Improved Approximation Algorithms for Box Contact Representations*

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

September 11, 2015

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 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 also consider several planar graph classes (namely 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. Finally, we show that the problem is APX-complete on bipartite graphs of bounded maximum degree.

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 US presidential candidates Obama and McCain. More recently, the German media used them to visualize the newly signed coalition agreement and to compare it to a similar agreement from 2009; see Figure 1. 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 names of towns and cities on geographic maps, for example). The

*A preliminary version of this paper has appeared in *Proc. 22nd Eur. Symp. Algorithms (ESA'14)*, volume 8737 of *Lect. Notes Comput. Sci.*, pages 87–99, Springer-Verlag. 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

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

Figure 1: Word cloud with words that were more important in the 2013 than in the 2009 coalition treaty concluded by the parties that formed the respective German governments (Spiegel Online [26]). Whereas the words concerning "digitalization" (in dashed boxes) form mostly one group, the words concerning "energy" are scattered over the whole cloud (in solid boxes).

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 [25] 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 [19]. In this case, it makes sense to place the two words close to each other in the word cloud that visualizes the given text.

Problem Statement. We capture the semantic connections between words 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 complexity reasons, we require the lengths of the boxes to be rational numbers. For each edge $e = (u, v)$ of G, we are given a positive number $p(e) = p(u, v)$ which 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 $\lambda \colon V \to \mathbb{R}^2$ such that for any two vertices $u \neq v$, it holds that $B_u(\lambda(u))$ and $B_v(\lambda(v))$ have disjoint interior. 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. [3] recently 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 $\alpha = e/(e-1) \approx 1.58$ [15] for the Generalized Assignment Problem (Gap), defined as follows: 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 [8].

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 triangle. These two conditions are also sufficient to guarantee a rectangle representation [7]. Rectangle representations play an important role in VLSI layout, cartography, and architecture (floor planning). In a recent survey, Felsner [14] reviews many rectangulation variants. Several interesting problems arise when the rectangles in the representation are restricted. Eppstein et al. [12] consider rectangle representations which can realize any given area-requirement on the rectangles, so-called *area-preserving rectangular cartograms*, which were introduced by Raisz [24] 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. [22] 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 [11, 17]. Most tag cloud and word cloud tools such as Wordle [25] do not show the semantic relationships between words, but force-directed graph layout heuristics are sometimes used to add such functionality [4, 10, 23, 27]. For an example output of such a tool, see Figure 2.

Model. We consider two different models. In the model with *proper contacts*, 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. This is the standard model used in work on rectangle contact representations. In the model with *point contacts*, we consider two boxes in contact if their intersection is a line segment or a point. In this model, a realized graph is not necessarily planar as four boxes can meet in a point and both diagonals correspond to edges of the input graph. It is, however, easy to see that the graphs that can be realized are 1-planar. If not stated otherwise, our approximation algorithms work for both models. We allow words only to be placed horizontally.

Our Contribution. Known results and our contributions to Max-Crown are shown in Table 1. Note that the results of Barth et al. [3] in column 1 are simply based on existing

Table 1: Previously known and new results for the unweighted and weighted versions of Max-Crown (for $\alpha \approx 1.58$ and any $\varepsilon > 0$). The exact approximation factors are denoted in the corresponding theorems.

	Weighted			Unweighted	
Graph class	Ratio [3]	Ratio [new]	Ref.	Ratio	Ref.
cycle, path	1				
star	α	$1+\varepsilon$	Thm. 1		
tree	2α NP-hard	$2+\varepsilon$	Thm. 1	2	Thm. 7
max-degree Δ	$\lfloor(\Delta+1)/2\rfloor$				
planar max-deg. Δ				$1+\varepsilon$	Thm. 8
outerplanar		$3+\varepsilon$	Thm. 1		
planar	5α	$5+\varepsilon$	Thm. 1		
bipartite		APX-complete	Thm. 10		
with proper contacts		≈ 8.4	Thm. 2		
with point contacts		≈ 9.5	Thm. 2		
general					
with proper contacts		≈ 16.9	Thm. 5	≈ 13.4	Thm. 9
with point contacts		≈ 19	Thm. 5	≈ 16.5	Thm. 9

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 planar separator result of Frederickson [16] to obtain a polynomial-time approximation scheme (PTAS) for degree-bounded planar graphs.

