A Postprint of an Edge Contraction Greedy Algorithm for the Wireless Network Interference Minimization Problem

Authors: Du Lei, Fu Ximei, Yang Wenhan

Date: 2019-04-01T00:00:00+00:00

Abstract

The problem of minimizing maximum interference in wireless sensor networks can be described as follows: given the positions of n points on a plane and a threshold for transmission radius, the objective is to set the transmission radii such that the maximum number of points covered by the transmission range of any other point is minimized. To effectively solve this problem, a novel greedy algorithm—the edge-contraction algorithm—is proposed. Unlike existing algorithms' graph construction approaches, this algorithm constructs the network communication topology graph through edge contraction, and accelerates the greedy algorithm by incorporating the batch processing concept from operating systems, thereby reducing the algorithm' s running time. Experimental verification demonstrates that this algorithm produces superior solutions compared to existing algorithms on randomly generated test instances.

Full Text

Preamble

Edge Shrinking Greedy Algorithm for Minimizing Interference in Wireless Networks

Du Lei[1], Fu Ximei[2], Yang Wenhan[1] (1. School of Computer, South China Normal University, Guangzhou 510631, China; 2. School of Mathematics & Statistics, Shaoguan University, Shaoguan Guangdong 512005, China)

Abstract: The problem of minimizing the maximum interference in wireless sensor networks can be described as follows: given the positions of n points on a plane and thresholds for transmission ranges, we need to set the transmission radius for each point such that the maximum number of other points whose transmission ranges cover any given point is minimized. To effectively solve this problem, this paper proposes a novel greedy algorithm called the edge shrinking algorithm. Unlike existing algorithms' graph construction methods, this algorithm constructs a desirable network communication topology by shrinking edges. By incorporating the idea of batch processing from operating systems to accelerate the greedy algorithm, the running time is significantly reduced. Experimental validation demonstrates that this algorithm produces better solutions than existing algorithms on randomly generated test cases.

Key words: wireless sensor network; greedy algorithm; interference minimization; batch processing; connected graph; shrinking strategy

---

0 Introduction

Wireless sensor networks (WSN) consist of large numbers of stationary or mobile sensors that form self-organizing, multi-hop wireless networks. These networks collaboratively sense, collect, process, and transmit information about monitored objects within their geographical coverage area, ultimately delivering this information to network owners. Wireless sensor networks employ various types of sensors to detect diverse phenomena in their surrounding environment, including seismic activity, electromagnetic fields, temperature, humidity, noise, light intensity, pressure, soil composition, and the size, speed, and direction of moving objects [1,2].

Since wireless sensor networks are typically deployed in locations that are inconvenient for direct human control, and sensor nodes generally operate on batteries that are difficult to recharge or replace, energy consumption becomes a critical factor limiting the service lifetime of wireless networks [3]. One method for conserving node energy is reducing interference caused by simultaneous signal transmissions from nearby nodes.

Current interference reduction methods include frequency division multiplexing, time division multiplexing, and topology control. While the first two methods can effectively reduce interference, they require sophisticated technology and incur high costs. Topology control, by contrast, reduces interference by assigning an appropriate transmission radius to each node to form a connected topology, thereby minimizing a certain objective function of the radius to reduce the network' s maximum interference. Consequently, topology control has become an important method for interference reduction in wireless sensor networks [4] and has attracted significant research attention in recent years.

The problem of computing minimum maximum interference for wireless sensor networks in the plane while maintaining network connectivity has been proven to be NP-hard [5], making it difficult to find exact solutions in polynomial time. Few algorithms have been proposed in the literature to solve this problem. Notable examples include: in 2008, Halldorsson proposed an approximation algorithm combining computational geometry and -net theory [6] with maximum interference of $\Delta \cdot$, where Δ is the maximum node degree in a network with identical transmission radii. In the same year, researchers proved that any transmission radius function producing a connected graph could be modified to yield a radius function with essentially unchanged interference, where all sensor node transmission radii are less than R , the smallest possible radius that maintains strong network connectivity [7]. In 2009, the nearest neighbor algorithm was proposed based on each node' s relationship with its closest neighbor [8]. In 2010, Aslanyan et al. presented an approximation algorithm based on partial set cover [9]. In 2014, an optimal neighbor algorithm with $O(n^2)$ time complexity was proposed based on Prim' s algorithm [10]. Subsequent work attempted to incorporate regret mechanisms into the optimal neighbor algorithm [11,12], improving accuracy to some extent at the cost of increased running time.

