Improved Approximation Algorithms for Semantic Word Clouds

Michael A. Bekos* Thomas C. van Dijk† Martin Fink† Philipp Kindermann†
Stephen Kobourov‡ Sergey Pupyrev‡ Joachim Spoerhase† Alexander Wolff†

December 3, 2013

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-hard on bipartite graphs of bounded maximum degree.

*Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany
†Lehrstuhl für Informatik I, Universität Würzburg, http://www1.informatik.uni-wuerzburg.de/en/staff/
‡Department of Computer Science, University of Arizona

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 [23]. 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 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 [22] 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 [16]. 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 leads to the problem CONTACT REPRESENTATION OF WORD NETWORKS (CROWN) that we study in this paper.

In CROWN, the input is a 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 $(u, v) \in V^2 \setminus E$.

Given a box B and a point $p = (x, y)$ in the plane, let $B(p)$ be a placement of B with lower left corner p. 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 $B_v(\lambda(B_v))$ are in contact. This yields the following problem.

Contact Representation of Word Networks (MAX-CROWN): Given an edge-weighted graph G whose vertices correspond to boxes, find a representation of G with the vertex boxes that maximizes the total profit (that is, the weight) of the realized edges. We also consider the unweighted version of the problem, where all desired contacts yield a profit of 1.

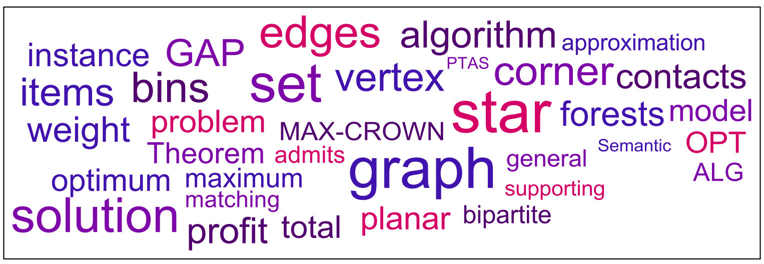


Figure 1: Semantics-preserving word cloud for the 35 most "important" words in this paper. Following the text processing pipeline of Barth et al. [3], these are the words ranked highest by LexRank [9], after removal of stop words such as "the". The edge profits are proportional to the relative frequency with which the words occur in the same sentences. The layout algorithm of Barth et al. [3] first extracts a heavy star forest from the weighted input graph as in Theorem 5 and then applies a force-directed post-processing.

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

Table 1: Previously known and new results for the unweighted and weighted versions of MAX-CROWN (for $\alpha \approx 1.58$ and any $\varepsilon > 0$). Note that Barth et al. [2] counted point contacts of boxes, while we count only proper contacts. To overcome this, the postprocessing presented in the proof of Theorem 3 can be applied to their results. In this case, α has to be replaced by $4\alpha/3$ in the results in column 1.

Previous Work. Barth et al. [2] 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 $\alpha = e/(e-1) \approx 1.58$ [11] 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 [5].

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 [4]. Rectangle representations play an important role in VLSI layout, cartography, and architecture (floor planning). In a recent survey, Felsner [10] reviews many rectangulation variants. Several interesting problems arise when the rectangles in the representation are restricted. Eppstein et al. [8] consider rectangle representations which can realize any given area-requirement on the rectangles, so-called *area-preserving rectangular cartograms*, which were introduced by Raisz [21] 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. [19] 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 [7, 13]. Most tag cloud and word cloud tools such as Wordle [22] do not show the semantic relationships between words, but force-directed graph layout heuristics are sometimes used to add such functionality [3, 6, 15, 20, 24].

Our Contribution. Known results and our contributions to MAX-CROWN are shown in Table 1. Our results rely on two main tools; (i) a PTAS for a special case of GAP and (ii) a lemma for combining results for subgraphs of the given input graph; see Section 2. The PTAS is based on rounding fractional LP solutions; it is one of our main results. The combination lemma is quite simple but very useful when a graph can be covered by few subgraphs that belong to graph classes that admit good approximations for MAX-CROWN. For stars, trees and planar graphs, it suffices to plug the GAP PTAS into the algorithms of Barth et al. [2] to improve their results. Our algorithm for outerplanar graphs, which have not been studied before, also relies on the GAP PTAS.

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) and a more involved deterministic algorithm. 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-hardness for bipartite graphs of maximum degree 9 (see Section 5) and list some open problems (see Section 6).

