Parallel Computing 114 (2022) 102984

Contents lists available at ScienceDirect

Parallel Computing

journal homepage: www.elsevier.com/locate/parco

Efficient parallel branch-and-bound approaches for exact graph edit distance problem

Adel Dabah [a], Ibrahim Chegrane [a,b,*], Saïd Yahiaoui [a], Ahcene Bendjoudi [a], Nadia Nouali-Taboudjemat [a]

[a] CERIST Research Center on Scientific and Technical Information, Algiers, Algeria
[b] CoBIUS Lab, Department of Computer Science, University of Sherbrooke Qc, Canada

ARTICLE INFO

Dataset link: https://github.com/chegrane/ged_parallel/

Keywords:
Parallel branch-and-bound
Graph matching
Graph edit distance

ABSTRACT

Graph Edit Distance (GED) is a well-known measure used in the graph matching to measure the similarity/dissimilarity between two graphs by computing the minimum cost of edit operations needed to transform one graph into another. This process, Which appears to be simple, is known NP-hard and time consuming since the search space is increasing exponentially. One way to optimally solve this problem is by using Branch and Bound (B&B) algorithms, Which reduce the computation time required to explore the whole search space by performing an implicit enumeration of the search space instead of an exhaustive one based on a pruning technique. nevertheless, They remain inefficient when dealing with large problem instances due to the impractical running time needed to explore the whole search space. To overcome this issue, We propose in this paper three parallel B&B approaches based on shared memory to exploit the multi-core CPU processors: First, a work-stealing approach where several instances of the B&B algorithm explore a single search tree concurrently achieving speedups up to 24× faster than the sequential version. Second, a tree-based approach where multiple parts of the search tree are explored simultaneously by independent B&B instances achieving speedups up to 28×. Finally, Due to the irregular nature of the GED problem, two load-balancing strategies are proposed to ensure a fair workload between parallel processes achieving impressive speedups up to 300×. all experiments have been carried out on well-known datasets

1. Introduction

Graph Edit Distance (GED) approach is a well-known technique used in graph matching to measure the minimum distance between two graphs. The goal of the GED is to compute the amount of dissimilarity between two graphs. In other words, it represents the cost of the best set of edit operations needed to transform one graph into another [1]. The allowed operations are insertion, deletion, and substitution, which are applied on vertices and edges. This problem is known to be very challenging due to its NP-hardness nature [2], which means that the time complexity of computing the minimum distance between two graphs increases exponentially with the number of vertices. The importance of the GED comes from its multitude of use cases. It can be used to find exact and also inexact matching, where some errors are tolerated. Moreover, the GED can be used in various areas [3], especially in areas related to pattern recognition, such as, handwriting recognition [4–6], person identification and authentication (example: fingerprint recognition) [7], documents analysis [5,8,9], and in graph database search [10]. It can also be found in machine learning, nearest-neighbor classification, and in data mining area [5].

To compute optimally the GED between two graphs, often A-Star [11] based search technique is used in the literature [3]. However, this latter needs huge memory resources, making it impossible to use for large graphs. The Branch and Bound (B&B) algorithms are well-known techniques for optimally solving optimization problems via an intelligent enumeration of the search space. This method models the search space as a tree using two components: branching and bounding. The branching is a recursive process that divides the search space of a given problem into several smaller sub-problems, which are treated the same way until solutions are found. After the branching process, the bounding operator evaluates the ability of each generated sub-problem to contain good solutions. The B&B algorithm uses several techniques (elimination and selection) to avoid exploring non-promising sub-problems (branches) and accelerate the search process. Due to the complexity of the GED problem, which is NP-hard [2] for general

* Corresponding author at: CoBIUS Lab, Department of Computer Science, University of Sherbrooke Qc, Canada.
E-mail addresses: adabah@cerist.dz (A. Dabah), ibrahim.chegrane@usherbrooke.ca (I. Chegrane), syahiaoui@cerist.dz (S. Yahiaoui), abendjoudi@cerist.dz (A. Bendjoudi), nnouali@cerist.dz (N. Nouali-Taboudjemat).

https://doi.org/10.1016/j.parco.2022.102984
Received 18 September 2021; Received in revised form 16 July 2022; Accepted 27 October 2022
Available online 3 November 2022
0167-8191/