Minimum Scan Cover and Variants – Theory and Experiments

Kevin Buchin ✉ 📧
Department of Mathematics & Computer Science,
TU Eindhoven, The Netherlands

Sándor P. Fekete ✉ 📧
Department of Computer Science,
TU Braunschweig, Germany

Alexander Hill ✉ 📧
Department of Computer Science,
TU Braunschweig, Germany

Linda Kleist ✉ 📧
Department of Computer Science,
TU Braunschweig, Germany

Irina Kostitsyna ✉ 📧
Department of Mathematics & Computer Science,
TU Eindhoven, The Netherlands

Dominik Krupke ✉ 📧
Department of Computer Science,
TU Braunschweig, Germany

Roel Lambers ✉ 📧
Department of Mathematics & Computer Science,
TU Eindhoven, The Netherlands

Martijn Struijs ✉ 📧
Department of Mathematics & Computer Science,
TU Eindhoven, The Netherlands

Abstract

We consider a spectrum of geometric optimization problems motivated by contexts such as satellite communication and astrophysics. In the problem MINIMUM SCAN COVER WITH ANGULAR COSTS, we are given a graph G that is embedded in Euclidean space. The edges of G need to be scanned, i.e., probed from both of their vertices. In order to scan their edge, two vertices need to face each other; changing the heading of a vertex incurs some cost in terms of energy or rotation time that is proportional to the corresponding rotation angle. Our goal is to compute schedules that minimize the following objective functions: (i) in MINIMUM MAKESPAN SCAN COVER (MSC-MS), this is the time until all edges are scanned; (ii) in MINIMUM TOTAL ENERGY SCAN COVER (MSC-TE), the sum of all rotation angles; (iii) in MINIMUM BOTTLENECK ENERGY SCAN COVER (MSC-BE), the maximum total rotation angle at one vertex.

Previous theoretical work on MSC-MS revealed a close connection to graph coloring and the cut cover problem, leading to hardness and approximability results. In this paper, we present polynomial-time algorithms for 1D instances of MSC-TE and MSC-BE, but NP-hardness proofs for bipartite 2D instances. For bipartite graphs in 2D, we also give 2-approximation algorithms for both MSC-TE and MSC-BE. Most importantly, we provide a comprehensive study of practical methods for all three problems. We compare three different mixed-integer programming and two constraint programming approaches, and show how to compute provably optimal solutions for geometric instances with up to 300 edges. Additionally, we compare the performance of different meta-heuristics for even larger instances.

1   Introduction

For many aspects of wireless communication, the relative direction, i.e., the angle of visibility between different locations, plays a crucial role. A particularly striking example occurs in the context of inter-satellite communication, which requires focused transmission, with communication partners facing each other with directional, paraboloid antennas or laser beams. This makes it impossible to exchange information with multiple partners at once. Moreover, a change of communication partner requires a change of heading, which is costly in the context of space missions with limited resources, making it worthwhile to invest in good schedules. Problems of this type do not only arise from long-distance communication. They also come into play when astro- and geophysical measurements are to be performed, in which groups of spacecraft can determine physical quantities not just at their current locations, but also along their common line of sight; see [21] for a description.

In previous theoretical work [15], we considered an optimization problem arising from this context: How can we schedule a given set of intersatellite communications, such that the overall timetable is as efficient as possible? In the problem MINIMUM SCAN COVER WITH ANGULAR COSTS (MSC), the task is to establish a collection of connections between a given set of locations, described by a graph $G = (V, E)$ that is embedded in space. For any connection (or scan) of an edge, the two involved vertices need to face each other; changing the heading of a vertex to cover a different connection takes an amount of time proportional to the corresponding rotation angle. In [15], the goal considered was to minimize the time until all tasks are completed, i.e., compute a geometric schedule of minimum makespan.

Given the importance of conserving energy on space (or drone) missions, this MINIMUM MAKESPAN SCAN COVER (MSC-MS) is not the only important objective: In MINIMUM TOTAL ENERGY SCAN COVER (MSC-TE), the goal is to minimize the sum of all rotation angles; in MINIMUM BOTTLENECK ENERGY SCAN COVER (MSC-BE), the task is to limit the energy used by any one vertex by minimizing the maximum total rotation at one vertex.

In this paper, we complement the previous theoretical results on MSC-MS (hardness and approximation) [15] by presenting an NP-hardness proof and a 2-approximation for MSC-TE and MSC-BE for bipartite graphs in two dimensions. For one-dimensional instances of MSC-TE and MSC-BE, we show a polynomial time algorithm and an upper bound independent of the chromatic number, which shows a fundamental difference to MSC-MS. Most importantly, we provide a comprehensive study of practical methods for all three objective functions. We compare three different mixed-integer programming (MIP) and two constraint programming (CP) approaches, and show how to compute provably optimal solutions for geometric instances with up to 300 edges. Additionally, we evaluate the practical performance of approximation algorithms and heuristics for even larger instances.

1.1   Previous Work

The use of directional antennas has introduced a number of geometric questions. The paper at hand expands on previous work of Fekete, Kleist, and Krupke [15], who investigated MSC-MS and identified a close connection to graph coloring and the (directed) cut cover number. More precisely, MSC-MS in 1D and 2D is in $\Theta(\log \chi(G))$, which implies that even in 1D, there exists no constant-factor approximation for MSC-MS. For 2D, they present a 4.5-approximation for bipartite instances and show inapproximability for a constant better than $3/2$. This yields an $O(c)$-approximation for k-colored graphs with $k \leq \chi(G)^c$.

Further problems involving directional antennas have been considered by Carmi et al. [12], who study the α-MST problem. This problem arises from finding orientations of directional antennas with α-cones, such that the connectivity graph yields a spanning tree of minimum

weight, based on bidirectional communication. They prove that for $\alpha < \pi/3$, a solution may not exist, while $\alpha \geq \pi/3$ always suffices. See Aschner and Katz [7] for more recent hardness proofs and constant-factor approximations for some specific values of α.

Many other geometric optimization problems deal with turn cost. Arkin et al. [5, 6] show hardness of finding an optimal milling tour with turn cost, even in relatively constrained settings, and give approximation algorithms. The complexity of finding an optimal cycle cover in a 2-dimensional grid graph was stated as *Problem 53* in *The Open Problems Project* [14] and shown to be NP-hard in [16], which also provides constant-factor approximations; practical methods and results are given in [17], and visualized in the video [9].

Finding a fastest roundtrip for a set of points in the plane for which the travel time depends only on the turn cost is called the ANGULAR METRIC TRAVELING SALESMAN PROBLEM. Aggarwal et al. [1] prove hardness and provide an $O(\log n)$ approximation algorithm. For the abstract version on graphs in which "turns" correspond to weighted changes between edges, Fellows et al. [19] show that the problem is fixed-parameter tractable in the number of turns, the treewidth, and the maximum degree. Fekete and Woeginger [18] consider the problem of connecting a set of points by a tour in which the angles of successive edges are constrained.

