# Parameterized Complexity of Efficient Sortation

**Robert Ganian**
Algorithms and Complexity Group, TU Wien, Austria

**Hung P. Hoang**
Algorithms and Complexity Group, TU Wien, Austria

**Simon Wietheger**
Algorithms and Complexity Group, TU Wien, Austria

**Abstract**

A crucial challenge arising in the design of large-scale logistical networks is to optimize parcel sortation for routing. We study this problem under the recent graph-theoretic formalization of Van Dyk, Klause, Koenemann and Megow (IPCO 2024). The problem asks – given an input digraph $D$ (the *fulfillment network*) together with a set of *commodities* represented as source-sink tuples – for a minimum-outdegree subgraph $H$ of the transitive closure of $D$ that contains a source-sink route for each of the commodities. Given the underlying motivation, we study two variants of the problem which differ in whether the routes for the commodities are fixed or can be chosen arbitrarily.

We perform a thorough parameterized analysis of the complexity of both problems, concentrating on three fundamental parameterizations:

1. When considering the target outdegree of $H$, we show that the problems are paraNP-hard even in highly restricted cases;
2. When parameterizing by the number of commodities, we utilize Ramsey-type arguments and the color-coding technique to obtain fixed-parameter algorithms for both problems;
3. When parameterizing by the structure of $D$, we establish fixed-parameter tractability for both problems w.r.t. the combined parameterization of treewidth, maximum degree and the maximum routing length. We complement this with lower bounds which show that omitting any of the three parameters results in paraNP-hardness.

**2012 ACM Subject Classification** Theory of computation → Parameterized complexity and exact algorithms

**Keywords and phrases** sort point problem, parameterized complexity, treewidth

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2026.72

**Related Version** *Full Version*: https://arxiv.org/abs/2404.16741

**Funding** *Robert Ganian*: Austrian Science Fund (FWF) Project 10.55776/Y1329.
*Hung P. Hoang*: Austrian Science Fund (FWF) Project 10.55776/ESP1136425.
*Simon Wietheger*: Austrian Science Fund (FWF) Project 10.55776/Y1329.

**Acknowledgements** We would like to thank the anonymous reviewers for their helpful comments.

## 1 Introduction

The task of finding optimal solutions to logistical challenges has motivated the study of a wide range of computational graph problems including, e.g., the classical Vertex and Edge Disjoint Paths [18, 19, 22, 27, 28] problems and Coordinated Motion Planning (also known as Multiagent Pathfinding) [9, 12, 13, 21, 25, 33]. And yet, when dealing with logistical challenges at a higher scale, collision avoidance (the main goal in these three problems) is no longer relevant and one needs to consider different factors when optimizing or designing a logistical network. In this paper, we focus on *parcel sortation*, a central aspect of contemporary large-scale logistical networks which has not yet been thoroughly investigated from an algorithmic and complexity-theoretic perspective.

In the considered setting, we are given an underlying fulfillment network and a set of commodities each represented as a source and destination node. The nodes in the fulfillment network typically represent facilities at various locations, and each commodity needs to be routed from its current facility $s_i$ (e.g., a large warehouse) to a facility $t_i$ in the vicinity of the end customer. However, when routing parceled commodities in the network, each parcel that travels from $s_i$ to $t_i$ via some internal node $u$ must be "sorted" at $u$ based on its subsequent downstream node. Hence, if multiple commodities arrive at $u$ and each needs to be routed to a different facility downstream, it is necessary to apply *sortation* at $u$ to subdivide the stream of incoming parcels between the next stops; in large-scale operations $u$ would typically be equipped with a designated *sort point* for each downstream node, and the number of sort points that a node can feasibly have is typically limited. On the other hand, if all the commodities arriving at $u$ were to be routed to the same downstream node, one can avoid the costly sortation step at $u$ (via applying *containerization* at a previous facility and using a process called *cross-docking* at $u$). We refer readers interested in a more detailed description of these processes to previous works on the topic [2, 5, 26].

In their recent article, Van Dyk, Klause, Koenemann and Megow [34] have shown that the task of optimizing parcel sortation in a logistical network can be modeled as a surprisingly "clean" digraph problem. Indeed, if one represents the network as a digraph $D$ and each commodity as $(s_i, t_i, P_i)$ where $P_i$ is an $s_i$-$t_i$-path in $D$, the aim is to find a *sorting network*[1] – a subgraph of the transitive closure of $D$ – with minimum outdegree. The sorting network captures the information of sort points at each node: in particular, it specifies the downstream nodes that the node has a sort point for. As we want to control the number of sort points at each node, this translates to the objective of minimizing the outdegree. Van Dyk, Klause, Koenemann and Megow [34] primarily focused their work on the task of computing the sorting plans when the physical routes of the commodities are fixed. In particular, they established NP-hardness even when the underlying undirected graph is a star and obtained several polynomial time algorithms for restricted classes of instances, including an exact algorithm if all commodities start at the same vertex and approximation algorithms for various cases where the underlying network is a tree.

In this work we consider both the lower-level optimization task with fixed physical routes as well as the higher-level optimization task where we can determine the routes and sort points. This gives rise to the following two problem formulations, where $\mathcal{T}(D)$ denotes the transitive closure of a directed graph $D$:

Min-Degree Sort Point Problem (MD-SPP)

**Input:** A digraph $D = (V, E)$, a target $T \in \mathbb{N}$ and a set $K$ of routed commodities each of which is a tuple of the form $(s, t, P)$ where $s, t \in V$ and $P$ is an $s$-$t$-path in $D$.

**Task:** Is there a subgraph $H$ of the transitive closure $\mathcal{T}(D)$ of $D$ such that the maximum outdegree of $H$ is at most $T$, and for each commodity $(s, t, P) \in K$, there is a directed $s$-$t$-path in $H \cap \mathcal{T}(P)$?

Min-Degree Routing and Sort Point Problem (MD-RSPP)

**Input:** A digraph $D = (V, E)$, a target $T \in \mathbb{N}$ and a set $K \subseteq V \times V$ of commodities.

**Task:** Is there a subgraph $H$ of $\mathcal{T}(D)$ such that the maximum outdegree of $H$ is at most $T$ and for each commodity $(s, t) \in K$, there is a directed $s$-$t$-path in $H$?

[1] These logistical networks should not be confused with number sorting networks [1].

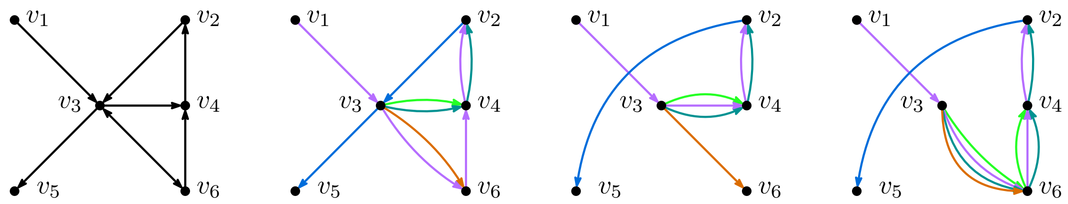


**Figure 1** Example network for MD-SPP and MD-RSPP. Multi-edges are only for illustration; a solution graph $H$ is in fact simple.
The left-most image shows an input graph $D$.
The second image visualizes five commodities for MD-SPP: $(v_1, v_2, (v_1, v_3, v_6, v_4, v_2)), (v_2, v_5, (v_2, v_3, v_5)), (v_3, v_2, (v_3, v_4, v_2)), (v_3, v_4, (v_3, v_4)), (v_3, v_6, (v_3, v_6))$.
The third image gives a minimum target ($T = 2$) solution graph $H$ to this MD-SPP instance.
The right-most image illustrates that for MD-RSPP, by not fixing the paths of commodities, solutions with a smaller target might exist (here $T = 1$).

