Tracking a Set of Moving Objects with Minimal Peak Power

Sándor P. Fekete
Department of Computer Science, TU Braunschweig, Germany
L3S Research Center, Hannover, Germany

Malte Hoffmann
Department of Computer Science, TU Braunschweig, Germany

Chek-Manh Loi
Department of Computer Science, TU Braunschweig, Germany

Michael Perk
Department of Computer Science, TU Braunschweig, Germany

Abstract

A common sensing problem is to use a set of stationary tracking locations to monitor a collection of moving devices. Given n objects that need to be tracked, each following its own trajectory, and m stationary traffic control stations, each with a sensing region that can be changed over time; how should we adjust the individual sensor ranges in order to optimize energy consumption? We illustrate how to combine geometric insights with mathematical optimization to find optimal solutions for the min max variant of the problem, which aims at minimizing peak power consumption. Instances with 500 moving objects and 25 stations can be solved in the order of seconds for scenarios that take minutes to play out in the real world, demonstrating real-time capability of our methods.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Set cover, kinetic problems, geometric optimization, exact optimization

Digital Object Identifier 10.4230/LIPIcs.SoCG.2026.102

Category Media Exposition

Related Version *Full Version*: https://arxiv.org/abs/2603.05286 [14, 15]
Video Link: https://youtu.be/YoLfsEbJO4Q

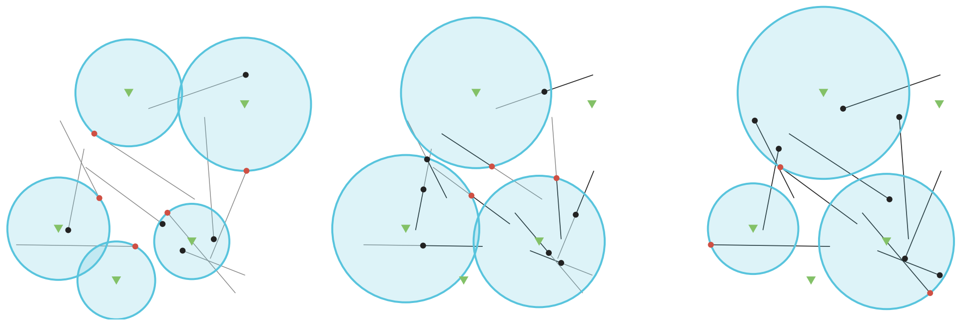
Supplementary Material *Software (and Data)*: https://doi.org/10.5281/zenodo.18955730 [9]

Funding This work was supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) as part of project Computational Geometry: Solving Hard Optimization Problems (CG:SHOP) – 444569951.

1 Introduction

Keeping track of a collection of moving objects is a fundamental problem with a long history in Computational Geometry. In air traffic control, stationary centers monitor planes with powerful tracking devices to coordinate overall motion. With the growing ubiquity of drones, air traffic control needs further development [4, 16], such as the use of less powerful tracking stations: How should we adjust sensing radii over time to minimize power consumption?

We illustrate exact and heuristic methods for the min max variant of the Kinetic Disk Cover problem (KDC): Given a set of m stationary sensors at positions $\mathcal{Y} \subset \mathbb{R}^2$, as well as a set $\mathcal{P}$ of n moving objects, each moving along a linear trajectory $p_i(t)$ over the time interval $t \in [0, 1]$. How should we adjust the sensor range $r_i(t)$ to each center $y_i \in \mathcal{Y}$ at each time t, such that all objects are always within range of some sensor, and the maximum sum of areas $\max_{t\in[0,1]} \sum_{i=1}^{m} \pi r_i(t)^2$ of all disks is minimized?



(a) $t = 0$ **(b)** $t = 0.5$ **(c)** $t = 1$

Figure 1 A solution for an instance of the Kinetic Disk Covering Problem for $n = 10$ and $m = 5$. Stations are shown as green triangles, points move along the line segments.

