# Multi-Covering a Point Set by $m$ Disks with Minimum Total Area

**Aaron T. Becker**[1], **Sándor P. Fekete**[2], **Mariem Guitouni**[1], **Chek-Manh Loi**[2], **and Michael Perk**[2]

1 **University of Houston**
`atbecker@uh.edu,mguitoun@CougarNet.uh.edu`
2 **Department of Computer Science, TU Braunschweig**
`s.fekete@tu-bs.de, {loi, perk}@ibr.cs.tu-bs.de`

**Abstract**

In general multi-coverage, we are given a set of $n$ points $p_1, \ldots, p_n$ in the plane. The task is to choose $m$ locations $q_1, \ldots, q_m$ and assign a radius $r_j$ to each $q_j$, such that each $p_i$ is covered by disks centered at $\kappa(p_i) \geq 1$ different $q_j$ with corresponding radius $r_j$, such that the sum of disk areas is minimized. We provide fast heuristics and exact methods that compute provably optimal solutions, which we extend to the generalization in which disk centers are subject to separation constraints.

**Related Version** arXiv:2502.13773

## 1 Introduction

Covering a set of geometric locations is an important optimization problem that arises in different areas. As shown in Figure 1a, this includes scenarios from robotics, e.g., controlling a set of ground locations from a finite set of drones with downward communication links [15, 7], requiring a set of different altitudes that balance safe separation between drones with reliable communication to the ground. The latter requires sufficient signal strength, so communication areas (and thus energy consumption) depend quadratically on the altitude. For any location, observation with more than one drone is often needed to ensure sufficiently robust coverage. Similar problems exist in diverse application domains, including wireless sensor networks [1, 3, 4, 17, 5], facility placement [2, 6] and pesticide application [14, 8].

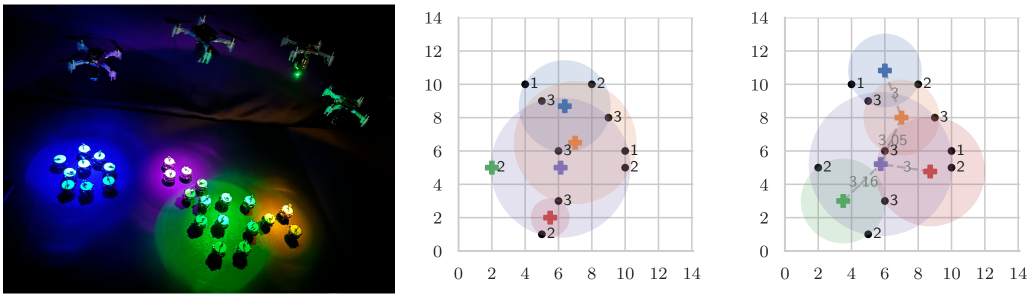
In the *general multi-coverage* (GMC) problem, we are given a point set $S$, $m$ sensors, and a coverage function $\kappa : S \to \mathbb{N}$; the goal is to assign a center $q_i$ and radius $r_i$ for each disk $i \in [1, \ldots, m]$ so that each $p_i \in S$ is covered by at least $\kappa(p_i)$ disks. The objective is to minimize the sum of disk areas $\pi \sum_{i=1}^{m} r_i^2$. An additional constraint arises by enforcing sufficient separation between coverage centers: For distance $\ell$, the *dispersive multi-coverage* problem (DGMC) asks for a GMC with $\|q_i - q_j\| \geq \ell$ for all $i \neq j$ and $i, j \in [1, \ldots, m]$. (See Figures 1b and 1c for examples of optimal solutions.)

## 2 Related Work

Alt et al. [2] studied the GMC with $\kappa(p) = 1$. A related, but simpler, problem explored in previous works uses a given set $C$ of disk centers; with given coverage multiplicities $\kappa(p_i)$, this is known as the non-uniform minimum-cost multi-cover (MCMC) problem. If $\forall p \in S, \kappa(p) = k$ it is referred to as the uniform MCMC.

