Optimization Methods and Software

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/goms20

Minimum point-overlap labelling*

Yuya Higashikawa , Keiko Imai , Takeharu Shiraga , Noriyoshi Sukegawa & Yusuke Yokosuka

OPTIMIZATION METHODS & SOFTWARE
https://doi.org/10.1080/10556788.2020.1833880

Minimum point-overlap labelling*

Yuya Higashikawa[a], Keiko Imai[b], Takeharu Shiraga[b], Noriyoshi Sukegawa[c] and Yusuke Yokosuka[d]

[a]School of Social Information Science, University of Hyogo, Hyogo, Japan; [b]Department of Information and System Engineering, Chuo University, Tokyo, Japan; [c]Faculty of Engineering, Tokyo University of Science, Tokyo, Japan; [d]Integrated Design Center, Mitsubishi Electric Corporation, Kanagawa, Japan

ABSTRACT
In an application of map labelling to air-traffic control, labels should be placed with as few overlaps as possible since labels include important information about airplanes. Motivated by this application, de Berg and Gerrits (Comput. Geom. 2012) proposed a problem of maximizing the number of free labels (i.e. labels not intersecting with any other label) and developed approximation algorithms for their problem under various label-placement models. In this paper, we propose an alternative problem of minimizing a degree of overlap at a point. Specifically, the objective of this problem is to minimize the maximum of $\lambda(p)$ over $p \in \mathbb{R}^2$, where $\lambda(p)$ is defined as the sum of weights of labels that overlap with a point p. We develop a 4-approximation algorithm by LP-rounding under the 4-position model. We also investigate the case when labels are rectangles with bounded height/length ratios.

ARTICLE HISTORY
Received 30 June 2019
Accepted 5 October 2020

KEYWORDS
Map labelling; air-traffic control; approximation algorithms

2010 MATHEMATICS SUBJECT CLASSIFICATIONS
68U05; 68W25

1. Introduction

In map labelling, we are given (i) a set of graphical features (such as regions, rivers and stations) represented as points, polylines or polygons, and (ii) a set of labels which include texts or symbols to indicate information about the graphical features. The goal of map labelling is to place labels so that the graphical features can be understood on a specified map. Map labelling has many applications including geographic information system (GIS), cartography and graph drawing.

This paper is motivated by an application of map labelling to air-traffic control where controllers use the results of map labelling to guide airplanes. In this application, labels indicate important information about airplanes (such as altitude and velocity), and hence it is desirable to place all the given labels so that there are as few overlaps as possible. This contrasts with the ordinary settings of map labelling where labels must not overlap each other (and to this end, some labels can be omitted or shrunken). The point is how to measure the degree of overlap; if some labels overlap, then controllers need to rearrange some

CONTACT Yusuke Yokosuka yusuke.yokosuka@gmail.com
*Preliminary versions of this work appeared in Proc. 10th International Conference on Algorithms and Complexity (CIAC 2017), pp. 334–344.

This article has been republished with minor changes. These changes do not impact the academic content of the article.

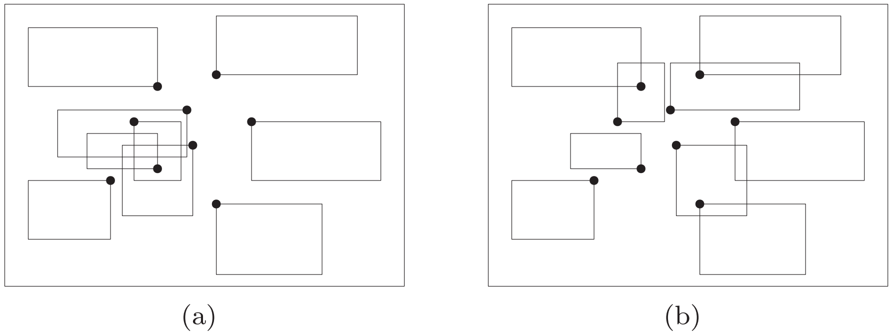

Figure 1. Optimal solutions of (a) FreeLabelMax and (b) PointOverlapMin for the same instance under the 4-position model.

labels by hand to read unreadable labels. This task is often time-consuming and hence can be an obstacle to controller's operation.

