Mathematica Balkanica

New Series Vol. 24, 2010, Fasc.3-4

A Metaheuristic Approach to Solving the Generalized Vertex Cover Problem

Marija Milanović

The topic is related to solving the generalized vertex cover problem (GVCP) by genetic algorithm. The problem is NP-hard as a generalization of well-known vertex cover problem which was one of the first problems shown to be NP-hard. The definition of the GVCP and basics of genetic algorithms are described. Details of genetic algorithm and numerical results are presented in [8]. Genetic algorithm obtained high quality solutions in a short period of time.

AMS Subj. Classification: 90C27, 05C85, 90C59

Key Words: vertex cover, genetic algorithm, evolutionary approach, combinatorial optimization, graph algorithms

1. **The vertex cover problem**

In 1972 Karp [5] showed that 24 diverse problems from graph theory and combinatorics are NP-complete. The vertex cover problem (VCP) was one of them. The VCP is defined over an undirected graph $G = (V, E)$ and searches for a set of vertices S such that for each edge $e \in E$ at least one of its endpoints belongs to S and $|S|$ is as small as possible.

The vertex cover problem has many real-world applications. Examples of the areas where this problem occurs are communications, civil and electrical engineering, and bioinformatics (the vertex cover problem finds applications in the construction of phylogenetic trees, in phenotype identification, and in analysis of microarray data).

Up to now, numerous researchers have studied this problem, mostly from the aspect of approximation. Nevertheless, there is significantly smaller number of researchers who have given experimental results.

In [2] Gilmour and Dras have presented a framework that allows the exploitation of existing techniques and resources to integrate structural knowledge of the problem into the ant colony system metaheuristic, where the structure was determined through the notion of kernelization from the field of parameterized complexity. They have given experimental results using vertex cover as the problem instance.

In [6] Kotecha and Gambhava have presented a hybrid genetic algorithm for solving the vertex cover problem. They have added local optimization technique to a genetic algorithm, and also developed a new heuristic vertex crossover operator especially for the vertex cover problem.

Pelikan et. al. [11] have analyzed the hierarchical Bayesian optimization algorithm (hBOA) on vertex cover for standard classes of random graphs and transformed SAT instances. The performance of hBOA was compared with that of the branch-and-bound problem solver, the simple genetic algorithm, and the parallel simulated annealing.

Richter [12] has introduced a novel stochastic local search algorithm for the vertex cover problem and tested its performance on a suite of problems drawn from the field of biology, and also on the commonly used DIMACS benchmarks for the related clique problem.

2. Generalizations

Several papers dealing with various generalizations of the vertex cover problem are [1, 3, 4, 10].

Broersma et. al. [1] have introduced the minimum weight processor assignment problem (MWPAP), showed that the MWPAP is a generalization of several problems known in the literature, including minimum multiway cut, graph k-colorability, and minimum (generalized) vertex covering. They have analyzed the complexity of the MWPAP, and showed that it is NP-hard, even when restricted to very specific classes of instances. For a number of classes of instances they have shown that the MWPAP is a polynomial.

In [3] Guo et. al. have investigated parameterized complexity of the following problems which all represent different generalizations of the vertex cover problem: connected vertex cover, capacitated vertex cover, and maximum partial vertex cover. They have shown that, with the size of the desired vertex cover as a parameter, the connected vertex cover and the capacitated vertex cover are both fixed-parameter tractable while the maximum partial vertex cover is W[1]-hard.

In his thesis [10], Moser has developed exact algorithms for the same three generalizations of the vertex cover problem.

3. The generalized vertex cover problem

In this paper, the formulation from [4] is chosen.

Let $G = (V, E)$ be an undirected graph, with three numbers $d_0(e) \geq d_1(e) \geq d_2(e) \geq 0$ for each edge $e \in E$. The solution is a subset $S \subseteq V$ and $d_i(e)$ represents the cost contributed to the solution by the edge e if exactly i of its endpoints are in the solution. The cost of including a vertex v in the solution is $c(v)$. The solution has a cost that is equal to the sum of the vertex costs and the edge costs. The generalized vertex cover problem (GVCP) is to compute a minimum cost set of vertices.

One of the problems that were a motivation for the generalized vertex cover problem is presented in [7]: given a budget that can be used to upgrade vertices, the goal is to upgrade a vertex set such that in the resulting network the minimum cost spanning tree is minimized.

In [4] Hassin and Levin have studied the complexity of GVCP with the costs $d_0(e) = 1$, $d_1(e) = \alpha$, $d_2(e) = 0$ for every $e \in E$ and $c(v) = \beta$ for every $v \in V$ for all possible values of α and β. They have also provided 2-approximation algorithms for the general case.