MSC-MS is a special case of scheduling in which the cost of a current job depends on the sequence of the already processed ones; e.g., Allahverdi et al. [2, 3, 4] provide a comprehensive overview, especially on practical work. In the context of earth observation, Li et al. [22] and Augenstein et al. [8] describe MIPs and heuristics to schedule image acquisition and downlink for satellites for which rotation and setup costs are taken into account.

1.2 Preliminaries and Problem Definitions

For all considered versions of MINIMUM SCAN COVER (MSC), the *input* consists of a (straight-line) embedded (not necessarily crossing-free) graph $G = (V, E)$ with a finite vertex set $V \subset \mathbb{R}^2$. We refer to the elements of V as *points* when their specific locations in $\mathbb{R}^2$ are relevant; if we focus on graph properties, we may also refer to them as *vertices*. We denote the undirected edge between $u, v \in V$ by uv. For $v \in V$, we let $N(v) = \{u \in V : uv \in E\}$ be all vertices adjacent to v, and $E(v) = \{uv : u \in N(v)\}$ be all edges incident to v. For two adjacent edges $uv, vw \in E(v)$, let $\alpha(uv, vw) \in [0, 180°]$ denote the smaller angle between the lines supporting the segments uv and vw. The *output* for each problem is a *scan cover* $S : E \to \mathbb{R}^+$, such that for all pairs of adjacent edges e, e', we have $|S(e) - S(e')| \geq \alpha(e, e')$. The geometric interpretation of a scan cover is that all points $v \in V$ have a *heading* that can change over time, and that if $S(uv) = t$ then u and v *face* each other at time t. In this case, we say that the edge uv is *scanned* at time t. Thus, the above condition on S guarantees that S complies with the necessary rotation time if rotation speed is bounded by 1.

A *rotation scheme* describes the geometric change of headings of the vertices over a time interval of length T, i.e., it is a map $r : V \times [0, T] \mapsto [0°, 360°]$. The *total rotation angle* of a vertex v in r is the total amount that v rotates over $[0, T]$. For a given scan cover S, we are particularly interested in edges that are scanned consecutively. Therefore, let $\nu_S(e, e') = 1$ if e and e' share exactly one vertex v and the edge e' is scanned directly after e at v; otherwise $\nu_S(e, e') = 0$.

The Problems. We consider the following three problems, defined by their respective objectives. For a given graph $G = (V, E)$ with vertices in the plane, find a scan cover S with

- Minimum Makespan (MSC-MS): $\qquad\qquad \min \max_{e \in E} S(e)$

- Minimum Total Energy (MSC-TE): $\quad \min \sum\limits_{v \in V} \sum\limits_{e,e' \in E(v)} \alpha(e,e') \cdot \nu_S(e,e')$

- Minimum Bottleneck Energy (MSC-BE): $\min \max\limits_{v \in V} \sum\limits_{e,e' \in E(v)} \alpha(e,e') \cdot \nu_S(e,e')$

Concentrating on the expensive and algorithmically challenging part of efficient rotations between the edges, we do not fix the initial heading of the satellites. In fact, all algorithms can be easily adapted to handle fixed initial headings. Furthermore, for every of the three objectives, an f-approximation can be converted into a $(f+1)$-approximation for the problem variant with fixed initial headings.

For a vertex v, we denote by $\Lambda(v)$ the minimum angle, such that a cone of this angle with apex v contains all edges in $E(v)$. We call such a cone a Λ-cone of v and call the complement of such a cone an *outer* cone of v. A Λ-*cover* is a scan cover for which every vertex v rotates in a single direction, either clockwise or counterclockwise, with a total rotation angle equal to $\Lambda(v)$. Note that different vertices can rotate in different directions. A Λ-cover minimizes both the MSC-TE and MSC-BE objectives.

1.3   Outline and Results

This paper consists of a theoretical part (Section 2) and a practical part (Section 3). Our theoretical results complement the work on MSC-MS [15] by hardness and approximation results for the two new objectives MSC-TE and MSC-BE. In Section 2, we show that both problems can be solved efficiently in 1D; on the other hand, we prove that they are NP-hard in 2D, even for bipartite graphs. Finally, we complement the hardness results by providing 2-approximations for bipartite graphs and $O(\log n)$-approximations for general graphs. Our practical study in Section 3 considers optimal solutions in Section 3.1 and heuristic solutions in Section 3.2. For optimal solutions, we develop three mixed integer linear programs (MIPs), as well as two constraint programs (CPs) and evaluate their practical performance on a suite of benchmark instances. Solving instances of MSC-TE and MSC-BE to provable optimality turned out to be quite difficult; for MSC-MS, we were able to solve instances with up to 300 edges, based on one CP. In addition, we compared the solution quality of four (meta-)heuristics and the approximation algorithms on larger instances with up to 800 edges. In our experiments, a genetic algorithm and the intermediate solution after timeout of one CP produced the best solutions. All omitted proofs can be found in the full version [11].

2   Complexity Results

Fekete, Kleist, and Krupke studied the computational complexity of MSC-MS [15]. In this section, we provide new results for MSC-BE and MSC-TE.

For MSC-MS in 1D, when all vertices are placed on a line, there exists no constant-factor approximation unless $P = NP$ [15]. In contrast, we show that MSC-TE and MSC-BE in 1D can be solved efficiently.

▶ **Theorem 1.** *MSC-TE and MSC-BE in 1D are in P. Moreover, denoting by k the number of vertices with neighbors to both sides, the objective value is 0 for $k = 0$, while for $k > 0$ it is $180° \cdot k$ for MSC-TE and $180°$ for MSC-BE.*

Next we show that for 2D instances of MSC-TE and MSC-BE, there does not exist an efficient algorithm, unless $P = NP$. Specifically, we show that MSC-TE and MSC-BE are NP-hard in 2D, even when the underlying graph $G = (V, E)$ is bipartite. Our proof is

based on the observation that if a Λ-cover exists, any scan cover optimal for MSC-TE is a Λ-cover. If additionally all vertices have the same $\Lambda(v)$, any scan cover optimal for MSC-BE is a Λ-cover. We show finding a Λ-cover is NP-hard via a reduction from the NP-complete problem MONOTONE NOT-ALL-EQUAL 3-SATISFIABILITY (MNAE3SAT) [23, 27], defined as follows: Given a set of Boolean variables X and a set of clauses $\mathcal{C}$ with at most 3 literals from X all of which are not negated, is there a 0/1-assignment to the variables in X, such that for each clause in $\mathcal{C}$, not all variables have the same value?