Motivated by this application, de Berg and Gerrits [6] proposed the *free-label maximization* problem (denoted as FreeLabelMax in this paper). In this problem, given a set of n sites and its n labels in the plane $\mathbb{R}^2$, we want to place the n labels so that the number of free labels (i.e. labels not intersecting any other label) is maximized. They developed constant-factor approximation algorithms and PTASs for this problem with unit-square labels under various label-placement models.

Figure 1(a) shows an instance of FreeLabelMax and its optimal solution under the 4-position model where each site must lie at one of the four corners of its label. The optimal value of FreeLabelMax for this instance is 5 since there are five free labels. On the other hand, we observe that the remaining four non-free labels overlap each other. One may need to spend a long time making these four non-free labels readable by hand when required.

1.1. Main contribution

In this paper, as an alternative to FreeLabelMax, we propose the following problem:

Definition 1.1 (PointOverlapMin): An instance I consists of a set P of n sites in the plane $\mathbb{R}^2$, denoted by $p_1, p_2, \ldots, p_n$. Each site p_i has an axis-parallel rectangular label ℓ_i and a positive weight w_i. Let $[n] = \{1, 2, \ldots, n\}$. For any $i \in [n]$, site p_i must lie at one of the four corners of label ℓ_i. Let Π be the set of all possible placements of $\{\ell_1, \ell_2, \ldots, \ell_n\}$, and for any placement $\pi \in \Pi$, let $\pi(i)$ denote the set of all points covered by ℓ_i (including its boundary). For a point $p \in \mathbb{R}^2$ under a given placement $\pi \in \Pi$, let $\lambda(p, \pi)$ denote the sum of weights of labels that overlap with p, i.e.

$$\lambda(p, \pi) \equiv \sum_{i \in [n]:\, p \in \pi(i)} w_i.$$

The problem is to find a placement $\pi \in \Pi$ that minimizes the maximum of $\lambda(p, \pi)$ over all $p \in \mathbb{R}^2$, i.e. $\max_{p \in \mathbb{R}^2} \lambda(p, \pi)$.

In what follows, we say that w_i is the weight of label ℓ_i, as well as the weight of site p_i for simplicity. When all labels are unweighted (i.e. $w_i = 1$ for any $i \in [n]$), the value of $\lambda(p, \pi)$

is exactly the number of labels that overlap with p for any placement $\pi \in \Pi$. We note that PointOverlapMin is NP-hard even when all labels are unit-squares; see e.g. [9,19] for problems that reduce to our problem.

Remark 1.1: Different from [6] where various label-placement models are discussed, we consider only the 4-position model (i.e. each site p_i must lie at one of the four corners of label ℓ_i) as stated in Definition 1.1. Extending our results for other label-placement models is left as future work.

Figure 1(b) shows an optimal solution of PointOverlapMin for the same instance used in Figure 1(a) where labels are unweighted. We see that in this optimal solution, all the nine labels are *almost* free (thanks to the novel objective function) meanwhile the number of free labels is only two. This toy example suggests that PointOverlapMin can be a promising alternative to FreeLabelMax.

We develop a 4-approximation algorithm for PointOverlapMin by LP-rounding. We also analyse the approximation ratio of a naive algorithm when the height/length ratios of labels are bounded.

1.2. Related work

Two typical problem settings of map labelling are the *label number maximization* and the *label size maximization*, see e.g. [20]. The former is to find a placement of a subset of labels with maximum cardinality, where the label sizes are fixed. The latter is to find a placement of all the labels so that their sizes are maximized under a global scale factor. In contrast, in FreeLabelMax and PointOverlapMin, all the labels need to be placed without changing the label sizes.

There are several models of how to place the labels. The 4-position model is one of the standards and is a special case of the *fixed-position model* [9]. In a fixed-position model, a finite number of label candidates are given for each site. Another important model is the *slider model* [19] which generalizes the fixed-position model. In a slider model, each label can slide as long as its specified sides touch the corresponding site. Extending our results for other label-placement models is left as future work.