Approximation algorithms with constant factors depending on $\kappa$ for uniform and non-uniform MCMC were given by Abu-Affash et al. [1] and BarYehuda and Rawitz [3], respectively. Bhowmick et al. [6] achieved constant approximation for non-uniform MCMC independent of $\kappa$. Huang et al. [12, 13] gave a PTAS for $\kappa(S) > 1$.

41st European Workshop on Computational Geometry, Liblice, Czech republic, April 9–11, 2025.
This is an extended abstract of a presentation given at EuroCG'25. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.




**(a)** Kilobot robots **(b)** GMC **(c)** DGMC

**Figure 1** (a) Ground-based Kilobot robots, commanded by overhead controllers via infrared communication [16]. Optimal (minimum total area) solutions with $n = 10$, $m = 5$, and $\kappa$ between 1 and 3. (b) Without separation constraints. (c) Enforcing a minimum distance of $\ell = 3$.

Also related is placing a minimum number of unit disks to multi-cover a point set of size $|S| = n$. Gao et al. [10] gave a 5-approximation with runtime $\mathcal{O}(n + \kappa_{\max})$, and a 4-approximation algorithm with runtime $\mathcal{O}(n^2)$. Filipov and Tomova studied coverage with the minimum number of unit disks [9], providing a stochastic algorithm with expected complexity $\mathcal{O}(n^2)$.

## 3 Solving GMC: Lower Bounds

We consider approaches for solving the GMC problem, implying lower bounds for the DGMC.

### 3.1 GMC heuristic

The heuristic starts with an initial $k$-means solution to partition the point set into $m$ clusters. These clusters are expanded to ensure each point $p$ is covered $\kappa(p)$ times and locally optimized to minimize the total area of the disks. See full version for a detailed description.

### 3.2 Integer Programming

To formulate the GMC as an IP, we need a discrete set of candidate sensor positions. We discuss computing a (preferably small) sufficient set $C$ of candidate disks in Section 3.2.1. Given $C$, we can formulate the integer program in Section 3.2.2.

#### 3.2.1 Computing the Candidate Set

Without separation constraints, there are three ways that a set of points $S' \subseteq S$ can be covered optimally (i.e., with minimum-area) by a disk.

a) For $S' = \{p_1\}$, a disk centered at $p_1$ with radius 0 is optimal.
b) For $S' = \{p_1, p_2\}$, there is a unique disk with radius $\frac{\|p_i - p_j\|}{2}$ centered at the midpoint of $\overline{p_1, p_2}$ that is the minimum-area disk that contains both points.
c) For $|S'| \geq 3$, any disk covering $S'$ can be shrunk until it has either (i) two points $p_1, p_2 \in S'$ on its boundary (see case b) or (ii) at least three points $p_1, p_2, p_3 \in S'$ on its boundary.

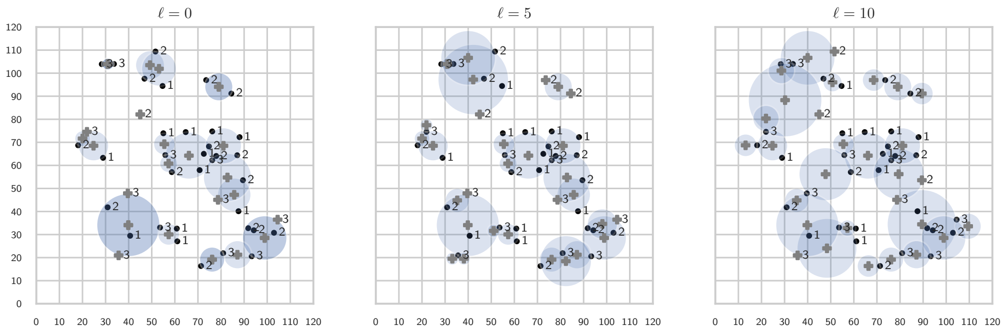



**Figure 2** Solutions from *uni_lg* with $n = 30$ and $m = 30$ and different separation constraints.

The above allows us to enumerate all necessary disks for the GMC: We add a disk with radius 0 and center $p_i$ for all points $p_i \in S$. Then for all pairs $p_i, p_j \in S$, we compute the disk centered between $p_i, p_j$. For all triples $p_i, p_j, p_k \in S$, we compute the unique disk that has $p_i, p_j$ and $p_k$ on its boundary. When the triangle between the points is obtuse, a disk in $C$ (defined by two of the points) already contains the third point and has a smaller area. Thus, we only add a disk defined by three points if we encounter an acute triangle.

