Fixed parameter tractable algorithms for corridor guarding problems

R. Subashini

Joint work with
Remi Raman
Subhasree Methirumangalath
NIT CALICUT

Recent Trends in Algorithms
National Institute of Science Education and Research(NISER)

Outline

Geometric Covering problems

- Motivated by the applications in VLSI design, and motion planning, geometric covering problems have been studied extensively.
- One has to cover geometric objects (e.g., points, lines, disks, squares or rectangles) with other geometric objects, satisfying some optimization requirements.

Motivation

- Applications in VLSI
  - Minimize the length of the wire used
  - Reduce the number of links(bends) in a path connecting two points in the board
- Most of the covering problems are NP-hard even in rectilinear domains(lines/line-segments parallel to x-axis or y-axis)[1]

[1] Jianxin Wang, Jinyi Yao, Qilong Feng, and Jianer Chen.Improved fpt algorithms for rectilinear k-links spanning path.In International Conference on Theory and Applications of Models of Computation, Springer,2012

Corridor Guarding problems

- Minimum corridor guarding problems (CMST/CTSP)
- Minimum link CTSP
- Minimum corridor connection problems

Minimum corridor guarding problems [3]

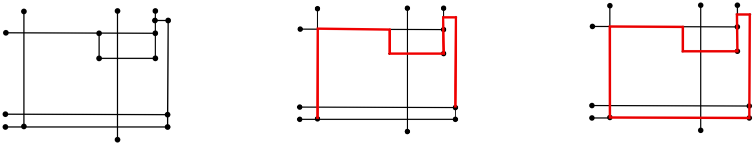
- Input: Connected orthogonal arrangement of line-segments
  Output: An optimal tree/closed walk, such that if a guard moves through the tree/closed walk, all the line-segments are visited[2] by the guard.
- If the guarding walk is a tree/closed walk, then the problem is referred to as Corridor-MST/Corridor-TSP(CMST/CTSP)
- Decision version of CMST/CTSP is proved to be NP-Complete.

---

[2] a line-segment l is said to be visited by a tree/walk, if any of the vertices in the tree/walk is incident to one of the endpoints or intersection points created by l with other line-segments

[3] Ning Xu.Complexity of minimum corridor guarding problems.Information Processing Letters, 2012.

Minimum corridor guarding problems



(a) (b) (c)

Figure: (a) represents input instance of CMST and CTSP. Red lines in (b) and (c) represent the tree and closed walk respectively

Minimum link CTSP

- Given an orthogonal connected arrangement L of line-segments, find a minimum link-distance closed walk visiting all the line-segments.
- Link-distance is the number of links or turns in a path/walk.

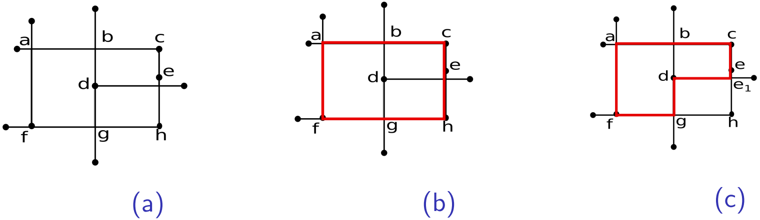


Figure: Input and Output Instances of MLC.
(a) The input arrangement of line-segments.
(b) closed walk in (a) with link-distance four(ac, ch, hf, and fa are the links)
(c) closed walk in (a) with six link-distance (ac, ce_1, e_1d, dg, gf and fa are the links) respectively.

Minimum corridor connection problems [4]

- Given a rectilinear polygon partitioned into rectilinear components or rooms, MCC asks for a minimum length tree along the edges of the partitions, such that every room is incident to at least one vertex of the tree.
- Decision version of the problem is shown to be NP-complete.

(a) (b)

Figure: Input and Output instances of MCC. (a) Rectilinear polygon partitioned into rooms. In (b) the red lines represent a minimal tree visiting all rooms

[4] Hans L Bodlaender et al.On the minimum corridor connection problem and other generalized geometric problems.Computational Geometry, 42(9), 2009.

