Theoretical Computer Science 640 (2016) 94–103

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

New models of graph-bin packing

Cs. Bujtás [a,f], Gy. Dósa [b], Cs. Imreh [c,d,*], J. Nagy-György [e], Zs. Tuza [a,f]

[a] Department of Computer Science and Systems Technology, University of Pannonia, H-8200 Veszprém, Egyetem u. 10, Hungary
[b] Department of Mathematics, University of Pannonia, H-8200 Veszprém, Egyetem u. 10, Hungary
[c] Department of Informatics, University of Szeged, H-6720 Szeged, Árpád tér 2, Hungary
[d] Szeged Research Group of TU Budapest, VIK, EIT, H-6720 Szeged, Árpád tér 2, Hungary
[e] Department of Mathematics, University of Szeged, H-6720 Szeged, Aradi Vértanúk tere 1, Hungary
[f] Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, H-1053 Budapest, Reáltanoda u. 13–15, Hungary

ARTICLE INFO

Article history:
Received 11 April 2016
Received in revised form 25 May 2016
Accepted 8 June 2016
Available online 14 June 2016
Communicated by D.-Z. Du

Keywords:
Graph theory
Packing
Online algorithm
Combinatorial games
Combinatorial optimization

ABSTRACT

In Bujtás et al. (2011) [4] the authors introduced a very general problem called Graph-Bin Packing (GBP). It requires a mapping $\mu : V(G) \to V(H)$ from the vertex set of an input graph G into a fixed host graph H, which, among other conditions, satisfies that for each pair u, v of adjacent vertices the distance of $\mu(u)$ and $\mu(v)$ in H is between two prescribed bounds. In this paper we propose two online versions of the Graph-Bin Packing problem. In both cases the vertices can arrive in an arbitrary order where each new vertex is adjacent to some of the previous ones. One version is a Maker–Breaker game whose rules are defined by the packing conditions. A subclass of Maker-win input graphs is what we call 'well-packable'; it means that a packing of G is obtained whenever the mapping $\mu(u)$ is generated by selecting an arbitrary feasible vertex of the host graph for the next vertex of G in each step. The other model is connected-online packing where we are looking for an online algorithm which can always find a feasible packing. In both models we present some sufficient and some necessary conditions for packability. In the connected-online version we also give bounds on the size of used part of the host graph.

1. Introduction

In paper [4] a very general problem was introduced, which is a common generalization of many fundamental problems studied to a great extent separately in thousands of papers in the literature of combinatorial optimization, theoretical computer science, graph theory, and operations research. Since *bin packing* and *graph homomorphism* are highly important instances of our problem, we named it *graph-bin packing*, or *graph H-bin packing* when the underlying structure (host graph) H has to be mentioned explicitly.

The input of the problem is a simple graph $G = (V, E)$ with lower and upper bounds on its edges and weights on its vertices. The vertices correspond to items which have to be packed into the vertices (bins) of a host graph, such that each host vertex can accommodate at most L weight in total, and if two items are adjacent in G, then the distance of their host vertices in H must be between the lower and upper bounds of the edge joining the two items.

* Corresponding author.
E-mail address: cimreh@inf.u-szeged.hu (Cs. Imreh).

http://dx.doi.org/10.1016/j.tcs.2016.06.008
0304-3975/

It has been noticed in [4] that Graph-Bin Packing is a very general framework; it includes many widely studied algorithmic problems like Bin Packing (with or without conflicts), Scheduling (with or without incompatible jobs), Graph Homomorphism, Subgraph- and Induced Subgraph Isomorphism, (k, d)-Coloring, Distance-Constrained Labeling, Channel Assignment, Bandwidth, Partition, and 3-Partition as subproblems expressible by suitable choices of the parameters.

In [4] some results are presented about the complexity of deciding whether G is packable into a given host graph, and some sufficient and also some necessary conditions are given on packability. The optimization version where the goal is to minimize the size of a connected extension[1] of the used part of the host graph is also studied.

We note that there are further papers considering extensions of the bin packing problem to graphs. The most closely related problem is the 'polyp packing problem' studied in [11,12] where the goal was to embed some paths in a vertex-disjoint or edge-disjoint way into some copies of a host graph. The host graphs, which were called polyps, consist of simple paths of the same length with a common endpoint. Those papers presented some complexity results and studied the extensions of the well-known First-Fit bin packing algorithm. Another problem, where the graphs present constraints on the possible packing, is the 'bin packing with conflicts' model (see [5,6]) where the items adjacent in the conflict graph cannot be packed into the same bin. A version where the graph presents only restrictions on the order in which the items can be packed into a bin is studied in [2].

An offline algorithm completely knows the input, and is allowed to use all pieces of information before a decision is made. In this paper we will consider definitions of packing the input graph into the host graph sequentially, where the packing algorithm has less power. On the other hand, in order to exclude some rather trivial classes of negative problem instances, we will consider only connected input sequences which are vertex orders where each prefix induces a connected subgraph of the input graph.

In Section 2 the notion of well-packable graphs is introduced and studied. First we define a Maker–Breaker game where two players are packing the input graph into the host graph, one player wants to find a suitable packing at the end and the other wants to prevent the packing. An input graph is called well-packable if the packing can be finished independently of the strategies of the players. We present several sufficient and some necessary conditions in special classes of graphs for being well-packable.

Section 3 is devoted to the study of connected-online packing. A graph is called connected-online packable if there exists an online algorithm that finds a feasible packing for every connected-online input sequence. We present some simple structural statements about connected-online packable graphs, and also give some results about the optimization version of the problem.

We close the paper with collecting some further questions in Section 4.

1.1. Standard notation

In a graph G the *distance* between two vertices x and y is defined as the length[2] of a shortest x–y path, and is denoted by $d_G(x, y)$. The diameter of graph G is $diam(G) = \sup_{x,y \in V(G)} d_G(x, y)$. A *geodesic path* means an x–y path of length $d_G(x, y)$.

The two-way infinite path is denoted by P_∞. As usual, we denote by C_n the cycle of length n, and by P_n the path on n vertices.

If $v \in V(G)$ then $G - v$ denotes the graph obtained from G by deleting v and its incident edges.

1.2. Problem definition for Graph H-Bin Packing

The problem called Graph H-Bin Packing can be specified with the following components:

- Host graph, H
  It is a fixed *connected* graph $H = (X, F)$ with vertex set X and edge set F, and vertex *capacity* $L > 0$.
  In the general setting the capacity $L(x)$ may depend on $x \in X$, and the edges may have different lengths; but in the present work we assume $L(x) = L$ for all x (possibly $L = \infty$), and unit length for all edges.
- Input graph, G
  It is a *finite* simple graph $G = (V, E)$ in which each vertex $v \in V$ has a given size $s(v)$ $(0 \le s(v) \le L)$ and each edge $e = uv \in E$ has a lower and an upper *edge-length bound* $a(e) = a(u, v)$ and $b(e) = b(u, v)$ which are integers or ∞, such that $0 \le a(u, v) \le b(u, v)$ holds.
- Packing, μ
  A mapping $\mu : V \to X$ is a *packing of* $G = (V, E)$ *into* $H = (X, F)$ if it satisfies the following two conditions.
  – For each vertex $x \in X$ of H,

$$\sum_{v \in V \,:\, \mu(v) = x} s(v) \le L.$$

[1] It means a *connected* subgraph G^+ of H, that contains all those vertices of H into which at least one vertex of G has been packed.

[2] That is the number of edges, or the sum of their lengths if a length function is given on the edge set of G. Throughout this paper all edges will be assumed to have unit length.