Related work. The stationary Disk Cover problem (DC) [1] and its generalization, the non-uniform minimum-cost multi-cover (MCMC) problem [12, 13] are known to be NP-hard. Recently, there has been some work on optimally multi-covering fixed points with disks of varying sizes [11]. Other work examines multi-covering points with the minimum number of unit disks, with centers at arbitrary locations [10]. Research on fundamental concepts for kinetic data structures [2] enabled algorithms for k-center and k-means problem for both kinetic stations and objects [3].

2 Algorithmic approach

We solve the kinetic problem with a primal–dual, iterative strategy. By combining exact stationary solutions with good heuristics for obtaining overall kinetic solutions, we generate a sequence of tighter and tighter upper and lower bounds and therefore provably near-optimal or even optimal solutions.

Our approach starts from a feasible, stationary solution at time $t = 0$. We can extend any solution to the DC problem at time t to a solution for the KDC problem over time by maintaining the same assignment of objects to stations and altering the supporting point of each station as needed to ensure coverage. This produces a feasible solution for all $t \in [0, 1]$, yielding an upper bound for the optimal objective value. The upper bound can be further improved by considering the times at which it is advantageous to switch objects between covering stations, so-called *handover points.* In the full version of our paper [14], we establish the following; this can be generalized to three or more stations.

▶ **Lemma 1.** *In $\mathcal{O}(m^2n^3)$ time, all $\mathcal{O}(m^2n^3)$ possible handovers of objects between two stations (when extending a stationary solution to $t \in [0, 1]$) can be computed.*

On the other hand, we can consider the time t' at which the peak total area is attained; an optimal solution for the static DC problem at t', which can be computed with the help of an Integer Programming (IP) solver, provides a lower bound for the kinetic problem.

$$
\begin{aligned}
\text{minimize} \quad & \pi \cdot \sum_{d_i \in C} r_i^2 x_i \\
\text{subject to} \quad & \sum_{\substack{d_i \in C \\ p_j \in d_i}} x_i \geq 1, \qquad \forall p_j \in \mathcal{P} \\
& x_i \in \{0, 1\}, \quad \forall d_i \in C
\end{aligned}
$$

In this formulation, C is the set of all possible disks that can be formed by a station and an object, r_i is the radius of the disk d_i, and x_i indicates whether d_i is selected. A solution for the stationary problem can again be extended to an overall solution for the kinetic problem. Combining it with the previous kinetic solution by extracting the lower envelope of both solutions yields an improved feasible solution, and thus a better upper bound, see Figure 2.

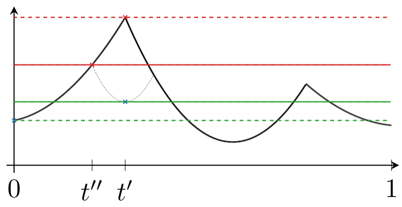


Figure 2 Improvement of the static solution at time t'. Green and red lines denote lower and upper bounds on the optimal solution.

3 Evaluation

We carried out extensive experiments for solving a wide range of benchmark instances; see our full paper [14, 15] for a detailed description.

We considered the following benchmark sets.

`fix_n` Fix number of objects to $n = 500$. For $m \in \{5, 10, \ldots, 50\}$ we generate 10 instances.

`fix_m` Fix number of sensors to $m = 25$. For $n \in \{50, 100, \ldots, 500\}$ we generate 10 instances.

`pub` Instances from well-known publicly available benchmarks used in [8], such as point sets from the CG:SHOP challenges [5, 6] and [7, 18, 19]. We scaled point sets to a fixed size and sampled $m = 25$ centers and point pairs (for trajectories) at random.