Map labelling is known to be computationally hard in most problem settings. For this reason, as in this study, approximation algorithms have been studied extensively so far; see e.g. [1,9,14,19]. Most of these studies treat unweighted unit-square or unit-height rectangular labels. In contrast, our 4-approximation algorithm is applicable to any weighted rectangular labels. We note that label weights are a natural requirement in practice (e.g. labels of cities can have priority over those of towns). For studies that treat the weighted case, we refer to [8,17].

Our algorithm makes use of an integer programming formulation. We note that the integer programming is a common approach in map labelling, see, e.g. [13,15,23]. We also note that as in this study, LP-rounding approaches have been discussed so far, see e.g. [4,5].

Finally, we note that the *dynamic* problem has attracted attention in the past decade due to its increasing importance in several applications such as personal mapping systems. We refer to [2,10–12,16,21,22] for problems with static sites and a dynamic map,

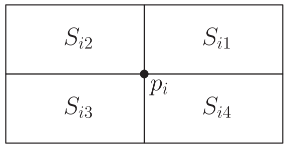

Figure 2. Label candidates.

and Buchin and Gerrits [3] for their hardness results. De Berg and Gerrits [7], on the other hand, discussed a trade-off between label speed and label overlap.

2. An approximation algorithm for arbitrary rectangular labels

In this section, we first formulate PointOverlapMin as an integer programming (IP) problem. We then describe a 4-approximation algorithm which solves a linear programming (LP) relaxation problem of this IP and rounds the resulting optimal solution in a natural way.

2.1. IP formulation

For each site p_i, let S_{ij} denote jth position on which label ℓ_i can be placed, where the four positions, upper right, upper left, lower left and lower right are indexed as 1, 2, 3 and 4, respectively; see Figure 2. Let $J = \{1, 2, 3, 4\}$. Let x_{ij} be a 0–1 variable that takes one if label ℓ_i is placed on S_{ij}, and zero otherwise.

To make the formulation polynomial size, we introduce a *cell* defined as follows. Draw horizontal lines through the top and bottom sides of each label candidate. These lines partition the plane into horizontal strips, which we call *slabs*; see Figure 3. Then, for each slab, draw vertical lines through left and right sides of each label candidate intersecting the slab. A cell is then defined as a region between consecutive vertical lines. The set of cells is denoted by $\mathcal{C}$. Note that $|\mathcal{C}| = O(n^2)$ since the number of slabs is $O(n)$ and the number of cells in a slab is $O(n)$.

Using these notations, PointOverlapMin can be formulated as an IP as follows:

$$
\begin{aligned}
\text{(IP)}\quad & \text{minimize} \quad \lambda \\
& \text{subject to} \quad \sum_{i\in[n],\, j\in J} \left\{ w_i x_{ij} \mid S_{ij} \cap C \neq \emptyset \right\} \le \lambda \quad (C \in \mathcal{C}), \\
& \qquad\qquad\quad \sum_{j\in J} x_{ij} = 1 \quad (i \in [n]), \\
& \qquad\qquad\quad x_{ij} \in \{0, 1\} \quad (i \in [n],\ j \in J).
\end{aligned}
$$

Recall that the objective function of PointOverlapMin is the maximum of $\lambda(p, \pi)$ over all $p \in \mathbb{R}^2$, and $\lambda(p, \pi)$ is defined as the sum of weights of labels that overlap with p under placement π. Thus, by the definition of the cells, $\lambda(p, \pi)$ remains unchanged for $p \in C$ for any C, implying that IP correctly formulates our problem. We note that IP resembles the one employed in [4,5] for developing approximation algorithms for the *maximum independent*

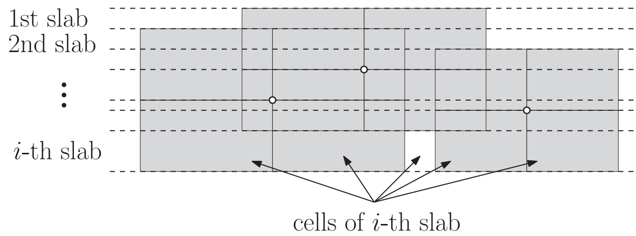

Figure 3. Slabs and cells.

