Minimizing Movement: Fixed-Parameter Tractability

Erik D. Demaine[1], MohammadTaghi Hajiaghayi[2], and Dániel Marx[3*]

[1] MIT Computer Science and Artificial Intelligence Laboratory,
32 Vassar St., Cambridge, MA 02139, USA, edemaine@mit.edu
[2] AT&T Labs — Research, 180 Park Ave.,
Florham Park, NJ 07932, USA, hajiagha@research.att.com
[3] Department of Computer Science and Information Theory, Budapest University of Technology and Economics, Budapest H-1521, Hungary, dmarx@cs.bme.hu

Abstract. We study an extensive class of movement minimization problems which arise from many practical scenarios but so far have little theoretical study. In general, these problems involve planning the coordinated motion of a collection of agents (representing robots, people, map labels, network messages, etc.) to achieve a global property in the network while minimizing the maximum or average movement (expended energy). The only previous theoretical results about this class of problems are about approximation, and mainly negative: many movement problems of interest have polynomial inapproximability. Given that the number of mobile agents is typically much smaller than the complexity of the environment, we turn to fixed-parameter tractability. We characterize the boundary between tractable and intractable movement problems in a very general set up: it turns out the complexity of the problem fundamentally depends on the treewidth of the minimal configurations. Thus the complexity of a particular problem can be determined by answering a purely combinatorial question. Using our general tools, we determine the complexity of several concrete problems and fortunately show that many movement problems of interest can be solved efficiently.

1 Introduction

In many applications, we have a relatively small number of mobile agents (e.g., a team of autonomous robots or people) moving cooperatively in a vast terrain or complex building to achieve some task. The number of cooperative agents is often small because of their expense: only small groups of people (e.g., emergency response or SWAT teams) can effectively cooperate, and autonomous mobile robots are currently quite expensive (in contrast to, e.g., immobile sensors). Nonetheless, an accurate model of the immense/intricate environment they traverse, and their ability to communicate or otherwise interact (say, by limited-range wireless radios or walkie-talkies), is complicated and results in a large problem input.

* Supported by the Magyary Zoltán Felsőoktatási Közalapítvány and the Hungarian National Research Fund (OTKA grant 67651).

Thus, to compute the most energy-efficient motion in such a scenario, we allow the running time to be relatively large (exponential) in the number of agents, but it must be small (polynomial or even linear) in the complexity of the environment. This set up motivates the study of *fixed-parameter tractability* (FPT) [10] for minimizing movement, with running time $f(k) \cdot n^{O(1)}$ for some function f, parameterized by the number k of mobile agents.

A movement minimization problem is defined by a class of target configurations that we wish the mobile agents to form and a movement objective function. For example, we may wish to move the agents to form a connected network (for communication), an independent set (either dispersing robots or placing map labels), or another topology. See Section 5 for more formal examples of problems and how our theory applies to them.

In the general formulation of the movement problem, we are given an arbitrary metric defining feasible motion, a graph defining "connectivity" (possibly according to the infinite Euclidean plane), and a desired property of the connectivity among the agents defined by a class $\mathcal{G}$ of graphs. We view the agents as "pebbles" located at vertices of the connectivity graph (and we use the two terms interchangeably). Our goal is to move the agents so that they induce a subgraph of the connectivity graph that possesses the desired property, that is, belongs to the class $\mathcal{G}$. There are three natural measures of agent motion that we might want to minimize: the total amount of motion, the maximum motion of any agent, and the number of moved agents. To obtain further generality and to model a wider range of problems, we augment this model with additional features: the agents have types, desired solutions can require certain types of agents, multiple agents can be located at the same vertex, and the cost of the movement can be different (even nonmetric) for the different agents.

To what level of generality can we solve these movement problems? Several versions have been studied from an approximation algorithms perspective in SODA 2007 [7] and FOCS 2008 [8], in addition to various specific problems considered less formally in practical scenarios [2,4,5,9,12,13,14]. Unfortunately, most forms of the movement problem are NP-complete, and furthermore are often hard to approximate even within polynomial factors [7]. Nonetheless, the problems are of significant practical interest, and the motion must be kept small in order to minimize energy consumption. Fortunately, as motivated above, the number of mobile agents is often small. Thus we have a natural context for considering fixed-parameter algorithms, i.e., algorithms with running time $f(k) \cdot n^{O(1)}$, where parameter k is the number of mobile agents.