Parameterized Complexity [5]

- A framework for solving NP-hard problems by measuring their time in terms of one or more parameters, in addition to the input size.
- A problem with input instance of size n, and with a non-negative integer parameter k, is *fixed-parameter tractable*(FPT), if it can be solved by an algorithm that runs in $O(f(k).n^c)$-time, where f is a computable function depending only on k, and c is a constant independent of k.

[5] Rolf Niedermeier.Invitation to fixed-parameter algorithms.2006

k-CMST/k-CTSP(k-Corridor-MST/k-Corridor-TSP)

Input: A connected arrangement of line-segments (corridors) $L = \{L_1, L_2, \ldots, L_n\}$, and an integer k
Parameter: k
Output: A minimum length tree/closed walk on at most k vertices, along the edges of the corridor, such that all the line-segments are visited.

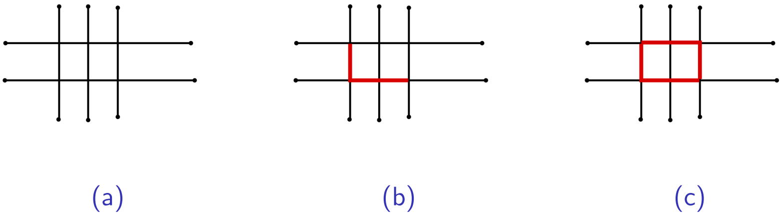
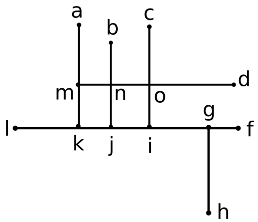


Figure: Red lines in (b) shows tree with k=4 and Red lines in (c) shows closed walk with $k = 6$ for input instance (a)

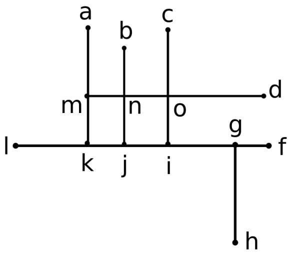
An FPT algorithm for k-CMST/k-CTSP

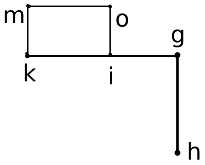
Input : Orthogonal Arrangement of line-segments



- Segment Vertices $V_s = \{ a, b, c \ldots o\}$ and Segment Edges $E_s = \{ am, bn, co, od, gf, lk, mn, no, kj, ji, mk, nj, oi, gh, ig\}$
- Isolated segment edges $E_{is} = \{ am, bn, co, od, gf, lk\}$
- Segment bounding rectangle: Rectangle formed by the set of topmost and bottommost horizontal line-segments , and leftmost and rightmost vertical line-segments when two or more horizontal(vertical) line-segments is intersected by three or more vertical(horizontal) line-segments. ($[mo, oi, ik, km]$ in the figure).

An FPT algorithm for k-CMST/k-CTSP

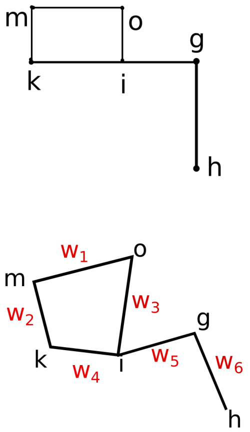




- Preprocess the input instance
  - Remove isolated-segment edges if any.
  - Remove those line segments which have both their end-points in the boundary of a segment-bounding rectangle, if any.

Parameter k is decreased by the number of line-segments removed. The updated parameter is referred to as l.

An FPT algorithm for k-CMST/k-CTSP



- Transform the preprocessed instance to graph instance G_{ls}.
  - The segment vertices and edges of the preprocessed instance is transformed into vertices and edges of the graph G_{ls}.
  - Length of the segment-edges are assigned as the weights of the corresponding edges in the graph.
- Find l-Tree cover and l-Tour cover of the graph instance

l-Tree cover/l-Tour cover (Weighted connected vertex cover)

- Input : A graph $G = (V, E, w)$ where $w : E \rightarrow \mathbb{R}^+$, an integer $l \geq 0$. Parameter : l, Number of vertices in the output tree/closed walk
  Output: A minimal Tree/closed walk $T = (V', E')$ of G with $V' \subseteq V$ and $E' \subseteq E$, $|V'| \leq l$ and V' is a vertex cover for G.
- Both l-Tree Cover and l-Tour Cover were shown to be FPT.