Model. As in most work on rectangle contact representations, we do not count point contacts of boxes. In other words, we consider two boxes in contact only if their intersection is a line segment of positive length. This type of contact is called *proper contact*. In this model, the contact graph of the boxes is clearly planar. With small modifications, our algorithms do, however, guarantee constant-factor approximations also in the model that allows and rewards point contacts. We discuss the differences in Appendix A.

Runtimes. Most of our algorithms involve approximating a number of GAP instances as a subroutine, using either the PTAS presented in Section 2.2 or the approximation algorithm of Fleischer et al. [11]. Because of this, the runtime of our algorithms consists mostly of approximating GAP instances. Both the PTAS and the existing algorithm solve linear programs, so we refrain from explicitly stating the runtime of these algorithms.

2 Preliminaries

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. [2] concerning 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 $G_1 = (V,E_1),\dots,G_k = (V,E_k)$ if $E = E_1 \cup \dots \cup E_k$.

Lemma 1. *Let graph $G = (V,E)$ be covered by graphs $G_1,G_2,\dots,G_k$. If, for $i = 1,2,\dots,k$, 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 $i = 1,\dots,k$, 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 $G,G_1,\dots,G_k$, let $\mathrm{OPT},\mathrm{OPT}_1,\dots,\mathrm{OPT}_k$ be the optimum profits and let $\mathrm{ALG},\mathrm{ALG}_1,\dots,\mathrm{ALG}_k$ be the profits of the approximate solutions. By definition, $\mathrm{ALG}_i \geq \mathrm{OPT}_i/\alpha_i$ for $i = 1,\dots,k$. Moreover, $\mathrm{OPT} \leq \sum_{i=1}^{k} \mathrm{OPT}_i$ because the edges of G are covered by the edges of $G_1,\dots,G_k$. Assume, w.l.o.g., that $\mathrm{OPT}_1/\alpha_1 = \max_i(\mathrm{OPT}_i/\alpha_i)$. Then

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

2.2 A PTAS for GAP with a Constant Number of Bins

Consider an instance of GAP with items $i = 1,\dots,n$ and bins $j = 1,\dots,m$. Bin j has capacity s_j and, for this bin, item i has size s_{ij} and profit p_{ij}. Note that the problem is NP-hard even for $m = 2$; PARTITION is a special case of GAP. We provide a PTAS for constant m. We use the following LP relaxation.

$$\max \quad \sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}x_{ij}$$

$$\text{s. t.} \quad \sum_{j=1}^{m} x_{ij} \le 1 \quad i = 1,\dots,n \tag{1}$$

$$\sum_{i=1}^{n} s_{ij}x_{ij} \le s_j \quad j = 1,\dots,m \tag{2}$$

$$x_{ij} \ge 0 \quad i = 1,\dots,n, j = 1,\dots,m \tag{3}$$

We select a positive integer k, the parameter that corresponds to the accuracy of the algorithm. The algorithm works as follows. We iterate over all possible assignments of at most k items to bins. (The idea is to guess the assignment of the k items giving the highest profit.) For each such assignment we restrict the solution space to extensions of the given partial assignment. This is achieved by the following pruning operation. Let I be the set of fixed items. Remove the items in I from the item set. Reduce the size of each bin by the size already occupied by the fixed items (as given by the partial assignment). If the profit p_{ij} of some remaining item i for some bin j is larger than the minimum profit of the items in the partial assignment then this profit it set to 0 (because the fixed items were assumed to be the ones with the highest profit). For the residual GAP instance, we solve the above LP relaxation. In fact, we compute an optimum extreme point solution $\mathbf{x}$. All fractional values in $\mathbf{x}$ are set to 0. We assign the items of the residual instance as given by the (now integral) solution $\mathbf{x}$.

We claim that the above algorithm is a PTAS if m is constant and the parameter k is chosen sufficiently large. First note that the exhaustive search in the above algorithm takes $O\left(\binom{n}{k}m^k\right) = O(n^k)$ steps (when m is treated as a constant), and solving the LP can be done in polynomial time.