2 Main Results

We develop general efficient fixed-parameter algorithms for a broad family of movement problems. Furthermore, we show our results are tight by characterizing, in a very general setting, the line between fixed-parameter tractability and intractability. It turns out that the notion of treewidth plays an important role in defining this boundary line. Specifically we show that, for problems closed under edge addition (i.e., adding an edge to the connectivity graph cannot destroy a solution), the complexity of the problem depends solely on whether the edge-deletion minimal graphs of the property have bounded treewidth. If they all have bounded treewidth, we show how to solve a very general formulation of the problem with an efficient fixed-parameter algorithm. If they have unbounded treewidth, we show that even very simple questions are W[1]-hard, meaning there is no efficient fixed-parameter algorithm under the standard parameterized complexity assumption FPT $\neq$ W[1]. In Section 5, we use these results to characterize the complexity of several concrete problems.

Our results apply to a more general model of agents, which in particular lets us capture facility-location types of problems where the number of facilities can be arbitrary large (not a fixed parameter). Such problems arise, e.g., in organizing a small team within a large infrastructure of wired network hubs or mobile satellites. The general model we consider divides the agents into three types—client, facility, and obnoxious agents—and the parameter is just the number of clients, which can be much smaller than the total number of agents. The clients can require collocated or nearby *facility agents*, among a potentially large set of facility agents, which themselves are mobile. Intuitively, facilities provide some service needed by clients. Clients can also require at most a certain number (e.g., zero) of collocated *obnoxious agents* (again among a potentially large, mobile set), which can represent dangerous or undesirable resources. In other words, adding facility agents or removing obnoxious agents does not affect a solution. More generally, there can be many different subtypes of client, facility, and obnoxious agents, and we may require a particular pattern of these types.

Formally, our results are as follows. A movement problem specifies a *multicolored graph property*: an (infinite) set $\mathcal{G}$ of desired configurations, each specifying a desired subgraph and how that subgraph should be populated by different types of agents (a *multicolored graph*). In this way, we can specify different types of client agents that need to interact in a particular way, or need particular types of nearby facility agents. The goal of the *movement problem* is to move the agents into a configuration containing at most ℓ vertices that contain all k client agents and induce a "good" target pattern: either the induced multicolored graph is in the set $\mathcal{G}$ or it is better than some multicolored graph $G \in \mathcal{G}$, i.e., contains more facility agents and fewer obnoxious agents at each vertex.

A mild technical condition that we require is that the multicolored graph property $\mathcal{G}$ is *regular*: for every fixed numbers k and ℓ, there are only finitely many graphs in $\mathcal{G}$ with at most ℓ vertices and at most k client agents (as we do not bound the number of obnoxious and facility agents here, this is a nontrivial restriction). In other words, there should be only finitely many minimal ways to satisfy a bounded number of clients in a bounded subgraph. For example, the propery requiring that the number of facility agents is not less than the number of obnoxious agents is *not* a regular property. Note that this restriction does not say that there is only a finite number of good configurations: as mentioned in the previous paragraph, we allow configurations having any number of extra facility vertices. Furthermore, our main algorithmic result considers properties that are closed under edge addition; this is certainly true for properties that model some notion of connectivity.

Theorem 1. *If $\mathcal{G}$ is a regular multicolored graph property that is closed under edge addition, and if the edge-deletion minimal graphs in $\mathcal{G}$ have bounded treewidth, then the movement problem can be solved in $f(k, \ell) \cdot n^{O(1)}$ time, assuming that the movement cost function is the same on any two agents of the same obnoxious type that are initially located on the same vertex.*

Here the *movement cost function* is an arbitrary (polynomially computable) function for each agent specifying the nonnegative integer cost of moving that agent to each vertex in the graph. This definition allows nonmetric terrains, agents of different speeds, immobile agents, regions impassable by certain agents, etc. In the movement problem, we are given an initial configuration (a multicolored graph), and we wish to minimize the total cost of all movement subject to reaching one of the desired target configurations in $\mathcal{G}$ with at most ℓ vertices, where both ℓ and the number k of client agents are parameters. This problem in particular captures the variations of minimizing the maximum movement and minimizing the number of moved agents. For the latter, we simply specify a movement cost function for each agent of 0 to remain stationary and 1 to make any move. For the former, we can binary search on the maximum movement cost τ, and modify the movement cost function to jump to ∞ whenever exceeding τ.