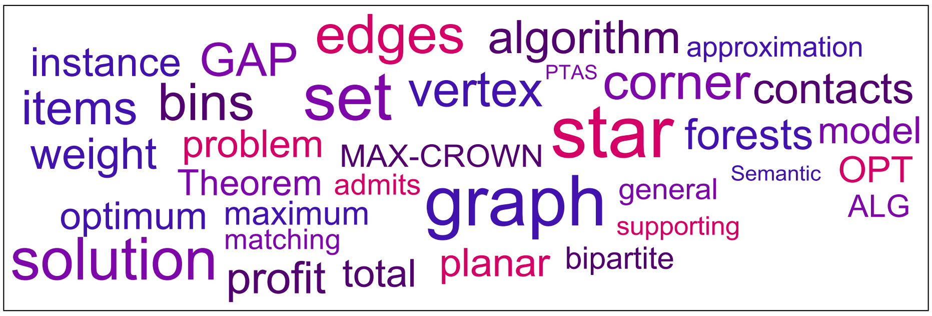


Figure 2: Semantics-preserving word cloud for the 35 most "important" words in this paper. Following the text processing pipeline of Barth et al. [4], these are the words ranked highest by LexRank [13], after removal of stop words such as "the". The edge profits are proportional to the relative frequency with which the words occur in the same sentences. The layout algorithm of Barth et al. [4] first extracts a heavy star forest from the weighted input graph as in Theorem 6 and then applies a force-directed post-processing.

Our other main result is the use of the combination lemma, which, among others, yielded the first approximation algorithms for bipartite and for general graphs; see Section 3. For general graphs, we present a simple randomized solution (based on the solution for bipartite graphs), which we then derandomize. For trees, planar graphs of constant maximum degree, and general graphs, we have improved results in the unweighted case; see Section 4. Finally, we show APX-completeness for bipartite graphs of maximum degree 9 (see Section 5) and list some open problems (see Section 6).

Runtimes. Most of our algorithms involve approximating a number of GAP instances as a subroutine, using either the PTAS [6] if the number of bins is constant or the approximation algorithm of Fleischer et al. [15] 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 [9], 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

In this section, we present two technical lemmas that will help us to prove our main results in the following two sections where we treat the weighted and unweighted cases of MAX-CROWN. The second lemma immediately improves the results of Barth et al. [3] for stars, trees, and planar graphs.

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

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

2.2 Improvement on existing approximation algorithms

The approximation algorithms for stars, trees, and planar graphs provided by Barth et al. [3] use an α-approximation algorithm for Gap instances. We prove that these instances require only a constant number of bins and thus can be approximated using the PTAS of Briest et al. [6].

Lemma 2 ([6])**.** *For any $\epsilon > 0$, there is a $(1+\epsilon)$-approximation algorithm for* Gap *with a constant number of bins. The algorithm takes $n^{O(1/\epsilon)}$ time.* □

Hakimi et al. [18] have studied the star arboricity of certain graph classes, that is, the number of star forests into which a graph can be partitioned. The algorithm of Barth et al. [3] uses some of these results to get approximation algorithms for trees and planar graphs.

Lemma 3 ([18])**.** *Any tree can be partitioned into two star forests, any outerplanar graph can be partitioned into three star forests, and any planar graph can be partitioned into five star forests. All these partitions can be found efficiently.*

Using Lemmas 1, 2 and 3, we improve the approximation algorithms of Barth et al. [3].

Theorem 1. *Weighted* Max-Crown *admits a $(1+\varepsilon)$-approximation algorithm on stars, a $(2+\varepsilon)$-approximation algorithm on trees, a $(3+\varepsilon)$-approximation algorithm on outerplanar graphs, and a $(5+\varepsilon)$-approximation algorithm on planar graphs.*

Proof. By combining Lemma 3 and Lemma 1, the claim for stars implies the other three claims. Note that a PTAS for stars immediately gives a PTAS for star forests since the algorithm can be applied to each star in the forest independently.

We now show that we can use Lemma 2 to get a PTAS for stars. First, we give the PTAS for the model with point contacts. Note that Barth et al. [3] already gave a similar reduction from Max-Crown on stars with point contacts to Gap. However, instead of trying all combinations for the adjacencies in the corners of the center box, we add an additional bin for each corner. We proceed to show that a constant number of bins suffices.

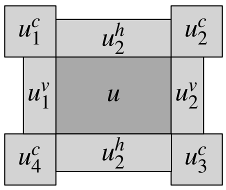


Figure 3: Notation for the PTAS for stars

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; see Figure 3. 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.

We now give a reduction from Max-Crown on stars with proper contacts to Max-Crown on stars with point contacts which has not been proven before. We first assume that all boxes have integral edge lengths, which can be accomplished by scaling. Consider a feasible solution with proper contacts. We modify the solution as follows. Each box that touches a corner of u is moved so that it has a point contact with this corner. Afterwards, we move some of the remaining boxes until all corners of u have point contacts or until we run out of boxes. This yields a solution with point contacts in which there are two opposite sides of u—say the two horizontal sides—which either do not touch any box or from which we removed one box during the modification. Now observe that, if we shrink the two horizontal sides by an amount of $1/2$, then all contacts can be preserved since there was a slack of at least 1 at both horizontal sides. Conversely, observe that any feasible solution with point contacts to the modified instance with shrunk horizontal sides can be transformed into a solution with proper contacts since we always have a slack of at least $1/2$ on both horizontal sides. This shows that there is a correspondence between feasible solutions with proper contacts and feasible solutions with point contacts to a modified instance where we either shrink the horizontal or the vertical sides by $1/2$. The PTAS for MAX-CROWN on stars consists in applying a PTAS to two instances of MAX-CROWN with point contacts where we shrink the horizontal or vertical sides, respectively, and in outputting the better of the two solutions. □