Given an instance I of the MNAE3SAT, we construct an MSC instance G_I (with the same $\Lambda(v)$ for all vertices) that has a Λ-cover if and only if I has a valid variable assignment. Recall that in a Λ-cover, the edges of each vertex are scanned in either clockwise or counterclockwise order. We encode variable assignment by the rotation direction of the vertices in G_I in a Λ-cover. A variable is encoded by a subgraph that contains a set of vertices that have the same rotation direction in a Λ-cover, and a clause by a subgraph that contains three vertices that cannot all have the same rotation direction in a Λ-cover. We connect variables to clauses via wires, which are encoded by a subgraph that contains two vertices that have the same rotation direction in a Λ-cover. See Figure 1 for an example of the construction.

▶ **Theorem 2.** *MSC-TE and MSC-BE in 2D are NP-hard, even for bipartite graphs.*
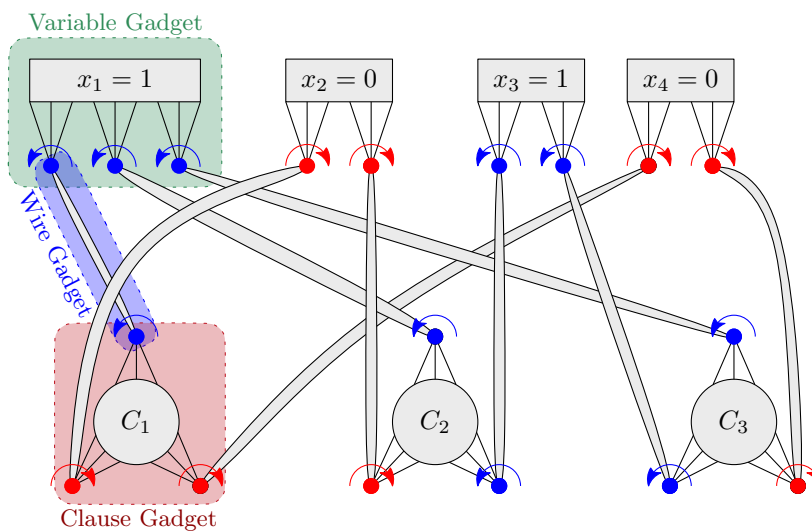
Figure 1 The constructed graph G_I for the instance $(x_1 \vee x_2 \vee x_4) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_3 \vee x_4)$ of MNAE3SAT. The gadgets are drawn symbolically; also shown are the directions of the connector vertices corresponding to the satisfying assignment $x_1 = x_3 = 1, x_2 = x_4 = 0$.

The construction in the proof of Theorem 2 establishes a gap between optimal and suboptimal solutions, which implies a constant-factor approximation lower bound for MSC-BE.

▶ **Corollary 3.** *MSC-BE in 2D is NP-hard to approximate within a factor of 1.04, even for bipartite graphs.*

Next, we complement the 4.5-approximation algorithm for MSC-MS in bipartite graphs in the plane [15] by presenting an approximation algorithms for both remaining objectives.

▶ **Theorem 4.** *There exists a 2-approximation algorithm for MSC-BE and MSC-TE for each bipartite graph $G = (V_1 \cup V_2, E)$ embedded in the plane.*

Proof. Defining $V := V_1 \cup V_2$, the values $\max_{v \in V} \Lambda(v)$ and $\sum_{v \in V} \Lambda(v)$ are clearly lower bounds on the value of a scan cover minimizing MSC-BE and MSC-TE, respectively.

We use the following geometric property based on alternating angles that is also used in [15]: Starting with opposite headings, two vertices face their edge at the same time when both start a full clockwise rotation simultaneously. Defining start headings $r(v, 0) := 0°$ for $v \in V_1$ and $r(v, 0) := 180°$ for $v \in V_2$, the clockwise rotation scheme induces a scan cover S by defining the scan time $S(e)$ of edge e as the time when its two vertices face each other.

We now show that in the rotation scheme r' induced by S, i.e., every vertex v starts to head towards its edge first scanned in S and then follows the order on $E(v)$ defined by S, the total rotation angle of each vertex v is at most $2\Lambda(v)$. To this end, we consider three types of vertices; for an illustration consider Figure 2.

(a) Case (a).   **(b)** Case (b).   **(c)** Case (c).
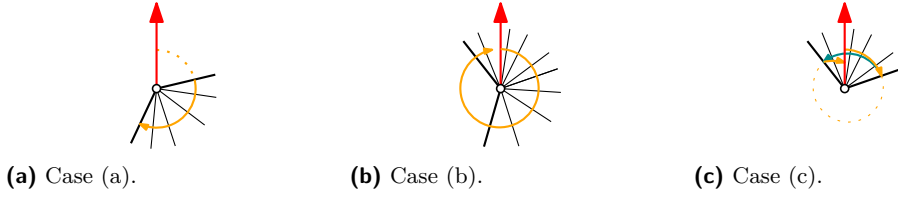
■ **Figure 2** Illustration for the rotation scheme r' in the proof of Theorem 4.

(a) Case: $r(v, 0)$ lies outside the Λ-cone of v. Then all edges of v are scanned by a clockwise rotation, one after the other. Hence, v has a total rotation angle of $\Lambda(v)$.

(b) Case: $r(v, 0)$ lies inside the Λ-cone of v and $\Lambda(v) \geq 180°$. Going over all edges clockwise takes at most a full rotation of $360° \leq 2\Lambda(v)$.

(c) Case: $r(v, 0)$ lies inside the Λ-cone of v and $\Lambda(v) < 180°$. Let e_1 and e_2 denote the bounding edges of the Λ-cone such that $S(e_1) \leq S(e_2)$. By definition, the minimal angle of e_1 and e_2 is $\Lambda(v) < 180°$. Splitting the Λ-cone of v into two halves at $r(v, 0)$, v scans the edges in each half in clockwise direction, rotating an angle of $\Lambda(v)$ counterclockwise between e_1 and e_2. It follows that the total rotation angle of v is at most $2\Lambda(v)$.

As the total rotation angle is at most $2\Lambda(v)$ for each vertex v, MSC-BE and MSC-TE are upper bounded by $2 \cdot \max_{v \in V} \Lambda(v)$ and $2 \cdot \sum_{v \in V} \Lambda(v)$. Together with the lower bounds provided above, this shows that this scan cover is a 2-approximation for *either* objective.   ◄

▶ **Corollary 5.** *Let $G = (V_1 \cup V_2, E)$ be a bipartite graph embedded in the plane such that the points of V_1 and V_2 can be separated by a line. Then an optimal MSC-BE and MSC-TE of G can be found in polynomial time.*

Proof. We follow the same technique as in the proof of Theorem 4. We may assume without loss of generality that the separating line is vertical and that the points of V_1 lie left of the line. Then, with the above definitions, every vertex is in case (a), i.e., the total rotation angle for each vertex v is $\Lambda(v)$. Consequently, the resulting scan cover is optimal for both MSC-BE and MSC-TE.   ◄

The insights from Theorem 4 yield an approximation algorithm for k-colored graphs.