We now analyze the approximation performance of the algorithm. Consider an optimum solution to the GAP instance and let I^* be the k items in this solution achieving the highest profit. If the optimum solution assigns less than k items then the solution will already be found by the exhaustive search phase. Otherwise, the exhaustive search phase of the algorithm will consider the set I^* and an assignment of it as in the optimum solution. Let P^* be the profit achieved by the items in I^* in the optimum solution. Note that the optimum solution also provides a feasible assignment of the remaining items for the pruned instance generated by the algorithm. The profit $\bar{P}$ achieved by this solution is the same as the profit of the items in the optimum solution that are not in I^*. Here, note that the items in I^* have the highest profits in the optimum solution. Therefore, the profits of the remaining items in the optimum solution are not affected by the modification of the profits in the pruning operation. Thus $\text{OPT} = P^* + \bar{P}$. The profit P achieved by the fractional optimum solution $\mathbf{x}$ can only be higher than $\bar{P}$ and, hence, $\text{OPT} \le P^* + P$.

We now analyze the effect of the rounding step. The crucial insight is that at most m fractional variables are set to 0 by this step and, hence, the loss is small in comparison to P^*.

The above LP has mn variables and $n+m+mn$ constraints (1), (2), (3). By standard polyhedral theory, an extreme point solution $\mathbf{x}$ satisfies at least mn of these constraints with equality. Let ℓ be the number of positive variables x_{ij} in $\mathbf{x}$. For these variables, constraint (3) is not tight. Hence at least ℓ of the constraints (1), (2) must be tight. This implies that at least $\ell - m$ of the constraints (1) are tight. Let ℓ' be the number of items i where constraint (1) is tight and all variables x_{ij} are integral. This means that exactly one of these variables is 1 while the remaining ones are 0. There are at least $\ell - m - \ell'$ items i where constraint (1) is tight but where there are non-integral variables x_{ij}. For these items at least two of their variables are positive. Since there are ℓ positive variables in total, we have that $\ell' + 2(\ell - m - \ell') \le \ell$, which implies $\ell' \ge \ell - m$. Consequently $\mathbf{x}$ has at most m fractional entries. Note that in the residual instance no profit p_{ij} is larger than P^*/k. Hence, the loss in profit when rounding the fractional variables in $\mathbf{x}$ down to 0 is bounded by mP^*/k. This yields a total profit of at least

$$P^* + P - \frac{mP^*}{k} \;\geq\; \left(1 - \frac{m}{k}\right)(P^* + P) \;\geq\; \left(1 - \frac{m}{k}\right)\mathrm{OPT}\,.$$

Thus, for any $\varepsilon > 0$, we achieve a $(1+\varepsilon)$-approximation by setting $k = \lceil (1+1/\varepsilon)m \rceil = \Theta(m/\varepsilon)$. This yields the following lemma.

Lemma 2. *For any $\varepsilon > 0$, there is a $(1+\varepsilon)$-approximation algorithm for* GAP *with a constant number of bins. The algorithm requires solving $n^{O(1/\varepsilon)}$ many LPs with $O(n)$ many variables and constraints each.*

Using the two above lemmas, we improve the approximation algorithms of Barth et al. [2].

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 [14].

We now show that we can use Lemma 2 to get a PTAS for stars. We first give the PTAS for the model with point contacts and then argue how to tackle 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; see Fig. 2. 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).

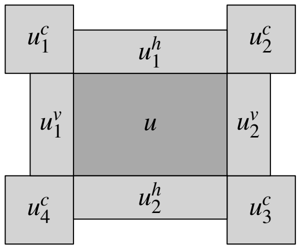


Figure 2: PTAS for stars

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.

Now we show how we can reduce the case without point contacts to the model with point contacts. We first assume that all boxes have integral edge lengths, which can be accomplished by scaling. Consider a feasible solution without point contacts. We now 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 shrunken horizontal sides can be transformed into a solution without point 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 without point 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 consider the weighted MAX-CROWN problem. 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 2. *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 [14]. 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 Lemmas 1 and 2 to these three star forests completes the proof. □

Theorem 3. *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 move the boxes in the corners a bit so that we have proper contacts. 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 finally 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. If we do this 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.

Analogously, 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 whose profit $\mathrm{ALG} = \max\{\mathrm{ALG}_1, \mathrm{ALG}_2\}$ is larger.

