Improved Approximation Algorithms for Box Contact Representations*

Michael A. Bekos[1], Thomas C. van Dijk[2], Martin Fink[2,5], Philipp Kindermann[2], Stephen Kobourov[3], Sergey Pupyrev[3,4], Joachim Spoerhase[2], and Alexander Wolff[2]*

[1] Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany.
[2] Lehrstuhl für Informatik I, Universität Würzburg, Germany.
[3] Department of Computer Science, University of Arizona, USA.
[4] Institute of Mathematics and Computer Science, Ural Federal University, Russia.
[5] Department of Computer Science, University of California, Santa Barbara, USA.

Abstract. We study the following geometric representation problem: Given a graph whose vertices correspond to axis-aligned rectangles with fixed dimensions, arrange the rectangles without overlaps in the plane such that two rectangles touch if the graph contains an edge between them. This problem is called CONTACT REPRESENTATION OF WORD NETWORKS (CROWN) since it formalizes the geometric problem behind drawing word clouds in which semantically related words are close to each other. CROWN is known to be NP-hard, and there are approximation algorithms for certain graph classes for the optimization version, MAX-CROWN, in which realizing each desired adjacency yields a certain profit. We show that the problem is APX-complete on bipartite graphs of bounded maximum degree. We present the first $O(1)$-approximation algorithm for the general case, when the input is a complete weighted graph, and for the bipartite case. Since the subgraph of realized adjacencies is necessarily planar, we consider several planar graph classes (stars, trees, outerplanar, and planar graphs), improving upon the known results. For some graph classes, we also describe improvements in the unweighted case, where each adjacency yields the same profit.

1 Introduction

In the last few years, word clouds have become a standard tool for abstracting, visualizing, and comparing text documents. For example, word clouds were used in 2008 to contrast the speeches of the U.S. presidential candidates. More recently, the German media used them to visualize the newly signed coalition agreement and to compare it to a similar agreement from 2009. A word cloud of a given document consists of the most important (or most frequent) words in that document. Each word is printed in a given font and scaled by a factor roughly proportional to its importance (the same is done with the

* The work of M. A. Bekos is implemented within the framework of the Action "Supporting Postdoctoral Researchers" of the Operational Program "Education and Lifelong Learning" (Action's Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State. Ph. Kindermann and A. Wolff acknowledge support by the ESF EuroGIGA project GraDR. S. Kobourov and S. Pupyrev are supported by NSF grants CCF-1115971 and DEB 1053573

A. Schulz and D. Wagner (Eds.): ESA 2014, LNCS 8737, pp. 87–99, 2014.

names of towns and cities on geographic maps, for example). The printed words are arranged without overlap and tightly packed into some shape (usually a rectangle). Tag clouds look similar; they consist of keyword metadata (tags) that have been attributed to resources in some collection such as web pages or photos.

Wordle [23] is a popular tool for drawing word or tag clouds. The Wordle website allows users to upload a list of words and, for each word, its relative importance. The user can further select font, color scheme, and decide whether all words must be placed horizontally or whether words can also be placed vertically. The tool then computes a placement of the words, each scaled according to its importance, such that no two words overlap. Generally, the drawings are very compact and aesthetically appealing.

In the automated analysis of text one is usually not just interested in the most important words and their frequencies, but also in the connections between these words. For example, if a pair of words often appears together in a sentence, then this is often seen as evidence that this pair of words is linked semantically [17]. In this case, it makes sense to place the two words close to each other in the word cloud that visualizes the given text. This is captured by an input graph $G = (V,E)$ of desired contacts. We are also given, for each vertex $v \in V$, the dimensions (but not the position) of a *box* B_v, that is, an axis-aligned rectangle. We denote the height and width of B_v by $h(B_v)$ and $w(B_v)$, respectively, or, more briefly, by $h(v)$ and $w(v)$. For each edge $e = (u,v)$ of G, we are given a positive number $p(e) = p(u,v)$, that corresponds to the *profit* of e. For ease of notation, we set $p(u,v) = 0$ for any non-edge $(u,v) \in V^2 \setminus E$ of G.

Given a box B and a point q in the plane, let $B(q)$ be a placement of B with lower left corner q. A *representation* of G is a map $\lambda : V \to \mathbb{R}^2$ such that for any two vertices $u \neq v$, it holds that $B_u(\lambda(u))$ and $B_v(\lambda(v))$ are interior-disjoint. Boxes may *touch*, that is, their boundaries may intersect. If the intersection is non-degenerate, that is, a line segment of positive length, we say that the boxes are *in contact*. We say that a representation λ *realizes* an edge (u,v) of G if boxes $B_u(\lambda(u))$ and $B_v(\lambda(v))$ are in contact.

This yields the problem *Contact Representation of Word Networks* (CROWN): Given an edge-weighted graph G whose vertices correspond to boxes, find a representation of G with the vertex boxes such that every edge of G is realized. In this paper, we study the optimization version of CROWN, MAX-CROWN, where the aim is to maximize the total profit (that is, the sum of the weights) of the realized edges. We also consider the unweighted version of the problem, where all desired contacts yield a profit of 1.