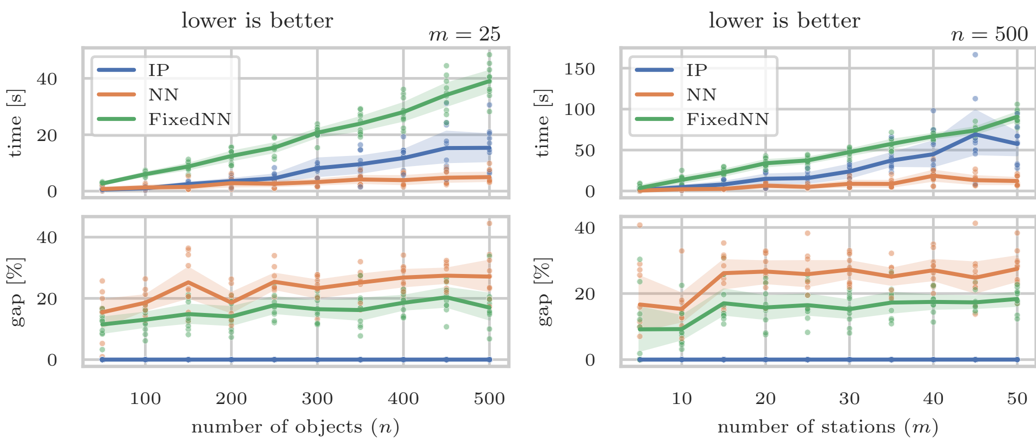


Figure 3 Performance of `IP`, `NN` and `FixedNN` on instances from the `fix_m` and `fix_n` datasets. While `IP` is significantly slower than `NN`, it produces optimal solutions for all instances. `FixedNN` is slightly better in terms of solution quality than `NN` but significantly slower.

Influence of m and n. We evaluate the influence of the number of objects n and the number of stations m on the performance of the algorithm. We execute the algorithm from Section 2 with both an IP solver and a nearest neighbor heuristic to solve the DC problem in each iteration. We denote the IP-based algorithm as `IP` and the nearest neighbor based algorithm as `NN`. Additionally, we implemented another heuristic that divides the time $[0, 1]$ into $k = 10$ evenly spaced intervals and computes the nearest neighbor solution at each border. The algorithm `FixedNN` then extends all solutions and reports the lower envelope. Figure 3 shows the performance of the different algorithms. Overall, `IP` is able to solve all instances to optimality in a matter of seconds, demonstrating real-time capability for scenarios that take minutes to play out in the real world.

Instances from Literature. We further assess the algorithm's performance on the `pub` instance set, as shown in Figure 4 and illustrated by Figure 5. The presence of degeneracies (such as equidistant points and collinearities), originating from real-world data, led to various numerical challenges in our implementation. To address these issues, we developed an alternative version of `IP` utilizing exact number types for computations; see the full paper [14, 15] for details. Our results show that 290 out of 302 instances with up to 700 objects could be solved to provable optimality within 600 s of computation time.

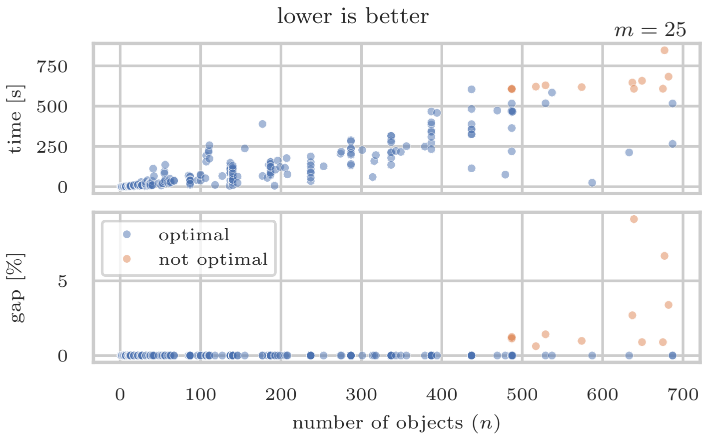
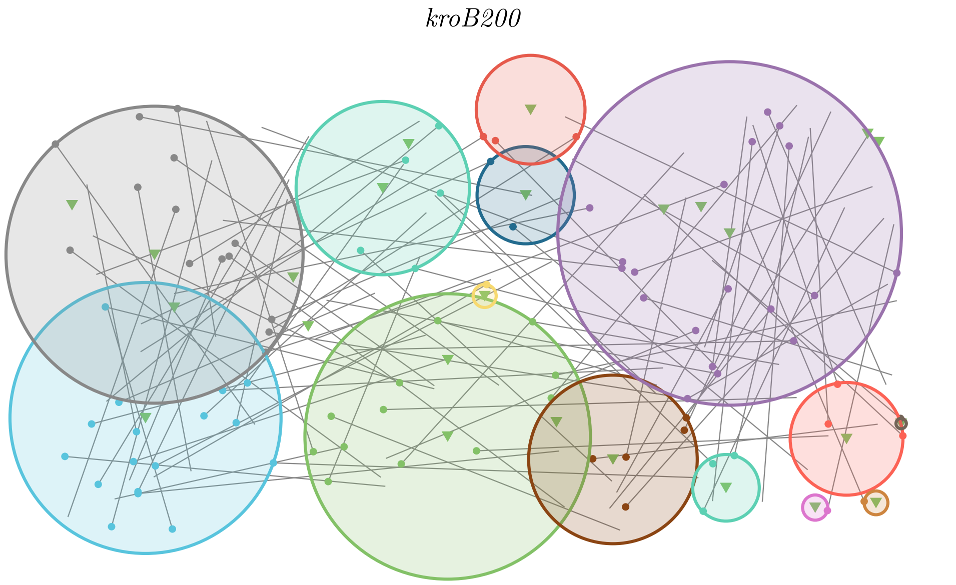