In total, this yields $\mathcal{O}(n^3)$ possible positions. Using a $k$-$d$- or ball-tree one can find the set $S' \subseteq S$ of points that intersect a given disk in $\mathcal{O}(\sqrt{n} + |S'|)$ time. This yields a worst-case runtime of $\mathcal{O}(n^4)$ to enumerate all elements of $C$, but with better performance in practice.

### 3.2.2 GMC IP Formulation

For every disk $d_i$ in the candidate set $C$, we define integer variables $x_i$ that encode how often each disk is used in the solution. The constraints ensure that at most $m$ disks are placed and every point $p_j \in S$ is covered by at least $\kappa(p_j)$ disks.

$$\begin{aligned}
\text{minimize} \quad & \pi \cdot \sum_{d_i \in C} r_i^2 x_i \\
\text{subject to} \quad & \sum_{d_i \in C} x_i \leq m \\
& \sum_{\substack{d_i \in C \\ p_j \in d_i}} x_i \geq \kappa(p_j), && \forall p_j \in S \\
& x_i \in \{0, \dots, m\}, && \forall d_i \in C
\end{aligned}$$

## 4 Upper Bounds: Enforcing separation constraints

While the DGMC can be formulated as a quadratic program with non-convex constraints for disk separation, solving this to optimality is challenging. Thus, we again work with a discretized candidate set $C$ and modify the Integer Programming formulation from Section 3.2.2. However, unlike for the GMC, $C$ does not necessarily contain disks of an optimal solution. It could even be that no selection of disks from $C$ provides any feasible solution to the DGMC. Therefore, we modify $C$ to improve the quality of our solutions; see Section 4.2.

We use the GMC as a lower bound to the DGMC. Comparing this to any DGMC solution with a discretized candidate set allows us to evaluate the quality of the solution for the (non-discretized) DGMC. We found our solutions to be very close to lower bounds provided by the GMC IP; see Section 5.

## 4.1 Introducing Separation Constraints

We start with the integer program from Section 3.2.2. Separation between two disks is achieved by using binary variables (to ensure that each disk can be selected at most once) and adding the following constraints to prevent two disks with distance $\leq \ell$ (for some distance $\ell$) from being selected.

$$x_i + x_j \leq 1 \quad \forall d_i, d_j \in C : \|q_i - q_j\| \leq \ell. \tag{1}$$

For $\mathcal{O}(n^3)$ possible disks, this would yield $\mathcal{O}(n^6)$ possible constraints, which is impractical for interesting instances. Thus, we only add the violating constraints in an iterative fashion.

Due to the separation requirement, we can further add a *clique constraint* for each violating disk that ensures at most one disk is selected within a distance $< \frac{\ell}{2}$. For some disk $d_i$ this clique constraint can be formulated as

$$\sum_{\substack{d_j \in C \\ d(q_i, q_j) < \ell/2}} x_j \leq 1. \tag{2}$$

Equation (2) includes separations from Equation (1), so we add Equation (1) for distant violating disk pairs, and Equation (2) for the $\frac{\ell}{2}$ neighbors of each violating disk. We limit the size of the resulting DGMC IP, by only adding Equation (2) for a disk $d_i$ if no clique was added for some other disk $d_j$ in the clique.