Previous Work. Barth et al. [1] introduced MAX-CROWN and showed that the problem is strongly NP-hard even for trees and weakly NP-hard even for stars. They presented an exact algorithm for cycles and approximation algorithms for stars, trees, planar graphs, and graphs of constant maximum degree; see the first column of Table 1. Some of their solutions use an approximation algorithm with ratio $\alpha = e/(e-1) \approx 1.58$ [13] for the GENERALIZED ASSIGNMENT PROBLEM (GAP): Given a set of bins with capacity constraints and a set of items that possibly have different sizes and values for each bin, pack a maximum-valued subset of items into the bins. The problem is APX-hard [6].

MAX-CROWN is related to finding *rectangle representations* of graphs, where vertices are represented by axis-aligned rectangles with non-intersecting interiors and edges correspond to rectangles with a common boundary of non-zero length. Every graph that can be represented this way is planar and every triangle in such a graph is a facial trian-

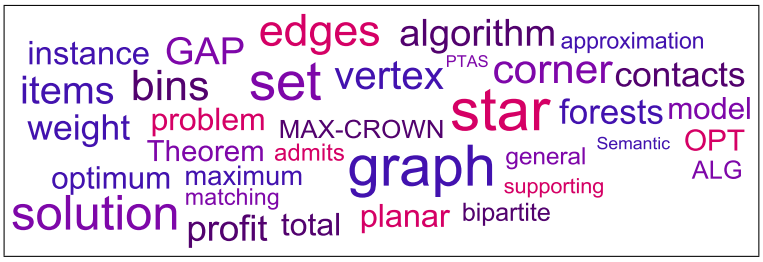


Fig. 1: Semantics-preserving word cloud for the 35 most "important" words in this paper. Following the text processing pipeline of Barth et al. [2], these are the words ranked highest by LexRank [11], after removal of stop words such as "the". The edge profits are proportional to the relative frequency with which the words occur in the same sentences. The layout algorithm of Barth et al. [2] first extracts a heavy star forest from the weighted input graph as in Theorem 6 and then applies a force-directed post-processing.

gle. These two conditions are also sufficient to guarantee a rectangle representation [5]. Rectangle representations play an important role in VLSI layout, cartography, and architecture (floor planning). In a recent survey, Felsner [12] reviews many rectangulation variants. Several interesting problems arise when the rectangles in the representation are restricted. Eppstein et al. [10] consider rectangle representations which can realize any given area-requirement on the rectangles, so-called *area-preserving rectangular cartograms*, which were introduced by Raisz [22] already in the 1930s. Unlike cartograms, in our setting there is no inherent geography, and hence, words can be positioned anywhere. Moreover, each word has fixed dimensions enforced by its importance in the input text, rather than just fixed area. Nöllenburg et al. [20] recently considered a variant where the edge weights prescribe the length of the desired contacts.

Finally, the problem of computing semantics-aware word clouds is related to classic graph layout problems, where the goal is to draw graphs so that vertex labels are readable and Euclidean distances between pairs of vertices are proportional to the underlying graph distance between them. Typically, however, vertices are treated as points and label overlap removal is a post-processing step [9,15]. Most tag cloud and word cloud tools such as Wordle [23] do not show the semantic relationships between words, but force-directed graph layout heuristics are sometimes used to add such functionality [2,8,21,24]. For an example output of such a tool, see Fig. 1.

Our Contribution. Known results and our contributions to MAX-CROWN are shown in Table 1. Note that the results of Barth et al. [1] in column 1 are simply based on existing decompositions of the respective graph classes into star forests or cycles.

Our results rely on a variety of algorithmic tools. First, we devise sophisticated decompositions of the input graphs into heterogeneous classes of subgraphs, which also requires a more general combination method than that of Barth et al. Second, we use randomization to obtain a simple constant-factor approximation for general weighted graphs. Previously, such a result was not even known for unweighted bipartite graphs. Third, to obtain an improved algorithm for the unweighted case, we prove a lower bound on the size of a matching in a planar graph of high average degree. Fourth, we use a

Table 1: Previously known and new results for the unweighted and weighted versions of MAX-CROWN (for $\alpha \approx 1.58$ and any $\varepsilon > 0$).

	Weighted			Unweighted	
Graph class	Ratio [1]	Ratio [new]	Ref.	Ratio	Ref.
cycle, path	1				
star	α	$1+\varepsilon$	Thm. 1		
tree	2α	$2+\varepsilon$	Thm. 1	2	Thm. 7
	NP-hard				
max-degree Δ	$\lfloor(\Delta+1)/2\rfloor$				
planar max-deg. Δ				$1+\varepsilon$	Thm. 8
outerplanar		$3+\varepsilon$	Thm. 3		
planar	5α	$5+\varepsilon$	Thm. 1		
bipartite		$16\alpha/3$ (≈ 8.4)	Thm. 4		
		APX-complete	Thm. 2		
general		$32\alpha/3$ (≈ 16.9; rand.)	Thm. 5	$5+16\alpha/3$	Thm. 9
		$40\alpha/3$ (≈ 21.1; det.)	Thm. 6		

planar separator result of Frederickson [14] to obtain a polynomial-time approximation scheme (PTAS) for degree-bounded planar graphs.

We start our paper with basic results on simple graph classes and prove that MAX-CROWN is APX-complete on bipartite graphs of maximum degree 9 (Section 2). Then, we tackle weighted graphs (Section 3). We obtain improved results for several unweighted graph classes (Section 4). Finally, we list some open problems (Section 5).