set of rectangles problem. Observe that IP has $O(n)$ variables and $O(n^2)$ constraints. We also consider the following LP relaxation problem of IP:

$$\begin{aligned}
\text{(LP)} \quad & \text{minimize} \quad \lambda \\
& \text{subject to} \quad \sum_{i\in[n],\, j\in J} \left\{ w_i x_{ij} \mid S_{ij} \cap C \neq \emptyset \right\} \leq \lambda \quad (C \in \mathcal{C}), \\
& \qquad \sum_{j\in J} x_{ij} = 1 \quad (i \in [n]), \\
& \qquad x_{ij} \geq 0 \quad (i \in [n],\ j \in J).
\end{aligned}$$

2.2. A 4-approximation algorithm

Let $\mathcal{I}$ be the set of instances of PointOverlapMin for the 4-position model. For a given instance $I \in \mathcal{I}$ and an algorithm ALG, let ALG(I) denote the objective value of the solution obtained by ALG. Similarly, let OPT(I) denote the optimal value of this instance. Now, let us consider the following algorithm:

LP-Rounding (LPR)
Step 1. Find an optimal solution $\bar{x}$ to LP.
Step 2. For each site p_i, pick one variable x_{ij} such that $\bar{x}_{ij} \geq \frac{1}{4}$ and set it to 1. All other variables are set to 0. Output the 0–1 vector x constructed as above.

The following is readily obtained.

Theorem 2.1: *For any instance $I \in \mathcal{I}$, we have* $\mathsf{LPR}(I) \leq 4\,\mathsf{OPT}(I)$.

Proof: Any output of the LPR algorithm is a feasible solution of IP because in Step 2 only one label is placed for each site. We now estimate the objective value $\mathsf{LPR}(I)$ of the output x of the LPR algorithm. If we denote by $\mathsf{OPT}_{\mathsf{LP}}(I)$ the optimal value of LP, then $\mathsf{LPR}(I)$ is at most $4\,\mathsf{OPT}_{\mathsf{LP}}(I)$ because x_{ij} is at least $\frac{1}{4}$ for each site p_i. As $\mathsf{OPT}_{\mathsf{LP}}(I) \leq \mathsf{OPT}(I)$, we have that $\mathsf{LPR}(I) \leq 4\,\mathsf{OPT}(I)$. ∎

Theorem 2.1 implies that the LPR algorithm gives a 4-approximation for PointOverlapMin under the 4-position model.

The following shows that the integrality gap of LP is 4, which matches the approximation ratio of the LPR algorithm.

Remark 2.1: Let I be an instance consisting of only one site p_1 with weight 1. Clearly, $\mathsf{OPT}(I) = 1$ while $\mathsf{OPT}_{\mathsf{LP}}(I) = \frac{1}{4}$ as one can set $x_{11}, x_{12}, x_{13}, x_{14} = \frac{1}{4}$ in LP. Hence, $4\,\mathsf{OPT}_{\mathsf{LP}}(I) = \mathsf{OPT}(I)$.

The time complexity of the LPR algorithm can be estimated as follows. The construction of $\mathcal{C}$ takes $O(n^2)$ time. One can solve LP in $O((N+M)^{1.5}NL)$ time where N and M are the number of variables and constraints, respectively, and L is a parameter defined based on the input length [18]. As for LP, $N = O(n)$ and $M = O(n^2)$. Thus Step 1 takes $O(n^4L)$ time. As Step 2 takes $O(n)$ time, the LPR algorithm runs in $O(n^4L)$ time. This time complexity is weakly polynomial for the weighted case and strongly polynomial for the unweighted case since, in this case, $L = O(n \log n)$.

