HUST_RTS_SOLVER: A heuristic algorithm of directed feedback vertex set problem

YuMing Du
School of Computer Science and Technology, Huazhong University of Science & Technology, China

QingYun Zhang
School of Computer Science and Technology, Huazhong University of Science & Technology, China

JunZhou Xu
Huawei TCS Lab Shanghai, China

ShunGen Zhang
School of Computer Science and Technology, Huazhong University of Science & Technology, China

Chao Liao
Huawei TCS Lab Shanghai, China

ZhiHuai Chen
Huawei TCS Lab Shanghai, China

ZhiBo Sun
School of Computer Science and Technology, Huazhong University of Science & Technology, China

ZhouXing Su
School of Computer Science and Technology, Huazhong University of Science & Technology, China

JunWen Ding
School of Computer Science and Technology, Huazhong University of Science & Technology, China

Chen Wu
Huawei TCS Lab Shanghai, China

PinYan Lu
Huawei TCS Lab Shanghai, China

ZhiPeng Lv
School of Computer Science and Technology, Huazhong University of Science & Technology, China

Abstract

A directed graph is formed by vertices and arcs from one vertex to another. The directed feedback vertex set problem (DFVSP) consists in making a given directed graph acyclic by removing as few vertices as possible. In this write-up, we introduce our heuristic solver HUST_RTS_SOLVER submitted for the heuristic track of the Computational Experiments (PACE) 2022 challenge, and briefly outline the core techniques used in the solver.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Feedback vertex set, Local search, Heuristic search

Digital Object Identifier 10.4230/LIPIcs.PACE.2022.1

Supplementary Material The source code is available on GitHub (https://github.com/Zhang-qingyun/pace_2022_HUST_solver.git)
Software (Source Code): `https://doi.org/10.5281/zenodo.6643002`

1 Problem Description

Let $G = (V, E)$ be a directed graph, where V is the vertex set and $E \subseteq V \times V$ is the edge set. The feedback vertex set problem is to find a minimum subset $X \subseteq V$ such that, when

all vertices of X and their adjacent edges are deleted from G, the remainder subset $G' \subseteq G$ is acyclic. The subset X is a feedback vertex set of the graph G. In the challenge of PACE 2022, for heuristic algorithms, the process was limited in 10 miniutes.

2 Reduction Rules

Reduction Rules play an important role in solving DFVSP because it improves the efficiency of algorithm. A contraction (reduction) operation reduces the original graph while it preserves the information necessary for finding the minimum feedback set. We adopt eight reduction rules to eliminate some vertices and edges, and deduce that some vertices must be included in the optimal solution. Five contraction operations have been proposed in Levy and Low (1988) [3]. More recently, three new contraction operations have been presented in Lin and Jou (1999) [4]. Our implementation adopted the directed graph reduction method in [4]. Specially, we applies repeatedly these reduction rules until nothing changes on the graph. The reductions simplifies the graph efficiently and does not take too long.

3 Local search

In order to tackle DFVSP, we propose a threshold search algorithm which is based on the local search framework. It generates an initial solution, than iteratively improves the incumbent solution by a local search procedure. At each iteration of the algorithm, it first evaluates the neighborhood of the current solution and determines whether to perform the current neighborhood move by the threshold accepting rule. If the current solution improves the best solution found so far, then the best solution is updated with the current. Finally, when the specified termination condition is met, the algorithm terminates and returns the best solution.

3.1 Initialization

This procedure is divided into two stages to generate an initial solution. In the first stage, the original problem is transformed into a vertex cover problem [1, 5] and then solves the problem by executing a heuristic algorithm under limited time. The execution time of the heuristic algorithm is determined according to the ratio r_b of bidirectional edges. The greater the value of r_b, the greater the execution time.

$$r_b = \frac{2 * \sum_{i,j \in V} (e_{ij} * e_{ji})}{|E|} \tag{1}$$

In the second stage, the descent heuristic algorithm is used to further optimize the solution. Finally, the initial solution X_0 of the problem is obtained.

3.2 Neighborhood Structure and Evaluation

A directed graph with no directed cycles is named a directed acyclic graph (DAG). Every DAG has a topological ordering, i.e. an ordering of its vertices such that the starting-point of every arc occurs earlier in the ordering than the endpoint of the arc. Conversely, the existence of a topological ordering in a graph proves that this graph is acyclic.

In order to solve DFVSP, we adopts a neighborhood move based on add and remove operations [2]. Specifically, an add operation consists of inserting a new vertex $v \notin X$ at some particular position into the sequence and, at the same time, a remove operation used

to remove the vertices that would now violate the precedence constraint. Based on the incumbent solution X, performing the operations produces a new neighborhood solution X'.

To obtain the best neighboring solution and improve the incumbent solution X, our solver uses the insert policies that in-coming and out-coming neighbors. Specifically, a vertex v can be inserted in the sequence at only two different positions, which are after its numbered in-coming neighbors, or just before its numbered out-going neighbors.

The threshold search algorithm consists of two phase, which are threshold-based exploration procedure and the descent-based improvement procedure. The improving and deteriorating solutions are allowed in the threshold-based exploration phase as long as its solution quality meets the threshold accepting rule. In the seconds phase, the descent-based improvement procedure starts with a solution returned by the threshold-based exploration procedure, it iteratively explores and accept the improved solution. The descent-based improvement procedure achieve a local optimum.

3.3 Perturbation Mechanism

When the optimal solution cannot be improved within a certain number of iterations, the perturbation mechanism is implemented. Specifically, we random select some vertices that do not belong to the topological ordering. Then, these vertices are inserted into topological sequence based on neighborhood move – add and remove operations.

References

1 Yuning Chen and Jin-Kao Hao. Dynamic thresholding search for minimum vertex cover in massive sparse graphs. *Eng. Appl. Artif. Intell.*, 82:76–84, 2019.

2 Philippe Galinier, Eunice Adjarath Lemamou, and Mohamed Wassim Bouzidi. Applying local search to the feedback vertex set problem. *J. Heuristics*, 19(5):797–818, 2013.

3 Hanoch Levy and David W. Low. A contraction algorithm for finding small cycle cutsets. *J. Algorithms*, 9(4):470–493, 1988.

4 Hen-Ming Lin and Jing-Yang Jou. Computing minimum feedback vertex sets by contraction operations and its applications on CAD. In *Proceedings of the IEEE International Conference On Computer Design, VLSI in Computers and Processors, ICCD '99, Austin, Texas, USA, October 10-13, 1999*, page 364. IEEE Computer Society, 1999.

5 Changsheng Quan and Ping Guo. A local search method based on edge age strategy for minimum vertex cover problem in massive graphs. *Expert Syst. Appl.*, 182:115185, 2021.