Model. As in most work on rectangle contact representations, we do not count *point contacts*, that is, we consider two boxes in contact only if their intersection is a line segment of positive length. Hence, the contact graph of the boxes is planar. Our algorithms can easily be modified to guarantee $O(1)$-approximations also in the model that allows and rewards point contacts [3]. We allow words only to be placed horizontally.

Runtimes. Most of our algorithms involve approximating a number of GAP instances as a subroutine, using either the PTAS [4] if the number of bins is constant or the approximation algorithm of Fleischer et al. [13] for general instances. Because of this, the runtime of our algorithms consists mostly of approximating GAP instances. Both algorithms to approximate GAP instances solve linear programs, so we refrain from explicitly stating the runtime of these algorithms.

For practical purposes, one can use a purely combinatorial approach for approximating GAP [7], which utilizes an algorithm for the KNAPSACK problem as a subroutine. The algorithm translates into a 3-approximation for GAP running in $O(NM)$ time (or a $(2+\varepsilon)$-approximation running in $O(MN\log 1/\varepsilon + M/\varepsilon^4)$ time), where N is the number of items and M is the number of bins. In our setting, the simple 3-approximation implies a randomized 32-approximation (or a deterministic 40-approximation) algorithm with running time $O(|V|^2)$ for MAX-CROWN on general weighted graphs.

2 Some Basic Results

We first present two technical lemmas that will help us prove our main results on weighted and unweighted MAX-CROWN. The second lemma immediately improves the results of Barth et al. [1] for stars, trees, and planar graphs. Finally, we prove APX-completeness of MAX-CROWN on bipartite graphs of bounded maximum degree.

2.1 A Combination Lemma

Several of our algorithms cover the input graph with subgraphs that belong to graph classes for which the MAX-CROWN problem is known to admit good approximations. The following lemma allows us to combine the solutions for the subgraphs. We say that a graph $G=(V,E)$ is *covered* by graphs $G_1=(V,E_1),\ldots,G_k=(V,E_k)$ if $E=E_1\cup\cdots\cup E_k$.

Lemma 1. *Let graph $G=(V,E)$ be covered by graphs $G_1,G_2,\ldots,G_k$. If, for $i=1,2,\ldots,k$, weighted* MAX-CROWN *on graph G_i admits an α_i-approximation, then weighted* MAX-CROWN *on G admits a $\left(\sum_{i=1}^{k}\alpha_i\right)$-approximation.*

Proof. Our algorithm works as follows. For $i=1,\ldots,k$, we apply the α_i-approximation algorithm to G_i and report the result with the largest profit as the result for G. We show that this algorithm has the claimed performance guarantee. For the graphs $G,G_1,\ldots,G_k$, let $\mathrm{OPT},\mathrm{OPT}_1,\ldots,\mathrm{OPT}_k$ be the optimum profits and let $\mathrm{ALG},\mathrm{ALG}_1,\ldots,\mathrm{ALG}_k$ be the profits of the approximate solutions. By definition, $\mathrm{ALG}_i\geq \mathrm{OPT}_i/\alpha_i$ for $i=1,\ldots,k$. Moreover, $\mathrm{OPT}\leq\sum_{i=1}^{k}\mathrm{OPT}_i$ because the edges of G are covered by the edges of $G_1,\ldots,G_k$. Assume, w.l.o.g., that $\mathrm{OPT}_1/\alpha_1=\max_i(\mathrm{OPT}_i/\alpha_i)$. Then

$$\mathrm{ALG}\;=\;\mathrm{ALG}_1\;\geq\;\frac{\mathrm{OPT}_1}{\alpha_1}\;\geq\;\frac{\sum_{i=1}^{k}\mathrm{OPT}_i}{\sum_{i=1}^{k}\alpha_i}\;\geq\;\frac{\mathrm{OPT}}{\sum_{i=1}^{k}\alpha_i}\,.$$ □

2.2 Improvement on existing approximation algorithms

Lemma 2 ([4]). *For any $\varepsilon>0$, there is a $(1+\varepsilon)$-approximation algorithm for* GAP *with a constant number of bins. The algorithm takes $n^{O(1/\varepsilon)}$ time.*

Using Lemmas 1 and 2, we improve the approximation algorithms of Barth et al. [1].

Theorem 1. *Weighted* MAX-CROWN *admits a $(1+\varepsilon)$-approximation algorithm on stars, a $(2+\varepsilon)$-approximation algorithm on trees, and a $(5+\varepsilon)$-approximation algorithm on planar graphs.*

Proof. By Lemma 1, the claim for stars implies the other two claims since a tree can be covered by two star forests and a planar graph can be covered by five star forests in polynomial time [16]. We now show that we can use Lemma 2 to get a PTAS for stars. Here, we give the PTAS for the model with point contacts; in the full version [3], we show how to handle the model without point contacts.

Let u be the center vertex of the star. We create eight bins: four *corner bins* u_1^c,u_2^c,u_3^c, and u_4^c modeling adjacencies on the four corners of the box u, two *horizontal bins* u_1^h

and u_2^h modeling adjacencies on the top and bottom side of u, and two *vertical bins* u_1^v and u_2^v modeling adjacencies on the left and right side of u. The capacity of the corner bins is 1, the capacity of the horizontal bins is the width $w(u)$ of u, and the capacity of the vertical bins is the height $h(u)$ of u. Next, we introduce an item $i(v)$ for any leaf vertex v of the star. The size of $i(v)$ is 1 in any corner bin, $w(v)$ in any horizontal bin, and $h(v)$ in any vertical bin. The profit of $i(v)$ in any bin is the profit $p(u,v)$ of the edge (u,v).