In the special case when $d_0(e) = 1$, $d_1(e) = d_2(e) = 0$ for every $e \in E$ and $c(v) = 1$ for every $v \in V$, the GVCP is reduced to the VCP. Thus, the generalized vertex cover problem is NP-hard as a generalization of the vertex cover problem which is proved to be an NP-hard problem. Hassin and Levin have also proved that there are some cases when the GVCP can be solved in polynomial time. Those cases are:

- $\frac{1}{2} \leq \alpha \leq 1$
- $\alpha < \frac{1}{2}$
- $\alpha < \frac{1}{2}$ and there exists an integer $d \geq 3$ such that $d(1-\alpha) \leq \beta \leq (d+1)\alpha$

4. **Mathematical formulation**

Let $G = (V, E)$ be an undirected graph. For every edge $e \in E$ three numbers $d_0(e) \geq d_1(e) \geq d_2(e) \geq 0$ are given and for every vertex $v \in V$ a number $c(v) \geq 0$ is given.

For a subset $S \subseteq V$ denote $\overline{S} = V \backslash S$, $E(S)$ is the set of edges whose both end-vertices are in S, $E(S, \overline{S})$ is the set of edges that connect a vertex from S with a vertex from $\overline{S}$, $c(S) = \sum_{v \in S} c(v)$, and for $i = 0, 1, 2$ $d_i(S) = \sum_{e \in E(S)} d_i(e)$ and $d_i(S, \overline{S}) = \sum_{e \in E(S,\overline{S})} d_i(e)$.

The generalized vertex cover problem is to find a vertex set $S \subseteq V$ that minimizes the cost $c(S) + d_2(S) + d_1(S, \overline{S}) + d_0(\overline{S})$. Thus, the value $d_i(e)$ represents the cost of the edge e if exactly i of its endpoints are included in the solution, and the cost of including a vertex v in the solution is $c(v)$.

v	1	2	3	4
c(v)	1	2	3	4

TABLE 1. c(v) costs

An integer programming formulation of the generalized vertex cover problem, introduced in [4], is shown below.

(1)
$$min \qquad \sum_{i=1}^{n} c(i)x_i \;+ \sum_{(i,j)\in E} \Big(d_2(i,j)z_{ij} + d_1(i,j)(y_{ij}-z_{ij}) + d_0(i,j)(1-y_{ij})\Big)$$

subject to:

(2)
$$y_{ij} \le x_i + x_j \qquad for\ every\ (i,j) \in E$$

(3)
$$z_{ij} \le x_i \qquad for\ every\ i \in V, (i,j) \in E$$

(4)
$$z_{ij} \le x_j \qquad for\ every\ j \in V, (i,j) \in E$$

(5)
$$x_i, y_{ij}, z_{ij} \in \{0,1\}$$

x_i is an indicator variable that is equal to 1 if vertex i is included in solution; y_{ij} is an indicator variable that is equal to 1 if at least one of the vertices i and j is included in the solution, z_{ij} is an indicator variable that is equal to 1 if both i and j are included in the solution.

Example 1 Let $|V| = 4$ and $|E| = 5$. The costs $c(v)$ of the including vertices in the solution are given in Table 1. For every edge its end-points and d_0, d_1, and d_2 costs are given in Table 2. The graph is shown in Figure 1.

The optimal objective value in this example is 15 and the generalized vertex cover consists of only one vertex (vertex 1). The corresponding vertex cost $c(S) = c(1) = 1$ and the edge costs are $d_1(S,\overline{S}) = d_1(1,2) + d_1(1,3) + d_1(1,4) = 3 + 4 + 2 = 9$, $d_0(\overline{S}) = d_0(2,3) + d_0(3,4) = 3 + 2 = 5$ and $d_2(S) = 0$. The optimal solution is obtained by CPLEX solver using the integer programming formulation (1)-(5). $\diamondsuit$

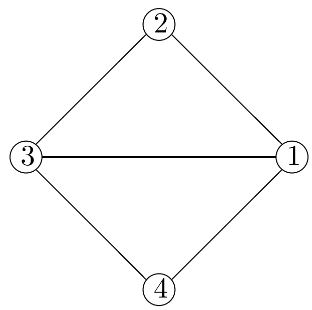

Figure 1. The graph

start	end	d_0	d_1	d_2
1	2	5	3	2
1	3	4	4	3
1	4	5	2	2
2	3	3	2	1
3	4	2	2	2

Table 2. Edges and their d_0, d_1, d_2 costs

5. Genetic algorithm

The genetic algorithms (GAs) are a class of optimization algorithms. The GAs solve problems by creating a population consisting of individuals. An individual represents an encoded solution of the problem. Individuals from the current population are evaluated by using a fitness function to determine their qualities. By applying genetic operators of selection, crossover, and mutation on the current generation, the next generation of individuals is produced. The process is repeated until stopping criterion is satisfied.