Figure 1 provides an example illustrating both problems.[2] For MD-RSPP the paths of commodities are not fixed, so commodities are defined as 2-tuples.

While both MD-SPP and MD-RSPP could be rather easily shown to be NP-complete on general graphs, Van Dyk, Klause, Koenemann and Megow showed that the former problem remains NP-complete even when restricted to orientations of stars [34]. In the rest of their article, they then focused on obtaining approximation as well as exact algorithms for MD-SPP on special classes of oriented trees. Apart from these individual results, the computational complexity of MD-RSPP and MD-SPP remains entirely unexplored.

## Contributions

The central question tackled by our work is whether the structural properties of inputs can be leveraged to circumvent the NP-hardness of MD-RSPP and MD-SPP. To do so, we analyze the problems through the more refined lens of *parameterized complexity* [6, 11]. There, the general aim is to design algorithms with running times which are not exponential in the whole input size, but only exponential in some well-defined integer *parameters* of the input. The most desirable complexity class in the parameterized setting is *fixed-parameter tractable* (FPT), which means that the problem can be solved in time $f(p) \cdot n^{\mathcal{O}(1)}$ where $f(p)$ is a computable function of the parameter $p$ and $n$ is the input size. Naturally, if we establish NP-hardness for a problem even under the restriction that the parameter $p$ is fixed to a constant, then this immediately rules out fixed-parameter tractability w.r.t. $p$; in this case, we say that the problem is paraNP-hard.

In our work, we consider three natural types of parameterizations for MD-RSPP and MD-SPP: parameterizing by the target solution quality (i.e., $T$), by the number of commodities, or by the structural properties of the input graph. The article is accordingly split into three parts, one for each of these perspectives.

In the first part of our article – Section 3 – we establish that MD-SPP and MD-RSPP remain NP-hard already when $T = 2$. We complement this lower bound by providing a polynomial-time algorithm solving the latter problem in the special case of $T = 1$, meaning that for MD-RSPP the NP-hardness result is tight. While the proofs of these results are

[2] For complexity-theoretic reasons, here we consider the decision variants. However, all obtained algorithms can solve the corresponding optimization tasks as well.

non-trivial and especially the algorithm requires us to obtain deeper insights into the problem, they do not rely on advanced technical machinery and can thus be seen as introductory. We note that the provided hardness result holds already for orientations of trees with a single source, and complements an earlier additive (+1)-approximation of Van Dyk, Klause, Koenemann and Megow for MD-SPP on this graph class [34, Theorem 3].

In Section 4, we turn towards a parameterized analysis of the considered problems w.r.t. the number of commodities. For MD-SPP, we begin by establishing fixed-parameter tractability for the case of $T = 1$ via a stand-alone algorithm, as our more involved arguments for the general cases do not seamlessly transfer to this simpler setting. We then obtain an involved fixed-parameter algorithm for the "more general" setting of $T \geq 2$ by a kernelization subroutine that is based on Ramsey-type arguments, but with a twist: to implement the main reduction rule used here, we need to transition to a more general variant of the problem where some of the vertices do not contribute towards the degree bound $T$.

As the algorithm used for MD-SPP relies on the fixed paths of the commodities, separate arguments have to be used in order to solve MD-RSPP when $T \geq 2$. Here, we obtain a fixed-parameter algorithm as well, this time by combining a delicate branching routine for solution templates with a color coding approach. It is perhaps worth noting that – in spite of some superficial similarity between the problems – the fixed-parameter tractability of our problems contrasts the known paraNP-hardness of Vertex Disjoint Paths (parameterized by the number of paths) on directed graphs [15, 29].

In Section 5, we target graph-structural parameterizations which would yield tractability for arbitrary choices of $T$ and arbitrarily many commodities. Given the previously established NP-hardness of MD-SPP on orientations of stars – and the fact that the same reduction also works for MD-RSPP– one could ask whether parameterizing by treewidth plus the maximum vertex degree (of the underlying undirected graph) suffices; after all, this combined parameterization has already been successfully employed to achieve fixed-parameter algorithms for a number of other challenging problems, including, e.g., Edge Disjoint Paths [19]. Unfortunately, our reductions in Section 3 already establish the NP-hardness of both problems of interest even on bounded-degree trees.

As our final contribution, we show that the intractability of both problems can be overcome if one takes the maximum length of any admissible route as a third parameter. In particular, we devise a fixed-parameter algorithm that relies on dynamic programming to solve both MD-SPP and MD-RSPP when parameterized by the treewidth and maximum degree of the input graph (or, more precisely, its underlying undirected graph) plus the maximum length of a route in a solution. We complement this algorithm with reductions which prove that all three of these parameters are necessary: dropping any of the three leads to paraNP-hardness for both problems.

A summary of our complexity results for the two problems is provided in Table 1. A careful reader may notice that although the techniques used to obtain our algorithms and lower bounds often vary between the two problems, their complexity under the comprehensive range of parameterizations considered here seems to be the same. We believe this outcome to be rather unexpected, and provide a further discussion in Section 6.

## 2 Preliminaries

For $k \in \mathbb{N}$, we denote by $[k]$ the set $\{1, \ldots, k\}$. We employ standard graph-theoretic terminology [10]. For a directed graph $D = (V, E)$, let $\Delta^+_D(v)$ and $\Delta^-_D(v)$ be the sets of out-neighbors and in-neighbors of $v$, respectively, for all $v \in V$. Let $\delta^+_D(v) = |\Delta^+_D(v)|$,

**Table 1** Complexity results for MD-SPP and MD-RSPP by different parameterizations. The *degree* and *treewidth* refer to the underlying undirected graph of the input graph $D$. The *path length* is the maximum length of any path fulfilling a commodity.

| Parameter | Variant | Complexity | Reference |
|---|---|---|---|
| Target $T$ | MD-RSPP | Polynomial time solvable for $T \leq 1$ | Thm. 2 |
| | both | NP-hard for $T \geq 2$ | Thm. 3 |
| Number $\lvert K\rvert$ of Commodities | both | FPT | Thm. 5, 10, & 11 |
| Structural | both | FPT w.r.t. degree + treewidth + path length | Thm. 12 |
| | | paraNP-hard w.r.t. degree + treewidth | Thm. 3 |
| | | paraNP-hard w.r.t. treewidth + path length | [34] & Fact 13 |
| | | paraNP-hard w.r.t. degree + path length | Thm. 14 & 15 |

$\delta^-_D(v) = |\Delta^-_D(v)|$, $\delta^+(D) = \max_{v\in V} \delta^+_D(v)$, and $\delta^-(D) = \max_{v\in V} \delta^-_D(v)$. We may omit the subscript where $D$ is clear from context. We use $D[X]$ to denote the sub(di)graph of $D$ induced on the vertex set $X$. The *transitive closure* of $D$, denoted by $\mathcal{T}(D)$, is the directed graph on the vertex set $V(D)$, whereas an edge $(i, j)$ exists in $\mathcal{T}(D)$ if and only if there is a directed path from $i$ to $j$ in $D$. The *underlying undirected graph* of a directed graph $D$, denoted by $\mathcal{G}(D)$, is the undirected simple graph such that $V(\mathcal{G}(D)) = V(D)$ and $ab \in E(\mathcal{G}(D))$ if $D$ contains at least one arc between $a$ and $b$. We use *SCC* as an abbreviation for a strongly connected component.