Note that any feasible solution to the MAX-CROWN instance can be normalized so that any box that touches a corner of u has a point contact with u. Hence, the above is an approximation-preserving reduction from weighted MAX-CROWN on stars (with point contacts) to GAP. By Lemma 2, we obtain a PTAS. □

2.3 APX-Completeness

The proof for the following theorem is given in the full version [3].

Theorem 2. *Weighted* MAX-CROWN *is APX-complete even if the input graph is bipartite of maximum degree 9, each edge has profit 1, 2 or 3, and each vertex corresponds to a square of one out of three different sizes.*

3 The Weighted Case

In this section, we provide new approximation algorithms for more involved classes of (weighted) graphs than in the previous section. Recall that $\alpha = e/(e-1) \approx 1.58$. First, we give a $(3+\varepsilon)$-approximation for outerplanar graphs. Then, we present a $16\alpha/3$-approximation for bipartite graphs. For general graphs, we provide a simple randomized $32\alpha/3$-approximation and a deterministic $40\alpha/3$-approximation.

Theorem 3. *Weighted* MAX-CROWN *on outerplanar graphs admits a* $(3+\varepsilon)$*-approximation.*

Proof. It is known that the star arboricity of an outerplanar graph is 3, that is, it can be partitioned into at most three star forests [16]. Here we give a simple algorithm for finding such a partitioning.

Any outerplanar graph has degeneracy at most 2, that is, it has a vertex of degree at most 2. We prove that any outerplanar graph G can be partitioned into three star forests such that every vertex of G is the center of only one star. Clearly, it is sufficient to prove the claim for maximal outerplanar graphs in which all vertices have degree at least 2. We use induction on the number of vertices of G. The base of the induction corresponds to a 3-cycle for which the claim clearly holds. For the induction step, let v be a degree-2 vertex of G and let (v,u) and (v,w) be its incident edges. The graph $G-v$ is maximal outerplanar and thus, by induction hypothesis, it can be partitioned into star forests F_1, F_2, and F_3 such that u is the center of a star in F_1 and w is the center of a star in F_2. Now we can cover G with three star forests: we add (v,u) to F_1, we add (v,w) to F_2, and we create a new star centered at v in F_3.

Applying Lemma 1 and Theorem 1 to the star forests completes the proof. □

Theorem 4. *Weighted* MAX-CROWN *on bipartite graphs admits a* $16\alpha/3$*-approximation.*

Proof. Let $G = (V,E)$ be a bipartite input graph with $V = V_1 \dot\cup V_2$ and $E \subseteq V_1 \times V_2$. Using G, we build an instance of GAP as follows. For each vertex $u \in V_1$, we create eight bins $u_1^c, u_2^c, u_3^c, u_4^c, u_1^h, u_2^h, u_1^v, u_2^v$ and set the capacities exactly as we did for the star center in Theorem 1. Next, we add an item $i(v)$ for every vertex $v \in V_2$. The size of $i(v)$ is, again, 1 in any corner bin, $w(v)$ in any horizontal bin, and $h(v)$ in any vertical bin. For $u \in V_1$, the profit of $i(v)$ is $p(u,v)$ in any bin of u.

It is easy to see that solutions to the GAP instance are equivalent to word cloud solutions (with point contacts) in which the realized edges correspond to a forest of stars with all star centers being vertices of V_1. Hence, we can find an approximate solution of profit $\mathrm{ALG}_1' \geq \mathrm{OPT}_1' / \alpha$ where OPT_1' is the profit of an optimum solution (with point contacts) consisting of a star forest with centers in V_1.

We now show how to get a solution without point contacts. If the three bins on the top side of a vertex u (two corner bins and one horizontal bin) are not completely full, we can slightly move the boxes in the corners so that point contacts are avoided. Otherwise, we remove the lightest item from one of these bins. We treat the three bottommost bins analogously. Note that in both cases we only remove an item if all three bins are completely full. The resulting solution can be realized without point contacts. We do the same for the three left and three right bins and choose the heavier of the two solutions. It is easy to see that we lose at most $1/4$ of the profit for the star center u: Assume that the heaviest solution results from removing weight w_1 from one of the upper and weight w_2 from one of the lower bins. As we remove the lightest items only, the remaining weight from the upper and lower bins is at least $2(w_1 + w_2)$. On the other hand, the weight in the two vertical at least $w_1 + w_2$; otherwise, dropping everything from these vertical bins would be cheaper. Hence, we keep at least weight $3(w_1 + w_2)$.

If we do so for all star centers, we get a solution with profit $\mathrm{ALG}_1 \geq 3/4 \cdot \mathrm{ALG}_1' \geq 3\,\mathrm{OPT}_1' / (4\alpha) \geq 3\,\mathrm{OPT}_1 / (4\alpha)$ where OPT_1 is the profit of an optimum solution (without point contacts) consisting of a star forest with centers in V_1.