## 4.2 Modifying the Candidate Set

The next idea is to enhance the candidate set $C$ by promising disks for coverage. In the GMC solution, single outlier points $p$ with $\kappa(p) > 1$ are often covered using $\kappa(p)$ many drones that cover only $p$. This is no longer possible when ensuring disk separation. For each point $p$ with $\kappa(p) > 1$, we extend the candidate set $C$ by $\kappa(p)$ small disks, that respect the separation constraints, i.e., we construct a regular $\kappa(p)$-gon with side length $\ell$ centered around $p$.

To speed up the solver, we can focus on *small* disks. To that end, we check for the largest disk $d_i$ used in the GMC solution and remove all other disks that have a radius that is greater than $\alpha \cdot r_i$. The factor $\alpha$ compromises between the size of $C$ and solution quality.

# 5 Results

Experiments were carried out on a regular desktop workstation with an AMD Ryzen 9 7900 (12×3.7 GHz) CPU and 88 GB of RAM. Code and data are available[1]. Instances were generated uniformly in a 100 m × 100 m canvas. Values of $\kappa(p)$ for all points were sampled uniformly from $\{1, 2, 3\}$. This yields the instances sets *uni_sm* ($n = 20, 30, \ldots, 200$ and $m = 20$), *uni_lg* ($n = 30, 40, \ldots, 300$ and $m = 30$), and *uni_fix_n* ($m = 5, 10, \ldots, 100$ and $n = 250$). We generated five instances for each parameter combination.

## 5.1 GMC

We compare the GMC heuristic and the integer program in terms of runtime and total area. GMC IP requires time to set up the solver, i.e., (i) computing the candidate set $C$ which takes $\mathcal{O}(n^4)$ and (ii) building the model which takes $\mathcal{O}(n|C|)$. The solver is executed on the resulting integer program.

[1] `https://gitlab.ibr.cs.tu-bs.de/alg/disc-covering`

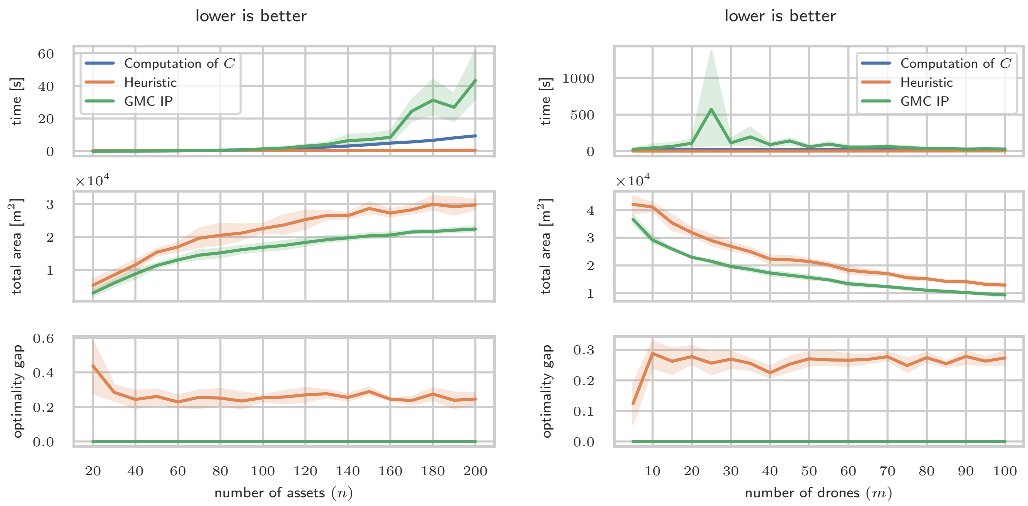



**Figure 3** Comparison of runtime, total area, and optimality gap between the GMC IP solver and the heuristic. On all plots, lower is better. (left) *uni_sm*; fixed $m = 20$ variable $n$. (right) *uni_fix_n*; fixed $n = 250$ variable $m$.

Figure 3 shows that the runtime of the GMC IP solver is significantly higher than of the GMC heuristic. Computing the candidate set is challenging for larger instances, but we do not observe the worst-case behavior in runtime. For *uni_fix_n*, having $m$ between 20 and 35 the GMC IP needs significantly more time to obtain provable optimal solutions.

