Graph Edit Distance Compacted Search Tree

Ibrahim Chegrane[1,3(✉)], Imane Hocine[1,2], Saïd Yahiaoui[1], Ahcene Bendjoudi[1], and Nadia Nouali-Taboudjemat[1]

[1] CERIST, Research Center on Scientific and Technical Information, 16306 Ben Aknoun, Algiers, Algeria
{syahiaoui,abendjoudi,nnouali}@cerist.dz
[2] Ecole Nationale Supérieure d'Informatique, BP 68M, 16309 Oued-Smar, Alger, Algeria
bi_hocine@esi.dz
[3] CoBIUS Lab, Department of Computer Science, University of Sherbrooke, Sherbrooke, QC, Canada
ibrahim.chegrane@usherbrooke.ca

Abstract. We propose two methods to compact the used search tree during the graph edit distance (GED) computation. The first maps the node information and encodes the different edit operations by numbers and the needed remaining vertices and edges by BitSets. The second represents the tree succinctly by bit-vectors. The proposed methods require 24 to 250 times less memory than traditional versions without negatively influencing the running time.

Keywords: Graph Edit Distance (GED) · Compacted GED search space

1 Introduction

The Graph Edit Distance (GED) is a well-known metric used to compute the degree of dissimilarity between two graphs g_1 and g_2. It is generally used in pattern recognition [12], such as handwriting recognition [9] and document analysis [4]. The GED is defined as the minimum-cost sequence of edit operations needed to transform graph g_1 into graph g_2 [3]. The allowed operations are insertion, deletion, and substitution, which are applied on vertices and their corresponding edges. The GED computation is an NP-hard problem [13]. It has an exponential time complexity due to the exponential size of the generated search tree.

Bunke and Allermann were the precursors for solving the GED problem [3]. The authors used an A* based algorithm where the search tree is generated dynamically. In [8], the authors proposed an approximation for the GED problem called *A*-Beamsearch*. By limiting the size of the A* priority queue to a certain size s. To speed up the A* search process, [10] presents an effective heuristic that gives the estimated cost h and concludes a lower bound. This heuristic, called bipartite heuristic [9], has been discussed and improved in [11] and [2] to compute a more accurate lower bound. Authors in [1] proposed an approach called (DF_GED)

T. Skopal et al. (Eds.): SISAP 2022, LNCS 13590, pp. 181–189, 2022.
https://doi.org/10.1007/978-3-031-17849-8_14