▶ **Corollary 6.** *For MSC-TE and MSC-BE of k-colored graphs embedded in 2D, there exists an $O(\log(k))$-approximation.*

Proof. The edges of a k-colored graph G can be covered by $\lceil \log_2(k) \rceil$ bipartite graphs G_i [25]. For each G_i, we use the 2-approximation of Theorem 4. Clearly, for both objectives, the optimal scan time of G is lower bounded by the optimal scan time of every subgraph. Consequently, scanning all G_i takes at most $\sum_i 2 \cdot OPT(G_i) \leq 2\lceil \log_2(k) \rceil \cdot OPT(G)$, where OPT denotes the optimum scan time for the respective objective. For adjusting the headings between the scan covers of the bipartite graphs, we need $(\lceil \log_2(k) \rceil - 1)$ transition phases each of which needs at most $OPT(G)$. Hence, the total scan time is upper bounded by $(3\lceil \log_2(k) \rceil - 1) \cdot OPT(G)$. ◄

3 Experiments

For our experimental evaluation, we considered two types of benchmark instances in 2D, which we call *random* and *celestial*. Random instances are generated by placing n points chosen uniformly at random from the unit square, with each edge chosen with probability p. Note that the visible area of a satellite constellation on the same altitude in low Earth orbit is fairly close to a set of co-planar points and hence the square (or plane) serves as a reasonable approximation.

Celestial instances are inspired by real-world instances of satellites in a shared orbit, in which they maintain their relative positions while orbiting around a central body like Earth, as long as no explicit orbit-changing maneuvers are carried out. They are characterized by a set of points on a circle and a central circular obstacle. The points on the circle are chosen uniformly at random; an edge exists if and only if its vertices *see* each other, i.e., the edge does not intersect the central obstacle. Examples and the distribution of the nearly 2000 instances with up to 800 edges used for our experiments can be seen in Figure 3.

All experiments were run on Intel Core i7-3770 with 3.4 GHz and 32 GB of RAM.

(a) A celestial instance. **(b)** Instance distribution. **(c)** A random instance.

Figure 3 Examples of instances with 15 vertices and ~ 70 edges: (a) celestial, (c) random, and (b) the instance distribution. Auxiliary lines in (b) indicate graphs with edge densities 50% (dashed) and 100% (solid) of complete graphs.

3.1 Exact Algorithms

We developed three mixed integer programs (MIPs) and two constraint programs (CPs) to solve instances to provable optimality. Note that not every program solves all three problems. An experimental evaluation is given at the end of this section. While we focus on 2-dimensional geometric instances, all formulations are applicable to all metric cost functions.

3.1.1   Mixed Integer Program 1 (MSC-MS, MSC-TE, MSC-BE)

Our first MIP, denoted by MIP-1, uses two types of variables. The first type are real variables $t_e \geq 0$ for all $e \in E$. The second type are Boolean variables $x_{(e,e')}$ for all ordered edge pairs $(e, e') \in E^2$. In a computed solution, the variables t_e define a scan cover in which $S(e) := t_e$ and the value of $x_{(e,e')}$ corresponds to $\nu_S(e, e')$. Because $\nu_S(e, e') = 0$ if $|e \cap e'| \neq 1$, we directly set $x_{(e,e')} := 0$ in these cases. Consequently, the objective functions can be expressed by substitution of $S(e)$ with t_e and $\nu_S(e, e')$ with $x_{(e,e')}$. Note that a min-max objective can be implemented by a single additional real variable and one additional constraint for each term in the objective.

We introduce a set of constraints to guarantee that the t-variables and the x-variables arise from a valid scan cover. Because the angle function α fulfills the triangle inequality, it suffices to ensure the time difference of the t-variables for all $x_{(e,e')} = 1$. We know that $M_1 := \log_2 n \cdot 360°$ is an upper bound on the minimal makespan for a graph G in 2D with n vertices [15]. Moreover, a makespan of $M_2 := |E| \cdot 180°$ allows to scan each edge individually, and thus an optimal scan cover of MSC-BE and MSC-TE can be realized in this makespan. Therefore, by inserting the correct M_i, we can enforce feasible scan times by using the Big-M method.

$$\forall v \in V, \forall (e, e') \in E(v) \times E(v), e \neq e': \quad t_{e'} \geq t_e + \alpha(e, e') - (1 - x_{(e,e')}) \cdot M_i. \tag{1}$$

This leaves us with ensuring that the x-variables correspond to a feasible scan cover. First, for every vertex v, an incident scanned edge e has at most one predecessor edge and one successor edge in the scan order.

$$\forall v \in V, e \in E(v): \sum_{e' \in E(v), e' \neq e} x_{(e,e')} \leq 1 \quad \text{and} \quad \sum_{e' \in E(v), e' \neq e} x_{(e',e)} \leq 1 \tag{2}$$

Second, the total number of scanned edges at vertex v is $|E(v)|$, i.e., the number of consecutively scanned edge pairs, is $|E(v)| - 1$.

$$\forall v \in V: \sum_{e,e' \in E(v) \times E(v), e \neq e'} x_{(e,e')} = |E(v)| - 1 \tag{3}$$

Together, Equations (2) and (3) enforce that every vertex has exactly one first and one last scanned edge in the induced scan order. Because Equation (1) enforces that the scan times obey the rotation times, there are no cycles in the sequence defined by x if all angles are positive. This fact is very similar to the Miller-Tucker-Zemlin formulation of the TSP [24]. In the presence of 0°-angles, we dynamically add the following constraint similar to the Dantzig formulation [13] to separate these cycles.

$$\forall v \in V, \forall S \subsetneq E(v), S \neq \emptyset: \sum_{e \in S, e' \in E(v) \setminus S} x_{(e,e')} + x_{(e',e)} \geq 1 \tag{4}$$

3.1.2   Mixed Integer Program 2 (MSC-MS)

The abstract definition of the MSC [15] can be directly implemented as a MIP, because absolute values can be implemented using a Boolean variable. Some modern solvers like Gurobi actually provide this functionality directly. Like for MIP-1 (Section 3.1.1), we have a real-valued variable $t_e \geq 0$ for each $e \in E$ that states its scan time. We try to keep the maximum value assigned to any $t_e, e \in E$ as low as possible. For every two incident edges

vw and vu, we only have the constraint that t_{vw} and t_{vu} have to be at least the time apart that v needs to rotate between these two. This results in the following MIP-2.

$$\min \qquad \max_{e \in E} t_e \qquad\qquad\qquad\qquad\qquad\qquad (5)$$

$$\text{s.t.} \qquad |t_{vw} - t_{vu}| \geq \alpha(vu, vw) \qquad\qquad \forall vw, vu \in E \qquad (6)$$

$$t_e \geq 0 \qquad\qquad\qquad \forall e \in E \qquad (7)$$

The main difference to MIP-1 is that we do not keep a record of the actually performed rotations. As a consequence, MIP-2 can only be used for MSC-MS. However, on the positive side, we do not need to dynamically add additional cycle constraints.