Figure 4 Performance of the exact version of `IP` on the `pub` dataset. Despite the variety and degeneracies, almost all instances can be solved to provable optimality.

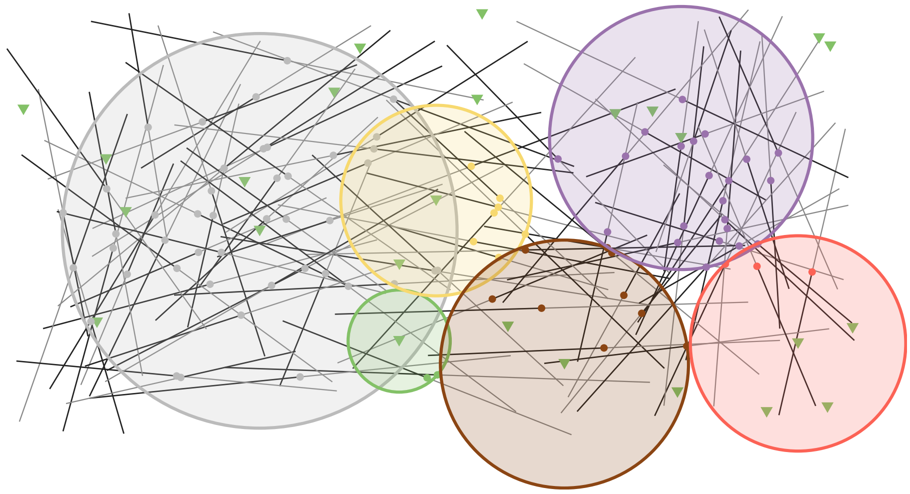
4 The video

We start by introducing and motivating the KDC problem before describing the primal-dual approach, illustrating the computation of upper and lower bounds as well as the overall iterative method. We proceed by showing some of the involved technical components and evaluations, concluding with a benchmark instance with 87 objects and 25 stations derived from a point set of the well-known TSPLIB benchmark library, as shown in Figure 5.

Overall video production was based on an integrated tool chain: Visual elements were produced with the free, open-source software Manim [20] for creating mathematical animations, which also supports conversion from scripted text to synchronized voiceover and animation. Voiceover was generated with the help of Microsoft Azure Speech [17].



(a) $t = 0$



(b) $t = 0.5$

(c) $t = 1$

Figure 5 A solution of an instance from `pub` instance set based on *kroB200* TSPLIB [18].

References

1 Helmut Alt, Esther M. Arkin, Hervé Brönnimann, Jeff Erickson, Sándor P. Fekete, Christian Knauer, Jonathan Lenchner, Joseph S. B. Mitchell, and Kim Whittlesey. Minimum-cost coverage of point sets by disks. In *Symposium on Computational Geometry (SoCG)*, pages 449–458, 2006. doi:10.1145/1137856.1137922.

2 Julien Basch, Leonidas J. Guibas, and John Hershberger. Data structures for mobile data. *J. Algorithms*, 31(1):1–28, 1999. doi:10.1006/JAGM.1998.0988.