Let $G^\star = (V, E^\star)$ be the contact graph realized by a fixed the 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 [17]; see Fig. 4 in Appendix B. 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 4. *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 3 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, $E[\overline{\text{OPT}}] = \text{OPT}/2$. Hence,

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

Theorem 5. *Weighted* MAX-CROWN *on general graphs admits a* $40\alpha/3$*-approximation.*

Proof. Let $G = (V, E)$ be the input graph. Similarly to the proof of Theorem 3, 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 $u \neq v$ 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 [14], 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 $G' = (V, E')$ with $(u, v) \in E'$ 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 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. In this case, we partition the vertices into two subgraphs; a star forest and the union of a star forest and a cycle; see Fig. 3. 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 3 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, $\text{ALG} \geq 3\,\text{OPT}/(40\alpha)$. $\square$

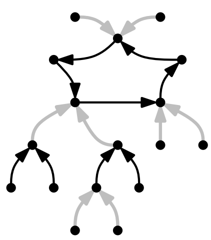


Figure 3: Partitioning a 1-tree into a star forest (gray) and the union of a cycle and a star forest (black).

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 $(5+4\alpha)$-approximation for general graphs.

Theorem 6. *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 $(u,v_1),\dots,(u,v_k)$ 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 factor $(k+1)/k \le 2$.

(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 $(1+\varepsilon)\,\mathrm{ALG}+1$ contacts. Thus, the approximation factor of our algorithm is $((1+\varepsilon)\,\mathrm{ALG}+1)/\,\mathrm{ALG} \le (1+\varepsilon)+1/4 < 2$.

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

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. [2].

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

The second lemma provides an exponential-time exact algorithm for MAX-CROWN.

Lemma 4. *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, with $\ell_B+w(B)=r_B$ and $b_B+h(B)=t_B$. For the x-axis, this gives a (non-strict) linear order on the values ℓ_B and r_B; 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, both of which can be seen from the orders.) The algorithm enumerates all possible combinations of these orders. A single order can be enumerated using a permutation of the variables and, between every two variables adjacent in this permutation, whether it is '$=$' or '$\le$'. The number of orders is bounded by $O(2^{2n}(2n)!)$, for a total of $2^{O(n\log n)}$ combinations. For any given pair of orders, it can be determined if they imply overlaps and what the objective value is: 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. □

Theorem 7. *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 [12] showed that we can find a vertex set $X \subseteq V$ (called *separator*) 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 $V_1,\dots,V_{n/r}$ such that (i) $|V_i|\le r$ for $i=1,\dots,n/r$ 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 a separator 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 $\mathrm{OPT}\ge 2(n-1)/(\Delta+1)=\Omega(n/\Delta)$. 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

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

We want this last term to be smaller than $1+\varepsilon$ for some prescribed error parameter $0<\varepsilon\le 1$. 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 of 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 clearly 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 $m'\ge m-2$ edges. By the inductive hypothesis G' (and hence, G, too) has a matching of size at least $(m'-2n')/3 \ge ((m-2)-2(n-1))/3 = (m-2n)/3$.

It remains to tackle the case where G is connected and has minimum degree 3. Nishizeki and Baybars [18] showed that any connected planar graph with at least $n\ge 10$ vertices and minimum degree 3 has a matching of size at least $\lceil (n+2)/3\rceil \ge n/3$. This shows the claim for $n\ge 10$ since $m\le 3n-6$. Finally, we consider the case that G is connected, has minimum degree 3 and $n\le 9$ vertices.

First, we assume that a maximum matching of G consists of a single edge $e=(u,v)$. Any edge in G is either equal to or incident on e. Since the minimum degree of G is 3, there is an edge $(u,x)\ne e$ incident on u and an edge $(v,y)\ne e$ incident on v. Since the matching is maximum, we have $x=y$. Hence, G must be a triangle, which is a contradiction.

Now we assume that the maximum matching consists of two edges $e=(u,v)$ and $e'=(u',v')$. We show that $n\le 5$, which completes the proof since then $6\le n\le 9$ guarantees a matching of size at least 3. Assume for a contradiction that there are vertices x and y on which e and e' are not incident. Due to the maximality of the matching $\{e,e'\}$, edges incident on x and y can only be incident on u, v, u', and v'. Since x has degree at least 3, G contains, w.l.o.g., the edges (x,u) and (x,v). Since y has also degree 3, y must be adjacent to at least one of the vertices u and v, say u. But then (x,v,u,y) is an augmenting path for the matching, contradicting its optimality. □

Theorem 8. *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 $(V', V\setminus V')$ since the matching M is maximal and hence every edge in $E\setminus E'$ is incident to a vertex of V' and to a vertex not in V'; see Fig. 5a in Appendix B. Hence, we can compute a $16\alpha/3$-approximation to $\bar{G}$ using the algorithm presented in Theorem 3.