This paper addresses the minimization of maximum interference in planar wireless sensor networks through topology control, proposing a novel greedy algorithm with a different graph construction approach from existing methods, and validating its effectiveness through extensive random test cases.

---

1 Problem Model and Related Algorithms

Multiple models have been proposed to address interference reduction in wireless communications [13]. One model is sender-centric, where interference refers to edge interference. In another receiver-centric model, interference refers to node interference, where the interference at any point v is defined as the number of other nodes whose transmission ranges cover v. The sender-centric model only considers the sender, whereas actual interference occurs at the receiver, making this model unable to reflect real-world conditions. This paper adopts the receiver-centric model and designs effective algorithms for interference minimization in wireless sensor networks through topology control.

We use graph theory concepts to describe the problem model, requiring familiarity with relevant terminology. Since wireless sensor networks are modeled as graphs, the terms "graph" and "network" are used interchangeably in this paper. For graph-related concepts, please refer to the literature [14]. Below we provide relevant definitions, followed by a formal problem statement and introduction to the minimum interference model.

A wireless sensor network in the plane can be described as a set of nodes, where each point i has two-dimensional coordinates $(x(i), y(i))$ and satisfies a radius function $r(i)$ $[0, R(i)]$, where $R(i)$ is the node' s radius threshold. Therefore, all transmission relationships among nodes derived from a transmission radius function r can be modeled as a graph $G = V, E$. For any two points u and v in V, edge (u, v) exists if and only if both $r(u)$ and $r(v)$ are not less than the Euclidean distance $d(u, v)$ between u and v, i.e., $r(u)$ $d(u, v)$ and $r(v)$ $d(u, v)$. For convenience, we refer to $G = V, E$ as the derived graph of transmission radius function r and make the following definitions.

Definition 1 For any two points u, v V, the transmission relationship derived via transmission radius function r is denoted as graph $G = V, E$, where edge (u, v) E exists if and only if the transmission radii $r(u)$ and $r(v)$ are not less than their Euclidean distance $d(u, v)$, i.e., $r(u)$ $d(u, v)$ and $r(v)$ $d(u, v)$. Therefore, $G = V, E$ is called the derived graph of transmission radius function r.

Definition 2 The interference degree of any point v in derived graph $G = V, E$ is defined as $I\ (v)$, representing the number of points in the graph whose transmission ranges can cover node v.

Definition 3 The interference degree of derived graph $G = V, E$ is defined as the maximum interference degree among all points in the graph, i.e., $I = \max\{I\ (v) \mid v \quad V\}$.

Based on Definitions 2 and 3, the interference at point v in Figure 1 [Figure 1: see original paper] is 3, and the interference of the derived graph is also 3.

1.1 Interference Minimization Model

Considering the fundamental requirement of connectivity, we have the following problem model:

Minimum Interference Problem (MIP): Given a node set V in the plane, where any point i V has coordinates $(x(i), y(i))$, and each point i V has a transmission radius threshold function $R(i)$ $[0, 1]$ such that the derived graph $G = V, E$ is connected. The goal is to determine a transmission radius function r such that the derived graph $G = V, E$ is connected and has minimum interference.

From the above definitions, for any transmission radius function r, the derived graph G is a subgraph of G . Moreover, if we reselect node v' s transmission radius as the distance to its farthest neighbor in G , the resulting derived graph has the same interference as G . Therefore, we can select v's transmission radius only from the different distances to its reachable neighbor nodes in G . Through this selection method, node v' s transmission radius becomes the distance to its farthest reachable neighbor in G .