We refer to the standard books [6, 11] for an introduction of the basics of parameterized complexity theory, including *fixed-parameter tractability*, *kernelization*, XP*-tractability*, *para*NP*-hardness*, and *treewidth*.

## Problem-Specific Definitions

For a routed commodity $(s, t, P)$ in an MD-SPP instance or a commodity $(s, t)$ in an MD-RSPP instance, we call $s$ its *source* and $t$ its *destination*. A graph $H$ satisfying all conditions given in the problem statements is called a *solution graph*. Given an MD-SPP or MD-RSPP instance $(D, K, T)$ and a commodity $\kappa \in K$, a *witness path* of $\kappa$ is a path in $\mathcal{T}(D)$ that satisfies the condition of the corresponding problem description for that commodity. In other words, if $(D, K, T)$ is an MD-SPP instance, then a witness path of a commodity $(s, t, P)$ is an $s$-$t$-path in $\mathcal{T}(P)$. If $(D, K, T)$ is an MD-RSPP instance, then a witness path of a commodity $(s, t)$ is an $s$-$t$-path in $\mathcal{T}(D)$. A *solution path cover* of $(D, K, T)$ is a set of witness paths, one for each commodity in $K$, such that its union is a solution graph of $(D, K, T)$. We provide an observation on trivial instances of the problems.

▶ **Observation 1.** *Each* MD-RSPP *instance with a commodity* $(s, t)$ *such that* $(s, t) \notin \mathcal{T}(D)$ *is a* NO*-instance. Apart from these, all instances of* MD-SPP *and* MD-RSPP *where the target is at least the maximum outdegree of the input graph or the maximum number of commodities starting at the same vertex are* YES*-instances.*

# 3 Complexity Classification with Respect to the Target

Our first – and in a sense introductory – technical section is dedicated to the study of MD-SPP and MD-RSPP when parameterized by the value of the target $T$. We begin by noting that both problems of interest are trivially solvable in linear time if $T = 0$. Next we consider the case of $T = 1$ and have the following tractability result for MD-RSPP.

▶ **Theorem 2.** MD-RSPP *is polynomial time solvable when restricted to instances with target* $T \leq 1$*.*

**Proof Sketch.** Define $\mathcal{I} := (D, K, 1)$. First, construct the graph $H = (V(D), K)$, that is, the graph on the vertices of $D$ with an edge $(s, t)$ for every commodity $(s, t) \in K$. For all non-trivial instances, $H$ is a solution graph except that its outdegree might exceed the target. We iteratively transform $H$ to reduce the outdegree as much as possible while still satisfying all commodities. In particular, we start by replacing each strongly connected component (SCC) in $H$ by an (arbitrary) cycle over the vertices in the SCC, while edges between distinct SCCs are preserved. Next, whenever a commodity requires such an edge between two such cycles, we *merge* the two cycles into a larger one (i.e., we replace them with an arbitrary single cycle over the union of their vertices in this auxiliary graph). This operation is used whenever the vertices $s$ and $t$ of a commodity $(s, t)$ are in distinct SCCs of $H$ but can reach each other in the underlying network $D$ – in particular, if such a merge is not realizable within the transitive closure of $D$, we reject the instance.

At the end of this step, there are no commodities leaving any cycle. Last, we remove all cycles and obtain a topological ordering of the vertices of the resulting acyclic graph. We then iterate through this ordering, consider several cases based on the commodities starting at the current vertex and, depending on the precise situation, decide whether we reject the instance, merge or connect some cycles, or reorder some subsequent vertices. We show that if the process above is successful, the final graph is indeed a solution graph for $\mathcal{I}$. ◀

In contrast to the result above, we show that both MD-RSPP and MD-SPP are paraNP-hard when parameterized by $T$. This complements the earlier additive (+1)-approximation of Van Dyk, Klause, Koenemann and Megow on orientations of trees with a single source [34, Theorem 3].

▶ **Theorem 3.** MD-SPP *and* MD-RSPP *are* NP*-hard, even when restricted to instances where* $T \leq 2$*, there is a single source, and* $\mathcal{G}(D)$ *is a tree of maximum degree at most* 4*.*

**Proof Sketch.** We reduce from the strongly NP-hard 3-Partition problem [20, 23], which asks: can a set of given $3m$ distinct positive integers $n_1, \ldots, n_{3m}$ such that $\sum_{i=1}^{3m} n_i = mB$ be partitioned into $m$ triples, each summing up to $B$? Given an instance of 3-Partition, we construct an instance of MD-RSPP with $T = 2$ as follows (for MD-SPP, we use the same construction and set each path to be the unique $s$-$t$-path in the constructed oriented tree).

For $D$, we begin by creating a path of $2mB + 2m$ vertices. Let $S$ be the set of the first $2m$ vertices with labels $s_1, s'_1, \ldots, s_m, s'_m$ (in the order they appear in the path). The next $mB$ vertices form the set $W$ and are indexed by the numbers $n_1, \ldots, n_{3m}$, specifically $w_1^1, \ldots, w_1^{n_1}, \ldots, w_{3m}^1, \ldots, w_{3m}^{n_{3m}}$. Among the remaining $mB$ vertices, which form the set $R$, there are $B$ vertices for each triple to be created, that is, $r_1^1, \ldots, r_1^B, \ldots, r_m^1, \ldots, r_m^B$. For all $i \in [m]$, create the commodities $(s_i, s'_i)$ and $\{(s_i, r_i^j) \mid j \in [B]\}$. For all $i \in [3m]$, create the commodities $\{(w_i^j, w_i^{j+1}) \mid j \in [n_i - 1]\}$. For all $i \in [3m]$, create a vertex $\tilde{w}_i$, and add the edge $(w_i^{n_i}, \tilde{w}_i)$ and a commodity $(w_i^{n_i}, \tilde{w}_i)$. For all $i \in [m]$ and $j \in [B]$, create vertices $\tilde{r}_i^j$ and $\bar{r}_i^j$, and add the two edges and two commodities $(r_i^j, \tilde{r}_i^j)$ and $(r_i^j, \bar{r}_i^j)$. The resulting graph $D$ is depicted in Figure 2.