3.1.3 Mixed Integer Program 3 (MSC-TE, MSC-BE)

The third MIP (defined by Equations (2)–(4) and (8)), denoted by MIP-3, is a variant of MIP-1 (Section 3.1.1) in which the t-variables and the corresponding Big-M based constraint (Equation (1)) are removed. As a consequence, we may use it for MSC-BE and MSC-TE, as they only need the x-variables.

It is possible that the scan orders at the individual vertices are cycle free, but that the overall schedule has a deadlock when the vertices wait for each other, see Figures 4a and 4b. We therefore prohibit directed cycles in the scan order defined by the x-variables (if not already separated by Equation (4)) dynamically via callbacks for every newly found integral solution. Violated constraints can be found via a simple DFS search.

$$\forall k \in \mathbb{N}_{|V|}, \forall (e_0, e_1, \ldots, e_{k-1}) \in E^k: \quad x_{(e_{k-1}, e_0)} + \sum_{i=0,1,\ldots k-2} x_{(e_i, e_{i+1})} \leq k - 1 \qquad (8)$$

Note that these cycles can also happen in MIP-1, but only with zero rotation costs between the involved edges. Thus, they are irrelevant for the solution, as all of these edges can be scanned at once.

3.1.4 Constraint Program 1 (MSC-MS)

Our first constraint program (denoted by CP-1) has the same formulation as MIP-2. The only difference between the CP version and the MIP version lies in the employed solver. In particular, absolute values can be modeled directly.

(a) Rotation scheme.   **(b)** Cycle in scan order.
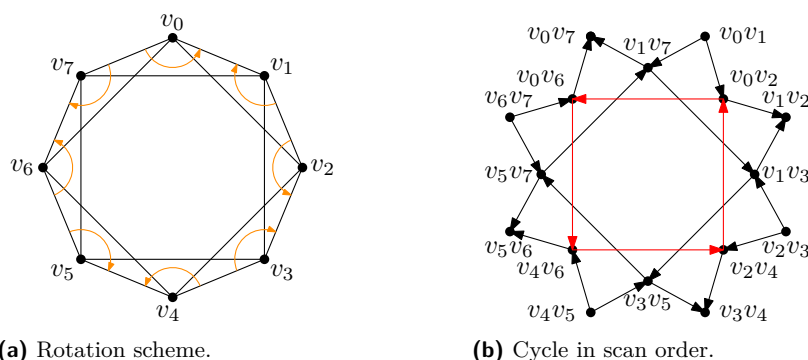
Figure 4 A globally infeasible edge order fulfilling Equations (2) and (3), i.e., it is cycle-free at each vertex: (a) its rotations scheme (b) the resulting edge order that contains a cycle. An arc (e, e') in this graph corresponds to an $x_{(e,e')} = 1$.

3.1.5   Constraint Program 2 (MSC-TE, MSC-BE)

Our second constraint program (defined by Equations (2), (3), and (9)), denoted by CP-2, is similar to MIP-3 described in Section 3.1.3. However, MIP-3 adds Equations (4) and (8) dynamically, which our CP does not support. Because adding all these constraints directly results in a prohibitively large formulation, we instead use a conditional variant of the Miller-Tucker-Zemlin [24] formulation to eliminate cycles in the scan order. Different from MIPs, we do not need the Big M method for CPs, but can implement conditional constraints directly. More precisely, we add the variables $o_e \in \mathbb{N}_{|E|}, e \in E$ that state the cycle-free scan order of the edges, which is enforced by the constraints

$$\forall (e, e') \in E \times E\colon \qquad o_{e'} - o_e \geq 1 \quad \text{if } x_{(e,e')} = 1. \tag{9}$$

3.1.6   Experimental Evaluation of Exact Algorithms

We used *Gurobi* (v9.0.1) for solving the MIPs and *CP-SAT* of Google's *or-tools* (v7.7.7810) for solving the CPs. CP-SAT, which is based on a SAT solver, requires all coefficients and variables to be integral for computational efficiency. We therefore convert the floating point values to integral values including the first eight floating point digits (rounded, decimal). While this weakens the accuracy, we calculated a theoretical maximal deviation of less than $1 \times 10^{-4}\,\%$, which we consider negligible and comparable to the accuracy of the MIP solver.

   We considered all solvers for the three objectives on the two instance types described in the preliminaries. We evaluated how many instances of which size could still be solved to provable optimality within a time limit of 900 sec; see Figure 5. For MSC-MS, CP-1 has a clear lead, solving $50\,\%$ of the instances with $242 \pm 5\%$ edges for random instances, and $125 \pm 5\%$ edges for celestial instances. In our experiments, neither MIPs was able to solve any instance with more than 70 edges to provable optimality. For MSC-TE, MIP-1 and MIP-3 performed better than CP-2, but all solvers could barely solve instances with more than 30 edges. While MIP-1 has a more direct objective without auxiliary constraints and variables as needed for MSC-MS, its actual performance was slightly worse. For MSC-BE, CP-2 performed considerably better; for celestial instances, it can solve instances nearly twice as large ($\geq 50\,\%$ at $48 \pm 5\%$ edges) than the MIPs. Surprisingly, MIP-1 was slightly better than CP-2 for random instances, being able to solve $50\,\%$ of the instances with $61 \pm 5\%$ edges. Overall, CPs appear to be considerably more effective than MIPs, and random instances show to be easier to solve than celestial ones.

### 3.2   Approximations and Heuristics

For larger instances (beyond the size that was solvable to provable optimality), we developed additional methods based on approximation algorithms and heuristics that provided good (but not provably optimal) solutions.