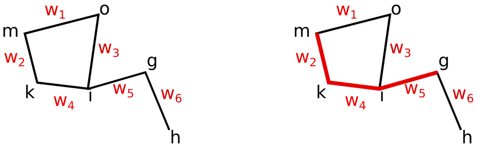


Figure: Red lines in (b) shows tree-cover with k=4 for graph in (a).

FPT result of k-CMST/k-CTSP

Lemma

l-Tree Cover and l-Tour Cover can be solved in $O((2l)^l)$ and $O((4l)^l)$-time, respectively.[a]

[a] Jiong Guo, Rolf Niedermeier, and Sebastian Wernicke. Parameterized complexity of generalized vertex cover problems. In Workshop on Algorithms and Data Structures, pages 3648. Springer, 2005.

Lemma

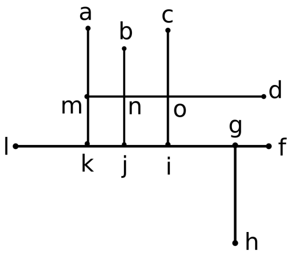
k-CMST/k-CTSP on an input instance (L', l) is an YES-instance iff l-Tree Cover/l-Tour Cover in its corresponding G_{ls} has an YES-instance.

Theorem

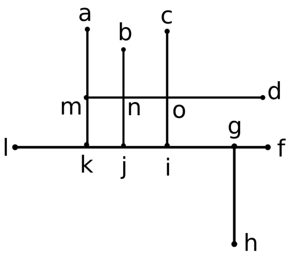
k-CMST and k-CTSP on an arrangement L is FPT with a run-time of $O^*(2k^k)$ and $O^*(4k^k)$ respectively.

An improved FPT algorithm for k-CMST/k-CTSP

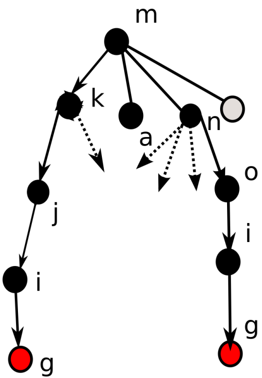
- Consider the geometric instance.
- Uses a search tree which starts with a segment-vertex with segment-degree ≥ 2.
- Each node has 4 branches, and each branch selects one segment edge.
- Branching is performed until all the line-segments in the arrangement are visited, S is a tree/closed walk and $k \geq 0$.



An improved FPT algorithm for k-CMST/k-CTSP



(a)



(b)

Figure: m is the start vertex. $m - k - j - i - g$ and $m - n - o - i - g$ are two trees with $k = 5$ vertices

An improved FPT algorithm for k-CMST/k-CTSP

- Initially, if we select a vertex which is not part of the tree/closed walk, the branching algorithm may return a NO, even when the input is a YES instance.

Lemma

If there is line-segment l in L intersected by more than k line-segments, then the instance (L, k) is a NO instance for k-CMST. If l is intersected by more than $k/2$ line-segments, then the instance is a NO instance for k-CTSP.

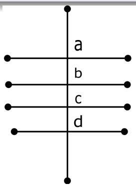


Figure: For $k < 4$ k-CMST returns a NO, and for $k < 8$ k-CTSP returns a NO

An improved FPT algorithm for k-CMST/k-CTSP

Corollary

The maximum intersections possible for a line-segment l in a YES instance of k-CMST is k, and k/2 for k-CTSP.

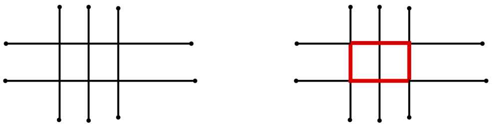
- The algorithm is invoked for a maximum of k times for k-CMST and $k/2$ times for k-CTSP (Maximum number of intersections is k and $k/2$ respectively).
- Running time : O*($k.4^k$)

Theorem

There is an $O^*(k.4^k)$-time algorithm for k-CMST and $O^*((k/2).4^k)$-time algorithm for k-CTSP. Consequently, these problems are FPT.

b-MLC (b-Minimum link Corridor-TSP)

Input: A connected arrangement of line-segments (corridors) $L = \{L_1, L_2, \ldots, L_n\}$ with bounded number of intersections m for every line-segment in L & an integer b
Parameter: b
Output: A minimum length closed walk on at most b link-distance along the edges of the corridor, such that all the line-segments are visited.