It remains to establish that the initial 3-Partition instance was a YES-instance if and only if so is the constructed instance. We note that as the constructed graph is acyclic, every edge $(u, v)$ for which there is a commodity from $u$ to $v$ needs to be in the solution. Thus, for each $i \in [m]$ there are three available edges from $s_i$ and $s'_i$ that can be used to select three subpaths in the middle section, representing three integers. Each subpath $w_j^1, \ldots, w_j^{n_j}$ allows

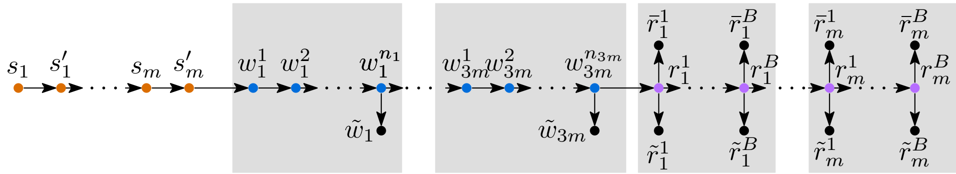

**Figure 2** Constructed graph $D$ for the proof of Theorem 3. Vertices in $S$ are colored orange, those in $W$ are blue and those in $R$ are violet.

for exactly $n_j$ outgoing edges, thus the subpaths have to be chosen such that their sum is at least $B$ to satisfy the commodities that need to reach the last part of the path from $s_i$. The formal correctness proof, however, needs to carefully account for all possible routings for the commodities and choices of $H$. ◀

## 4 Parameterizing by the Number of Commodities

This section is dedicated to the study of the considered parcel sortation problems when parameterized by the number of commodities. This line of enquiry can be seen as analogous to the fundamental questions that have been investigated for other prominent examples of problems typical for logistical networks, such as the study of Vertex and Edge Disjoint Paths parameterized by the number of paths or of Coordinated Motion Planning parameterized by the number of robots. We remark that the former has been shown to be fixed-parameter tractable in Robertson and Seymour's seminal work [32], while the fixed-parameter tractability of the latter has only been established on planar grid graphs in a recent work of Eiben, Ganian and Kanj [12].

We start by bounding the size of a hypothetical solution by a function of the number of commodities – an insight which will be used later on in this section.

▶ **Lemma 4.** *Every YES-instance $(D, K, T)$ of* MD-SPP *or* MD-RSPP *has a solution path cover whose unions consists of at most $2^{|K|} + |K|$ vertices.*

**Proof Sketch.** Let $k := |K|$ and $K = \{\kappa_1, \ldots, \kappa_k\}$. By deleting excess vertices and edges from any solution, there is a solution path cover with the paths $Q_1, \ldots, Q_k$, where $Q_i$ is a path that satisfies $\kappa_i$ for $i \in [k]$. Let $H$ be the solution graph that is the union of the solution path cover. For every non-source vertex $v \in V(H)$, we define $\mathcal{Q}(v) := \{i \mid v \in Q_i\}$. In particular, note that for every non-source vertex $v$ in the graph, we have $\mathcal{Q}(v) \neq \emptyset$.

Suppose there are non-source vertices $u, v$ such that $\mathcal{Q}(u) = \mathcal{Q}(v)$. Consider the following modification of $H$. For every path $Q_i$ that visits $u$ before $v$, we remove the subpath between $u$ (inclusive) and $v$ (exclusive) from $Q_i$. In other words, if $Q_i = (v_1, \ldots, v_\ell, u, v_{\ell+1}, \ldots, v_j, v, v_{j+1}, \ldots, v_t)$, then we replace this path by $(v_1, \ldots, v_\ell, v, v_{j+1}, \ldots, v_t)$. Similarly, for every path $Q_i$ that visits $v$ before $u$, we remove the subpath between $v$ (inclusive) and $u$ (exclusive) from $Q_i$. Note that after the modification, the two vertices $u$ and $v$ no longer satisfy $\mathcal{Q}(u) = \mathcal{Q}(v)$. We can show that the graph $H'$ as the union of all $k$ paths after the modification is still a solution graph to $(D, K, T)$.

Apply the modification above repeatedly as long as there are two non-source vertices $u, v$ such that $\mathcal{Q}(u) = \mathcal{Q}(v)$. This procedure terminates after less than $k \cdot |V(D)|$ iterations as each time at least one of the $Q_i$ paths is shortened. At the end, we obtain a solution graph such that every non-source vertex has a distinct $\mathcal{Q}$ value. The lemma then follows from the facts that (i) the codomain of $\mathcal{Q}$ is the set of all subsets of $[k]$ and (ii) there are at most $k$ sources. ◀

## 4.1 A Fixed-Parameter Algorithm for MD-SPP

In this subsection, we establish the fixed-parameter tractability of MD-SPP by the number of commodities. Our proof relies on three main ingredients. First, Theorem 5 below allows us to assume $T \geq 2$ in the rest of the section.

▶ **Theorem 5.** MD-SPP *restricted to instances such that* $T \leq 1$ *is in* FPT *w.r.t. the number of commodities.*

**Proof.** We prove the statement by providing an $f(|K|)$-kernel. In particular, we show that any MD-SPP instance $(D, K, 1)$ can be reduced to the equivalent instance $(D', K', 1)$, where $V(D')$ contains only vertices that are the source or destination of a commodity in $K$, $E(D') = \{(v, v') \mid v, v' \in V(D'), (v, v') \in \mathcal{T}(D)\}$, and $K'$ is obtained from $K$ by deleting all vertices from all paths that are neither start nor destination of any commodity.

As $\mathcal{T}(D')$ is a subgraph of $\mathcal{T}(D)$, any solution to $\mathcal{I}' = (D', K', 1)$ is a solution to $\mathcal{I} = (D, K, 1)$. In the reversed direction, we show that if $\mathcal{I}$ is a YES-instance then it has a solution that does not use any of the deleted vertices by an exchange argument. Let $H$ be a solution graph for $\mathcal{I}$ and let $v$ be a non-source, non-destination vertex in $H$. If $v$ has no out-neighbor in $H$, it does not contribute to the solution and can be removed. Otherwise, as $H$ has outdegree 1, $v$ has a unique out-neighbor $w$ in $H$. Create a new solution graph $H'$ by letting $V(H') = V(H) \setminus \{v\}$, deleting the edge $(v, w)$, and replacing each edge $(u, v) \in E(H)$ by an edge $(u, w) \in E(H')$. Then the outdegree of each vertex in $H'$ is at most as large as its outdegree in $H$ and thus at most 1. Moreover, for every commodity routed along an $s$-$t$-path $P$ there is still an $s$-$t$-path in $H' \cap \mathcal{T}(P)$. In either case, $H'$ still resolves all commodities in $K$. By repeatedly applying this procedure, we can create a solution graph $H'$ in which all non-source, non-destination vertices are removed. Thereby $H'$ is a solution graph for $\mathcal{I}'$.

Clearly, the above reduction is computable in polynomial time and leaves a graph with at most $2|K|$ vertices. By brute forcing the reduced instance, this kernel yields FPT runtime. ◀

Second, we recall the following generalization of Ramsey's Theorem [31]:

▶ **Fact 6.** *There is a function* RAM*:* $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *with the following property. For each pair of integers* $r, s$ *and each clique* $G$ *of size at least* RAM$(r, s)$ *such that each edge has a single label (color) from a set of* $r$ *possible labels, it holds that* $G$ *has a subclique* $H$ *of size at least* $s$ *such that every edge in* $H$ *has the same label.*

In particular, we remark that RAM has a computable upper bound.

Third, we define the following generalization of MD-SPP.

SUBSET-MIN-DEGREE SORT POINT PROBLEM (SMD-SPP)

**Input:** A digraph $D = (V, E)$, a set $F \subseteq V$ of vertices, a target $T \in \mathbb{N}$, and a set $K$ of routed commodities each of which is a tuple of the form $(s, t, P)$ where $s, t \in V$ and $P$ is an $s$-$t$-path in $D$.