Similarly, we can find a solution of profit $\mathrm{ALG}_2 \geq 3\,\mathrm{OPT}_2 / (4\alpha)$ with star centers in V_2, where OPT_2 is the maximum profit that a star forest with centers in V_2 can realize. Among the two solutions, we pick the one with larger profit $\mathrm{ALG} = \max\{\mathrm{ALG}_1, \mathrm{ALG}_2\}$.

Let $G^\star = (V, E^\star)$ be the contact graph realized by a fixed optimum solution, and let $\mathrm{OPT} = p(E^\star)$ be its total profit. We now show that $\mathrm{ALG} \geq 3\,\mathrm{OPT} / (16\alpha)$. As $G^\star$ is a planar bipartite graph, $|E^\star| \leq 2n - 4$. Hence, we can decompose $E^\star$ into two forests H_1 and H_2 using a result of Nash-Williams [18]. We can further decompose H_1 into two star forests S_1 and S_1' in such a way that the star centers of S_1 are in V_1 and the star centers of S_1' are in V_2. Similarly, we decompose H_2 into a forest S_2 of stars with centers in V_1 and a forest S_2' of stars with centers in V_2. As we decomposed the optimum solution into four star forests, one of them—say S_1—has profit $p(S_1) \geq \mathrm{OPT}/4$. On the other hand, $\mathrm{OPT}_1 \geq p(S_1)$. Summing up, we get

$$\mathrm{ALG} \;\geq\; \mathrm{ALG}_1 \;\geq\; 3\,\mathrm{OPT}_1 / (4\alpha) \;\geq\; 3p(S_1)/(4\alpha) \;\geq\; 3\,\mathrm{OPT}/(16\alpha). \qquad \square$$

Theorem 5. *Weighted* MAX-CROWN *on general graphs admits a randomized* $32\alpha/3$*-approximation.*

Proof. Let $G = (V,E)$ be the input graph and let OPT be the weight of a fixed optimum solution. Our algorithm works as follows. We first randomly partition the set of vertices into V_1 and $V_2 = V \setminus V_1$, that is, the probability that a vertex v is included in V_1 is $1/2$. Now we consider the bipartite graph $G' = (V_1 \dot\cup V_2, E')$ with $E' = \{(v_1,v_2) \in E \mid v_1 \in V_1 \text{ and } v_2 \in V_2\}$ that is induced by V_1 and V_2. By applying Theorem 4 on G', we can find a feasible solution for G with weight $\text{ALG} \geq 3\,\text{OPT}'/(16\alpha)$, where OPT' is the weight of an optimum solution for G'.

Any edge of the optimum solution is contained in G' with probability $1/2$. Let $\overline{\text{OPT}}$ be the total weight of the edges of the optimum solution that are present in G'. Then, $E[\overline{\text{OPT}}] = \text{OPT}/2$. So, $E[\text{ALG}] \geq 3E[\text{OPT}']/(16\alpha) \geq 3E[\overline{\text{OPT}}]/(16\alpha) = 3\,\text{OPT}/(32\alpha)$. □

Theorem 6. *Weighted* MAX-CROWN *on general graphs admits a* $40\alpha/3$*-approximation.*

Proof. Let $G = (V,E)$ be the input graph. As in the proof of Theorem 4, our algorithm constructs an instance of GAP based on G. The difference is that, *for every vertex* $v \in V$, we create *both eight bins and an item* $i(v)$. Capacities and sizes remain as before. The profit of placing item $i(v)$ in a bin of vertex u, with $u \neq v$, is $p(u,v)$.

Let OPT be the value of an optimum solution of MAX-CROWN in G, and let OPT_{GAP} be the value of an optimum solution for the constructed instance of GAP. Since any optimum solution of MAX-CROWN, being a planar graph, can be decomposed into five star forests [16], there exists a star forest carrying at least $\text{OPT}/5$ of the total profit. Such a star forest corresponds to a solution of GAP for the constructed instance; therefore, $\text{OPT}_{\text{GAP}} \geq \text{OPT}/5$. Now we compute an α-approximation for the GAP instance, which results in a solution of total profit $\text{ALG}_{\text{GAP}} \geq \text{OPT}_{\text{GAP}}/\alpha \geq \text{OPT}/(5\alpha)$. Next, we show how our solution induces a feasible solution of MAX-CROWN where every vertex $v \in V$ is either a bin or an item.

Consider the directed graph $G_{\text{GAP}} = (V, E_{\text{GAP}})$ with $(u,v) \in E_{\text{GAP}}$ if and only if the item corresponding to $u \in V$ is placed into a bin corresponding to $v \in V$. A connected component in G_{GAP} with n' vertices has at most n' edges since every item can be placed into at most one bin. If $n' = 2$, we arbitrarily make one of the vertices a bin and the other an item. If $n' > 2$, the connected component is a 1-tree, that is, a tree and an edge. We partition the edges into two subgraphs: a star forest and the disjoint union of a star forest and a cycle. Note that both subgraphs can be represented by touching boxes if we allow point contacts because the stars correspond to a GAP solution. Hence, choosing a subgraph with larger weight and post-processing the solution as in the proof of Theorem 4 results in a feasible solution of MAX-CROWN with no point contacts. Initially, we discarded at most half of the weight and the post-processing keeps at least $3/4$ of the weight, so $\text{ALG} \geq 3\,\text{ALG}_{\text{GAP}}/8$. Therefore, $\text{ALG} \geq 3\,\text{OPT}/(40\alpha)$. □

4 The Unweighted Case

In this section, we consider the unweighted MAX-CROWN problem, that is, all desired contacts have profit 1. Thus, we want to maximize the number of edges of the input graph realized by the contact representation. We present approximation algorithms for different graph classes. First, we give a 2-approximation for trees. Then, we present a PTAS for planar graphs of bounded degree. Finally, we provide a $(5+16\alpha/3)$-approximation for general graphs.