3.2.1   Bipartite Approximation Algorithms with Coloring Partition

The constant-factor approximation algorithms for bipartite graphs extend to general graphs by partitioning them into bipartite graphs and applying the corresponding approximation algorithm to each of the bipartite subgraphs. More specifically, assigning a vector over $\{0, 1\}$ with $\lceil \log_2 k \rceil$ bits to each color class of a k-colored graph induces a covering of its edge set with $\lceil \log_2 k \rceil$ bipartite graphs; for more details see Motwani and Naor [25]. For MSC-MS, this even preserves the approximation factor [15]. We use the well-engineered
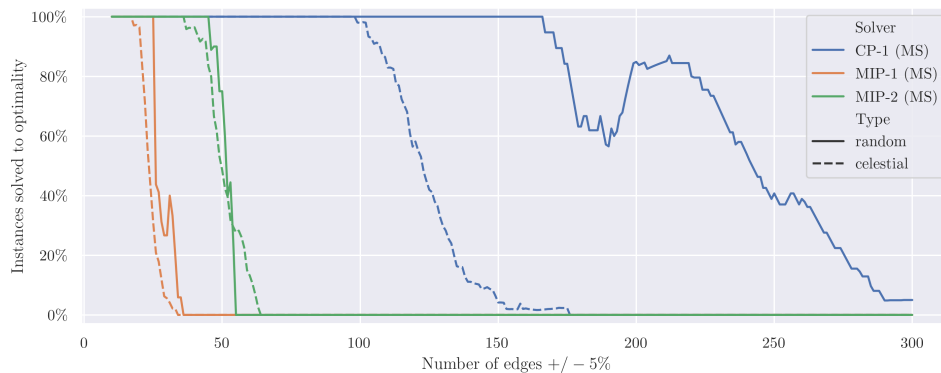
(a) Makespan.
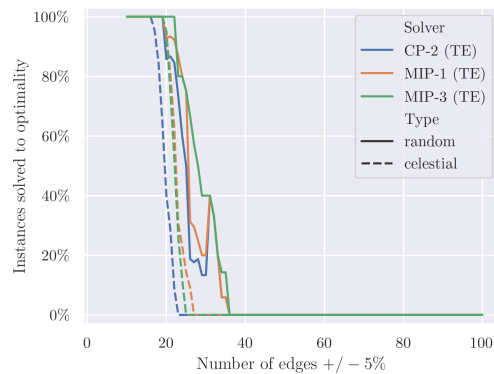
(b) Total Energy.
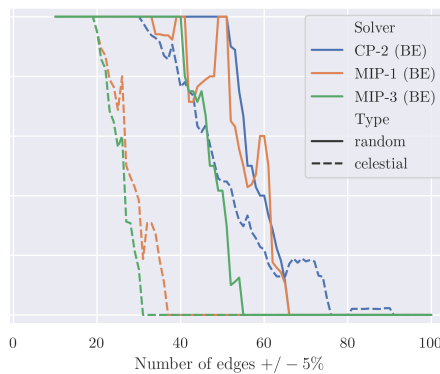
(c) Bottleneck Energy.

Figure 5 Performance of the exact solver measured in how many instances with $m \pm 5\%$ edges can be solved to provable optimality within $900\,\mathrm{sec}$. The bump for CP-1 starting at 200 can be explained by the instance distribution that at this point includes more instances with lower degree.

dsatur heuristic [10] for the graph coloring problem, which is shipped with the *pyclustering*-package [26]. Concatenating the solutions of the bipartite graphs yields a feasible scan cover; here we use a greedy approach to minimize the transition costs. We denote this method by APX.

3.2.2 (Meta-)Heuristics

We also considered a number of (meta-)heuristics for optimizing the three objectives.

Greedy: Scan the first edge regarding a given or random order and then scan the edge that increases the objective the least, until all edges are scanned. If multiple edges are equally good, the first one regarding the order is selected. Many edges can be inserted without extra cost and thus the initial edge order has a strong influence on the result.

Iterated Local Search (ILS): This simple but potentially slow heuristic considers for a given start solution (in this case of *Greedy*) all possible swaps of edges; the locally best swap is carried out, until no further improvement is possible.

Simulated Annealing (SA): This common variation of *Iterated Local Search* performs swaps according to a probability based on the Boltzmann function $\mathbf{Boltzmann}(T, s_1, s_2) = e^{1/T \cdot (s_2 - s_1)}$, where s_1 is the objective value of the current best solution, s_2 is the objective value of the considered solution, and $T \in \mathbb{R}^+$ is the current *temperature*. The temperature

decreases over time and with it the likelihood of a worse solution being used. If the objective does not improve for some time, the temperature is increased in order to escape the local minimum. Due to randomization, we can run multiple searches in parallel. We terminate if the solution has not improved for some time.

Genetic Algorithm (GA): We start with an initial population of 200 solutions generated by a randomized Greedy. A solution is encoded by assigning each edge a fractional number between 0.0 and 1.0, similar to [20]. The scan order is determined by sorting the edges by these numbers. In each round, we build a new population by selecting the best 10% of the old population (*elitism*) and then fill the rest of the population by crossovers of the old generation. For a crossover, we select two solutions of the old generation with a probability matching their objective values (*uniform selection*) and for each edge we choose with equal probability either the number from the first or second solution (*uniform crossover*). If by chance, two edges get the same number, we randomly change one of them without influencing the order. Of the new generation of solutions, 3% are selected for mutation. A mutation applies Greedy with a probability of 60% (the old order is used as initial edge order) or changes each edge with a 3% probability to a new random number. This is repeated until we either reach a time limit of 900 sec, 300 generations, or 60 generations without improvement. The best solution found during this process is then returned.

### 3.2.3   Experimental Evaluation of Approximations and Heuristics

Figure 6 shows experimental results for heuristically solving instances with up to 800 edges with a 900 sec time limit (at which point the current solution is returned). For MSC-MS, CP-1 yields the best results even for larger instances (where it is aborted by the time limit) by a margin of 25 % to 50 % to the next best algorithm, GA. For MSC-TE, the genetic algorithm turned out to be the best approach for celestial instances by a margin of over 50 % for the larger instances. Surprisingly, CP-1 (optimizing for MSC-MS) yields slightly better solutions than the genetic algorithm for random instances of MSC-TE. The most interesting results are for MSC-BE. Here, CP-1 achieves the best results by a margin of over 20 % for random instances, and GA (TE) the best results for celestial instances by a margin of over 40 %. The excellent performance of CP-1 can be explained by a strong correlation of MSC-MS and MSC-BE for random graphs, as shown in Figure 7. The fact that GA (TE) is actually better in optimizing MSC-BE than GA (BE) can be explained by the weaker gradients of bottleneck objectives, because only a small part of the solution (the most expensive vertex) actually contributes to the value. However, the initial bump, at which the exact solver of MSC-BE still yields (better) solutions, indicates that these solutions could be far from optimal and that there may still be room for improvement.

Overall, either CP-1 or GA (TE) yields the best solutions. CP-1 is especially strong on random instances for all three objectives. The approximation algorithm is usually among the worst. For MSC-MS, the algorithm performs a full rotation for nearly all instances, as $\max_{v \in V} \Lambda(v)$ is usually above 180°. Note that the factor can be worse than the approximation factor 4.5 (resp. 2), because these are not bipartite graphs.