**Task:** Is there a subgraph $H$ of the transitive closure $\mathcal{T}(D)$ of $D$ such that every vertex in $V(D) \setminus F$ has outdegree at most $T$ in $H$, and for every commodity $(s, t, P) \in K$, there exists a directed $s$-$t$-path in $H \cap \mathcal{T}(P)$?

We call the vertices in $F$ in the definition above *flexible*, as their outdegrees are not constrained. The terms solution graph and solution path cover are defined analogously for SMD-SPP. The size of a solution can be bounded by a function of the number of commodities using an identical argument as in the proof of Lemma 4.

▶ **Lemma 7.** *Every YES-instance $(D, K, T)$ of* SMD-SPP *has a solution path cover whose union consists of at most $2^{|K|} + |K|$ vertices.*

For an MD-SPP instance $(D, K, T)$ and $v \in V(D)$, let the *type* $\mathcal{P}(v)$ of $v$ be the set of routed commodities with paths using $v$ (i.e., $\mathcal{P}(v) = \{(s, t, P) \mid (s, t, P) \in K, P \text{ contains } v\}$). In Lemma 9 below, we reduce MD-SPP instances to SMD-SPP with a nontrivial set $F$ such that there are only few non-flexible vertices. To define $F$, we build on the following definition.

▶ **Definition 8** ($q$-enclosure)**.** *Let $\mathcal{I} = (D, K, T)$ be an* MD-SPP *instance and $q$ be a natural number. A vertex $v$ is $q$-*enclosed *w.r.t. $\mathcal{I}$, if there exist $U \subseteq K$, $U \neq \emptyset$, and distinct vertices $v_1, \dots, v_{2q+1}$ of type $U$ in $D$ such that $v_{q+1} = v$, and for all commodities $(s, t, P) \in U$, the directed path $P$ contains $(v_1, \dots, v_{2q+1})$ or $(v_{2q+1}, \dots, v_1)$ as a subsequence.*

▶ **Lemma 9.** *For an* MD-SPP *instance $(D, K, T)$ with $T \geq 2$, let $k := |K|$, and $q = \left\lceil (2^k + k)\left(1 + \frac{k}{T-1}\right) \right\rceil$. Suppose that $|V(D)| > q$. Further suppose $F$ is a set of $q$-enclosed vertices w.r.t. $(D, K, T)$. Then $(D, K, T)$ is a YES-instance if and only if the* SMD-SPP *instance $(D, F, K, T)$ is a YES-instance.*

**Proof Sketch.** Trivially, a solution path cover of $(D, K, T)$ is also a solution path cover of $(D, F, K, T)$. Hence, we only need to prove the other direction.

Suppose that $C$ is a solution path cover of $(D, F, K, T)$. By Lemma 7, we can assume that the union of $C$ has at most $2^k + k$ vertices. Let $X$ be the set of vertices not in $C$. Note that $C$ is also a solution path cover of $(D, F \setminus X, K, T)$. We can prove by induction that there exists a sequence $C_0, C_1, \dots, C_{|F \setminus X|}$ such that for $i \in \{0, \dots, |F \setminus X|\}$, $C_i$ has at most $2^k + k + ki/(T-1)$ distinct vertices, and $C_i$ is a solution path cover of $(D, F_i, K, T)$, where $|F_i| = |F \setminus X| - i$ and $F_i \subseteq F \setminus X$. The idea is, given a solution path cover $C_i$, to select a $q$-enclosed vertex $v$ from $F_i$ and reduce its outdegree to $2 \leq T$. More precisely, the commodities passing through $v$ are divided in two sets depending on the order in which they pass through the vertex sequence of $v$'s $q$-enclosure. For the commodities passing from $v_1$ to $v_{2q+1}$ we select vertices that succeed $v$ in the sequence and not yet have $T$ out-edges. These are connected in a path starting at $v$ and each vertex resolves as many commodities directly to their destination as possible. Analogously, the commodities that pass from $v_{2q+1}$ to $v_1$ are handled using vertices that preceed $v$ in the sequence. This way we obtain a solution path cover of $(D, F_{i+1} \setminus X, K, T)$ where $F_{i+1} = F_i \setminus \{v\}$. Note that $(D, F_{|F \setminus X|}, K, T)$ is equivalent to $(D, K, T)$. Hence, $C_{|F \setminus X|}$ is a solution path cover of $(D, K, T)$. ◀

We now have all components to establish the main result of this subsection.

▶ **Theorem 10.** MD-SPP *is in* FPT *w.r.t. the number of commodities.*

**Proof Sketch.** Let $(D, K, T)$, $k$, and $q$ be as in Lemma 9. The proof hinges on finding a set $F$ of $q$-enclosed vertices such that $V(D) \setminus F \leq f(k)$ for some computable function $f$. For sufficiently large graphs, there is a large set of vertices visited by the same commodities. Now, we ensure that there exists a sufficiently long sequence such that all commodities traverse it in a “left-to-right” or “right-to-left” fashion. To find this sequence, we apply an edge labeling which intuitively captures the direction in which the routed commodities traverse the two incident vertices, and then invoke Fact 6. Exhaustive repetition yields such a set $F$. Then by Lemma 9, our task reduces to solving $(D, F, K, T)$. As the outdegrees of vertices in $F$ are unrestricted in the solution and $V(D) \setminus F \leq f(k)$, we can solve this by brute force. ◀

## 4.2 A Fixed-Parameter Algorithm for MD-RSPP

We use the color-coding technique with derandomization [6, Subsections 5.2 and 5.6] to establish fixed-parameter tractability of MD-RSPP with respect to the number of commodities. Intuitively, we solve a restricted version of MD-RSPP, namely COLORFUL-MD-RSPP, where the vertices in $D$ are colored and a solution graph $H$ is not allowed to contain two vertices of the same color. By Lemma 4, this requires $2^{|K|} + |K|$ distinct colors. Then, by applying color coding, any fixed-parameter algorithm for the colorful variant w.r.t. the number of colors plus $|K|$ implies fixed-parameter tractability of MD-RSPP w.r.t. $|K|$ alone.

We note that the algorithm underlying Theorem 10 cannot be applied to Theorem 11, as it hinges on identifying "irrelevant" ($q$-enclosed) vertices. The crucial criterion for this relevance is the way in which the routed commodities use these vertices in their paths – some information which is not present for instances of MD-RSPP. On the other hand, the color-coding based algorithm for the theorem below does not trivially adapt to MD-SPP as it requires to rearrange the subgraphs of a solution induced by SCCs into paths, which does not violate unrouted commodities but might violate routed ones.

▶ **Theorem 11.** MD-RSPP *is in* FPT *w.r.t. the number of commodities.*

**Proof Sketch.** Let $(D, K, T)$ be an MD-RSPP instance and $k := |K|$. By Theorem 2, we can assume that $T \geq 2$. Applying the color-coding technique as mentioned above, we only need to give an algorithm to solve the colorful variant, where there is additionally a coloring $c : V(D) \to [2^k + k]$ and any solution can use each color at most one time.

Let $S$ and $R$ be the set of sources of commodities in $K$ and the set of sources or destinations of commodities in $K$, respectively. We can show that any colorful YES-instance has a solution graph $H'$ in which each SCC either has size one or contains a vertex in $S$.