Theorem 7. *Unweighted* MAX-CROWN *on trees admits a 2-approximation.*

Proof. Let T be the input tree. We first decompose T into edge-disjoint stars as follows. If T has at most two vertices, then the decomposition is straight-forward. So, we assume w.l.o.g. that T has at least three vertices and is rooted at a non-leaf vertex. Let u be a vertex of T such that all its children, say $v_1,\ldots,v_k$, are leaf vertices. If u is the root of T, then the decomposition contains only one star centered at u. Otherwise, denote by π the parent of u in T, create a star S_u centered at u with edges $(u,\pi),(u,v_1),\ldots,(u,v_k)$ and call the edge (u,π) of S_u the *anchor edge* of S_u. The removal of $u,v_1,\ldots,v_k$ from T results in a new tree. Therefore, we can recursively apply the same procedure. The result is a decomposition of T into edge-disjoint stars covering all edges of T.

We next remove, for each star, its anchor edge from T. We apply the PTAS of Theorem 1 to the resulting star forest and claim that the result is a 2-approximation for T. To prove the claim, consider a star S'_u of the new star forest, centered at u with edges $(u,v_1),\ldots,(u,v_k)$ and let ALG be the total number of contacts realized by the $(1+\varepsilon)$-approximation algorithm on S'_u. We consider the following two cases.

(a) $1 \le k \le 4$: Since it is always possible to realize four contacts of a star, ALG $\ge k$. Note that an optimal solution may realize at most $k+1$ contacts (due to the absence of the anchor edge from S'_u). Hence, our algorithm has approximation ratio $(k+1)/k \le 2$.
(b) $k \ge 5$: Since it is always possible to realize four contacts of a star, we have ALG ≥ 4. On the other hand, an optimal solution realizes at most $(1+\varepsilon)\,\text{ALG}+1$ contacts. Thus, the approximation ratio is $((1+\varepsilon)\,\text{ALG}+1)/\,\text{ALG} \le (1+\varepsilon)+1/4 < 2$.

The theorem follows from the fact that all edges of T are incident to the star centers. □

Next, we develop a PTAS for bounded-degree planar graphs. Our construction needs two lemmas, the first of which was shown by Barth et al. [1].

Lemma 3 ([1]). *If the input graph $G=(V,E)$ has maximum degree Δ then* OPT $\ge 2|E|/(\Delta+1)$.

The second lemma provides an exponential-time exact algorithm for MAX-CROWN. The proof is given in the full version [3].

Lemma 4. *There is an exact algorithm for unweighted* MAX-CROWN *with running time* $2^{O(n\log n)}$.

Theorem 8. *Unweighted* MAX-CROWN *on planar graphs with maximum degree Δ admits a PTAS. More specifically, for any $\varepsilon > 0$ there is an $(1+\varepsilon)$-approximation algorithm with linear running time $n2^{(\Delta/\varepsilon)^{O(1)}}$.*

Proof. Let r be a parameter to be determined later. Frederickson [14] showed that we can find a vertex set $X \subseteq V$ (called *r-division*) of size $O(n/\sqrt{r})$ such that the following holds. The vertex set $V \setminus X$ can be partitioned into n/r vertex sets $V_1,\ldots,V_{n/r}$ such that (i) $|V_i| \le r$ for $i=1,\ldots,n/r$ and (ii) there is no edge running between any two distinct vertex sets V_i and V_j. In what follows, we assume w.l.o.g. that G is connected, as we can apply the PTAS to every connected component separately.

We apply the result of Frederickson to the input graph and compute an r-division X. By removing the vertex set X from the graph, we remove $O(n\Delta/\sqrt{r})$ edges from G.

Now, we apply the exact algorithm of Lemma 4 to each of the induced subgraphs $G[V_i]$ separately. The solution is the union of the optimum solutions to $G[V_i]$.

Since no edge runs between the distinct sets V_i and V_j, the subgraphs $G[V_i]$ cover $G-X$. Let $E^\star$ be the set of edges realized by an optimum solution to G, let $\mathrm{OPT} = |E^\star|$, and let $\mathrm{OPT}' = |E^\star \cap E(G-X)|$. By Lemma 3, we have that $\mathrm{OPT} \geq 2(n-1)/(\Delta+1) = \Omega(n/\Delta)$. When we removed X from G, we removed $O(n\Delta/\sqrt{r})$ edges. Hence, $\mathrm{OPT} = \mathrm{OPT}' + O(n\Delta/\sqrt{r})$ and $\mathrm{OPT}' = \Omega(n(1/\Delta - \Delta/\sqrt{r}))$.