Our main algorithm uses several tools from fixed-parameter tractability, color coding, and graph structure theory, in particular treewidth. This combination of techniques seems interesting in its own right.

We prove a matching hardness result for Theorem 1: if the edge-deletion minimal graphs in $\mathcal{G}$ have unbounded treewidth, then it is hard to answer even some very simple questions. Thus treewidth plays an essential role in the complexity of the problem, which is not apparent at first sight.

Theorem 2. *If $\mathcal{G}$ is any (possibly regular) multicolored graph property that is closed under edge addition, and for every $w \geq 1$, there is an edge-deletion minimal graph $G_w \in \mathcal{G}$ with treewidth at least w and at least one client agent on each vertex (but no other type of agent), then the movement problem is* W[1]*-hard with the combined parameter (k, ℓ), already in the special case where each agent is allowed to move at most one step.*

3 Further Results

In addition to our general classification, we present many additional fixed-parameter results. These results capture situations where the general classification cannot be applied directly, or the general results apply but problem-specific approaches enable more efficient algorithms. Specifically, we consider situations where the graphs are more specific (e.g., almost planar), the property is not closed under edge addition, or the number of client agents is not bounded. Our aim is to demonstrate that there are many problem variants that can be explored and that there is a vast array of algorithmic techniques that become relevant when studying movement problems. In particular, the fast set convolution algorithm of Björklund et al., results from algorithmic graph minor theory, Courcelle's Theorem, bidimensionality, Canny's Roadmap Algorithm, and a result of Khot and Raman all find uses in this framework.

Planar graphs and H-minor-free graphs. Our general characterization makes no assumptions on the connectivity structure: it is an arbitrary graph. However, significantly stronger results can be achieved if we have some restriction on the connectivity graph. For example, many road networks, fiber networks, and building floorplans can be accurately represented by planar graphs. We show that, for planar graphs, the fixed-parameter algorithms of Theorem 1 work even if we remove the requirement that $\mathcal{G}$ is closed under edge addition.

In many cases, approximation and fixed-parameter tractability results for planar graphs generalize to arbitrary surfaces, to bounded local treewidth graphs, and to H-minor-free graph classes. These generalizations are made possible by the algorithmic consequences of the Graph Minor Theorem [6]. To obtain maximum generality, we state the result on planar graphs generalized to arbitrary H-minor-free classes:

Theorem 3. *If $\mathcal{G}$ is a regular multicolored graph property, then for every fixed graph H, the movement problem can be solved on H-minor-free graphs in $f(k,\ell)\cdot n^{O(1)}$ time, assuming that the movement cost function is the same on any two agents of the same obnoxious type that are initially located on the same vertex.*

One possible application scenario where these generalizations of planar graphs play a role is the following. The terrain is a multi-level building, where the connectivity graph is planar on each level, and there are at most d connections between two adjacent levels. Now the graph is K_{d+1}-free for $d \geq 4$ (as a K_{d+1} minor would be contained on one level). Thus, for every fixed value of d, Theorem 3 applies for such connectivity graphs.

We also consider two specific problems in the context of planar graphs.

Bidimensionality. We consider parameterizing by the sum of all movement, instead of the number of pebbles, for the problem of DISPERSION (see Section 5). This parameterization is likely hard in general, but we show that it becomes fixed-parameter tractable in planar graphs, even in linear time (for every fixed maximum sum k). The proof uses a combination of bidimensionality theory, parameter-treewidth bounds, grid-minor theorems, Courcelle's Theorem, and monadic second-order logic.