3 The Weighted Case

In this section, we provide new constant-factor 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 present an approximation algorithm for bipartite graphs. Then, for general graphs, we provide a simple randomized approximation algorithm and show how to derandomize it to get a deterministic approximation algorithm that makes $O(n)$ many calls to the α-approximation for GAP. Finally, we provide a deterministic approximation algorithm for general graphs that has a slightly worse approximation factor but only makes a single call to the α-approximation for GAP.

Theorem 2. *Weighted* MAX-CROWN *on bipartite graphs admits*

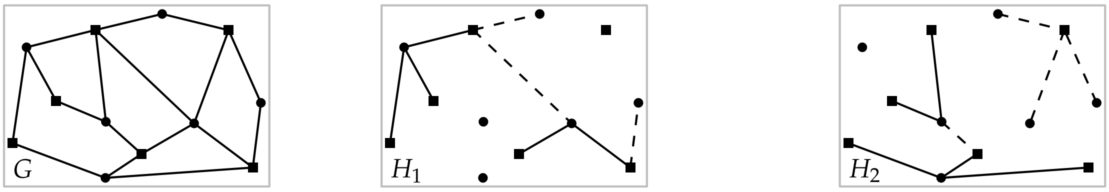
(i) a $16\alpha/3 (\approx 8.4)$-approximation with proper contacts and

(ii) a $6\alpha (\approx 9.5)$-approximation with point contacts.

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. Similarly, we can find a solution of profit $\mathrm{ALG}_2 \geq \mathrm{OPT}_2/\alpha$ with star centers in V_2, where OPT_2 is the maximum profit that a star forest with centers in V_2 can realize.

We first show how to get a solution with point contacts. Among the two solutions described above, 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 $\mathrm{ALG} \geq \mathrm{OPT}/(6\alpha)$. Recall that a realized graph in



(a) The graph $G^\star$ realized by an optimum solution is planar and bipartite.

(b) $G^\star$ can be decomposed into two forests H_1 and H_2 and further into four star forests S_1, S_2 (black) with centers in V_1 (disks) and S'_1, S'_2 (dashed) with centers in V_2 (boxes).

Figure 4: Partitioning the optimum solution in the proof of Theorem 2

this model is 1-planar. Hence, $G^\star$ is a bipartite 1-planar graph. Following a very recent proof of Ackerman [1], we can cover $G^\star$ by a planar bipartite graph G_1^* and a tree G_2^*. As $G_1^\star$ is a planar bipartite graph, each (connected) subgraph $G' = (V', E')$ of G_1^* is also planar bipartite and thus has at most $|E'| \leq 2|V'| - 4$ edges. Hence, we can decompose G_1^* into two forests H_1 and H_2 using a result of Nash-Williams [20]; see Figure 4.

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, and G_2^* into a forest S_3 of stars with centers in V_1 and a forest S'_3 of stars with centers in V_2. As we decomposed the optimum solution into six star forests, one of them—say S_1—has profit $p(S_1) \geq \mathrm{OPT}/6$. On the other hand, $\mathrm{OPT}_1 \geq p(S_1)$. Summing up, we get

$$\mathrm{ALG} \;\geq\; \mathrm{ALG}_1 \;\geq\; \mathrm{OPT}_1/(6\alpha) \;\geq\; p(S_1)/(6\alpha) \;\geq\; \mathrm{OPT}/(6\alpha).$$

We now show how to get a solution with proper contacts. Recall the Gap instance that we constructed at the start of the proof. 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 with proper 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 bins is 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 (with proper 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, using the same argument as above, we can decompose it into four star forests S_1, S'_1, S_2, S'_2 in such a way that the star centers of S_1 and S_2 are in V_1 and the star centers of S'_1 and S'_2 are 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 3. *Weighted* Max-Crown *on general graphs admits*

(i) a randomized $32\alpha/3(\approx 16.9)$*-approximation with proper contacts and*