Since we solved each sub-instance $G[V_i]$ optimally and since these sub-instances cover $G-X$, the solution created by our algorithm realizes at least OPT' many edges. Using this fact and the above bounds on OPT and OPT', the total performance of our algorithm can be bounded by

$$\frac{\mathrm{OPT}}{\mathrm{OPT}'} \;=\; \frac{\mathrm{OPT}' + O(n\Delta/\sqrt{r})}{\mathrm{OPT}'} \;=\; 1 + O\left(\frac{n\Delta/\sqrt{r}}{n(1/\Delta - \Delta/\sqrt{r})}\right) \;=\; 1 + O\left(\frac{\Delta^2}{\sqrt{r}-\Delta^2}\right).$$

We want this last term to be smaller than $1+\varepsilon$ for some prescribed error parameter $0 < \varepsilon \leq 1$. It is not hard to verify that this can be achieved by letting $r = \Theta(\Delta^4/\varepsilon^2)$. Since each of the subgraphs $G[V_i]$ has at most r vertices, the total running time for determining the solution is $n2^{(\Delta/\varepsilon)^{O(1)}}$. □

Before tackling the case of general graphs, we need a lower bound on the size of maximum matchings in planar graphs in terms of the numbers of vertices and edges.

Lemma 5. *Any planar graph with n vertices and m edges contains a matching of size at least* $(m-2n)/3$.

Proof. Let G be a planar graph. Our proof is by induction on n. The claim holds for $n = 1$.

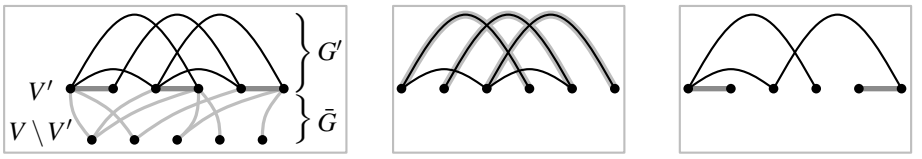
For the inductive step assume that $n > 1$. If G is not connected, the claim follows by applying the inductive hypothesis to every connected component. Now assume that G has a vertex u of degree less than 3. Consider the graph $G' = G - u$ with $n' = n-1$ vertices and $m' \geq m-2$ edges. By the induction hypothesis, G' (and hence, G, too) has a matching of size at least $(m'-2n')/3 \geq ((m-2)-2(n-1))/3 = (m-2n)/3$.

It remains to tackle the case where G is connected and has minimum degree 3. Nishizeki and Baybars [19] showed that any connected planar graph with at least $n \geq 10$ vertices and minimum degree 3 has a matching of size at least $\lceil (n+2)/3 \rceil \geq n/3$. This shows the claim for $n \geq 10$ since $m \leq 3n-6$.

In the remaining cases, G has $n \leq 9$ vertices. Due to planarity, we have $(m-2n)/3 \leq (n-6)/3 \leq 1$. Hence, any nonempty matching is large enough. □

Theorem 9. *Unweighted* MAX-CROWN *on general graphs admits a* $(5+16\alpha/3)$*-approximation.*

Proof. The algorithm first computes a maximal matching M in G. Let V' be the set of vertices matched by M, let G' be the subgraph induced by V', and let E' be the edge set of G'. Note that $\bar{G} = G - E'$ is a bipartite graph with partition $(V', V \setminus V')$. This is because the matching M is maximal, which implies that every edge in $E \setminus E'$ is



(a) G is covered by $\bar{G}$ (bipartite, gray) and G'. The graph G' is induced by the matching M (gray, bold).

(b) maximum matching M'' (gray/black) in $G'' = G' - M$

(c) optimum solution to G': graph G^* (black) and part of M (gray)

Fig. 2: Partitioning the input graph and the optimum solution in the proof of Theorem 9

incident to a vertex in V' and to a vertex not in V'; see Fig. 2a. Hence, we can compute a $16\alpha/3$-approximation to $\bar{G}$ using the algorithm presented in Theorem 4.

Consider the graph $G'' = (V', E' \setminus M)$ and compute a maximum matching M'' in G''; see Fig. 2b. The edge set $M \cup M''$ is a set of vertex-disjoint paths and cycles and can therefore be completely realized [1]. The algorithm realizes this set. Below, we argue that this realization is in fact a 5-approximation for G', which completes the proof (due to Lemma 1 and since G is covered by G' and $\bar{G}$).

Let $n' = |V'|$ be the number of vertices of G'. Let E^* be the set of edges realized by an optimum solution to G', and let $\mathrm{OPT} = |E^*|$. Consider the subgraph $G^* = (V', E^* \setminus M)$ of G''; see Fig. 2c. Note that G^* is planar and contains at least $\mathrm{OPT} - n'/2$ many edges. Applying Lemma 5 to G^*, we conclude that the maximum matching M'' of G'' has size at least $(\mathrm{OPT} - 5n'/2)/3$. Hence, by splitting OPT appropriately, we obtain

$$\mathrm{OPT} \;=\; (\mathrm{OPT} - 5n'/2) + 5n'/2 \;\leq\; 3|M''| + 5|M| \;\leq\; 5|M'' \cup M|\,. \qquad \square$$

5 Conclusions and Open Problems

We presented approximation algorithms for the MAX-CROWN problem, which can be used for constructing semantics-preserving word clouds. Apart from improving approximation factors for various graph classes, many open problems remain. Most of our algorithms are based on covering the input graph by subgraphs and packing solutions for the individual subgraphs. Both subproblems—covering graphs with special types of subgraphs and packing individual solutions together—are interesting problems in their own right. Practical variants of the problem are also of interest, for example, restricting the heights of the boxes to predefined values (determined by font sizes), or defining more than immediate neighbors to be in contact, thus considering non-planar "contact" graphs. Another interesting variant is when the bounding box of the representation has a certain fixed size or aspect ratio.