In Figure 7 (first row, fourth and last column) we can additionally see that for MSC-MS the objective correlates strongly with the number of edges for celestial instances and with the average degree for random instances. Total energy seems to primarily correlate with the number of edges for both types; our random instances are on average twice as expensive. For MSC-BE, only random graphs seem to have a significant correlation to MSC-MS and the average degree.
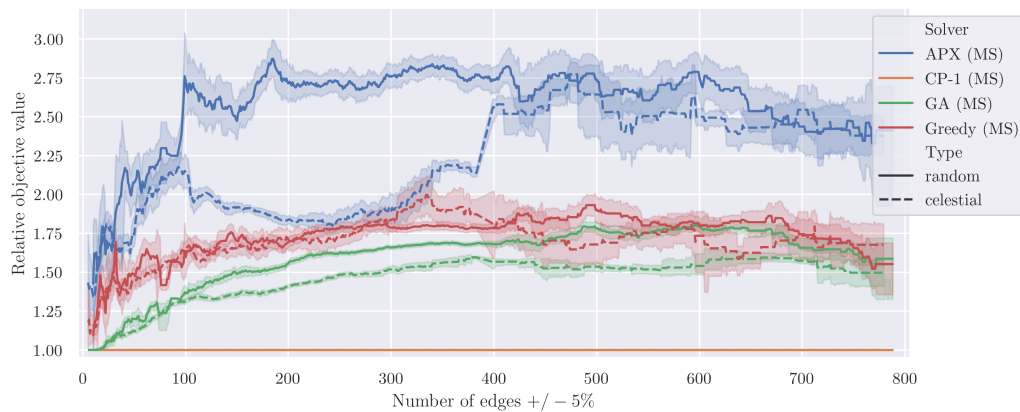
(a) Makespan.
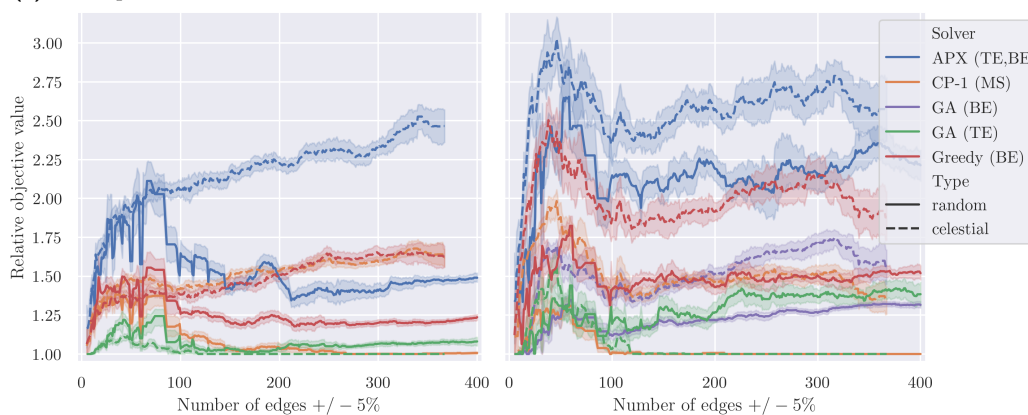
**(b)** Total energy. **(c)** Bottleneck energy.

Figure 6 Relative performance of the non-exact methods, measured by the obtained objective value divided by the best known value. We used the same instances for the exact solver, so the better denominator creates a small bump for smaller instance sizes, in particular for MSC-BE. Except for CP-1, the exact solvers did not yield good solutions for larger instances, if any at all, and are thus excluded for readability. The plots show the mean and the corresponding 95 % confidence interval. We highlight the difference between the two instance types by using different styles for the lines. Note that because these are relative values, a comparison of the performance over the different objectives is not possible. ILS and SA are excluded for readability and perform only slightly better than Greedy.

4 Conclusion and Open Problems

We studied problems of minimum scan cover with three different practically relevant objective functions, providing both theoretical and practical contributions: complexity and algorithmic results for the new objectives (MSC-TE and MSC-BE), and practical methods for computing provably optimal solutions for smaller and near-optimal solutions for larger instances.

In particular, we developed multiple MIP and CP formulations and demonstrated that instances of MSC-MS can be solved reliably for instances with more than 100 edges using constraint programming which performs much better than our MIP approaches. While this approach generalizes also to 3D, we only tested 2D instances; it is open whether these results also carry over to 3D. MSC-TE and MSC-BE can only be solved to optimality for much smaller instances. For solving larger instances without guarantee of optimality, we
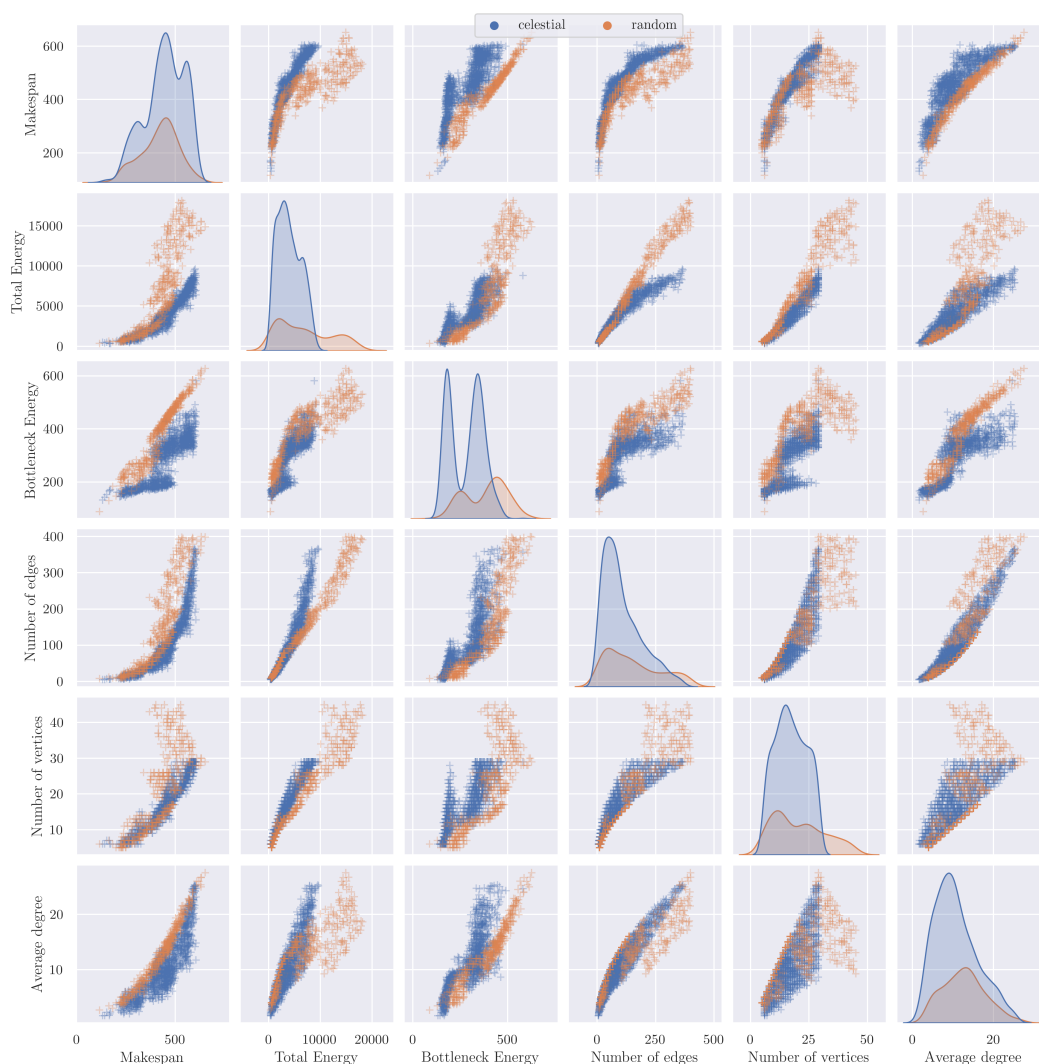
Figure 7 Correlation and distribution of the best known objectives and instance properties. The diagonal shows the density distribution of the x-values. The scatter plots have a point for every existing value pair, which allows to detect correlations.

evaluated approximation algorithms and a spectrum of meta-heuristics. Within the given time limit, CP-1 provided the best solutions for all MSC-MS instances, and even the random instances for MSC-TE and MSC-BE, despite only optimizing for MSC-MS. For celestial instances of MSC-TE and MSC-BE, the genetic algorithm optimizing for MSC-TE provides the best solutions. However, the results indicate perspectives for improving the optimization of MSC-BE.