(a) (b)

Figure: Red lines in (c) shows closed walk with $b = 4$ for input instance (a)

Hardness result of b-MLC

Theorem

b-MLC is NP-complete.

Candidate problem: Point covering rectilinear tour of b links or b link point-tour.
Input: A set of n points in a plane
Question: Is there a rectilinear tour of at most b link-distance which covers all the points?

- b-link point tour is proven to be NP-Complete[6].

[6] Jianxin Wang, Jinyi Yao, Qilong Feng, and Jianer Chen.Improved fpt algorithms for rectilinear k-links spanning path.In International Conference on Theory and Applications of Models of Computation, pages 560571. Springer,2012

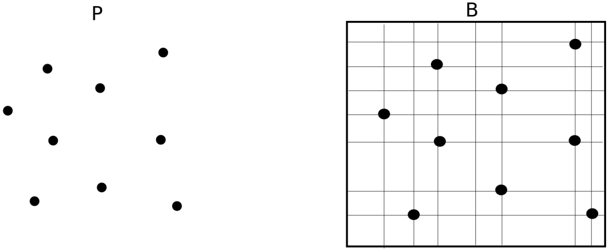


Figure: Example of reduction from point covering by a b-link tour to b-MLC.

- Enclose the points in a rectangular bounding box and build an orthogonal line arrangement of the points.
- The endpoints in the line-segments of b-MLC is either one of the original n points, or the intersection points made by the lines with the bounding box.

Hardness result of b-MLC

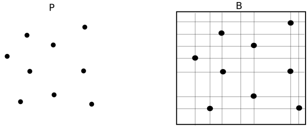


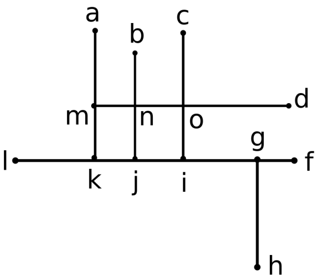
Figure: Example of reduction from point covering by a b-link tour to b-MLC.

- Every point in the input of point covering corresponds to four line-segments in b-MLC.
- It is obvious from the construction, that each of the line-segments share one of its endpoints with at least one of the n points.
- So, if there is a b-link tour connecting the n points, then there is a closed walk visiting all $4n$ line-segments with at most b link-distance.
- The decision version of the problem is in NP, the verifying algorithms checks if a sequence of line-segments forms a closed walk, visits all the line-segments, and has at most b link-distance.
- b-MLC is NP-Complete.

An FPT algorithm for b-MLC

- Uses a search tree
- Each node has 4(m+1) branches where m is the bound in number of intersections in one line-segment, and each branch selects one link.
- Branching is performed until all the line-segments in the arrangement are visited, S is a closed walk and $b \geq 0$.
- Initially, if we select a vertex which is not part of the closed walk, the branching algorithm may return a NO, even when the input is a YES instance.

An FPT algorithm for b-MLC



- The maximum number of intersections for a line-segment in the figure is 4.
- Suppose we start with the vertex l, the possible links are lk, lj, li, lg, and lf .
- If we start with m, one of the solutions is $m - o - i - g$ with 3 link-distance.

An FPT algorithm for b-MLC

- The algorithm is invoked for a maximum of m times since the maximum bound on intersection is m.
- Running time : $O(m.(4(m+1))^b)$

Input: A rectilinear polygon P partitioned into $\{P_1, P_2, \ldots, P_k\}$ rectilinear components or rooms.
Parameter: k, The number of partitions or rooms.
Output: A minimum length tree along the edges of the partitions such that all k rooms are visited.[7]

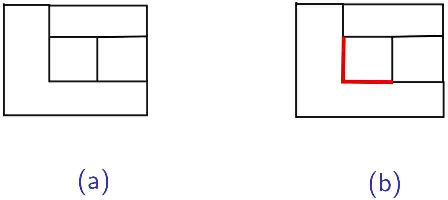
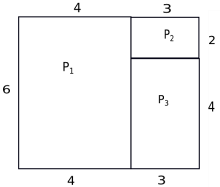


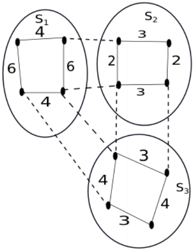
Figure: Red lines in (b) shows tree with k=4 for the input in (a).