Planar STEINER CONNECTIVITY. In the STEINER CONNECTIVITY problem (see Section 5), the goal is to connect one type of agents ("terminals") using another type of agents ("connectors"). Our general characterization shows that this problem is fixed-parameter tractable if the numbers of both types of agents are bounded. The problem becomes W[1]-hard if only the number of connector agents is bounded and the number of terminal pebbles is unbounded. On the other hand, we show that this version of the problem is fixed-parameter tractable for planar graphs, using problem-specific techniques.

Geometric graphs. In some of the applications, the environment can be naturally modeled by the infinite geometric graph defined by Euclidean space, where vertices correspond to points and edges connect two vertices that are within a fixed distance of each other, say 1. In this case, we develop efficient algorithms in a very general setting, even though the graph is infinite:

Theorem 4. *If $\mathcal{G}$ is any regular graph property, then the movement problem can be solved in Euclidean d-space up to multiplicative error $1 + \varepsilon$ in $f(k, d) \cdot n^{O(1)} \lg(1/\varepsilon)$ time, where k is the total number of agents (including facility and obnoxious agents).*

The main tool for proving this theorem is Canny's Roadmap Algorithm for motion planning in Euclidean space [3], which lets us manipulate bounded-size semi-algebraic sets.

Hereditary properties. In addition to properties closed under edge addition, we investigate another general class of properties, *hereditary properties*, where if some $G \in \mathcal{G}$, then every induced subgraph of G is also in the property $\mathcal{G}$. For example, independence (having no edges) is such a property. We prove another general hardness result for hereditary properties:

Theorem 5. *Let $\mathcal{G}$ be a hereditary property where each vertex has exactly one client pebble and there are no other type of pebbles. If the maximum clique size is bounded in $\mathcal{G}$, then the movement problem is* W[1]*-hard with the combined parameter (k, ℓ), already in the special case where each agent is allowed to move at most one step in the graph.*

The proof of Theorem 5 uses a hardness result by Khot and Raman [11] on the parameterized complexity of finding induced subgraphs with hereditary properties. The theorem in particular establishes W[1]-hardness of DISPERSION (moving to an independent set); see Section 5.

Improving CONNECTIVITY with fast subset convolution. Finally, we optimize one particularly practical problem, CONNECTIVITY: moving the agents so that they form a connected subgraph. Our general characterization implies that this problem is fixed-parameter tractable. Using the recent algorithm of Björklund et al. [1] for fast subset convolution in the min-sum semiring, we design a more efficient algorithm for this problem: the exponential factor of the running time is only $O(2^k)$.

In summary, our results form a systematic study of the movement problem, using powerful tools to classify the complexity of the different variants. Our algorithms are general, so may not be optimal for any specific version of the problem, but they nonetheless characterize which problems are tractable, and lead the way for future investigation into more efficient algorithms for practical special cases.

4 Model and Definitions

In this section, we make precise the model described in the Introduction and introduce some additional notation.

Definition 1. *We fix three finite sets of colors: C_m (main colors), C_f (facility colors), C_o (obnoxious colors).*

Definition 2. *A* multicolored graph *is a graph with a multiset of colored pebbles assigned to each vertex (a vertex can be assigned multiple pebbles with the same color). We denote by $n_G(c, v)$ the number of pebbles with color c at vertex v in G. A* multicolored graph property *is a (possibly infinite) recursively enumerable set $\mathcal{G}$ of multicolored graphs. A graph property $\mathcal{G}$ is* regular *if for every fixed k, ℓ there is only a finite number of graphs in $\mathcal{G}$ with at most ℓ vertices and at most k main pebbles and there is an algorithm that, given k and ℓ, enumerates these graphs. A graph property $\mathcal{G}$ is* hereditary *if, for every $G \in \mathcal{G}$, every induced subgraph of G is also in $\mathcal{G}$. A graph property $\mathcal{G}$ is* closed under edge addition *if whenever G is in $\mathcal{G}$ and G' is obtained from G by connecting two nonadjacent vertices, then G' is also in $\mathcal{G}$. A graph $G \in \mathcal{G}$ is* edge-deletion minimal *if there is no graph $G' \in \mathcal{G}$ that can obtained from G by edge deletions.*