Consider the graph $G''=(V', E'\setminus M)$ and compute a maximum matching M'' in G''; see Fig. 5b. The edge set $M\cup M''$ is a set of vertex-disjoint paths and cycles and can therefore be completely realized [2]. 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. 5c. 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 $(\mathrm{OPT}-5/2n')/3$. Hence, by splitting OPT appropriately, we obtain

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

5 APX-Hardness

Theorem 9. *Weighted* MAX-CROWN *is APX-hard 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.*

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 [11]. 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 Fig. 6 in Appendix B. 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. 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.

References

[1] E. Ackerman. A note on 1-planar graphs. Available at http://sci.haifa.ac.il/~ackerman/ publications/1planar.pdf, Nov. 2013. 12

[2] 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. Viola, editor, *Proc. 11th Latin American Theoret. Inform. Symp. (LATIN'14)*, LNCS. Springer, 2014. To appear. Available at arxiv.org/abs/1311.4778. 2, 3, 5, 8, 9

[3] L. Barth, S. Kobourov, and S. Pupyrev. An experimental study of algorithms for semantics-preserving word cloud layout. Technical report, University of Arizona, 2013. Available at ftp://ftp.cs.arizona.edu/reports/2013/TR13-02.pdf. 1, 2

[4] A. L. Buchsbaum, E. R. Gansner, C. M. Procopiuc, and S. Venkatasubramanian. Rectangular layouts and contact graphs. *ACM Trans. Algorithms*, 4(1), 2008. 2

[5] C. Chekuri and S. Khanna. A PTAS for the multiple knapsack problem. In *Proc. 11th ACM-SIAM Symp. Discrete Algorithms (SODA'00)*, pages 213–222, 2000. 2

[6] W. Cui, Y. Wu, S. Liu, F. Wei, M. Zhou, and H. Qu. Context-preserving dynamic word cloud visualization. *IEEE Comput. Graphics Appl.*, 30(6):42–53, 2010. 2

[7] T. Dwyer, K. Marriott, and P. J. Stuckey. Fast node overlap removal. In *Proc. 13th Int. Symp. Graph Drawing (GD'05)*, volume 3843 of *LNCS*, pages 153–164. Springer, 2005. 2

[8] D. Eppstein, E. Mumford, B. Speckmann, and K. Verbeek. Area-universal and constrained rectangular layouts. *SIAM J. Comput.*, 41(3):537–564, 2012. 2

[9] 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. 1

[10] S. Felsner. Rectangle and square representations of planar graphs. In J. Pach, editor, *Thirty Essays on Geometric Graph Theory*, pages 213–248. Springer, 2013. 2

[11] L. Fleischer, M. X. Goemans, V. S. Mirrokni, and M. Sviridenko. Tight approximation algorithms for maximum separable assignment problems. *Math. Oper. Res.*, 36(3):416–431, 2011. 2, 3, 10

[12] G. N. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM J. Comput.*, 16(6):1004–1022, 1987. 8

[13] E. R. Gansner and Y. Hu. Efficient, proximity-preserving node overlap removal. *J. Graph Algortihms Appl.*, 14(1):53–74, 2010. 2

[14] S. L. Hakimi, J. Mitchem, and E. F. Schmeichel. Star arboricity of graphs. *Discrete Math.*, 149:93–98, 1996. 5, 6, 7

[15] K. Koh, B. Lee, B. H. Kim, and J. Seo. Maniwordle: Providing flexible control over Wordle. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1190–1197, 2010. 2

[16] H. Li and N. Abe. Word clustering and disambiguation based on co-occurrence data. In *Proc. 17th Int. Conf. Comput. Linguistics (COLING'98)*, volume 2, pages 749–755, Stroudsburg, PA, USA, 1998. ACL. 1

[17] C. Nash-Williams. Decomposition of finite graphs into forests. *J. London Math. Soc.*, 39:12, 1964. 6

[18] T. Nishizeki and I. Baybars. Lower bounds on the cardinality of the maximum matchings of planar graphs. *Discrete Math.*, 28(3):255–267, 1979. 9

[19] M. Nöllenburg, R. Prutkin, and I. Rutter. Edge-weighted contact representations of planar graphs. In *Proc. 20th Int. Symp. Graph Drawing (GD'12)*, volume 7704 of *LNCS*, pages 224–235. Springer, 2013. 2

[20] 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. 2

[21] E. Raisz. The rectangular statistical cartogram. *Geogr. Review*, 24(3):292–296, 1934. 2

[22] F. B. Viégas, M. Wattenberg, and J. Feinberg. Participatory visualization with Wordle. *IEEE Trans. Vis. Comput. Graphics*, 15(6):1137–1144, 2009. 1, 2

[23] 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. 1

[24] Y. Wu, T. Provan, F. Wei, S. Liu, and K.-L. Ma. Semantic-preserving word clouds by seam carving. *Comput. Graphics Forum*, 30(3):741–750, 2011. 2

Appendix

A Model with Point Contacts

In this model, adjacencies between boxes are allowed to be realized by a *point contact*, that is, by a contact of the boxes only in two corners.