3. An analysis of a naive algorithm for labels with bounded height/length ratios

In this section, we consider the case when labels are rectangles with bounded height/length ratios. For a label ℓ_i, let h_i and l_i denote its height and length, respectively. For a given set of n labels, let

$$h_{\min} = \min_{i \in [n]} h_i, \quad h_{\max} = \max_{i \in [n]} h_i, \quad l_{\min} = \min_{i \in [n]} l_i, \quad l_{\max} = \max_{i \in [n]} l_i.$$

We call a set of labels (α, β)*-rectangular labels* if $h_{\max} \le \alpha h_{\min}$ and $l_{\max} \le \beta l_{\min}$ hold. By the definitions, α and β are both greater than or equal to 1. Note that the case of unit-square labels is a special one of (α, β)-rectangular labels with $\alpha = \beta = 1$. In this case, we show that a naive linear time algorithm achieves a good approximation ratio. We call this algorithm the PUL algorithm (where PUL stands for 'place upper left '). Namely, the PUL algorithm places the label of each site at the upper left position. We prove the following theorem.

Theorem 3.1: *The* PUL *algorithm is an* $(\lceil\alpha\rceil + 1)(\lceil\beta\rceil + 1)$*-approximation algorithm for* PointOverlapMin *with* (α, β)*-rectangular labels.*

Proof: Fix an arbitrary point $p \in \mathbb{R}^2$. Let $G(p)$ denote a set of an infinite number of horizontal parallel lines and vertical parallel lines satisfying the following conditions:

(1) The distance between any two adjacent horizontal lines is $h_{\min}$.
(2) The distance between any two adjacent vertical lines is $l_{\min}$.
(3) There exist a horizontal line and a vertical line intersecting at point p.

Figure 4 shows an example of $G(p)$. We say that a point (x, y) is a *grid point* of $G(p)$ if a horizontal line and a vertical line of $G(p)$ intersect at (x, y). We also let

$$H = \lceil\alpha\rceil h_{\min}, \quad L = \lceil\beta\rceil l_{\min}.$$

Note that $h_{\max} \le H$ and $l_{\max} \le L$ by the definitions of α and β.

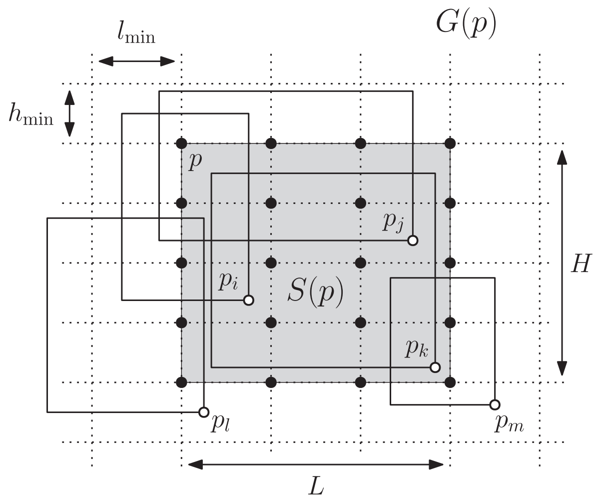

Figure 4. $G(p)$ (dotted lines), $S(p)$ (shaded area) and $Q(p)$ (solid points). In the figure, labels ℓ_i and ℓ_j overlap with p and label ℓ_k does not while all the corresponding three sites p_i, p_j and p_k are in $S(p)$. The labels whose sites are outside $S(p)$, like ℓ_l and ℓ_m in the figure, cannot overlap with p.

Let $S(p)$ denote the rectangle (including the boundary and the interior points) with the four corners $p+(L,0), p, p+(0,-H)$ and $p+(L,-H)$, see Figure 4. Let $Q(p)$ denote a set of all the grid points of $G(p)$ in $S(p)$. Clearly,

$$|Q(p)| = (\lceil\alpha\rceil + 1)(\lceil\beta\rceil + 1). \tag{1}$$

Observe that if some label ℓ_i overlaps with p under $\pi_{\mathsf{PUL}}(I)$, then the corresponding site p_i must be contained in $S(p)$ since $h_{\max} \le H$ and $l_{\max} \le L$ as shown in Figure 4. Therefore it holds

$$\lambda(p, \pi_{\mathsf{PUL}}(I)) \le \sum_{i\in[n]} \{w_i \mid p_i \in S(p)\}. \tag{2}$$