Definition 3. *Let G_1 and G_2 be two multicolored graphs whose underlying graphs are isomorphic. G_2* dominates *G_1 if there is an isomorphism $\phi : V(G_1) \to V(G_2)$ such that, for every $v \in V(G_1)$,*

1. *for every $c \in C_m$, vertices v and $\phi(v)$ have the same number of pebbles with color c;*
2. *for every $c \in C_f$, vertex $\phi(v)$ has at least as many pebbles with color c as v; and*
3. *for every $c \in C_o$, vertex $\phi(v)$ has at most as many pebbles with color c as vertex v.*

Definition 4. *For every set $\mathcal{G}$ of multicolored graphs, the* movement problem *has the following inputs:*

1. *a multicolored graph $G(V, E)$, P is the set of pebbles, k is the number of main pebbles;*
2. *a movement cost function $c_p : V \to Z^+$ for each pebble $p \in P$;*
3. *integer ℓ, the maximum solution size; and*
4. *integer C, the maximum cost.*

The task is to find a movement plan $m : P \to V$ such that

1. *the total cost $\sum_{p \in P} c_p(m(p))$ of the moves is at most C; and*
2. *after the movements, there is a set S of at most ℓ vertices such that S contains all the main pebbles and the multicolored graph $G[S]$ dominates some graph in $\mathcal{G}$.*

By using different movement cost functions, we can express various goals:

1. if $c_p(v)$ is the distance of p from v, then we have to minimize the sum of movements,
2. if $c_p(v) = 0$ if v is at distance at most d from p and ∞ otherwise, then we have to find a solution where p moves at most d steps,

3. if $c_p(v) = 0$ if v is the initial location of p and $c_p(v) = 1$ for every other vertex, then we have to minimize the number of pebbles that move.

Of course, we can express combinations of these goals or the different pebbles can have different movement graphs, etc. The formulation is very flexible.

5 Sample Problems of Interest

To illustrate the generality of our model and characterization, we define several specific movement problems similar to those mentioned informally in the Introduction, and determine their fixed-parameter tractability using Theorems 1 and 2. Using these tools, if a movement problem can be modeled with colored pebbles and the target patterns are closed under adding edges, then the complexity of the problem can be determined by solving the (sometimes nontrivial) combinatorial question of whether the minimal configurations have bounded treewidth. The minimal configurations are those pebbled graphs that are acceptable solutions, but removing any edge makes them unacceptable.

Example: CONNECTIVITY. Move the pebbles (agents) so that they are connected and on distinct vertices. The parameter is the number k of pebbles. Now there is only one, main color of pebbles, and $\mathcal{G}$ contains all connected graphs with exactly one pebble on each vertex. Clearly, $\mathcal{G}$ is closed under edge addition and the edge-deletion minimal graphs are trees. Trees have treewidth 1, hence by Theorem 1, this movement problem is fixed-parameter tractable for any movement cost function. The variant of the problem where it is not required that the pebbles are on distinct vertices is also FPT: in this case, $\mathcal{G}$ contains all connected graphs with *at least* one pebble on each vertex. □

Example: GRID. Move the k pebbles so that they form a $\lfloor\sqrt{k}\rfloor \times \lfloor\sqrt{k}\rfloor$ square grid. The parameter is the number k of pebbles. Again there is only one, main color of pebbles, and $\mathcal{G}$ contains all graphs containing a spanning square grid subgraph with exactly one pebble on each vertex. Clearly, $\mathcal{G}$ is closed under edge addition and the edge-deletion minimal graphs are grids, which have arbitrarily large treewidth. Thus Theorem 2 implies that it is W[1]-hard, parameterized by (k, ℓ), to decide whether there is a solution where each pebble moves at most one step. □

Example: s-t CONNECTIVITY (few pebbles). Move the pebbles to form a path of pebbled vertices between fixed vertices s and t. The parameter is the number k of pebbles. Now there are two main colors of pebbles, call them red and blue, and $\mathcal{G}$ consists of all graphs containing exactly two red pebbles and a path between them using only vertices with blue pebbles. We reduce s-t CONNECTIVITY to this movement problem by putting red pebbles at s and t, and giving them an infinite movement cost to any other vertices. Clearly, $\mathcal{G}$ is closed under edge addition and the edge-deletion minimal graphs are paths. Paths have treewidth 1, so by Theorem 1, this problem is fixed-parameter tractable. □

In the next example, we show that a much more general version of s-t CONNECTIVITY is FPT: instead of parameterizing by the number k of pebbles, we can parameterize by the maximum length L of the path. Thus we can have arbitrarily many pebbles that might form the path, as long as the formed path itself is small.