Simple cases. As a first consequence, the PTAS for stars gets a bit simpler as we do not have to care about avoiding point contacts. The approximation factor does not change there as well as for all classes of planar or bounded degree graphs. Note that the APX-hardness proof also holds for this model without any modification.

Bipartite and general graphs. For these graph classes, we do, on the one hand, no longer need the post-processing that we applied in Theorems 3 and 5 (and implicitly also in Theorem 4). This post-processing cost us up to a quarter of the total profit. Hence, we can (for now) replace α by $3\alpha/4$, which improves the approximation factors for these cases.

On the other hand, a realized graph is now 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. This means that an optimal solution has at most $4n-8$ edges in the case of general graphs and at most $3n-6$ edges in the case of bipartite graphs. Furthermore, Ackerman [1] showed very recently that a 1-planar graph can be covered by a planar graph and a tree. Hence, we can cover a 1-planar graph with seven star forests and a bipartite 1-planar graph with six star forests (via a bipartite planar graph and a tree).

If our approximation algorithm for bipartite graphs uses this decomposition into six star forests, we easily get a 6α-approximation for this case. As a consequence, we get (as in Theorem 4) a randomized 12α-approximation for general graphs. Similarly, decomposing an optimum 1-planar solution into seven star forests (instead of five star forests for planar graphs), we get a deterministic 14α-approximation for general graphs.

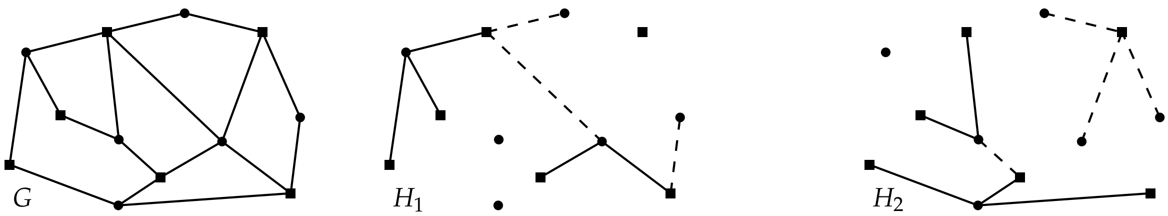
Unweighted general graphs. In order to modify the algorithm for the unweighted case, we use the new decomposition of bipartite graphs. It is easy to prove that any 1-planar graph with m edges and n vertices contains a matching of size at least $(m-3n)/3$: we planarize the graph (by removing at most n edges) and then apply Lemma 5. This results in a $(7+6\alpha)$-approximation for unweighted general graphs.

Table 2 shows the approximation factors for the model with point contacts; in the cases not mentioned in this table, the approximation ratio is the same as in the model without point contacts shown in Table 1.

graph class	weighted	unweighted
bipartite	6α	
general	14α (det.)	$7+6\alpha$
	12α (rand.)	

Table 2: New approximation factors for the version of MAX-CROWN where point contacts are allowed.

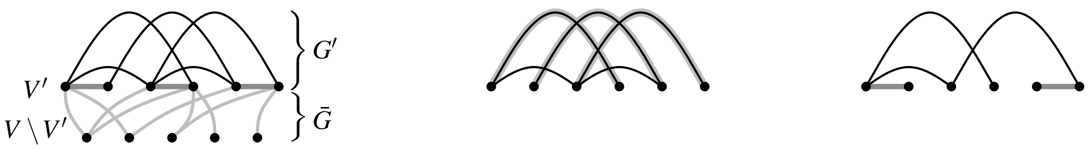
B Additional Figures



(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 3.



(a) G is covered by $\bar{G}$ (bipartite, gray) and G' with perfect 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 5: Partitioning the input graph and the optimum solution in the proof of Theorem 8.

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

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

Figure 6: The two possible configurations of a hyperedge $e = (x, y, z)$ in the APX-hardness proof (Theorem 9).