[7] A room is said to be visited by a tree when it is incident to one of the vertices of the tree.

An FPT algorithm for k-MCC

Transform the input instance to a graph instance where the vertices are divided as k groups of terminals.





- Corresponding to each of the partitions $\{P_1, P_2, \ldots, P_k\}$ in P, group of terminals $S_1, S_2, \ldots, S_k$ in G_{pd} is created.
- Edge weights in G_{pd} are added corresponding to the length of the line-segments in the partitions of P.
- The dotted lines corresponds to the 0 weight edges which are added between vertices shared by partitions.

FPT algorithm for k-edgewt-Group Steiner tree

- In G_{pd}, find a group Steiner tree visiting all k groups.

k-edgewt-GST
Input: A connected undirected graph $G = (V, E, w)$ where $w : E \rightarrow \mathbb{R}^+$, vertex-disjoint subsets $\{S_1, S_2, \ldots, S_k\}$ where each $S_i \subseteq V \; \forall \; 1 \leq i \leq k$.
Parameter: k
Output: A minimal tree in G that includes at least one vertex from each $S_i \; \forall \; 1 \leq i \leq k$.

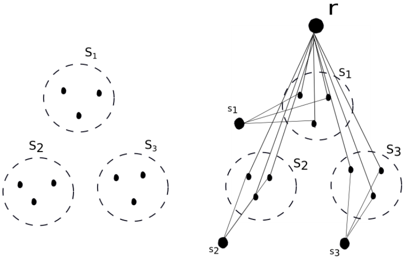
FPT algorithm for k-edgewt-Group Steiner tree

- Reduce k-edgewt-GST to k-edgewt-DST.
  k-edgewt-DST
  Input: A Directed graph $G' = (V', E', w')$ where $w^{'} : E^{'} \rightarrow \mathbb{R}^{+}$, a distinguished vertex r $\in V$, a set of terminals $S \subseteq V$ where $|S| = k$.
  Parameter: k
  Output: A minimal out-tree in G' that is rooted at r and that contains all the vertices of S.

Lemma

k-edgewt-GST has a parameter preserving reduction to k-edgewt-DST.

Transformation of instance from weighted GST in G to weighted DST



- Additional $k+1$ vertices $\{s_1, s_2, \ldots, s_k, r\}$ are included in DST instance.
- For each edge (u, v) in G, edges (u, v) and (v, u) with the same edge weights is added in D.
- An arc of length 1 is added from r to all vertices in S_i.
- An arc of length 1 is added from vertices of S_i to corresponding s_i, $\forall\ 1 \leq i \leq k$.

Transformation of instance from weighted GST in G to weighted DST

- If G contains a tree T with minimal edge-weight m that includes at least one vertex from each S_i, then this tree with the same weight m is also contained in D which can be accessed from r using one of the (r, u) arc for some $u \in V$.
- Thus we have a directed out-tree with edge-weight $(m + k + 1)$ containing r and all vertices in S.
- Also, if any one of the group S_i is omitted, then T must omit s_i.
- Thus, there is a parameterized preserving reduction from k-edgewt-GST to k-edgewt-DST.

k-MCC

Theorem

There is a $O^*(2^{O(klogk)})$-time algorithm for k-edgewt-DST.[a]

[a] Fedor V Fomin, Fabrizio Grandoni, Dieter Kratsch, Daniel Lokshtanov, and Saket Saurabh. Computing optimal steiner trees in polynomial space. Algorithmica, 2013

Theorem

k-MCC is solved in $O^*(2^{O(klogk)})$-time. Consequently, it is FPT.

Summary

Problem	Complexity Status	FPT results
k-CMST	NP-Complete [Xu12]	$O^*(2k^k)$, $O^*(k(4^k))$
k-CTSP	NP-Complete [Xu12]	$O^*(4k^k)$, $O^*((k/2)4^k)$
b-MLC	NP-Complete	$O^*(m(4(m+1))^b)$
k-MCC	NP-Complete[BFG+09]	$O^*(2^k logk)$

Future work

- To incorporate an option of visibility of rooms, in addition to the notion of visiting rooms.