The lower row of Figure 3 shows a comparison between GMC heuristic and GMC IP in terms of solution quality. The plot shows the optimality gap that is the relative gap between the found solution versus the optimal solution ($(C_{\text{alg}} - C_{\text{ip}})/C_{\text{alg}}$). For both fixed $m$ and $n$, the optimality gap remains stable at around 27.5 % in different instances. The only exception being the case where $m \geq n$ in which the iterative algorithm gives slightly worse results.

## 5.2 DGMC

In Section 4.2 we presented different candidate set strategies that are now compared in terms of solution quality and runtime. For all the experiments, we enable clique constraints and extend the candidate set by small disks.

We ran these exploratory experiments on the benchmark set *uni_sm*, as the workstations ran out of memory for larger instances. After identifying the best parameters, we ran another experiment on the bigger benchmark set *uni_lg*, i.e., for more points and more drones. For all the experiments, we chose a fixed $\ell = 5$. For a fixed value of $\ell$, it is more reasonable to variate the number of points $n$, as more drones increase the difficulty of sparsifying the solution and lead to more infeasibilities.

### 5.2.1 Parameters of DGMC IP

Figure 4 (left) shows the described tradeoff for the $\alpha$ parameter: $\alpha$ controls the size of the largest disk in the candidate set $C$. The top row shows that reducing $C$ improves the solver times significantly. At the same time, the bottom row shows how the solution quality decreases with a smaller set. Setting $\alpha = 1.2$ provides excellent tradeoff between solution quality and runtime, reducing runtime while almost maintaining the same solution quality as the original set, i.e., $\alpha = \infty$.

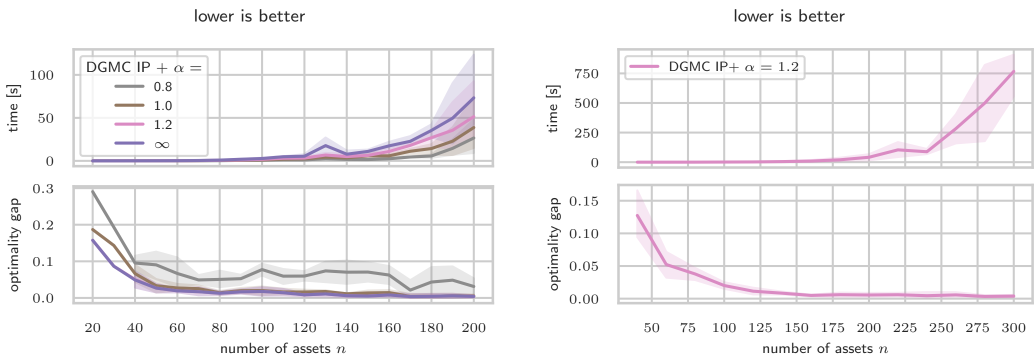



**Figure 4** (left) Tradeoff between reducing the candidate set $C$ and the optimality gap for the DGMC IP on *uni_sm*; $m = 20$ and $\ell = 5$. For comparison, we only display instances that were feasible for all $\alpha$ values, removing 14 of the 95 instances. (right) Runtime and optimality gap of the DGMC IP on *uni_lg*; $m = 30$ and $\ell = 5$.

### 5.2.2 Large benchmark *uni_lg*

First we ran the GMC IP on the benchmark set to obtain the lower bounds for the DGMC. There is a single instance with $n = 300$ that could not be solved within the memory limit; for the remainder of this section, we will exclude this instance.

Based on the results from *uni_sm*, we ran the DGMC IP on the larger benchmark set *uni_lg* with a time limit of 900 s for the solver and set $\alpha = 1.2$. Note that without reducing $C$ (i.e. with $\alpha = \infty$), we cannot reliably solve the larger instances, i.e., instances with more than 250 points, as the integer program requires too much memory.