Example: s-t CONNECTIVITY (bounded length). Move the pebbles to form a path of pebbled vertices of length at most L between fixed vertices s and t. The parameter is the length L. Now we define one main color of pebbles, red, and one facility color of pebbles, blue, and we define $\mathcal{G}$ as in the previous example. Again by Theorem 1, this problem is fixed-parameter tractable in the combined parameter (k, ℓ); in the example, we have $k = 2$ and $\ell = L + 1$. □

Example: STEINER CONNECTIVITY. Connect the red pebbles (representing terminals) by moving the blue pebbles to form a Steiner tree. The parameter is the number of red pebbles plus the number of blue pebbles in the *solution* Steiner tree. This is simply a generalization of s-t CONNECTIVITY to more than two red pebbles. Again by Theorem 1 the problem is fixed-parameter tractable (the edge-deletion minimal graphs are trees), even when the number of blue pebbles is very large. □

Example: 2-CONNECTIVITY. Move the pebbles so that they induce a 2-connected graph and the pebbles are on distinct vertices. The parameter is the number k of pebbles. Now $\mathcal{G}$ contains all 2-connected graphs and clearly $\mathcal{G}$ is closed under edge addition. The edge-deletion minimal graphs have unbounded treewidth: subdividing every edge of a clique gives an edge-deletion-minimal 2-connected graph. Thus by Theorem 2, it is W[1]-hard to decide whether there is a solution where each pebble moves at most one step. □

Example: s-t d-CONNECTIVITY. Move the pebbles so that there are d vertex-disjoint paths using pebbled vertices between two fixed vertices s and t. The parameter is the total length L of the solution paths. Now we use one main color, red, and one facility color, blue, and $\mathcal{G}_d$ consists of all graphs containing two vertices with a red pebble on each, and having d vertex-disjoint paths between these two vertices, with blue pebbles on each path vertex. In the input instance, there are red pebbles on s and t, and the cost of moving them is infinite. Clearly, $\mathcal{G}_d$ is closed under edge addition and the edge-deletion minimal graphs are series-parallel (as they consist of d internally vertex disjoint paths connecting two vertices), which have treewidth 2. Hence, by Theorem 1, this movement problem is fixed-parameter tractable with respect to L, for every fixed d. Again the number of blue pebbles can be arbitrarily large. □

The previous example shows that s-t d-CONNECTIVITY is FPT for every fixed value of d. Furthermore, we can show that the problem remains FPT even if d appears as part of the input.

Example: s-t d-CONNECTIVITY (unbounded version). Move the pebbles so that there are d vertex-disjoint paths using pebbled vertices between two fixed vertices s and t, where d is a number given in the input. The parameter is the total length L of the solution paths. First, if d is larger than the bound on the total length of the paths, then there is no solution. Otherwise, we can assume d is a fixed parameter. Now we use two main colors, red and green, and one facility color, blue. A graph G is in $\mathcal{G}$ if the blue pebbles form d vertex-disjoint paths between two vertices containing red pebbles, where d is the number of green pebbles in G. Thus we use green pebbles to "label" a graph G in $\mathcal{G}$ according to what level of connectivity it attains. Again $\mathcal{G}$ is closed under edge addition and the edge-deletion minimal graphs are series-parallel, which have treewidth 2, so by Theorem 1, the movement problem is fixed-parameter tractable with respect to $k := 2$ and $\ell := L$. In the initial configuration, we put red pebbles on s and t with infinite movement cost, and we place d green pebbles arbitrarily in the graph. The target configuration we obtain will have exactly d green pebbles, and thus d vertex-disjoint paths, because these are main pebbles. □