MIP has been proven to be NP-hard. Even finding an optimal instance for the special case where all nodes lie on a line is difficult, though its hardness has not been formally proven. For special cases of two-dimensional networks, such as grid networks, the problem remains NP-hard. The difficulty of MIP stems from its combinatorial structure, as many intuitive properties such as minimum degree, nearest neighbor, and best neighbor fail to produce satisfactory solutions. Therefore, designing greedy algorithms is a viable approach.

1.2 Typical Algorithms for Interference Minimization

Numerous researchers have studied the interference minimization problem and proposed relevant algorithms. This paper primarily introduces the following algorithms.

1.2.1 Nearest Neighbor Algorithm (NNA) The basic idea of greedy algorithms is to start from an initial solution and proceed step by step, ensuring a locally optimal solution at each step based on some optimization measure. In interference minimization, it is intuitive to assume that smaller node transmission radii result in lower network interference. The nearest neighbor algorithm is based on this intuitive property and incorporates minimum spanning tree concepts by adjusting the transmission radii of the two closest nodes to connect currently disconnected components in the network. This step is repeated until the entire network becomes connected.

1.2.2 Best Neighbor Algorithm (BNA) While the nearest neighbor algorithm performs adequately in general networks, its effectiveness is poor in special networks (exponential chain networks). To address this limitation, the literature [10] proposed an improved version called the best neighbor algorithm. The main idea is to connect two disconnected components by adjusting the transmission radii of the node pair that causes the smallest increase in the current network' s maximum interference. This process is repeated until the entire network is connected.

The best neighbor algorithm has comparable time complexity to the nearest neighbor algorithm. However, extensive experimental results show that BNA not only eliminates the drawbacks of NNA in special networks but also outperforms NNA in general networks.

1.2.3 Regret Greedy Algorithm (RGA) Sun [12] proposed the regret greedy algorithm in 2017. Its main steps follow the greedy algorithm workflow, attempting to apply regret strategies at each iteration. In each iteration of the greedy algorithm, if the current network' s maximum interference changes (i.e., the maximum interference of the derived graph increases), the algorithm applies a regret strategy to adjust the transmission radii of certain nodes that have already been determined by the greedy algorithm, targeting the node with maximum interference to reduce its interference and thereby decrease the current maximum interference.

1.2.4 Carousel Greedy Algorithm (CGA) The carousel greedy algorithm based on basic greedy algorithm G selects a certain proportion of elements from the feasible solution obtained by G as a partial solution. At each step, it removes the first element of the partial solution and re-adds elements through greedy algorithm G to update the partial solution. This process is repeated until the iteration ends. Finally, greedy algorithm G is used to obtain a complete solution.

1.2.5 Set Cover Algorithm Based on Partial Membership (SCA) Kuhn et al. proposed the minimum membership set cover problem in 2005 and applied it to interference problems in cellular networks. In literature [9], Aslanyan et al. extended this to minimum partial membership set cover in 2010, applying it to wireless network interference minimization and theoretically proving an interference upper bound of $O(\text{opt}^2 \times \ln^2 n)$ and an approximation ratio of $O(\text{opt} \times \ln^2 n)$.

The partial membership set cover is defined as follows: Given an n-element set $S = S \quad S$ and a collection $C = \{C \,,\, C \,,\, \cdots,\, C\}$ of subsets of S, where $S \quad S$ $=$, $|S| = n$, and $|S| = n$, find a collection $C \quad C$ satisfying: (1) C covers all elements in S , expressed as $S \quad\; \{C \quad C\}$ C *; and (2)* C *minimizes the maximum coverage count of elements in* S *, i.e., minimizes* $max\{u \quad S\}$ $M(u, C)$, where the coverage count of element $u \quad S$ by collection C is defined as the membership value $M(u, C) = |\{C \quad C : u \quad C\}|$.