Now, let us consider the optimal placement $\pi_{\mathsf{PUL}}(I)$. We clearly have

$$\lambda(q, \pi_{\mathsf{OPT}}(I)) \le \mathsf{OPT}(I) \quad (\forall q \in Q(p)). \tag{3}$$

On the other hand, under $\pi_{\mathsf{OPT}}(I)$, any label ℓ_i with $p_i \in S(p)$ must overlap with at least one of the points of $Q(p)$, which implies

$$\sum_{i\in[n]} \{w_i \mid p_i \in S(p)\} \le \sum_{q\in Q(p)} \lambda(q, \pi_{\mathsf{OPT}}(I)). \tag{4}$$

Combining (1), (2), (3) and (4), we obtain

$$\lambda(p, \pi_{\mathsf{PUL}}(I)) \le (\lceil\alpha\rceil + 1)(\lceil\beta\rceil + 1) \cdot \mathsf{OPT}(I),$$

which implies $\mathsf{PUL}(I) \le (\lceil\alpha\rceil + 1)(\lceil\beta\rceil + 1)\cdot \mathsf{OPT}(I)$. This concludes the proof. ■

For the case with unit-square labels (i.e. $\alpha = \beta = 1$), we can obtain the following immediately from Theorem 3.1.

Corollary 3.2: *The* PUL *algorithm is a 4-approximation algorithm for* PointOverlapMin *with unit-square labels.*

4. Conclusion

We proposed a map labelling problem called the point-overlap minimization problem, denoted as PointOverlapMin, with a view to its application to air-traffic control. We believe that our problem can be a promising alternative to the free-label maximization problem introduced by de Berg and Gerrits [6].

We developed a 4-approximation algorithm for PointOverlapMin. This algorithm is based on LP-rounding and runs in weakly polynomial and strongly polynomial time for the weighted and unweighted cases, respectively. We also presented an analysis of a naive algorithm when all labels are rectangles with bounded height/length ratio.

Extending our results to other label-placement models (such as the slider model), and developing combinatorial algorithms (e.g, by using the primal–dual method) that are as good as the 4-approximation algorithm are left as open problems.

Acknowledgments

The authors would like to thank the two anonymous reviewers for their insightful and detailed comments that have helped improve the paper. In particular, the authors are grateful to one reviewer who simplified and improved our previous analysis to establish the special case of Theorem 3.1, i.e. Corollary 3.2.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by Japan Society for the Promotion of Science [16K00024,19H00808,20K 19746].

Notes on contributors

Yuya Higashikawa received the B. Eng, M. Eng. and Dr. Eng. degrees in Engineering from Kyoto University, in 2008, 2010 and 2014, respectively. From 2010 to 2011 he worked at a television production company in Japan. From 2014 to 2015 he was a JSPS Research Fellow (PD) at Kyoto University. From 2015 to 2018 he was an assistant professor at Chuo University. In 2018 he moved to University of Hyogo, where he is currently an associate professor at the School of Social Information Science.

Keiko Imai received the B.S., M.S. and Dr. S. degrees in Mathematics from Tsuda College in 1980, 1982 and 1991, respectively. Since 1992 she has been with Chuo University, where she is currently as a full professor in the Department of Information and System Engineering.

Takeharu Shiraga obtained the Doctor of Engineering from Kyushu University in 2017. He is a research associate at Department of Information and System Engineering, Chuo University.

Noriyoshi Sukegawa obtained the Doctor of Engineering from Tokyo Institute of Technology in 2015 and is an assistant professor at Department of Information and Computer Technology, Tokyo University of Science.

Yusuke Yokosuka received the B.E, M.E. and Ph.D. degrees in Engineering from Chuo University, in 2004, 2006 and 2018, respectively. and is a senior researcher at Mitsubishi Electric Corporation.