We can also consider the edge-disjoint version of s-t connectivity. We need the following combinatorial lemma to characterize the minimal graphs:

Lemma 6. *Let G be a connected graph and assume that there are d edge-disjoint paths between vertices s and t in G, but for any edge $e \in E(G)$, there are at most $d-1$ edge-disjoint paths between s and t in $G \setminus e$. Then the treewidth of G is at most $O(d^2)$.*

Example: s-t d-EDGE-CONNECTIVITY. Move the pebbles so that there are d edge-disjoint paths of pebbled vertices between s and t. The parameter is the total length L of the paths. Now we use one main color, red, and one facility color, blue, and $\mathcal{G}_d$ contains all graphs containing two vertices with a red pebble on each and having d edge-disjoint paths between these two vertices, with blue pebbles on each path vertex. By Lemma 6, the edge-deletion minimal graphs have treewidth $O(d^2)$. Hence, by Theorem 1, the movement problem is fixed-parameter tractable with respect to L. □

The previous example shows that s-t d-EDGE-CONNECTIVITY is FPT for every fixed value of d. Somewhat surprisingly, unlike in the vertex-disjoint case, the problem becomes hard if d is part of the input:

Example: s-t d-EDGE-CONNECTIVITY (unbounded version). Move the pebbles so that there are d edge-disjoint paths of pebbled vertices between s and t, where d is a number given in the input. We use three main colors: red, green, and blue. A graph G is in $\mathcal{G}$ if the blue pebbles form d edge-disjoint paths between two vertices containing red pebbles, where d is the number of green pebbles in G. We show that $\mathcal{G}$ contains edge-deletion minimal graphs of arbitrary large treewidth, so by Theorem 2, it is W[1]-hard to decide whether there is a solution where each of the k pebbles move at most one step each. Assume d is even and let G be a graph consisting of vertices s, t, and d vertex-disjoint paths between s and t such that vertices $p_{i,1}, \ldots, p_{i,d}$ are the internal vertices of the ith path. Now for every odd i and odd j, identify vertices $p_{i,j}$ and $p_{i+1,j}$, and for every even $i < d$ and even j, identify $p_{i,j}$ and $p_{i+1,j}$. There are d edge-disjoint s-t paths in this graph, but there are at most $d-1$ such paths after the deletion of every edge. (It is easy to see that every edge is in an s-t cut of exactly d edges.) Thus G is an edge-deletion minimal member of $\mathcal{G}$. Furthermore, if for every odd i and odd j, we contract the edge $p_{i,j}p_{i,j+1}$, then we get a $d/2 \times d/2$ grid, so the treewidth is $\Omega(d)$. □

Example: FACILITY LOCATION (collocation version). Move client and facility pebbles so that each client pebble is collocated with at least one facility pebble and the client pebbles are at distinct locations. The parameter is the number of client pebbles. We use one main color, red, for the clients, and one facility color, blue, for the facilities, and $\mathcal{G}$ contains all graphs in which every vertex contains exactly one red and one blue pebble. The edge-deletion minimal graphs in $\mathcal{G}$ have no edges, so have treewidth 0. By Theorem 1, the movement problem is fixed-parameter tractable parameterized by the number of main pebbles, i.e., the number of clients. The number of facilities can be unbounded, which is useful, e.g., to organize a small team within a large infrastructure of wired network hubs or mobile satellites. □

Example: FACILITY LOCATION (distance-d version). Move client and facility pebbles so that each client pebble is within distance at most d from at least one facility pebble and the client pebbles are at distinct locations. Now we use two main colors, red and green, and one facility color, blue. Let $\mathcal{G}$ contain all graphs that contain some number d of green pebbles and each red pebble is at distance at most d from some blue pebble. Given a graph with k client (red) pebbles and some number of facility (blue) pebbles, we add d dummy green pebbles and ask whether there is a solution on $\ell := k(d+1)+d$ vertices. If we move the pebbles so that each red pebble is at distance d from some blue pebble, then there are $k(d+1)+d$ vertices that contain all d of the green pebbles and induce a graph in $\mathcal{G}$. We claim that the edge-deletion minimal graphs in $\mathcal{G}$ are forests, and hence have treewidth 1. Consider an edge-deletion minimal graph $G \in \mathcal{G}$, and for each vertex v without a blue pebble, select an edge uv that goes to a neighbor u that is closer to some blue pebble than v. If an edge is not selected in this process, then it can be removed (it does not change the distance to the blue pebbles), so by the minimality of G, every edge is selected. Each connected component contains at least one blue pebble. This means that, in each connected component, the number of selected edges is strictly smaller than the number of vertices, i.e., each component is a tree. Thus, by Theorem 1, the movement problem is FPT. □