(ii) a randomized $12\alpha(\approx 19)$*-approximation with point contacts.*

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 two sets V_1 and $V_2 = V \setminus V_1$. To this end we assign every $v \in V$ either to V_1 or to V_2 with probability 1/2 each, so that the random decisions for the nodes are mutually independent. 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 2 on G', we can find a feasible solution for G with weight $\mathrm{ALG} \geq 3\,\mathrm{OPT}'/(16\alpha)$ in the model with proper contacts or weight $\mathrm{ALG} \geq \mathrm{OPT}'/(6\alpha)$ in the model with point contacts, 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 $\overline{\mathrm{OPT}}$ be the total weight of the edges of the optimum solution that are present in G'. Then, $E[\overline{\mathrm{OPT}}] = \mathrm{OPT}/2$. Hence,

$$E[\mathrm{ALG}] \;\geq\; 3E[\mathrm{OPT}']/(16\alpha) \;\geq\; 3E[\overline{\mathrm{OPT}}]/(16\alpha) \;=\; 3\,\mathrm{OPT}/(32\alpha)$$

in the model with proper contacts and

$$E[\mathrm{ALG}] \;\geq\; E[\mathrm{OPT}']/(6\alpha) \;\geq\; E[\overline{\mathrm{OPT}}]/(6\alpha) \;=\; \mathrm{OPT}/(12\alpha)$$

in the model with point contacts. $\square$

In order to derandomize the above algorithms, we make use of the following classical result.

Theorem 4 (Folklore [2])**.** *Let* $n = 2^k - 1$ *and* $d = 2t + 1 \leq n$*. Then there exists a probability space* Ω *of size* $2(n+1)^t$ *and* d*-wise independent random variables* $X_1, \ldots, X_n$ *over* Ω *each of which takes values* 0 *and* 1 *with probability* 1/2*. The space and the variables can be constructed in* $O(tn^{t+1} \log n)$ *time.*

Proof. The construction is described, for example, by Alon and Spencer [2] (Theorem 16.2.1). They don't state the running time of the construction explicitly, so we will do this now.

First, they set up a $(1 + kt) \times n$ matrix H over the field $\mathbb{Z}_2$. Computing an entry of H takes $O(tk^2)$ time, hence $O(t^2 n \log^3 n)$ time in total.

Second, they construct the probability space Ω and the variables $X_1, \ldots, X_n$. To this end, they compute the linear combinations of all $2^{kt+1} = 2(n+1)^t$ many subsets of rows of H, each of which takes $O(nkt)$ time. Using $k = O(\log n)$, this sums up to $O(tn^{t+1} \log n)$, which dominates the first step since $t \leq n$. $\square$

Theorem 5. *Weighted* Max-Crown *on general graphs admits*

(i) a (deterministic) $32\alpha/3(\approx 16.9)$*-approximation with proper contacts and*

(ii) a (deterministic) $12\alpha(\approx 19)$*-approximation with point contacts.*

Both approximation algorithms make $O(n)$ *many calls to the* α*-approximation algorithm for* GAP *on instances with* $O(n)$ *items and bins.*

Proof. Let $G = (V, E)$ be the input graph and let $V = \{v_1, \ldots, v_n\}$. We set $t = 1$ and apply Theorem 4 to construct, in $O(n^2 \log n)$ time, a probability space of size $O(n)$ and random variables $X_1, \ldots, X_n$ that are 3-wise (and thus pairwise) independent.

We now run the algorithms described in the proof of Theorem 3 with the only difference that we do not assign the vertices to V_1 or V_2 mutually independently but by sampling from the space Ω. In doing so, we add v_i to V_1 if $X_i = 0$ and to V_2 if $X_i = 1$. Let (v_i, v_j) be an edge of the optimum solution. Since the variables X_i and X_j are independent, the probability that the edge (v_i, v_j) ends up in the bipartite graph G' is precisely $1/2$ as in the original algorithms. Using an analogous analysis, we can prove that both algorithms retain their expected approximation performance if we use the space Ω. Hence, there must be at least one elementary event in Ω that achieves the expected approximation ratio.

The crucial point is that we can exhaustively check the space Ω running the algorithms for each of the elementary events to *deterministically* find a solution whose approximation ratio is bounded by the expected approximation ratio. □

Note that the above derandomization makes $O(n)$ many calls to the α-approximation algorithm for GAP on $O(n)$ items and bins, two for each of the $O(n)$ many bipartite graphs that are considered. The α-approximation algorithm for GAP is based on LP rounding [9], which might be time consuming. Below, we give a deterministic approximation algorithm solving only one GAP instance with $O(n)$ items and bins. The algorithm obtains a weaker but still constant approximation ratio.

Theorem 6. *Weighted* MAX-CROWN *on general graphs admits*

(i) a $40\alpha/3(\approx 21.1)$*-approximation with proper contacts and*

(ii) a $14\alpha(\approx 22.1)$*-approximation with point contacts.*

Both approximation algorithms make a single call to the α*-approximation algorithm for* GAP *on an instance with* $O(n)$ *items and bins.*

Proof. Let $G = (V, E)$ be the input graph. As in the proof of Theorem 2, 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 $p(u, v)$.