3 Sergei Bespamyatnikh, Binay K. Bhattacharya, David G. Kirkpatrick, and Michael Segal. Mobile facility location. In *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, pages 46–53, 2000. doi:10.1145/345848.345858.

4 Adam P. Cohen, Susan A. Shaheen, and Emily M. Farrar. Urban air mobility: History, ecosystem, market potential, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(9):6074–6087, 2021. doi:10.1109/TITS.2021.3082767.

5 Erik D Demaine, Sándor P Fekete, Phillip Keldenich, Dominik Krupke, and Joseph S. B. Mitchell. Computing convex partitions for point sets in the plane: The CG:SHOP Challenge 2020. *arXiv preprint arXiv:2004.04207*, 2020. arXiv:2004.04207.

6 Erik D Demaine, Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Joseph S. B. Mitchell. Area-optimal simple polygonalizations: The CG Challenge 2019. *Journal of Experimental Algorithmics (JEA)*, 27(2):1–12, 2022. doi:10.1145/3504000.

7 Günther Eder, Martin Held, Steinþór Jasonarson, Philipp Mayer, and Peter Palfrader. Salzburg database of polygonal data: Polygons and their generators. *Data in Brief*, 31:105984, 2020. doi:10.1016/j.dib.2020.105984.

8 Sándor P. Fekete, Phillip Keldenich, and Michael Perk. Exact Algorithms for Minimum Dilation Triangulation. In *Symposium on Computational Geometry (SoCG)*, pages 48:1–48:18, 2025. doi:10.4230/LIPIcs.SoCG.2025.48.

9 Sándor Fekete, Malte Hoffmann, Chek-Manh Loi, and Michael Perk. Drone air traffic control: Tracking a set of moving objects with minimal power, 2026. doi:10.5281/zenodo.18955730.

10 Xuening Gao, Longkun Guo, and Kewen Liao. Fast approximation algorithms for multiple coverage with unit disks. In *Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 185–193, 2022. doi:10.1109/WoWMoM54355.2022.00037.

11 Mariem Guitouni, Chek-Manh Loi, Sándor P. Fekete, Michael Perk, and Aaron T. Becker. Multi-covering a point set by m disks with minimum total area. In *ICRA*, pages 3000–3006. IEEE, 2025. doi:10.1109/ICRA55743.2025.11127835.

12 Ziyun Huang, Qilong Feng, Jianxin Wang, and Jinhui Xu. PTAS for minimum cost multi-covering with disks. In *Symposium on Discrete Algorithms (SODA)*, pages 840–859, 2021. doi:10.1137/1.9781611976465.53.

13 Ziyun Huang, Qilong Feng, Jianxin Wang, and Jinhui Xu. PTAS for minimum cost multicovering with disks. *SIAM Journal on Computing*, 53(4):1181–1215, 2024. doi:10.1137/22M1523352.

14 Chek-Manh Loi, Michael Perk, Malte Hoffmann, and Sándor P. Fekete. Drone air traffic control: Tracking a set of moving objects with minimal power. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. To appear.

15 Chek-Manh Loi, Michael Perk, Malte Hoffmann, and Sándor Fekete. Drone air traffic control: Tracking a set of moving objects with minimal power, 2026. doi:10.48550/arXiv.2603.05286.

16 Franco Mazzenga, Romeo Giuliano, and Alessandro Vizzarri. 5G-based synchronous network for air traffic monitoring in urban air mobility. *IEEE Access*, 12:188542–188559, 2024. doi:10.1109/ACCESS.2024.3513212.

17 Microsoft. Azure Speech in Foundry Tools, 2026. URL: https://azure.microsoft.com/en-us/products/ai-foundry/tools/speech.

18 G. Reinelt. TSPLIB–A Traveling Salesman Problem Library. *ORSA Journal of Computing*, 3(4):376–384, 1991. doi:10.1287/ijoc.3.4.376.

19 Andre Rohe. VLSI data set. URL: https://www.math.uwaterloo.ca/tsp/vlsi/index.html.

20 The Manim Community Developers. Manim – Mathematical Animation Framework, 2026. URL: https://www.manim.community/.