Acknowledgement. We thank an anonymous reviewer for pointing out a simpler analysis for the last case in the proof of Lemma 5.

References

1. Barth, L., Fabrikant, S.I., Kobourov, S., Lubiw, A., Nöllenburg, M., Okamoto, Y., Pupyrev, S., Squarcella, C., Ueckerdt, T., Wolff, A.: Semantic word cloud representations: Hardness and

approximation algorithms. In: Pardo, A., Viola, A. (eds.) LATIN 2014. LNCS, vol. 8392, pp. 514–525. Springer, Heidelberg (2014)
2. Barth, L., Kobourov, S., Pupyrev, S.: Experimental comparison of semantic word clouds. In: Gudmundsson, J., Katajainen, J. (eds.) SEA 2014. LNCS, vol. 8504, pp. 247–258. Springer, Heidelberg (2014)
3. Bekos, M., van Dijk, T., Fink, M., Kindermann, P., Kobourov, S., Pupyrev, S., Spoerhase, J., Wolff, A.: Improved approximation algorithms for box contact representations. Arxiv report (2014), arxiv.org/abs/1403.4861
4. Briest, P., Krysta, P., Vöcking, B.: Approximation techniques for utilitarian mechanism design. SIAM J. Comput. 40(6), 1587–1622 (2011)
5. Buchsbaum, A.L., Gansner, E.R., Procopiuc, C.M., Venkatasubramanian, S.: Rectangular layouts and contact graphs. ACM Trans. Algorithms 4(1) (2008)
6. Chekuri, C., Khanna, S.: A PTAS for the multiple knapsack problem. In: 11th ACM-SIAM Symp. Discrete Algorithms (SODA). pp. 213–222. SIAM (2000)
7. Cohen, R., Katzir, L., Raz, D.: An efficient approximation for the generalized assignment problem. Inf. Process. Lett. 100(4), 162–166 (2006)
8. Cui, W., Wu, Y., Liu, S., Wei, F., Zhou, M., Qu, H.: Context-preserving dynamic word cloud visualization. IEEE Comput. Graph. Appl. 30(6), 42–53 (2010)
9. Dwyer, T., Marriott, K., Stuckey, P.J.: Fast node overlap removal. In: Healy, P., Nikolov, N.S. (eds.) GD 2005. LNCS, vol. 3843, pp. 153–164. Springer, Heidelberg (2005)
10. Eppstein, D., Mumford, E., Speckmann, B., Verbeek, K.: Area-universal and constrained rectangular layouts. SIAM J. Comput. 41(3), 537–564 (2012)
11. Erkan, G., Radev, D.R.: Lexrank: graph-based lexical centrality as salience in text summarization. J. Artif. Int. Res. 22(1), 457–479 (2004)
12. Felsner, S.: Rectangle and square representations of planar graphs. In: Pach, J. (ed.) Thirty Essays on Geometric Graph Theory, pp. 213–248. Springer, Heidelberg (2013)
13. Fleischer, L., Goemans, M.X., Mirrokni, V., Sviridenko, M.: Tight approximation algorithms for maximum separable assignment problems. Math. Oper. Res. 36(3), 416–431 (2011)
14. Frederickson, G.N.: Fast algorithms for shortest paths in planar graphs, with applications. SIAM J. Comput. 16(6), 1004–1022 (1987)
15. Gansner, E.R., Hu, Y.: Efficient, proximity-preserving node overlap removal. J. Graph Algorithms Appl. 14(1), 53–74 (2010)
16. Hakimi, S.L., Mitchem, J., Schmeichel, E.F.: Star arboricity of graphs. Discrete Math. 149(1–3), 93–98 (1996)
17. Li, H.: Word clustering and disambiguation based on co-occurrence data. J. Nat. Lang. Eng. 8(1), 25–42 (2002)
18. Nash-Williams, C.: Decomposition of finite graphs into forests. J. L. Math. Soc. 39, 12 (1964)
19. Nishizeki, T., Baybars, I.: Lower bounds on the cardinality of the maximum matchings of planar graphs. Discrete Math. 28(3), 255–267 (1979)
20. Nöllenburg, M., Prutkin, R., Rutter, I.: Edge-weighted contact representations of planar graphs. J. Graph Algorithms Appl. 17(4), 441–473 (2013)
21. Paulovich, F.V., Toledo, F.M.B., Telles, G.P., Minghim, R., Nonato, L.G.: Semantic wordification of document collections. Comput. Graph. Forum 31(3), 1145–1153 (2012)
22. Raisz, E.: The rectangular statistical cartogram. Geogr. Review 24(3), 292–296 (1934)
23. Viégas, F.B., Wattenberg, M., Feinberg, J.: Participatory visualization with Wordle. IEEE Trans. Vis. Comput. Graph. 15(6), 1137–1144 (2009)
24. Wu, Y., Provan, T., Wei, F., Liu, S., Ma, K.L.: Semantic-preserving word clouds by seam carving. Comput. Graph. Forum 30(3), 741–750 (2011)