Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Approximation algorithms for the scaffolding problem and its generalizations

Zhi-Zhong Chen[a,*], Youta Harada[a], Fei Guo[b], Lusheng Wang[c,1]

[a] *Division of Information System Design, Tokyo Denki University, Hatoyama, Saitama 350-0394, Japan*
[b] *School of Computer Science and Technology, Tianjin University, Tianjin, China*
[c] *Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Hong Kong*

ARTICLE INFO

Article history:
Received 17 July 2016
Received in revised form 5 February 2017
Accepted 16 March 2017
Available online xxxx

Keywords:
Approximation algorithms
Randomized algorithms
Scaffolding
Matchings

ABSTRACT

Scaffolding is one of the main stages in genome assembly. During this stage, we want to merge contigs assembled from the paired-end reads into bigger chains called *scaffolds*. For this purpose, the following graph-theoretical problem has been proposed: Given an edge-weighted complete graph G and a perfect matching D of G, we wish to find a Hamiltonian path P in G such that all edges of D appear in P and the total weight of edges in P but not in D is maximized. This problem is NP-hard and the previously best polynomial-time approximation algorithm for it achieves a ratio of $\frac{1}{2}$. In this paper, we design a new polynomial-time approximation algorithm achieving a ratio of $\frac{5-5\epsilon}{9-8\epsilon}$ for any constant $0 < \epsilon < 1$. Several generalizations of the problem have also been introduced in the literature and we present polynomial-time approximation algorithms for them that achieve better approximation ratios than the previous bests. In particular, one of the algorithms answers an open question.

1. Introduction

Sequencing the whole genome of an organism is a vital component for detailed molecular analysis of the organism, and genome projects are now underway or complete [8]. Unfortunately, with current genome-sequencing technologies, it is impossible to continuously read from one end of a long chromosome to the other. So, a commonly used method for sequencing a chromosome is to first randomly shear multiple copies of the chromosome into many small fragments of varying sizes, then accurately sequence the fragments to obtain *reads*, and further assemble the reads into a sequence of the whole chromosome. The assembling process typically consists of two steps. The first step is called *contigging*, where we use confident overlaps between the reads to piece together larger segments of continuous sequences called *contigs* (each of which consists of two strands, namely, the *forward strand* and the *reverse strand*). The second step is called *scaffolding*, where we link contigs together into *scaffolds* by using longer fragments of a known length whose ends are sequenced (called *paired-end reads*). A recent comprehensive evaluation of available software tools shows that scaffolding is still computationally intractable [6].

The scaffolding problem can be formulated as the problem of finding a special Hamiltonian path in an edge-weighted complete graph G as follows [7]. For each contig c, G has two vertices f_c and r_c, where f_c corresponds to the forward strand of c while r_c corresponds to the reverse strand of c. The edge $\{f_c, r_c\}$ is assigned a weight of 0 in G and is called a

[*] Corresponding author.
E-mail addresses: zzchen@mail.dendai.ac.jp (Z.-Z. Chen), guofeieileen@163.com (F. Guo), cswangl@cityu.edu.hk (L. Wang).
[1] Also with University of Hong Kong Shenzhen Research Institute, Shenzhen Hi-Tech Industrial Park, Nanshan District, Shenzhen, PR China.

http://dx.doi.org/10.1016/j.tcs.2017.03.042
0304-3975/© 2017 Elsevier B.V. All rights reserved.

dummy edge for convenience. Each non-dummy edge $\{u, v\}$ of G corresponds to a bundle of paired-end reads each of which connects strand u (of a contig c) with strand v (of another contig c'), and the weight of $\{u, v\}$ in G is equal to the size of the corresponding bundle. In case no such bundle exists for $\{u, v\}$, then the weight of $\{u, v\}$ in G is 0. Note that the set D of dummy edges is a perfect matching of G. A Hamiltonian path P in G is D-*valid* if P contains all dummy edges of G. Since G is a complete graph and D is a perfect matching of G, we can always transform D into a D-valid Hamiltonian path of G by adding edges of G to D. Given G and D, the objective is to compute a D-valid Hamiltonian path in G such that the total weight of edges in P is maximized over all D-valid Hamiltonian paths in G.

The scaffolding problem is NP-hard [1,2]. Indeed, the problem is APX-hard because the maximum asymmetric traveling salesman problem is APX-hard [9] and can be reduced to the scaffolding problem as follows. Given an edge-weighted digraph G, we construct an (undirected) graph H from G by splitting each vertex u of G into two vertices u_{in} and u_{out} so that (1) $\{u_{\text{in}}, u_{\text{out}}\}$ is an edge (of weight 0) in both H and D and (2) each arc (u, v) in G is transformed into an edge $\{u_{\text{out}}, v_{\text{in}}\}$ (of the same weight as (u, v)) in H. Mandric and Zelikovsky [7] propose two heuristics for the scaffolding problem. One of them is based on maximum-weight matching and the other is based on the greedy method. Although they claim that their heuristics perform well in practice, the heuristics are not shown to have a worst-case performance guarantee.