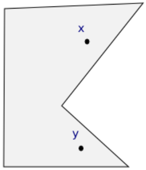


Figure: Notion of visibility: x and y is not visible to each other since the line-segment xy is not completely inside the polygon

- Another direction of work related to MLC problem is finding a tree with minimum number of links or link-diameter(maximum link-distance between any two points in the tree.)

Thank You

References I

[AGN01] Jochen Alber, Jens Gramm, and Rolf Niedermeier.
Faster exact algorithms for hard problems: a parameterized point of view.
Discrete Mathematics, 229(1-3):3–27, 2001.

[AHH93] Esther M Arkin, Magnús M Halldórsson, and Rafael Hassin.
Approximating the tree and tour covers of a graph.
Information Processing Letters, 47(6):275–282, 1993.

[BFG+09] Hans L Bodlaender, Corinne Feremans, Alexander Grigoriev, Eelko Penninkx, René Sitters, and Thomas Wolle.
On the minimum corridor connection problem and other generalized geometric problems.
Computational Geometry, 42(9):939–951, 2009.

[Bod98] Hans L Bodlaender.
A partial k-arboretum of graphs with bounded treewidth.
Theoretical computer science, 209(1-2):1–45, 1998.

[CFK+15] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh.
Parameterized algorithms, volume 3.
Springer, 2015.

[CG89] Bernard Chazelle and Leonidas J Guibas.
Visibility and intersection problems in plane geometry.
Discrete & Computational Geometry, 4(6):551–581, 1989.

[CN86] Wei-Pang Chin and Simeon Ntafos.
Optimum watchman routes.
In *Proceedings of the second annual symposium on Computational geometry*, pages 24–33. ACM, 1986.

[Cyg12] Marek Cygan.
Deterministic parameterized connected vertex cover.
In *Scandinavian Workshop on Algorithm Theory*, pages 95–106. Springer, 2012.

[DBVKOS97] Mark De Berg, Marc Van Kreveld, Mark Overmars, and Otfried Schwarzkopf.
Computational geometry.
In *Computational geometry*, pages 1–17. Springer, 1997.

[DELM03] Moshe Dror, Alon Efrat, Anna Lubiw, and Joseph SB Mitchell.
Touring a sequence of polygons.
In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 473–482. ACM, 2003.

[Deo17] Narsingh Deo.
Graph theory with applications to engineering and computer science.
Courier Dover Publications, 2017.

[DF13] Rodney G Downey and Michael R Fellows.
Fundamentals of parameterized complexity, volume 4.
Springer, 2013.

[DO00] Erik D Demaine and Joseph ORourke.
Open problems from cccg99.
In *Proc. 12th Canad. Conf. Comput. Geom*, pages 269–272, 2000.

[DT12] Adrian Dumitrescu and Csaba D Tóth.
Watchman tours for polygons with holes.
Computational Geometry, 45(7):326–333, 2012.

[DVV04] CW Duin, A Volgenant, and Stefan Voß.
Solving group steiner problems as steiner problems.
European Journal of Operational Research, 154(1):323–329, 2004.

[DW71] Stuart E Dreyfus and Robert A Wagner.
The steiner problem in graphs.
Networks, 1(3):195–207, 1971.

References III

[ECHS10a] Vladimir Estivill-Castro, Apichat Heednacram, and Francis Suraweera.
Np-completeness and fpt results for rectilinear covering problems.
J. UCS, 16(5):622–652, 2010.

[ECHS10b] Vladimir Estivill-Castro, Apichat Heednacram, and Francis Suraweera.
The rectilinear k-bends tsp.
In *International Computing and Combinatorics Conference*, pages 264–277. Springer, 2010.

[FG06] Jörg Flum and Martin Grohe.
Parameterized complexity theory.
Springer Science & Business Media, 2006.

[FGK+13] Fedor V Fomin, Fabrizio Grandoni, Dieter Kratsch, Daniel Lokshtanov, and Saket Saurabh.
Computing optimal steiner trees in polynomial space.
Algorithmica, 65(3):584–604, 2013.