The problem is then formulated as an integer programming problem and solved using linear programming relaxation techniques. Therefore, C can be viewed as a solution to the problem, where for each $C \quad C$, a variable $x \quad \{0, 1\}$ is assigned, with $x = 1$ indicating $C \quad C$. The objective is to minimize z subject to *{*C C*,* u C *}* x *1 for all* u S *and* $\{C \quad C, u \quad C\}$ $x \quad z$ for all $u \quad S$, with $x \quad \{0, 1\}$ for all $C \quad C$.

The specific steps are: (a) Initialize graph $G = \quad V, E$ with $E = \quad$ and $l = 0$; (b) Increment l by 1, identify the corresponding S , S , and C for the problem, and use linear programming; (c) Adjust radii on edges in C , update $E = E$ C , and update graph $G = \quad V, E$; (d) Repeat steps (b) and (c) until the graph becomes connected.

---

2 Edge Shrinking Greedy Algorithm

2.1 Basic Idea

The edge shrinking greedy algorithm (hereinafter referred to as the shrinking algorithm) starts from the derived connected graph $G = \quad V, E$ with maximum transmission radius R. It gradually shrinks the transmission radii of nodes in the graph to find a connected subgraph $G = \quad V, E$ with lower interference. The algorithm initializes all nodes to their maximum threshold radii, i.e., $r(i) = R[i]$ for $i \quad V$, ensuring the initial derived graph is connected. While maintaining connectivity, the algorithm reduces nodes' transmission radii according to a shrinking strategy until no further reduction is possible. In contrast, existing algorithms initialize all node radii to 0 and gradually expand certain nodes' radii to connect them to the current connected component—essentially a "edge addition" approach. The shrinking algorithm can thus be viewed as continuously "removing edges" from the current derived connected graph to reduce interference.

Since the final derived graph must remain connected, the primary condition for selecting nodes for edge shrinking is that connectivity must not be destroyed. Based on this condition, the intuitive selection should favor nodes with large out-degree whose maximum interference points lie on the periphery of their coverage circles. Large out-degree suggests good connectivity, so reducing such a node' s transmission radius has minimal impact on overall connectivity. Meanwhile, having the maximum interference point on the coverage periphery means a slight radius reduction can eliminate interference to that point with minimal impact on connectivity. However, considering either "large out-degree" or "maximum interference point on periphery" alone can lead to extreme cases that severely affect connectivity. To avoid these extremes, we comprehensively consider both factors when selecting shrinking nodes.

Let u be the current maximum interference point, $N(u)$ be the set of nodes that interfere with u, i.e., $N(u) = \{v \quad V \mid r \quad d(u, v)\}$, and $l(u, v)$ be the rank of u when sorted by distance from v among all nodes in $N(u)$, where $v \quad N(u)$. The selected shrinking point is:

[Figure 2: see original paper] illustrates the rationale behind selecting shrinking points using equation (3). Assuming u is the current maximum interference point with interference value 3 (for clarity, all points within transmission range are placed on the same line), points A, B, and C are the three nodes interfering with u. Node B has the largest out-degree, while u lies on the periphery of node C' s transmission range. Shrinking either node B or C would disconnect the shrinking node from only one other node, destroying graph connectivity. According to equation (3), point A is selected, and shrinking its transmission radius preserves connectivity. Choosing either B or C would severely damage connectivity.

2.2 Algorithm Description

This section describes and analyzes the shrinking algorithm, comparing solution quality across different algorithms on identical test cases.

Algorithm 1 presents the pseudocode for the shrinking algorithm.

Algorithm 1: Edge Shrinking Algorithm

Input: Node set $V = \{1, 2, \cdots, n\}$ with known planar coordinates; for each point $i \quad V$, a maximum transmission radius threshold function $R \quad [0, 1]$ such that graph $G \; = (V, E \;)$ is connected.

Output: Transmission radius vector $r = (r \;, r \;, \cdots, r \;)$ such that graph $G \; = (V, E \;)$ is connected with relatively low interference.