We first show how to get a solution with proper contacts. 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 is a planar graph, it can be decomposed into five star forests following a result of Hakimi et al. [18]. Hence, 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 $\text{ALG}_{\text{GAP}} \geq \text{OPT}_{\text{GAP}}/\alpha$. 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 $G_{\mathrm{GAP}} = (V, E_{\mathrm{GAP}})$ with $(u,v) \in E_{\mathrm{GAP}}$ 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 one an item. If $n' > 2$, the connected component is a 1-tree, that is, a tree and an edge. In this case, we partition the edges into two subgraphs; a star forest and the disjoint union of a star forest and a cycle; see Figure 5. Note that both subgraphs can be represented by touching boxes if we allow point contacts. This is due to the fact that the stars correspond to a solution of GAP. Hence, choosing a subgraph with larger weight and post-processing the solution as in the proof of Theorem 2 results in a feasible solution of Max-Crown with proper contacts. Initially, we discarded at most half of the weight and the post-processing keeps at least 3/4 of the weight, so

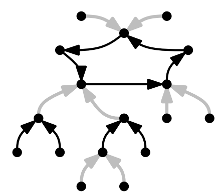


Figure 5: Partitioning a 1-tree into a star forest (gray) and the union of a cycle and a star forest (black)

$$\mathrm{ALG} \;\geq\; 3\,\mathrm{ALG}_{\mathrm{GAP}}/8 \;\geq\; 3\,\mathrm{OPT}_{\mathrm{GAP}}/(8\alpha) \;\geq\; 3\,\mathrm{OPT}/(40\alpha).$$

We now show how to get a solution with point contacts. Since any optimal solution of Max-Crown is a 1-planar graph, it can be decomposed into seven star forests by using the decomposition into a planar graph and a tree proven by Ackerman [1]; thus, $\mathrm{OPT}_{\mathrm{GAP}} \geq \mathrm{OPT}/7$. We proceed with the same algorithm as in the model with proper contacts. However, we do not need the post-processing that discards up to 1/4 of the weight. Therefore,

$$\mathrm{ALG} \;\geq\; \mathrm{ALG}_{\mathrm{GAP}}/2 \;\geq\; \mathrm{OPT}_{\mathrm{GAP}}/(2\alpha) \;\geq\; \mathrm{OPT}/(14\alpha). \qquad \square$$

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, for general graphs, we provide a $(5 + 16\alpha/3)$-approximation with proper contacts and a $(7 + 6\alpha)$-approximation with point contacts.

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 $v_1, \dots, v_k$, 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 $(u,\pi), (u,v_1), \dots, (u,v_k)$ and call the edge (u,π) of S_u the *anchor edge* of S_u. The removal of $u, v_1, \dots, v_k$ 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) $1 \leq k \leq 4$: Since it is always possible to realize four contacts of a star, ALG $\geq k$. 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 $(k+1)/k \leq 2$.

(b) $k \geq 5$: 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 $(1+\varepsilon)\,\mathrm{ALG}+1$ contacts. Thus, the approximation ratio is $((1+\varepsilon)\,\mathrm{ALG}+1)/\,\mathrm{ALG} \leq (1+\varepsilon)+1/4 < 2$.

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, one of which was shown by Barth et al. [3].

Lemma 4 ([3]). *If the input graph has maximum degree* Δ*, then* $\mathrm{OPT} \geq 2|E|/(\Delta+1)$.

The other lemma provides an exponential-time exact algorithm for Max-Crown.

Lemma 5. *There is an exact algorithm for unweighted* Max-Crown *with running time* $2^{O(n \log n)}$.

Proof. Consider a placement which assigns a position $[\ell_B, r_B] \times [b_B, t_B]$ to every box B, with $\ell_B + w(B) = r_B$ and $b_B + h(B) = t_B$. For the x-axis, this gives a (possibly nonstrict) linear order on the values ℓ_B and r_B, where some might be equal. An order on the y-axis is implied similarly. Together, these two orders fully determine the combinatorial structure of overlaps and contacts: for contact, two boxes must have a side of equal value and a side with overlap.

The algorithm enumerates all possible combinations of two such orders using the representation sketched above. On a single axis, this is a permutation of $2n$ variables and, between every two variables adjacent in this permutation, whether they are equal or the second variable has strictly larger value. This representation demonstrates that the number of distinct orders in one dimension is bounded by $O((2n)! \cdot 2^{2n})$, which is $2^{O(n \log n)}$. The number of combinations of two such orders also satisfies this bound.

For any given pair of orders, it can be determined if they imply overlaps and what the objective value is: count the number of profitable contacts. If there are no overlaps, the existence of an actual placement realizing the orders is tested using linear programming. As these tests run in polynomial time, an optimal placement can be found in $2^{O(n \log n)}$ time. □

We will now utilize these two lemmas to obtain a PTAS for planar graphs with bounded degree.