At this point, fully dynamic instances (in which the vertices change their relative positions to each other over time, such as for satellites with different orbit parameters) are yet to be explored. These promise to be even more challenging, due to bigger gaps between optimal and suboptimal solutions, resulting from possibly long delays when a limited communication window has been missed.

References

1 Alok Aggarwal, Don Coppersmith, Sanjeev Khanna, Rajeev Motwani, and Baruch Schieber. The angular-metric traveling salesman problem. *SIAM J. Comp.*, 29(3):697–711, 1999.

2 Ali Allahverdi. The third comprehensive survey on scheduling problems with setup times/costs. *Eur. J. Oper. Res.*, 246(2):345–378, 2015.

3 Ali Allahverdi, Jatinder N.D. Gupta, and Tariq Aldowaisan. A review of scheduling research involving setup considerations. *Omega*, 27(2):219–239, 1999.

4 Ali Allahverdi, C.T. Ng, T.C. Edwin Cheng, and Mikhail Y. Kovalyov. A survey of scheduling problems with setup times or costs. *Eur. J. Oper. Res.*, 187(3):985–1032, 2008.

5 Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Sándor P. Fekete, Joseph S. B. Mitchell, and Saurabh Sethia. Optimal covering tours with turn costs. In *Symp. Disc. Alg. (SODA)*, pages 138–147, 2001.

6 Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Sándor P. Fekete, Joseph S.B. Mitchell, and Saurabh Sethia. Optimal covering tours with turn costs. *SIAM J. Comp.*, 35(3):531–566, 2005.

7 Rom Aschner and Matthew J. Katz. Bounded-angle spanning tree: modeling networks with angular constraints. *Algorithmica*, 77(2):349–373, 2017.

8 Sean Augenstein, Alejandra Estanislao, Emmanuel Guere, and Sean Blaes. Optimal scheduling of a constellation of Earth-imaging satellites, for maximal data throughput and efficient human management. In *Int. Conf. Automated Planning & Scheduling*, pages 345–352, 2016.

9 Aaron T. Becker, Mustapha Debboun, Sándor P. Fekete, Dominik Krupke, and An Nguyen. Zapping Zika with a mosquito-managing drone: Computing optimal flight patterns with minimum turn cost. In *Symp. Comp. Geom. (SoCG)*, pages 62:1–62:5, 2017.

10 Daniel Brélaz. New methods to color the vertices of a graph. *Comm. ACM*, 22(4):251–256, 1979.

11 Kevin Buchin, Sándor P. Fekete, Alexander Hill, Linda Kleist, Irina Kostitsyna, Dominik Krupke, Roel Lambers, and Martijn Struijs. Minimum scan cover and variants – theory and experiments, 2021. `arXiv:2103.14599`.

12 Paz Carmi, Matthew J. Katz, Zvi Lotker, and Adi Rosén. Connectivity guarantees for wireless networks with directional antennas. *Comp. Geom.*, 44(9):477–485, 2011.

13 George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Op. Res. Soc. of America*, 2(4):393–410, 1954.

14 Erik D. Demaine, Joseph S. B. Mitchell, and Joseph O'Rourke. The Open Problems Project. URL: `http://cs.smith.edu/~orourke/TOPP/`.

15 Sándor P. Fekete, Linda Kleist, and Dominik Krupke. Minimum scan cover with angular transition costs. In *Symp. Comp. Geom. (SoCG)*, volume 164, pages 43:1–43:18, 2020. `doi:10.4230/LIPIcs.SoCG.2020.43`.

16 Sándor P. Fekete and Dominik Krupke. Covering tours and cycle covers with turn costs: Hardness and approximation. In *Int. Conf. Algor. Complexity (CIAC)*, pages 224–236, 2019.

17 Sándor P. Fekete and Dominik Krupke. Practical methods for computing large covering tours and cycle covers with turn cost. In *Alg. Engin. Exp. (ALENEX)*, pages 186–198, 2019.

18 Sándor P. Fekete and Gerhard J. Woeginger. Angle-restricted tours in the plane. *Comp. Geom.*, 8:195–218, 1997.

19 Mike Fellows, Panos Giannopoulos, Christian Knauer, Christophe Paul, Frances A. Rosamond, Sue Whitesides, and Nathan Yu. Milling a graph with turn costs: A parameterized complexity perspective. In *Worksh. Graph Theo. Conc. Comp. Sci. (WG)*, pages 123–134, 2010.

20 M. Gholami, M. Zandieh, and A. Alem-Tabriz. Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns. *Int. J. Adv. Manufact. Tech.*, 42(1-2):189–201, 2009.

21 Haje Korth, Michelle F. Thomsen, Karl-Heinz Glassmeier, and W. Scott Phillips. Particle tomography of the inner magnetosphere. *J. Geophys. Res.: Space Phys.*, 107(A9):SMP–5, 2002.

22   Guoliang Li, Lining Xing, and Yingwu Chen. A hybrid online scheduling mechanism with revision and progressive techniques for autonomous Earth observation satellite. *Acta Astronautica*, 140:308–321, 2017.

23   László Lovász. Coverings and colorings of hypergraphs. In *Southeastern Conf. Combin., Graph Th., Comput. (SEICCGTC)*, pages 3–12, 1973.

24   Clair E. Miller, Albert W. Tucker, and Richard A. Zemlin. Integer programming formulation of traveling salesman problems. *J. ACM*, 7(4):326–329, 1960.

25   Rajeev Motwani and Joseph (Seffi) Naor. On exact and approximate cut covers of graphs. Technical report, Stanford University, Stanford, CA, USA, 1994.

26   Andrei Novikov. PyClustering: Data mining library. *J. Open Source Softw.*, 4(36):1230, 2019.

27   Thomas J. Schaefer. The complexity of satisfiability problems. In *Symp. Th. Comp. (STOC)*, pages 216–226, 1978. `doi:10.1145/800133.804350`.