Begin 1. $r \; = R$ for $i \quad \{1, 2, \cdots, n\}$ 2. Do 3. $u \leftarrow$ current point with maximum interference degree 4. For $v \quad V$, $N(u) \leftarrow \{v \quad V \mid r \quad d(u, v)\}$ 5. While $N(u)$ 6. If graph remains connected after shrinking v then 7. Do update r 8. Else

9. $N(u) \leftarrow N(u)$ $\{v\}$ 10. End if 11. End while 12. While interference of u decreases 13. End while

The transmission radius of a shrinking point is reduced from r to $d(u, v)$ - , where is an arbitrarily small positive real number, and the derived graph G $=$ V, E remains connected.

The following theorem establishes the correctness and time complexity of the shrinking algorithm.

Theorem 1 The algorithm produces a transmission radius function r in $O(n^3 \times d)$ time, and the derived graph $G = V, E$ is connected.

Proof: First, we prove the time complexity. The algorithm' s first step finds the current maximum interference node u, requiring $O(n)$ time. The second step identifies all nodes whose out-degree includes u, also requiring $O(n)$ time. Finally, to ensure graph connectivity, each radius reduction requires connectivity verification, taking $O(n)$ time. Since these steps are repeated until no further reduction is possible, and the number of repetitions d is generally much smaller than n, the overall time complexity is $O(n^3 \times d)$.

Second, we prove connectivity preservation. The algorithm ensures the graph is connected upon initialization, and each selected node for radius reduction must satisfy the primary condition of not destroying connectivity. Therefore, the chosen points do not affect graph connectivity, and the algorithm produces a connected derived graph $G = V, E$.

2.3 Example Analysis

Example 1 Consider the execution of the shrinking algorithm on a one-dimensional exponential chain network instance with $n = 6$. Upon initialization, all nodes connect to their farthest neighbors, as shown in Figure 3(a), where arcs represent each node' s farthest reachable node.

At this stage, all six nodes have interference degree 5, making them all maximum interference points. When multiple nodes have maximum interference, we process them in order starting with the smallest index. The first node requiring interference reduction is v , and the selected shrinking node is v . Subsequently, nodes requiring reduction are v , v , and v , with corresponding shrinking nodes v , v , and v . After these operations, node transmission radii appear as shown in Figure 3(b). Following this series of shrinkings, both v and v have reduced their transmission radii, decreasing the current maximum interference to 4.

Repeating these steps until no further shrinking is possible yields the final result shown in Figure 3(c). In Figure 3(c), v , v , v , v , and v cover only their two nearest neighbors (except endpoints), while v covers its farthest neighbor v . The derived graph is clearly connected with maximum interference degree 2 and average interference degree 1.83.

Example 2 Consider a two-dimensional case with $n = 10$ randomly generated points. The positions of points v through v are shown in Figure 4, with coordinates (1.11, 0.17), (1.64, 1.84), (1.92, 1.72), (0.46, 0.79), (1.94, 0.81), (0.29, 1.84), (0.24, 1.30), (0.60, 1.00), (1.56, 0.37), and (1.54, 2.26).

For this node set $V = \{1, 2, \cdots, 10\}$ with known planar coordinates, the maximum transmission radius threshold function ensures $G = (V, E)$ is connected. The shrinking algorithm initializes by setting each node's transmission radius to reach its farthest connectable node, as shown in Figure 5, where arrows indicate each node' s farthest reachable node.

The transmission radius changes at each iteration of the shrinking algorithm are as follows: - (a) $r(1)$: $0.91 \to 0.89$ - (b) $r(8)$: $0.91 \to 0.89$ - (c) $r(10)$: $0.66 \to 0.43$ - (d) $r(7)$: $0.55 \to 0.54$ - (e) $r(8)$: $0.89 \to 0.25$ - (f) $r(6)$: $0.89 \to 0.54$

The final result is shown in Figure 6. For clarity, only the farthest nodes reachable within each node's final transmission radius are marked. The derived graph is connected, with multiple maximum interference points: v, v, v, v, v, v, v, and v, each having interference value 2.