[FLL03] Corinne Feremans, Martine Labbé, and Gilbert Laporte.
Generalized network design problems.
European Journal of Operational Research, 148(1):1–13, 2003.

[Gan99] Joseph L Ganley.
Computing optimal rectilinear steiner trees: A survey and experimental evaluation.
Discrete Applied Mathematics, 90(1-3):161–171, 1999.

[GGJ76] Michael R Garey, Ronald L Graham, and David S Johnson.
Some np-complete geometric problems.
In *Proceedings of the eighth annual ACM symposium on Theory of computing*, pages 10–22. ACM, 1976.

[GJ02] Michael R Garey and David S Johnson.
Computers and intractability, volume 29.
wh freeman New York, 2002.

References IV

[GNW05] Jiong Guo, Rolf Niedermeier, and Sebastian Wernicke.
Parameterized complexity of generalized vertex cover problems.
In *Workshop on Algorithms and Data Structures*, pages 36–48. Springer, 2005.

[Hee10] Apichat Heednacram.
The NP-hardness of covering points with lines, paths and tours and their tractability with FPT-algorithms.
Griffith University, 2010.

[HK14] Mathias Hauptmann and Marek Karpinski.
A compendium on steiner tree problems (cit. on p. 25).
2014.

[HRW92] Frank K Hwang, Dana S Richards, and Pawel Winter.
The Steiner tree problem, volume 53.
Elsevier, 1992.

[Jia12] Minghui Jiang.
On covering points with minimum turns.
In *Frontiers in Algorithmics and Algorithmic Aspects in Information and Management*, pages 58–69. Springer, 2012.

[KKPS04] Jochen Könemann, Goran Konjevod, Ojas Parekh, and Amitabh Sinha.
Improved approximations for tour and tree covers.
Algorithmica, 38(3):441–449, 2004.

[LYW94] DT Lee, Chung-Do Yang, and CK Wong.
On bends and distances of paths among obstacles in two-layer interconnection model.
IEEE Transactions on Computers, 43(6):711–724, 1994.

[MPR+12] Neeldhara Misra, Geevarghese Philip, Venkatesh Raman, Saket Saurabh, and Somnath Sikdar.
Fpt algorithms for connected feedback vertex set.
Journal of Combinatorial Optimization, 24(2):131–146, 2012.

References V

[MRR08] Daniel Mölle, Stefan Richter, and Peter Rossmanith.
Enumerate and expand: Improved algorithms for connected vertex cover and tree cover.
Theory of Computing Systems, 43(2):234–253, 2008.

[Ned09] Jesper Nederlof.
Fast polynomial-space algorithms using möbius inversion: Improving on steiner tree and related problems.
In *International Colloquium on Automata, Languages, and Programming*, pages 713–725. Springer, 2009.

[Nie06] Rolf Niedermeier.
Invitation to fixed-parameter algorithms.
2006.

[o'R98] Joseph o'Rourke.
Computational geometry in C.
Cambridge university press, 1998.

[PS12] Franco P Preparata and Michael I Shamos.
Computational geometry: an introduction.
Springer Science & Business Media, 2012.

[W+01] Douglas Brent West et al.
Introduction to graph theory, volume 2.
Prentice hall Upper Saddle River, 2001.

[Wag06] David Phillip Wagner.
Path planning algorithms under the link-distance metric.
2006.

[WPN92] Chin Wei-Pang and Simeon Ntafos.
The zookeeper route problem.
Information Sciences, 63(3):245–259, 1992.

References VI

[WTY$^+$14] Jianxin Wang, Peiqiang Tan, Jinyi Yao, Qilong Feng, and Jianer Chen.
On the minimum link-length rectilinear spanning path problem: complexity and algorithms.
IEEE Transactions on Computers, 63(12):3092–3100, 2014.

[WYFC12] Jianxin Wang, Jinyi Yao, Qilong Feng, and Jianer Chen.
Improved fpt algorithms for rectilinear k-links spanning path.
In *International Conference on Theory and Applications of Models of Computation*, pages 560–571. Springer, 2012.

[Xu12] Ning Xu.
Complexity of minimum corridor guarding problems.
Information Processing Letters, 112(17-18):691–696, 2012.