the comparaison amoung GA-VNS and KEA algorithms regarding the cardinal measure. Each algorithm is run ten times for all the six instances of the problem. For this fact, we have found useful to group in each line of the Table II the average results found among all the runs. We note that, the GA-VNS and KEA generate their own sets of non-dominated spanning trees, which do not necessarily belong to R.
It is worth observing that for all the instances, the GA-VNS algorithm out performs the KEA algorithm taking 81% of the set R. Hence, it determines a greater number of solutions in the set R compared to KEA algorithm whose contribution is only 25%.

IV. Conclusion

In this study, a suitable GA-VNS algorithm based on a hybridization of the two well known metaheuristics NSGA-II and VNS is developed. The results obtained by this new hybrid algorithm are efficient and show the superiority of this algorithm compared to other existing methods knwing that the KEA algorithm is one of the most pwerful. This encourages us to apply the GA-VNS algorithm to other combinatorial optimization problems and to develop, in future studies of research, hybrid algorithms with other meta-heuristic methods.

References

[1] A. Boumesbah, M.E-A. Chergui, AN EXACT METHOD TO GENERATE ALL NONDOMINATED SPANNING TREES, RAIRO Operations Research, DOI: 10.1051/ro/2016060, 50, 857-867 (2016).

[2] P. M. Camerini, G. Galbiati and F. Maffioli, The complexity of weighted multi-constrained spanning tree problems. Colloquium the Theory of Algorithms, Colloquium. L. Lovàsz,Ed. Amsterdam: North-Holland. 53-101 (1984).

[3] M. David-Moradkhan and W. Browne, Evolutionary Algorithms for the Multi Criterion Minimum Spanning Tree Problem, Y. Tenne and C.-K. Goh (Eds.): Computational Intel. in Expensive Opti. Prob., ALO 2, 423-452 (2010).

[4] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, A fast and elitist multi-objective genetic algorithm for multi-objective optimization. IEEE Trans. Evol.Comut. 6, 181-197 (2002).

[5] P. Erdös and A. Rényi, On the evolution of random graphs, Publications of the Mathematical Institute of the Hungarian Academy of Sciences. 5 (1960).

[6] H. W. Hamacher and G. Ruhe, On spanning tree problems with multiple objectives. Anals of Operations Research. 52, 209-230 (1994) .

[7] P. Hansen and N. Mladenović, Variable neighborhood search. Computers and Operations Research. 24 (11), 1097-1100 (1997).

[8] J. Knowles and D. Corne, Enumeration of pareto optimal multi-criteria spanning trees - a proof of the incorrectness of Zhou and Gen's proposed algorithm, European Journal of Operational Research, tm. 143, pp. 543-547 (2002).

[9] G. Zhou and M. Gen, Genetic algorithm approach on multi-criteria minimum spanning tree problem, European Journal of Operational Research, tm. 114, pp. 141-152 (1999).