On this test case, the "edge addition" algorithms mentioned earlier all produced identical results with maximum interference 3, as shown in Figure 7 [Figure 7: see original paper].

Comparing Figures 6 and 7 reveals that the only difference is that the shrinking algorithm sets $r(8) = 0.25$, while the "edge addition" algorithm sets $r(8) = 0.47$, leading to different maximum interference values. Thus, on this instance, the shrinking algorithm yields a better solution than the edge addition approach.

---

3 Acceleration Strategy

3.1 Acceleration Approach

Numerous enhancement strategies for greedy algorithms exist, such as iterated greedy algorithms applied to permutation flow shop scheduling problems, and randomized greedy algorithms applied to MLST problems by Xiong et al. [15,16]. These two types are easy to implement but risk destroying graph connectivity through randomization in the MIP context. Recently, Carmine et al. proposed the carousel greedy algorithm, which is easy to implement and has successfully enhanced greedy algorithms for typical problems [17]. Chapter 2 explained the carousel greedy algorithm, and experiments were conducted for comparison.

This paper considers an enhancement strategy combining batch processing to improve the shrinking algorithm. Batch processing enhancement executes multiple steps simultaneously, similar to the fundamental concept of batch processing in computing, which reduces algorithm running time and broadens applicability.

Factors affecting the shrinking algorithm' s running time consist of four main components: (a) finding the current maximum interference point; (b) identifying all nodes whose out-degree contains the current maximum interference point; (c) shrinking edges and verifying graph connectivity; and (d) repeating these steps until no shrinkable nodes remain.

Regarding (a), since the maximum interference node in the current connected graph changes globally after one shrinking operation, a full graph traversal is required each time. Regarding (b), indexing could be employed, but this requires matrix storage, increasing space complexity by $O(n^2)$, and matrix updates after each shrinking operation make optimization difficult. Therefore, we optimize components (c) and (d) by minimizing connectivity verification operations and accelerating maximum interference convergence.

From the algorithm description, connectivity verification occurs after shrinking a single node' s transmission radius. Extensive experiments reveal that multiple points often share the current maximum interference degree simultaneously, meaning multiple nodes could undergo shrinking operations. However, the original shrinking algorithm processes these sequentially, requiring connectivity verification after each operation and causing excessive running time. This paper investigates whether batch processing concepts can be integrated: performing shrinking operations on all nodes with current maximum interference simultaneously, then verifying connectivity once. Algorithm 2 presents the pseudocode for this batch shrinking approach.

Algorithm 2: Batch Edge Shrinking Algorithm

Input: Node set $V = \{1, 2, \cdots, n\}$ with known planar coordinates; for each point i V, a maximum transmission radius threshold function R [0, 1] such that graph $G = (V, E)$ is connected.

Output: Transmission radius vector $r = (r , r , \cdots, r)$ such that graph $G =$ (V, E) is connected with relatively low interference.

Begin 1. $r = R$ for i $\{1, 2, \cdots, n\}$, VI $=$ 2. Do 3. I $\leftarrow$ current maximum interference of derived graph 4. For each v V do 5. If $I = I$ then 6. VI $\leftarrow$ VI $\{v\}$ 7. End if 8. End for 9. If graph remains connected after shrinking all points in VI then 10. Update radii of all shrinking points 11. Else 12. Apply the unimproved shrinking algorithm to u VI 13. End if 14. While I decreases 15. End while

3.2 Accelerated Experimental Results

To demonstrate the effectiveness of this improvement, we conducted comparative experiments on the batch shrinking algorithm, with results shown in Table 3 and Figure 9 [Figure 9: see original paper]. In test cases (n, d), n and d represent the number of nodes and average degree, respectively.

The tables and figures show that while the batch shrinking algorithm' s solution quality is slightly inferior to the original shrinking algorithm, the difference is negligible and still superior to other algorithms. In terms of running time, although the batch acceleration strategy cannot reduce the time complexity from $O(n^3)$ by an order of magnitude, the actual running time is far lower than the original shrinking algorithm. This makes the algorithm more practical while maintaining superior solution quality within comparable timeframes.