The algorithm for a colorful instance $(D, K, T, c)$ goes as follows. We branch on all possible *templates* for a solution. Formally, a template $\Gamma$ is a graph of up to $2^k + k$ vertices such that $R \subseteq V(\Gamma)$, all vertices in $V(\Gamma)$ have a unique color from $[2^k + k]$ and vertices in $R$ have the same color in $\Gamma$ as in $D$. Further, we only consider templates that are the union of one $s$-$t$-path per commodity $(s, t) \in K$, which implies that each vertex has out-degree at most $k$. There are fewer than $\binom{2^k+k}{k}^{2^k+k} \cdot (2^k + k)!$ such templates. Discard a template $\Gamma$ if its maximum outdegree is larger than $T$; it contains an SCC of size at least 2 without a vertex from $S$; or there are $v, w \in R$ such that $(v, w) \in E(\Gamma)$ but $(v, w) \notin \mathcal{T}(D)$.

For every remaining template $\Gamma$, we verify whether it yields a solution to the colorful $(D, K, T, c)$ instance: We mark all vertices in $R$ as *viable.* Consider a topological sorting of the SCCs in $\Gamma$. Construct a total order of the vertices in $V(D) \setminus R$ according to a linearization of that topological sorting, where vertices inside an SCC of size at least 2 are placed in arbitrary order at the position of the SCC. We visit the vertices in that order from back to front; that is, we start with a vertex with no out-edges except inside its SCC or to $R$.

Let $v'$ be the currently visited vertex, and let $U$ be the set of all vertices in $D$ that share the same color with $v'$. If $v'$ is in an SCC $C$ of size at least 2, then we choose an arbitrary vertex $s$ in $C \cap S$, and we mark all $v \in U$ as viable as long as $(v, s)$ and $(s, v)$ are edges in $\mathcal{T}(D)$. Otherwise, for each edge $(v', w') \in E(\Gamma)$, we mark all $v \in U$ as viable, if (i) there is a viable vertex $w$ in $D$ of the same color as $w'$ in $\Gamma$ such that the edge $(v, w)$ is in $\mathcal{T}(D)$, and (ii) the edge $(w'', v)$ is in $\mathcal{T}(D)$ for each edge $(w'', v') \in E(\Gamma)$ with $w'' \in R$. We have a YES-instance if and only if there is $\Gamma$ with viable vertices for all used colors. ◀

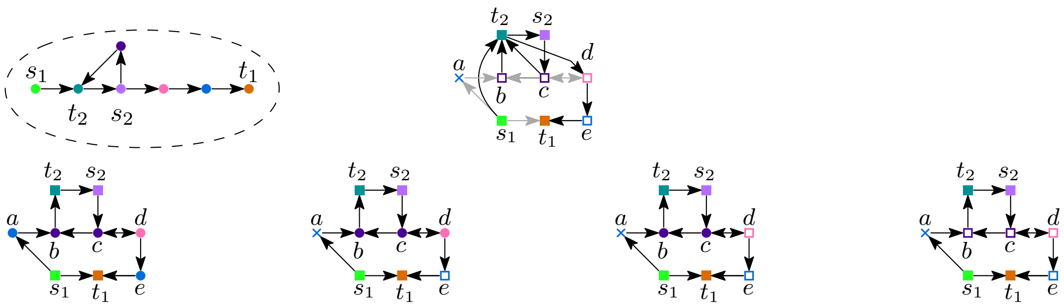

**Figure 3** Visualization of the algorithm for Theorem 11. A template $\Gamma$ is displayed in the top-left. The bottom row shows how vertices in $D$ are marked as viable (square) or not viable (cross). First, only sources and destinations are marked as viable (left). Only the blue vertex $e$ has a path to the orange vertex $t_1$ and is marked as viable (center-left). The pink vertex $d$ is viable as it is reachable from $s_2$ and has a path to the viable blue vertex $e$ (center-right). As both purple vertices $b$ and $c$ share an SCC with $s_2$ in both $\gamma$ and $D$, both are viable (right). Choosing out-edges according to the template might result in a solution graph $H$ as displayed in the top-right. Note that both viable purple vertices have an edge to $t_2$ but only (arbitrary) one of them has $s_2$ as an in-neighbor.

## 5 Structural Parameters

In our final section, we turn towards a graph-structural analysis of the considered parcel sortation problems. At first glance, it may seem difficult to identify a "reasonable" graph-structural parameter that could be used to solve MD-SPP or MD-RSPP: the NP-hardness of both problems on an orientation of a star ([34] in combination with Fact 13) rules out not only the use of treewidth and its directed variants [16, 17], but also a range of other much more restrictive parameterizations such as treedepth [30] and the vertex cover number [3, 7, 35].

The above situation is not unique: there is a well-known example of another routing problem – Edge Disjoint Paths – which suffers from a similar difficulty, notably by being paraNP-hard even on undirected graphs with vertex cover number 3 [14]. But for Edge Disjoint Paths, one can achieve fixed-parameter tractability when parameterizing by the combination of treewidth and the maximum degree [19], while for MD-SPP and MD-RSPP we can exclude tractability even under this combined parameterization by recalling Theorem 3.

The above considerations raise the question: Are there any structural restrictions under which we can efficiently solve instances of MD-SPP or MD-RSPP involving a large number of commodities and target? As the main contribution of this section, we answer this question positively by identifying the *maximum route length* as the missing ingredient required for tractability. Indeed, from an application perspective, restricting one's attention only to physical routes which do not involve too many intermediary nodes is well-aligned with practical considerations for the routing of most goods. For the purposes of this section we will assume that the parameterized instances of MD-SPP and MD-RSPP come equipped with an additional integer parameter $p$ called *path length* and require:

- for MD-SPP, the path $P$ of every routed commodity has length at most $p$;
- for MD-RSPP, for each commodity $(s, t) \in K$, there is a directed $s$-$t$-path $P'$ in $H$ with a corresponding *base path* $P$ in $D$ (formally, $P' \subseteq \mathcal{T}(P)$), and $P$ has length at most $p$. To formally distinguish it from MD-RSPP, we call this enriched problem MD-RSPP$_{\text{PL}}$ (the problems coincide for $p \geq |V(G)|$).

As our first result, we establish the fixed-parameter tractability of both problems of interest when $p$ is included in the parameterization.

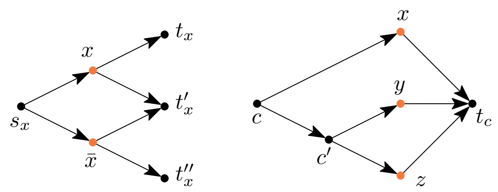


**Figure 4** Variable gadget (left) and clause gadget (right) for the proof of Theorem 14. Orange colored vertices are shared between different gadgets.

