The Minimum Weight Vertex Cover Problem

PROGETTO

Laboratorio di Intelligenza Artificiale

Prof. Mario Pavone

4 Febbraio 2022

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 $\omega : V \to \Re^+$ 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 for objective function evaluations. Please, include in the results table also the average number of objective function evaluations needed for reaching the optimal solution. 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-progetto20220204.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. Genetic Algorithm (GA);
3. Iterated Local Search (ILS).

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 Note Finali al Progetto

Al candidato é data facoltá di sviluppare gli operatori di ricerca e, nel caso lo richiedesse, trovare il miglior settaggio dei parametri su cui si basa l'algoritmo implementato. I valori dei parametri da cui dipende l'algoritmo e il valore ad essi settato devono essere indicati nella relazione, giustificando e motivando la scelta fatta (punto obbligatorio).

É data facoltá al candidato di:

- scegliere quale *term memory* utilizzare se si é scelto di sviluppare l'algoritmo Tabu Search;
- scegliere la *modalitá di selezione* degli elementi della popolazione per l'iterazione successiva e quale operatore di perturbazione utilizzare (Crossover e/o mutazione), se si é scelto di sviluppare l'algoritmo genetico. Si invita, in questo caso, il candidato a descrivere tali scelte nella relazione giustificandone la scelta;
- scegliere la *"Local Search"* piú opportuna da adottare, nel caso si scelga di implementare ILS or GA.

Nella relazione da consegnare insieme al progetto (min. 4 pag - max 12 pag), preferibilmente scritta in Latex, é fortemente **sconsigliata** l'inclusione di codici. É invece fortemente **consigliata** la descrizione (minuziosa) degli operatori utilizzati; algoritmo implementato, con relativo pseudo-codice; caratteristiche (chiave) dell'algoritmo implementato; motivazione delle scelte fatte; eventuali originalitá introdotte nell'elaborato; e (molto importante) risultati ottenuti e analisi sui risultati presentati. **É fortemente richiesta** l'inclusione di una sezione in cui porre le vostre conclusioni sull'elaborato svolto e i risultati ottenuti.

Si ricorda che la valutazione del progetto e della relazione si baserá su: 1) *originalitá introdotta* nell'elaborato; 2) complessitá dell'algoritmo sviluppato; 3) risultati ottenuti; 4) qualità de!a relazione; e 5) contenuti presenti ne!a relazione.