Theorem 8. *Unweighted* Max-Crown *on planar graphs with n vertices and maximum degree Δ admits a PTAS. More specifically, for any $\varepsilon > 0$ there is an $(1+\varepsilon)$-approximation algorithm with linear running time* $n 2^{(\Delta/\varepsilon)^{O(1)}}$.

Proof. Let $G = (V, E)$ be a graph and let r be a parameter to be determined later. Frederickson [16] showed that we can find a vertex set $X \subseteq V$ (called *r-division*) of size $O(n/\sqrt{r})$ in $O(n \log n)$ time 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) $|V_i| \leq r$ 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 5 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 $\mathrm{OPT}' = |E^\star \cap E(G-X)|$. Since G is connected, it has $|E| \geq n-1$ edges. Thus, by Lemma 4, 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 6. *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. If 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.

Nishizeki and Baybars [21] showed that any connected planar graph with at least $n \geq 10$ vertices and minimum degree 3 has a matching of size at least $\lceil (n+2)/3 \rceil \geq n/3 > (n-6)/3 \geq (m-2n)/3$ since $m \leq 3n-6$.

It remains to tackle the case where G has at least $n \geq 10$ vertices and is not connected or has a vertex of degree less than 3. Our proof is by induction on n. 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 inductive hypothesis G' (and hence, G, too) has a matching of size at least

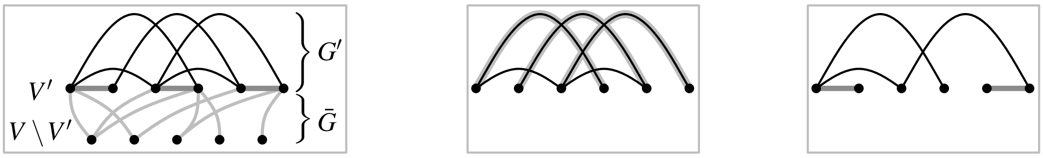
$$(m' - 2n')/3 \geq ((m-2) - 2(n-1))/3 = (m-2n)/3.$$ □

We are now ready to present an approximation algorithm for general graphs.

Theorem 9. *Unweighted* Max-Crown *on general graphs admits*

(i) a $(5 + 16\alpha/3)(\approx 13.4)$-approximation with proper contacts and

(ii) a $(7 + 6\alpha)(\approx 16.4)$-approximation with point contacts.



(a) G is covered by $\bar{G}$ (bipartite, gray) and G'; maximal 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).

Figure 6: Partitioning the input graph and the optimum solution in the proof of Theorem 9

Proof. We first show how to get a solution with proper contacts. 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 incident to a vertex in V' and to a vertex not in V'; see Figure 6a. Hence, we can compute a $16\alpha/3$- approximation to $\bar{G}$ using the algorithm presented in Theorem 2.

Consider the graph $G'' = (V', E' \setminus M)$ and compute a maximum matching M'' in G''; see Figure 6b. The edge set $M \cup M''$ is a set of vertex-disjoint paths and cycles and can therefore be completely realized [3]. 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 Figure 6c. Note that G^* is planar and contains at least $\mathrm{OPT} - n'/2$ many edges. Applying Lemma 6 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

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

We now show how to get a solution with point contacts. We use the same algorithm as described for the model with proper contacts, so we only have to adjust the analysis. We can compute a 6α- approximation to the bipartite graph $\bar{G}$ using the algorithm presented in Theorem 2.

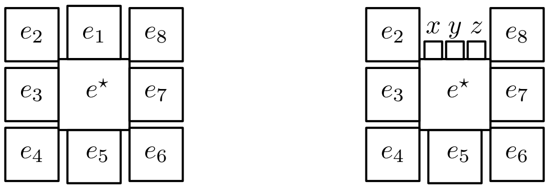
It is easy to prove that any 1-planar graph with m edges and n vertices contains a matching of size at least $(m - 3n)/3$: we planarize the graph (by removing at most n edges) and then apply Lemma 6. Thus, the maximum matching M'' of G'' has size at least $(\mathrm{OPT} - 7n'/2)/3)$. Hence, by splitting OPT appropriately, we obtain

$$\mathrm{OPT} \;=\; (\mathrm{OPT} - 7n'/2) + 7n'/2 \;\leq\; 3|M''| + 7|M| \;\leq\; 7|M'' \cup M|\,. \qquad \square$$

5 APX-Completeness

In this section, we prove APX-completeness of weighted Max-Crown by giving a reduction from 3-dimensional matching. This reduction works both in the model without and in the model with point contacts.