4 Conclusion

Under the receiver-centric interference model, this paper proposes an edge shrinking greedy algorithm for interference minimization in wireless sensor networks based on topology control. The algorithm employs a different graph construction method from existing approaches, constructing the network by shrinking edges on the derived graph where each node' s transmission radius is at maximum threshold. This approach produces better solutions on most randomly generated test cases. To address its long running time, we incorporated batch processing strategies, significantly reducing execution time and enabling the algorithm to obtain superior results within timeframes comparable to other algorithms.

References

[1] Sun Li-mining, Li Jian-zhong, Chen Yu. *Wireless Sensor Networks* [M]. Beijing: Tsinghua University Press, 2005: 3-23.

[2] Bilò D, Proietti G. On the complexity of minimizing interference in ad-hoc and sensor networks [J]. *Theoretical Computer Science*, 2008, 402(1): 43-55.

[3] Akyildiz I F, Su Weilian, Sankarasubramaniam Y, et al. Wireless sensor networks: a survey [J]. *Computer Networks*, 2002, 38(4): 393-422.

[4] Lou Tiancheng, Tan Haisheng, Wang Yuexuan, et al. Minimizing average interference through topology control [C]// Proc of International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics. Berlin: Springer, 2011, 7111: 115-129.

[5] Buchin K. Minimizing the maximum interference is hard [J]. arXiv: 0802.2134v2, 2008.

[6] Von Rickenbach P, Schmid S, Wattenhofer R, et al. A robust interference model for wireless ad-hoc networks [C]// Proc of the 19th IEEE International Parallel and Distributed Processing Symposium. Piscataway, NJ: IEEE Press, 2005, 13(4): 239-246.

[7] Benkert M, Gudmundsson J, Haverkort H, et al. Constructing minimum-interference networks [J]. *Computational Geometry*, 2008, 40(3): 179-194.

[8] Von Rickenbach P, Wattenhofer R, Zollinger A. Algorithmic models of interference in wireless ad hoc and sensor networks [J]. *IEEE/ACM Trans on Networking*, 2009, 17(1): 172-185.

[9] Halldorsson M M, Tokuyama T. Minimizing interference of a wireless ad-hoc network in a plane [J]. *Theoretical Computer Science*, 2008, 402(1): 29-42.

[10] Zhang Xiaoke, Chen Weidong. An effective algorithm for interference minimization in wireless sensor networks [C]// Proc of International Conference on Intelligent Computing. [S.l.]: Springer International Publishing, 2014: 572-581.

[11] Hassin R, Levin A. A better-than-greedy approximation algorithm for the minimum set cover problem [J]. *SIAM Journal on Computer*, 2005, 35(1): 189-200.

[12] Sun Peixin. A regret greedy algorithm for the interference minimization problem in wireless sensor networks [J]. *Computer Engineering and Science*, 2017, 39(12): 2217-2223.

[13] Shetty D P, Lakshmi M P. Algorithms for minimizing the receiver interference in a wireless sensor network [C]// Proc of IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics. 2016: 1-4.

[14] West D B. *Introduction to Graph Theory* [M]. Upper Saddle River, NJ: Prentice-Hall, 1996.

[15] Karabulut K. A hybrid iterated greedy algorithm for total tardiness minimization in permutation flowshops [J]. *Computers & Industrial Engineering*, 2016, 98: 300-307.

[16] Xiong Yupei, Golden B, Wasil E. Improved heuristics for the minimum label spanning tree problem [J]. *IEEE Trans on Evolutionary Computation*, 2006, 10(6): 700-703.

[17] Cerrone C, Cerulli R, Golden B. Carousel greedy: a generalized greedy algorithm with applications in optimization [J]. *Computers & Operations Research*, 2017, 85: 97-112.

Note: Figure translations are in progress. See original paper for figures.