▶ **Theorem 12.** MD-SPP *and* MD-RSPP$_{\text{PL}}$ *are in* FPT *w.r.t. the combination of the treewidth and maximum degree of* $\mathcal{G}(D)$ *and the path length.*

**Proof Sketch.** We compute a width-optimal nice tree-decomposition $(\mathtt{T}, \chi)$ using known techniques [4, 24] and then use the "standard" dynamic programming technique that starts at the leaves of the decomposition and traverses up to the root. The difficulty is that the solution graph $H$ is a subgraph of the transitive closure $\mathcal{T}(D)$ and not of $D$ itself. Thus, a bag need not be a separator for the sought-after solution graph. To remedy this issue, in the dynamic program we consider bounded-radius balls of vertices around each bag, and show that these are separators for $H$, whereas their size is bounded by a function of the maximum degree and the path length. The records essentially enumerate all possible local solutions within a ball. Intuitively, such a local solution depends only on the subgraph induced by the ball on a hypothetical solution graph $H$ as well as the number of edges to vertices outside the ball in $H$, for each vertex in the ball. To conclude the proof, we show how these records can be computed in a leaf-to-root fashion along $(\mathtt{T}, \chi)$ and establish correctness. ◀

We complement Theorem 12 by showing that not bounding any one of the three parameters (treewidth, degree, and path length) yields NP-hardness for both problems. Theorem 3 in Section 3 already establishes this for instances of large path length. Van Dyk, Klause, Koenemann, and Megow [34] proved MD-SPP to be NP-hard even when the input graph is an orientation of a (high-degree) star. As in such graphs, each commodity has a unique path connecting its source and destination, the same construction can be used for MD-RSPP$_{\text{PL}}$.

▶ **Fact 13.** MD-RSPP$_{\text{PL}}$ *and* MD-SPP *with path length* $p = 2$ *on stars are* NP*-hard.*

Last, we prove NP-hardness for instances with constant degree and path length by reducing from 3-SAT-(2,2): in this NP-complete restriction of 3-SAT, each clause has exactly three literals and each literal occurs in exactly two clauses [8].

▶ **Theorem 14.** MD-RSPP$_{\text{PL}}$ *is* NP*-hard, even when restricted to instances with* $T \leq 2$*, path length at most* $4$*, and maximum degree in* $\mathcal{G}(D)$ *at most* $7$*.*

**Proof Sketch.** We reduce from 3-SAT-(2,2) with $m$ clauses and $n$ variables. For every variable $x$, we use the variable gadget as in Figure 4 (left) and add the following commodities: $(s_x, t)$ for $t \in \{t_x, t'_x, t''_x\}$. For every clause $c$ over the literals $x, y$, and $z$, we use the clause gadget as in Figure 4 (right). There, the vertices of $x, y$, and $z$ correspond to the respective literal vertex ($x$ or $\overline{x}$) in the variable gadgets. Further, we add the following commodities: $(c, t)$ for $t \in \{x, y, z, t_c\}$. The constructed MD-RSPP$_{\text{PL}}$ instance with $p = 4$ has a solution with target at most 2 if and only if the 3-SAT-(2,2) instance is a YES-instance. ◀

We modify the proof of Theorem 14 using more sophisticated gadgets to account for the fixed paths and obtain a similar result for MD-SPP.

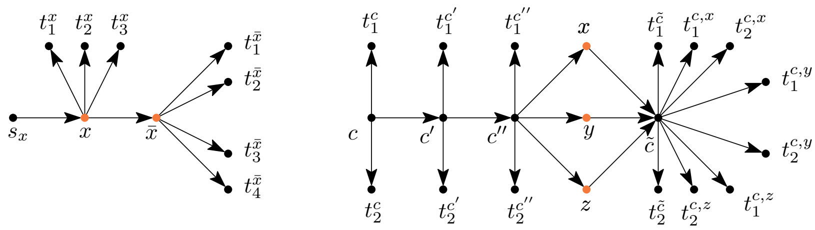


**Figure 5** Variable gadget (left) and clause gadget (right) for the proof of Theorem 15. Orange colored vertices are shared between different gadgets.

▶ **Theorem 15.** MD-SPP *is* NP*-hard, even when restricted to instances with* $T \leq 4$*, path length at most* 5 *and maximum degree in* $\mathcal{G}(D)$ *at most* 11*.*

**Proof Sketch.** We provide a reduction which constructs the following instance of MD-SPP from an instance of 3-SAT-(2,2). For every variable $x$, we use the variable gadget as in Figure 5 (left) and add the following commodities to $K$: $(s_x, t)$ for $t \in \{t_1^x, t_2^x, t_3^x, t_1^{\bar{x}}, t_2^{\bar{x}}, t_3^{\bar{x}}, t_4^{\bar{x}}\}$. The directed paths for these commodities are exactly the unique directed paths in the gadget. For every clause $c$ over the literals $x, y$, and $z$, we use the clause gadget as in Figure 5 (right) and add the following commodities to $K$. First, add $(a, t)$ for $a \in \{c, c', c'', \tilde{c}\}$ and $t \in \{t_1^a, t_2^a\}$. Second, add $(c, c')$ and $(c', c'')$. All of these commodities are assigned their unique respective path. Last, add $(c, t)$ for $t \in \{t_1^{c,a}, t_2^{c,a} | a \in \{x, y, z\}\}$. For $a \in \{x, y, z\}$, the directed path for $(c, t_1^{c,a})$ (or $(c, t_2^{c,a})$) is the unique directed path from $c$ to $t_1^{c,a}$ (or $t_2^{c,a}$) that contains $a$. ◀

## 6 Concluding Remarks and Discussion

The complexity-theoretic behavior of both MD-RSPP and MD-SPP w.r.t. graph-structural parameterizations of is somewhat unique: there seem to be no structural parameterizations of $D$ alone which would yield XP or fixed-parameter algorithms without bounding the input size. Indeed, we achieve tractability in Theorem 12 only by the addition of the path length as a parameter. Luckily, this parameter is well-motivated, as in practice virtually all commodities are routed via rather short paths in the fulfillment network due to its hierarchical nature.

The main technical open question arising from our work is the classical complexity of MD-SPP in the baseline case of $T = 1$. Another aspect which we consider noteworthy is that MD-SPP and MD-RSPP seem to have the same parameterized complexity landscape: they are both FPT w.r.t. $|K|$, paraNP-hard w.r.t. $T$, and appear indistinguishable from the perspective of structural parameterizations (see Table 1). Is there a deeper connection between the two problems and could this be exploited to obtain unified algorithms for both problems which avoid, e.g., the heavy machinery of Ramsey's Theorem and color coding?

## References


1 Miklós Ajtai, János Komlós, and Endre Szemerédi. An o(n log n) sorting network. In *STOC*, pages 1–9. ACM, 1983. doi:10.1145/800061.808726.

2 Jan Van Belle, Paul Valckenaers, and Dirk Cattrysse. Cross-docking: State of the art. *Omega-international Journal of Management Science*, 40:827–846, 2012. URL: https://api.semanticscholar.org/CorpusID:154347787.

3 Sujoy Bhore, Robert Ganian, Fabrizio Montecchiani, and Martin Nöllenburg. Parameterized algorithms for book embedding problems. *J. Graph Algorithms Appl.*, 24(4):603–620, 2020. doi:10.7155/JGAA.00526.

**4** Hans L. Bodlaender and Ton Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *J. Algorithms*, 21(2):358–402, 1996. doi:10.1006/JAGM.1996.0049.

**5** Tzu-Li Chen, James C. Chen, Chien-Fu Huang, and Ping-Chen Chang. Solving the layout design problem by simulation-optimization approach-a case study on a sortation conveyor system. *Simul. Model. Pract. Theory*, 106:102192, 2021. doi:10.1016/J.SIMPAT.2020.102192.