Theorem 10. *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.*



(a) profit $7 \cdot 3 + 2 = 23$ (b) profit $7 \cdot 3 + 3 \cdot 1 = 24$

Figure 7: The two possible configurations of a hyperedge $e = (x, y, z)$ in the proof of Theorem 10

Proof. We give a reduction from 3-dimensional matching (3DM). An instance of this problem is given by three disjoint sets X, Y, Z with cardinalities $|X| = |Y| = |Z| = k$ and a set $E \subseteq X \times Y \times Z$ of hyperedges. The objective is to find a set $M \subseteq E$, called *matching*, such that no element of $V = X \cup Y \cup Z$ is contained in more than one hyperedge in M and such that $|M|$ is maximized.

The problem is known to be APX-hard [15]. More specifically, for the special case of 3DM where every $v \in V$ is contained in at most three hyperedges (hence $|E| \leq 3k$) it is NP-hard to decide whether the maximum matching has cardinality k or only $k(1 - \varepsilon_0)$ for some constant $0 < \varepsilon_0 < 1$. We reduce from this special case of 3DM to Max-Crown.

To this end, we construct the following Max-Crown instance from a given 3DM instance. We create, for each $v \in V$, a square of side length 1. For each hyperedge $e \in E$, we create nine squares $e^\star, e_1, \ldots, e_8$ where $e^\star$ has side length 3.5 and $e_1, \ldots, e_8$ have side length 3. In the desired contact graph, we create an edge $(e^\star, e_1)$ of profit 2 and, for $i = 2, \ldots, 8$, an edge $(e^\star, e_i)$ of profit 3. We also create an edge $(e^\star, v)$ of profit 1 if v is incident to e in the 3DM instance.

Consider an optimum solution to the above Max-Crown instance. It is not hard to verify that, for any hyperedge $e = (x, y, z)$, the solution will realize the edges $(e^\star, e_i)$ for $i = 2, \ldots, 8$. Moreover, we can assume w.l.o.g. that the solution either realizes all three adjacencies $(e^\star, x)$, $(e^\star, y)$, and $(e^\star, z)$ of total profit 3 or the adjacency $(e^\star, e_1)$ of profit 2; see Figure 7. We call such a solution *well-formed*.

Assume that there is a solution M to the 3DM instance of cardinality k. Then this can be transformed into a well-formed solution to Max-Crown of profit $(7 \cdot 3 + 2)|E| + |M| = 23|E| + k$.

Conversely, suppose that the maximum matching has cardinality at most $(1 - \varepsilon_0)k$. Consider an optimum solution to the respective Max-Crown instance. We may assume that the solution is well-formed. Let M be the set of hyperedges $e = (x, y, z)$ for which all three adjacencies $(e^\star, x), (e^\star, y), (e^\star, z)$ are realized. Then, the profit of this solution is $(7 \cdot 3 + 2)|E| + |M| = 23|E| + |M|$. Note that M is in fact a matching because the solution to Max-Crown was well-formed. Thus, the optimum profit is bounded by $23|E| + (1 - \varepsilon_0)k$.

Hence, it is NP-hard to distinguish between instances with OPT $\geq 23|E| + k$ and instances with OPT $\leq 23|E| + (1 - \varepsilon_0)k$. Using $|E| \leq 3k$, this implies that there cannot be any approximation algorithm of ratio less than

$$\frac{23|E| + k}{23|E| + (1 - \varepsilon_0)k} = 1 + \frac{\varepsilon_0 k}{23|E| + (1 - \varepsilon_0)k} \geq 1 + \frac{\varepsilon_0 k}{(70 - \varepsilon_0)k} = 1 + \frac{\varepsilon_0}{70 - \varepsilon_0},$$

which is a constant strictly larger than 1. □

6 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 which may lead to algorithms with better guarantees. 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.

Acknowledgments. We thank the anonymous reviewers for helping us to improve the presentation of our paper. We particularly thank the reviewer who contributed the idea to derandomize our algorithms for the general weighted case using Theorem 4.

References

[1] E. Ackerman. A note on 1-planar graphs. *Discrete Appl. Math.*, 175:104–108, 2014. 8, 11

[2] N. Alon and J. Spencer. *The Probabilistic Method.* John Wiley, 1992. 9

[3] L. Barth, S. I. Fabrikant, S. Kobourov, A. Lubiw, M. Nöllenburg, Y. Okamoto, S. Pupyrev, C. Squarcella, T. Ueckerdt, and A. Wolff. Semantic word cloud representations: Hardness and approximation algorithms. In A. Pardo and A. Viola, editors, *Proc. 11th Latin American Symp. Theor. Inform. (LATIN'14)*, volume 8392 of *Lect. Notes Comput. Sci.*, pages 514–525. Springer, Heidelberg, 2014. 3, 4, 5, 6, 12, 14

[4] L. Barth, S. Kobourov, and S. Pupyrev. Experimental comparison of semantic word clouds. In J. Gudmundsson and J. Katajainen, editors, *Proc. 13th Int. Symp. Experimental Algorithms (SEA'14)*, volume 8504 of *Lect. Notes Comput. Sci.*, pages 247–258. Springer, Heidelberg, 2014. 3, 4

[5] P. Briest, P. Krysta, and B. Vöcking. Approximation techniques for utilitarian mechanism design. In H. N. Gabow and R. Fagin, editors, *Proc. 37th Ann. ACM Symp. Theory of Computing (STOC'05)*, pages 39–48. ACM, 2005.