Figure 4 (right) shows that we can solve all instances close to provable optimality. For smaller instances with $n \leq 100$ the optimality gaps are higher than 2 %. For larger instances, the DGMC IP solver was unable to find optimal solutions for the discretized DGMC (see Section 4) and was terminated due to a timeout. Despite early termination, DGMC IP found solutions with an optimality gap below 0.7 % for all these instances. Note that as we are comparing against the GMC IP solutions (without separation constraints), the gaps to an optimal solution of the DGMC are smaller than what can be seen here.

## 6 Conclusions

Directions for future work include an extension to covered assets in 3D, which is natural for domains such as flying robots, space applications, or undersea sensor networks. Calculating candidate centers is still possible, but more complicated [11]. The high speed of the iterative approximation may be applicable for dynamic targets, or for adjusting the sensor positions and radii when a sensor is added or deleted. A quadratic program for DGMC is suitable for small problems, and could be initialized with the DGMC IP solutions to speed computation.

## References


1 A Karim Abu-Affash, Paz Carmi, Matthew J Katz, and Gila Morgenstern. Multi cover of a polygon minimizing the sum of areas. *International Journal of Computational Geometry & Applications*, 21(06):685–698, 2011.

2 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.

3 Reuven Bar-Yehuda and Dror Rawitz. A note on multicovering with disks. *Computational Geometry*, 46(3):394–399, 2013.

4 Francesco Bernardini, Daniel Biediger, Ileana Pineda, Linda Kleist, and Aaron T. Becker. Strongly-connected minimal-cost radio-networks among fixed terminals using mobile relays and avoiding no-transmission zones. In *IEEE International Conference on Automation Science and Engineering*, 2024.

5 Francesco Bernardini, Mohammadreza Shahsavar, Aaron T. Becker, and Julien Leclerc. Joint optimization of target tracking and communication in a shared network. In *2024 IEEE Texas Symposium on Wireless and Microwave Circuits and Systems (WMCS)*, pages 1–6, April 2024. doi:10.1109/WMCS62019.2024.10619030.

6 Santanu Bhowmick, Kasturi Varadarajan, and Shi-Ke Xue. A constant-factor approximation for multi-covering with disks. In *Symposium on Computational Geometry (SoCG)*, pages 243–248, 2013.

7 K. Boudjit and C. Larbes. Detection and implementation autonomous target tracking with a quadrotor ar.drone. In *12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pages 223–230, 2015.

8 Jahid Chowdhury Choton and Pavithra Prabhakar. Optimal multi-robot coverage path planning for agricultural fields using motion dynamics. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11817–11823, May 2023. doi:10.1109/ICRA48891.2023.10160265.

9 Stefan M. Filipov and Fani N. Tomova. Covering a set of points with a minimum number of equal disks via simulated annealing. In Ivan Georgiev, Maria Datcheva, Krassimir Georgiev, and Geno Nikolov, editors, *Numerical Methods and Applications*, pages 134–145, Cham, 2023. Springer Nature Switzerland.

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. IEEE, 2022.

11 Bernd Gärtner. Fast and robust smallest enclosing balls. In *European symposium on algorithms*, pages 325–338. Springer, 1999.

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. SIAM, 2021.

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.

14 Ratan Lal and Pavithra Prabhakar. Time-optimal multi-quadrotor trajectory planning for pesticide spraying. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7965–7971. IEEE, 2021.

15 Zhihao Liu, Yuanyuan Shang, Timing Li, Guanlin Chen, Yu Wang, Qinghua Hu, and Pengfei Zhu. Robust multi-drone multi-target tracking to resolve target occlusion: A benchmark. *IEEE Transactions on Multimedia*, 25:1462–1476, 2023.

16 Michael Rubenstein, Christian Ahler, and Radhika Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE international conference on robotics and automation*, pages 3293–3298. IEEE, 2012.

17 Martin Zoula and Jan Faigl. Wireless communication infrastructure building for mobile robot search and inspection missions. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5970–5976, May 2024. doi:10.1109/ICRA57147.2024.10611561.