References

[1] P.K. Agarwal, M.J. van Kreveld, and S. Suri, *Label placement by maximum independent set in rectangles*, Comput. Geom. 11 (1998), pp. 209–218.
[2] K. Been, M. Nöllenburg, S.H. Poon, and A. Wolff, *Optimizing active ranges for consistent dynamic map labeling*, Comput. Geom. 43 (2010), pp. 312–328.
[3] K. Buchin and D.H.P. Gerrits, *Dynamic point labeling is strongly PSPACE-complete*, Int. J. Comput. Geometry Appl. 24 (2014), pp. 373–395.
[4] P. Chalermsook and J. Chuzhoy, *Maximum independent set of rectangles*, Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA. 2009, pp. 892–901.
[5] T.M. Chan and S. Har-Peled, *Approximation algorithms for maximum independent set of pseudo-disks*, Discr. Comput. Geom. 48 (2012), pp. 373–392.
[6] M. de Berg and D.H.P. Gerrits, *Approximation algorithms for free-label maximization*, Comput. Geom. 45 (2012), pp. 153–168.
[7] M. de Berg and D.H.P. Gerrits, *Labeling moving points with a trade-off between label speed and label overlap*, Algorithms–ESA 2013–21st Annual European Symposium, Sophia Antipolis, France. Proceedings, 2013, pp. 373–384.
[8] T. Erlebach, T. Hagerup, K. Jansen, M. Minzlaff, and A. Wolff, *Trimming of graphs, with application to point labeling*, Theory Comput. Syst. 47 (2010), pp. 613–636.
[9] M. Formann and F. Wagner, *A packing problem with applications to lettering of maps*, Proceedings of the Seventh Annual Symposium on Computational Geometry, North Conway, NH, USA, 1991, pp. 281–288.
[10] A. Gemsa, M. Nöllenburg, and I. Rutter, *Sliding labels for dynamic point labeling*, Proceedings of the 23rd Annual Canadian Conference on Computational Geometry, Toronto, Ontario, Canada, 2011.
[11] A. Gemsa, B. Niedermann, and M. Nöllenburg, *Trajectory-based dynamic map labeling*, Algorithms and Computation--24th International Symposium, ISAAC 2013, Hong Kong, China, Proceedings, 2013, pp. 413–423.
[12] A. Gemsa, M. Nöllenburg, and I. Rutter, *Consistent labeling of rotating maps*, J. Comput. Geom. 7 (2016), pp. 308–331.
[13] J.H. Haunert and A. Wolff, *Beyond maximum independent set: an extended integer programming formulation for point labeling*, ISPRS Int. J. Geo Inf 6 (2017). article 342, 20 pages.
[14] J.W. Jung and K.Y. Chwa, *Labeling points with given rectangles*, Inf. Process. Lett. 89 (2004), pp. 115–121.
[15] G.W. Klau and P. Mutzel, *Optimal labeling of point features in rectangular labeling models*, Math. Program. (Ser. B) 94 (2003), pp. 435–458.
[16] C. Liao, C. Liang, and S. Poon, *Approximation algorithms on consistent dynamic map labeling*, Theor. Comput. Sci. 640 (2016), pp. 84–93.
[17] S. Poon, C. Shin, T. Strijk, T. Uno, and A. Wolff, *Labeling points with weights*, Algorithmica 38 (2004), pp. 341–362.
[18] P.M. Vaidya, *Speeding-up linear programming using fast matrix multiplication (extended abstract)*, 30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA. 1989, pp. 332–337.
[19] M.J. van Kreveld, T. Strijk, and A. Wolff, *Point labeling with sliding labels*, Comput. Geom. 13 (1999), pp. 21–47.
[20] A. Wolff and T. Strijk, *The map-labeling bibliography* (2009). Available at http://i11www.iti.uni-karlsruhe.de/map-labeling/bibliography/.
[21] Y. Yokosuka and K. Imai, *Polynomial time algorithms for label size maximization on rotating maps*, J. Inform. Process. 25 (2017), pp. 572–579.

[22] X. Zhang, S. Poon, M. Li, and V.C.S. Lee, *On maxmin active range problem for weighted consistent dynamic map labeling*, Proceedings of The Seventh International Conference on Advanced Geographic Information Systems, Applications, and Services, GEOProcessing 2015, Lisbon, Portugal, 2015, pp. 32–37.
[23] S. Zoraster, *The solution of large 0-1 integer programming problems encountered in automated cartography*, Oper. Res. 38 (1990), pp. 752–759.