[6] P. Briest, P. Krysta, and B. Vöcking. Approximation techniques for utilitarian mechanism design. *SIAM J. Comput.*, 40(6):1587–1622, 2011. 5, 6

[7] A. L. Buchsbaum, E. R. Gansner, C. M. Procopiuc, and S. Venkatasubramanian. Rectangular layouts and contact graphs. *ACM Trans. Algorithms*, 4(1), 2008. 3

[8] C. Chekuri and S. Khanna. A PTAS for the multiple knapsack problem. In *Proc. 11th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA'00)*, pages 213–222. SIAM, 2000. 3

[9] R. Cohen, L. Katzir, and D. Raz. An efficient approximation for the generalized assignment problem. *Inform. Process. Lett.*, 100(4):162–166, 2006. 5, 10

[10] W. Cui, Y. Wu, S. Liu, F. Wei, M. Zhou, and H. Qu. Context-preserving dynamic word cloud visualization. *IEEE Comput. Graph. Appl.*, 30(6):42–53, 2010. 3

[11] T. Dwyer, K. Marriott, and P. J. Stuckey. Fast node overlap removal. In P. Healy and N. S. Nikolov, editors, *Proc. 13th Int. Symp. Graph Drawing (GD'05)*, volume 3843 of *Lect. Notes Comput. Sci.*, pages 153–164. Springer, Heidelberg, 2005. 3

[12] D. Eppstein, E. Mumford, B. Speckmann, and K. Verbeek. Area-universal and constrained rectangular layouts. *SIAM J. Comput.*, 41(3):537–564, 2012. 3

[13] G. Erkan and D. R. Radev. Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, 2004. 4

[14] S. Felsner. Rectangle and square representations of planar graphs. In J. Pach, editor, *Thirty Essays on Geometric Graph Theory*, pages 213–248. Springer, Heidelberg, 2013. 3

[15] L. Fleischer, M. X. Goemans, V. Mirrokni, and M. Sviridenko. Tight approximation algorithms for maximum separable assignment problems. *Math. Oper. Res.*, 36(3):416–431, 2011. 3, 5, 15

[16] G. N. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM J. Comput.*, 16(6):1004–1022, 1987. 4, 12

[17] E. R. Gansner and Y. Hu. Efficient, proximity-preserving node overlap removal. *J. Graph Algorithms Appl.*, 14(1):53–74, 2010. 3

[18] S. L. Hakimi, J. Mitchem, and E. F. Schmeichel. Star arboricity of graphs. *Discrete Math.*, 149(1–3):93–98, 1996. 6, 10

[19] H. Li. Word clustering and disambiguation based on co-occurrence data. *J. Nat. Lang. Eng.*, 8(1):25–42, 2002. 2

[20] C. Nash-Williams. Decomposition of finite graphs into forests. *J. London Math. Soc.*, 39:12, 1964. 8

[21] T. Nishizeki and I. Baybars. Lower bounds on the cardinality of the maximum matchings of planar graphs. *Discrete Math.*, 28(3):255–267, 1979. 13

[22] M. Nöllenburg, R. Prutkin, and I. Rutter. Edge-weighted contact representations of planar graphs. *J. Graph Algorithms Appl.*, 17(4):441–473, 2013. 3

[23] F. V. Paulovich, F. M. B. Toledo, G. P. Telles, R. Minghim, and L. G. Nonato. Semantic wordification of document collections. *Comput. Graph. Forum*, 31(3):1145–1153, 2012. 3

[24] E. Raisz. The rectangular statistical cartogram. *Geogr. Review*, 24(3):292–296, 1934. 3

[25] F. B. Viégas, M. Wattenberg, and J. Feinberg. Participatory visualization with Wordle. *IEEE Trans. Visual. Comput. Graphics*, 15(6):1137–1144, 2009. 2, 3

[26] S. Weiland. Der Koalitionsvertrag im Schnellcheck (Quick overview of the [German] coalition agreement). Spiegel Online, www.spiegel.de/politik/deutschland/was-der-koalitionsvertrag-deutschland-bringt-a-935856.html. Click on "Fotos", 27 Nov. 2013. 2

[27] Y. Wu, T. Provan, F. Wei, S. Liu, and K.-L. Ma. Semantic-preserving word clouds by seam carving. *Comput. Graph. Forum*, 30(3):741–750, 2011. 3