In order to take the desired structure of the genome (namely, the number of circular or linear chromosomes) into consideration, Chateau and Giroudeau [1,2] generalizes the scaffolding problem as follows. In addition to G and D, we are also given two nonnegative integers σ_p and σ_c. Instead of a single D-valid Hamiltonian path in G, we want to find a collection of exactly σ_p paths and exactly σ_c cycles such that the paths and cycles are disjoint and contain all edges of D. For convenience, we refer to such a collection as a D-*valid* (σ_p, σ_c)-*cover* of G. Note that a D-valid Hamiltonian path in G is just a D-valid $(1, 0)$-cover of G. Moreover, G has a D-valid (σ_p, σ_c)-cover if and only if $\sigma_p + \sigma_c \geq 1$ and $|D| \geq \sigma_p + 2\sigma_c$ [2]. So, we can hereafter assume that the input $(G, D, \sigma_p, \sigma_c)$ always satisfies $\sigma_p + \sigma_c \geq 1$ and $|D| \geq \sigma_p + 2\sigma_c$. The new objective is to compute a D-valid (σ_p, σ_c)-cover C of G such that the total weight of edges in C is maximized over all D-valid (σ_p, σ_c)-covers of G. We call this generalization the *generalized scaffolding problem* (GSP for short).

In the special case of GSP where the input satisfies $|D| = \sigma_p + 2\sigma_c$, a D-valid (σ_p, σ_c)-cover of G is simply a collection of disjoint edges and cycles with 4 edges and hence can be found by computing a maximum-weight matching in a suitably constructed graph [2]. Moreover, in the special case where $(\sigma_p, \sigma_c) = (0, 1)$, a very simple $O(n^3)$-time approximation algorithm achieving a ratio of $\frac{1}{2}$ can be designed [1,2], where n is the number of vertices in the input graph. This algorithm is also applicable to the scaffolding problem, i.e., the special case of GSP where $(\sigma_p, \sigma_c) = (1, 0)$. Furthermore, in the special case where the input satisfies $|D| \geq 2(\sigma_p + 2\sigma_c)$, an $O(n^3)$-time approximation algorithm achieving a ratio of $\frac{1}{3}$ can be designed [2]. However, the approximability of the remaining case where $\sigma_p + 2\sigma_c < |D| < 2(\sigma_p + 2\sigma_c)$ was left as an open question in [2].

In this paper, we improve the algorithmic results in [1,2] and answer the above open question in [2]. More specifically, we first design a new $O(n^3)$-time approximation algorithm for the scaffolding problem that achieves a ratio of $\frac{5-5\epsilon}{9-8\epsilon}$ for any constant $0 < \epsilon < 1$. This is done by first designing a randomized algorithm and then derandomizing it. The randomized algorithm finds two D-valid Hamiltonian paths and outputs the better one between the two paths. The randomized algorithm is inspired by the algorithm in [5] for the maximum traveling salesman problem. We also show that our analysis is almost tight. We then design an $O(n^3)$-time approximation algorithm for GSP that *always* achieves a ratio of $\frac{1}{3}$. A simple but crucial idea behind the algorithm is to first compute a maximum-weight matching M in the input graph G such that $M \cap D = \emptyset$ and $|M| = |D| - \sigma_p$. With a minor modification, the algorithm achieves a ratio of $\frac{1}{2}$ for the special case of GSP where $|D| \geq \sigma_p + 3\sigma_c$. With another minor modification, the algorithm achieves a ratio of $\min\left\{\frac{2}{5}, \frac{1+2\epsilon}{3}\right\}$ for the special case where $|D| \geq \sigma_p + (2 + \epsilon)\sigma_c$ for any constant $0 < \epsilon < 1$.

We also modify the approximation algorithm for the scaffold problem so that it works for two special cases of GSP. For the special case of GSP where the input satisfies $|D| \geq 9(\sigma_p + \sigma_c)$ (respectively, $|D| \geq 6(\sigma_p + \sigma_c)$), the modified algorithm runs in $O\left(\left(\sigma_c^2 + 1\right) n^3\right)$ time and achieves a ratio of $\frac{5-4\epsilon}{9}$ (respectively, $\frac{7-6\epsilon}{13}$) for any constant $0 < \epsilon < 1$.

Weller et al. [10] defined a different generalization of the scaffolding problem as follows. The input is the same as to GSP but without the condition $|D| \geq \sigma_p + 2\sigma_c$, and the objective is to find a D-valid (σ'_p, σ'_c)-cover C of G with $\sigma'_p \leq \sigma_p$ and $\sigma'_c \leq \sigma_c$ such that the total weight of edges in C is maximized over all D-valid (σ''_p, σ''_c)-covers of G with $\sigma''_p \leq \sigma_p$ and $\sigma''_c \leq \sigma_c$. We call this generalization the *loosely generalized scaffolding problem* (LGSP for short). The previously best approximation algorithm for LGSP achieves a ratio of $\frac{1}{2}$ [10] and runs in $O(n^3)$ time. In this paper, we show that the approximation algorithm for the scaffold problem can be modified to approximate LGSP as well without altering the approximation ratio and the time complexity.

The remainder of this paper is organized as follows. Section 2 gives basic definitions that will be used in the remainder of the paper. Section 3 presents approximation algorithms for the scaffolding problem, Section 4 presents approximation algorithms for LGSP, and Section 5 presents approximation algorithms for GSP and its special cases.

2. Basic definitions

Throughout this paper, a graph means an undirected graph without parallel edges or self-loops. Let G be a graph. We denote the vertex set of G by $V(G)$, and denote the edge set of G by $E(G)$. For a subset U of $V(G)$, $G[U]$ denotes the graph

Please cite this article in press as: Z.-Z. Chen et al., Approximation algorithms for the scaffolding problem and its generalizations, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.03.042