Individuals in the initial population are generated randomly or heuristically. The appropriate representation (encoding scheme) of the problem is the most important factor for a successful application of GA.

the selection operator favors better individuals to survive through the generations. The crossover operator provides a recombination of the genetic material by exchanging portions between the parents' genetic codes with the chance that good solutions can generate even better ones. The mutation causes sporadic and random changes by modifying individual's genetic material with some small probability.

The outline of genetic algorithm is shown below:

```
Input_data();
Population_init();
while not Stopping_criterion() do
    Objective_function();
    Fitness_function();
    Selection();
    Crossover();
    Mutation();
endwhile
Output_data();
```

In [9], a detailed description of GA can be found.

6. Numerical experiments

In [8], the very first numerical results for the GVCP were presented. The problem was solved by using an evolutionary based approach. A binary representation and standard genetic operators were used along with the appropriate objective function. The experiments were carried out on randomly generated instances with up to 500 vertices and 10000 edges. The performance of the genetic algorithm was compared with the CPLEX solver and 2-approximation algorithm based on LP relaxation.

The integer programming formulation (1)-(5) was implemented and tested by the CPLEX solver in order to obtain optimal solutions. Also, 2-approximation algorithm introduced in [4] was implemented and tested. The 2-approximation algorithm is based on LP relaxation of (1)-(5) integer program, fixing all relaxed binary variables with values greater or equal then $\frac{1}{2}$ to 1. Other variables (with relaxed value less than $\frac{1}{2}$) were fixed to 0. For solving this LP relaxation, the CPLEX solver was also used.

The generation of instances was performed in such way that the cases solvable in polynomial time were omitted. A detailed description of this process can be found in [8].

Numerical experiments have shown that genetic algorithm outperformed both the CPLEX solver and the 2-approximation heuristic.

7. Conclusions

The experimental results from [8] indicate that the genetic algorithm approach seems to be a good candidate for solving the GVCP.

Future research will be directed to parallelization of the GA, incorporation in exact methods, and application for solving similar problems.

References

[1] H.J.Broersma, D.Paulusma, G.J.M.Smit, F.Vlaardingerbroek, G.J.Woeginger, The computational complexity of the minimum weight processor assignment problem. *Lecture Notes in Computer Science,* **3353**, 2004, 189–200.

[2] S.Gilmour, M.Dras, Kernelization as heuristic structure for the vertex cover problem. *Lecture Notes in Computer Science*, **4150**, 2006, 452–459.

[3] J. Guo, R. Niedermeier, S. Wernicke, Parameterized complexity of generalized vertex cover problems. *Lecture Notes In Computer Science*, **3608**, 2005, 36–48.

[4] R. Hassin, A. Levin , The minimum generalized vertex cover problem. *ACM Transactions on Algorithms*, **2**, 2006, 66–78.

[5] R. M. Karp, Reducibility among combinatorial problems. *Complexity of Computer Computations, Plenum Press*, 1972, 85-103.

[6] K. Kotecha, N. Gambhava , A hybrid genetic algorithm for minimum vertex cover problem. *Proceedings of the First Indian International Conference on Artificial Intelligence*, 2003, 904–913.

[7] S.O. Krumke, M.V. Marathe, H. Noltemeier, R. Ravi, S.S. Ravi, R. Sundaram, H.C. Wirth, Improving minimum cost spanning trees by upgrading nodes. *Journal of Algorithms*, **3**, 1999, 92-111.

[8] M. Milanović, Solving the generalized vertex cover problem by genetic algorithm. *Computing and Informatics*, in press.

[9] M. Mitchell, *Introduction to genetic algorithms.* MIT Press, Cambridge, Massachusetts, 1999.

[10] H. Moser, *Exact algorithms for generalizations of vertex cover.* M.Sc. thesis, Friedrich-Schiller-University Jena, Faculty of Mathematics and Informatics, 2005.

[11] M.Pelikan, R.Kalapala, A.K.Hartmann, Hybrid evolutionary algorithms on minimum vertex cover for random graphs.*Proceedings of the Genetic and Evolutionary Computation Conference-GECCO(2007)*,547–554.

[12] S. Richter, M. Helmert, C. Gretton, A stochastic local search approach to vertex cover. *Lecture Notes in Computer Science*, **4667**, 2007, 412–426.

Faculty of Mathematics,
University of Belgrade,
Studentski trg 16/IV
11 000 Belgrade, Serbia
E-Mail: marija.milanovic@gmail.com