Minimum Weight Vertex Cover Problem

PROGETTO

Intelligenza Artificiale :: Artificial Intelligence

Prof. Mario Pavone

21^{st} January, 2025

1 Weight Vertex Cover Problem

The minimum weight vertex cover (MWVC) problem is a fundamental graph problem with many important real-life applications such as, for example, in wireless communication, circuit design and network flows. Given an undirected graph $G = (V, E)$ with weights on its vertices, the **goal** of the minimum weight vertex cover problem is to find among all subsets $S \subset V$ that are vertex covers a subset S^* for which the sum of the weights of the vertices is minimal. Note that the MWVC is NP-complete problem.

A problem instances (G, ω) of the MWVC problem is a tuple that consists of an undirected graph $G(V, E)$, where V is the set of vertex, E is the set of edges, and a function $\omega : V \rightarrow \Re^+$ that associates a positive weight value $\omega(v)$ to each vertex $v \in V$. The MWVC problem can be formally defined as follows:

$$\textbf{minimize} \quad \omega(S) = \sum_{v \in S} \omega(v),$$

$$\text{such that: } \forall \ (v_i, v_j) \in E, \ v_i \in S \text{ or } v_j \in S \text{ with } S \in V.$$

A candidate solution S is a valid solution, called *Vertex Cover*, if each edge in G has at least one endpoint in S. The objective function value $\omega(S)$ of a candidate solution S is defined as the sum of the weights of the vertices in S. The objective of the MWVC problem is then to find a valid candidate solution that minimizes the objective function.

2 Benchmark and Experimental Protocol

In this section are reported a set of benchmark instances to consider for testing the algorithm developed. Each considered instance for this project consists of an undirected, vertex-weighted graph with n vertices and m edges. These instances are grouped into three different classes, with respect to their number of vertices: *i*) SPI, i.e. *small problem instances*; *ii*) MPI, i.e. *moderate-size problem instances*; and *iii*) LPI, i.e. *large problem instances.* In each class a whole range of instances exists, including rather sparse graphs as well as rather dense graphs. The instances of the first two classes share the characteristics to have 10 problem instances randomly generated per combination of n and m, which means that the results must be the average over the objective function values obtained for the 10 instances, and the weight $\omega(v)$ of each vertex $v \in V$ is randomly drawn from a uniform distribution in the range $[20, 120]$. In contrast, class LPI instead only consists of one problem instance per combination of n and m and the vertex weights are generated similarly as in the case of previous instances. For this class type the average quality over 10 independent runs are generally reported as quality measure.

For the experiments to be included in the final report, the *stopping criterion* to consider is $FE = 2 \times 10^4$ as the **maximum number of objective function evaluations**. Please, include in the results table also the average number of objective function evaluations needed for reaching the optimal solution, or the best solution found. Would be good also if the candidate can include in the report some plots on the convergence behavior of the developed algorithm.

The instances to be considered can be found at the following link:
https://www.dmi.unict.it/mpavone/lab-ai-progetto-mwvc.html

3 Algorithm to be Developed

In order to tackle and solve the MWVC, the candidate is invited to select and choose only one algorithm on the list below:

1. Tabu Search (TS);
2. Branch-and-Bound;
3. Genetic Algorithm (GA) with *uniform crossover* and *roulette-wheel selection*;
4. Genetic Algorithm (GA) with *one-point crossover* and *k-tournament selection*;

It is important to point out that the choice of the algorithm must be motivated in the final report. Furthermore, it is also important to recall that the difficulty of the developed algorithm together with the designed originality will be the important basis for the final evaluation.

4 Final Remarks

The candidate is given the opportunity to find the best parameter setting on which the algorithm is based, highlighting them in the report, justifying and motivating the choice made (mandatory point).

For those implementing the Tabu Search algorithm, the candidate must select the most suitable *term memory* to use, explaining such motivation in the report (mandatory point).

Recall that the report must be written in Latex (min. 4 page - max 12 page). In the report is strongly **discouraged** the inclusion of codes in programming language, but it is strongly suggested to include pseudo-code. Specifically, in the report the candidate must explain **in detail** (mandatory): (*i*) algorithm developed and its key features; (*ii*) reasons about choices made; (*iii*) novelty and originalities introduced in the project; (*iv*) analysis of the obtained results and personal comments on them; finally (*v*) personal conclusions on the project.

Project (source code) and report (.pdf and .tex - including eventually figures) must be sent by email to mario.pavone@unict.it no later than February 11, 2025, at 5:00pm.