**6** Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*. Springer, Cham, 2015. doi:10.1007/978-3-319-21275-3.

**7** Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. On cutwidth parameterized by vertex cover. *Algorithmica*, 68(4):940–953, 2014. doi:10.1007/S00453-012-9707-6.

**8** Andreas Darmann and Janosch Döcker. On simplified NP-complete variants of Monotone 3-Sat. *Discrete Appl. Math.*, 292:45–58, 2021. doi:10.1016/j.dam.2020.12.010.

**9** Argyrios Deligkas, Eduard Eiben, Robert Ganian, Iyad Kanj, and M. S. Ramanujan. Parameterized algorithms for coordinated motion planning: Minimizing energy. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, Tallinn, Estonia, July 8-12, 2024*, volume 297 of *LIPIcs*, pages 53:1–53:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ICALP.2024.53.

**10** Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in math.* Springer, 2012.

**11** Rodney G. Downey and Michael R. Fellows. *Fundamentals of parameterized complexity.* Texts in Computer Science. Springer, London, 2013. doi:10.1007/978-1-4471-5559-1.

**12** Eduard Eiben, Robert Ganian, and Iyad Kanj. The parameterized complexity of coordinated motion planning. In Erin W. Chambers and Joachim Gudmundsson, editors, *39th International Symposium on Computational Geometry, SoCG 2023, June 12-15, 2023, Dallas, Texas, USA*, volume 258 of *LIPIcs*, pages 28:1–28:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.SOCG.2023.28.

**13** Foivos Fioravantes, Dusan Knop, Jan Matyás Kristan, Nikolaos Melissinos, and Michal Opler. Exact algorithms and lowerbounds for multiagent path finding: Power of treelike topology. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 17380–17388. AAAI Press, 2024. doi:10.1609/AAAI.V38I16.29686.

**14** Krzysztof Fleszar, Matthias Mnich, and Joachim Spoerhase. New algorithms for maximum disjoint paths based on tree-likeness. *Math. Program.*, 171(1-2):433–461, 2018. doi:10.1007/S10107-017-1199-3.

**15** Steven Fortune, John E. Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10:111–121, 1980. doi:10.1016/0304-3975(80)90009-2.

**16** Robert Ganian, Petr Hlinený, Joachim Kneis, Alexander Langer, Jan Obdrzálek, and Peter Rossmanith. Digraph width measures in parameterized algorithmics. *Discret. Appl. Math.*, 168:88–107, 2014. doi:10.1016/J.DAM.2013.10.038.

**17** Robert Ganian, Petr Hlinený, Joachim Kneis, Daniel Meister, Jan Obdrzálek, Peter Rossmanith, and Somnath Sikdar. Are there any good digraph width measures? *J. Comb. Theory, Ser. B*, 116:250–286, 2016. doi:10.1016/J.JCTB.2015.09.001.

**18** Robert Ganian and Sebastian Ordyniak. The power of cut-based parameters for computing edge-disjoint paths. *Algorithmica*, 83(2):726–752, 2021. doi:10.1007/S00453-020-00772-W.

**19** Robert Ganian, Sebastian Ordyniak, and M. S. Ramanujan. On structural parameterizations of the edge disjoint paths problem. *Algorithmica*, 83(6):1605–1637, 2021. doi:10.1007/S00453-020-00795-3.

**20** Michael R. Garey and David S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman, N.Y., 1979.

**21** Tzvika Geft and Dan Halperin. Refined hardness of distance-optimal multi-agent path finding. In Piotr Faliszewski, Viviana Mascardi, Catherine Pelachaud, and Matthew E. Taylor, editors, *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022*, pages 481–488. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2022. `doi:10.5555/3535850.3535905`.

**22** Petr A. Golovach and Dimitrios M. Thilikos. Paths of bounded length and their cuts: Parameterized complexity and algorithms. *Discret. Optim.*, 8(1):72–86, 2011. `doi:10.1016/J.DISOPT.2010.09.009`.

**23** Heather Hulett, Todd G. Will, and Gerhard J. Woeginger. Multigraph realizations of degree sequences: Maximization is easy, minimization is hard. *Operations Research Letters*, 36(5):594–596, September 2008. `doi:10.1016/j.orl.2008.05.004`.

**24** Tuukka Korhonen and Daniel Lokshtanov. An improved parameterized algorithm for treewidth. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 528–541. ACM, 2023. `doi:10.1145/3564246.3585245`.

**25** Daniel Kornhauser, Gary L. Miller, and Paul G. Spirakis. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications. In *25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984*, pages 241–250. IEEE Computer Society, 1984. `doi:10.1109/SFCS.1984.715921`.

**26** Cristiana L. Lara, Jochen Könemann, Yisu Nie, and Cid C. de Souza. Scalable timing-aware network design via lagrangian decomposition. *Eur. J. Oper. Res.*, 309(1):152–169, 2023. `doi:10.1016/J.EJOR.2023.01.018`.

**27** Daniel Lokshtanov, Pranabendu Misra, Michal Pilipczuk, Saket Saurabh, and Meirav Zehavi. An exponential time parameterized algorithm for planar disjoint paths. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1307–1316. ACM, 2020. `doi:10.1145/3357713.3384250`.

**28** Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Efficient graph minors theory and parameterized algorithms for (planar) disjoint paths. In Fedor V. Fomin, Stefan Kratsch, and Erik Jan van Leeuwen, editors, *Treewidth, Kernels, and Algorithms - Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, volume 12160 of *Lecture Notes in Computer Science*, pages 112–128. Springer, 2020. `doi:10.1007/978-3-030-42071-0_9`.

**29** Raul Lopes and Ignasi Sau. A relaxation of the directed disjoint paths problem: A global congestion metric helps. *Theor. Comput. Sci.*, 898:75–91, 2022. `doi:10.1016/J.TCS.2021.10.023`.

**30** Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. `doi:10.1007/978-3-642-27875-4`.

**31** F. P. Ramsey. On a Problem of Formal Logic. *Proc. London Math. Soc. (2)*, 30(4):264–286, 1929. `doi:10.1112/plms/s2-30.1.264`.

**32** Neil Robertson and Paul D. Seymour. Graph minors .xiii. the disjoint paths problem. *J. Comb. Theory, Ser. B*, 63(1):65–110, 1995. `doi:10.1006/JCTB.1995.1006`.

**33** Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Roman Barták, and Eli Boyarski. Multi-agent pathfinding: Definitions, variants, and benchmarks. In Pavel Surynek and William Yeoh, editors, *Proceedings of the Twelfth International Symposium on Combinatorial Search, SOCS 2019, Napa, California, 16-17 July 2019*, pages 151–158. AAAI Press, 2019. `doi:10.1609/SOCS.V10I1.18510`.

34 Madison Van Dyk, Kim Klause, Jochen Koenemann, and Nicole Megow. Fast combinatorial algorithms for efficient sortation. In *Integer programming and combinatorial optimization*, volume 14679 of *Lecture Notes in Comput. Sci.*, pages 418–432. Springer, Cham, [2024] ©2024. doi:10.1007/978-3-031-59835-7_31.

35 Meirav Zehavi. Maximum minimal vertex cover parameterized by vertex cover. *SIAM J. Discret. Math.*, 31(4):2440–2456, 2017. doi:10.1137/16M109017X.