On the other hand, FACILITY LOCATION becomes W[2]-hard if the parameter is the number of facilities, while the number of clients can be unbounded. We cannot obtain this result using Theorem 2 because, in this setting, the parameter is the number of facility pebbles.

Theorem 7. *For every fixed $d \geq 0$, FACILITY LOCATION (distance d version) is* W[2]*-hard parameterized by the number of facilities, even if each pebble is allowed to move at most one step in the graph.*

Example: MATCHING. Move the pebbles so that the pebbles are on distinct vertices and there is a perfect matching in the graph induced by the pebbles. The parameter is the number of pebbles. Now there is just one, main pebble color, and $\mathcal{G}$ contains all graphs that have a perfect matching. The edge-deletion minimal graphs are perfect matchings, so they have treewidth 1. By Theorem 1, the movement problem is FPT. □

Example: SEPARATION. Move client pebbles (say, representing population) and/or obnoxious pebbles (say, representing power plants) so that each client pebble is collocated with at most o obnoxious pebbles. The parameter is the number of client pebbles. Here $\mathcal{G}$ contains all graphs with the desired bounds, so the edge-deletion minimal graphs have no edges, which have treewidth 0. By Theorem 1, the movement problem is fixed-parameter tractable. As in previous examples, we can make o an input to the problem. □

Example: DISPERSION. Move the pebbles to distinct vertices and such that no two pebbles are adjacent. The parameter is the number k of pebbles. Here $\mathcal{G}$ contains all independent sets with exactly one pebble on each vertex. Because $\mathcal{G}$ is hereditary and the maximum clique size is 1, Theorem 5 implies that the movement problem is W[1]-hard, even in the case when each pebble is allowed to move at most one step. □

References

1. A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: fast subset convolution. In *STOC 2007*, pp. 67–74.
2. J. L. Bredin, E. D. Demaine, M. Hajiaghayi, and D. Rus. Deploying sensor networks with guaranteed capacity and fault tolerance. In *MOBIHOC 2005*, pp. 309–319.
3. J. F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, 1987.
4. P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme. Autonomous deployment of a sensor network using an unmanned aerial vehicle. In *ICRA 2004*, New Orleans, USA.
5. P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme. Deployment and connectivity repair of a sensor net with a flying robot. In *ISER 2004*, Singapore.
6. E. D. Demaine, M. Hajiaghayi, and K. Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *FOCS 2005*, pp. 637–646.
7. E. D. Demaine, M. Hajiaghayi, H. Mahini, A. S. Sayedi-Roshkhar, S. Oveisgharan, and M. Zadimoghaddam. Minimizing movement. *ACM Trans. Algorithms*.
8. Z. Friggstad and M. R. Salavatipour. Minimizing movement in mobile facility location problems. In *FOCS 2008*, pp. 357–366.
9. T.-R. Hsiang, E. M. Arkin, M. A. Bender, S. P. Fekete, and J. S. B. Mitchell. Algorithms for rapidly dispersing robot swarms in unknown environments. In *WAFR 2003*, pp. 77–94.
10. F. Hüffner, R. Niedermeier, and S. Wernicke. Techniques for practical fixed-parameter algorithms. *Comput. J.*, 51(1):7–25, 2008.
11. S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theoret. Comput. Sci.*, 289(2):997–1008, 2002.
12. S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
13. J. H. Reif and H. Wang. Social potential fields: a distributed behavioral control for autonomous robots. In *WAFR 2005*, pp. 331–345.
14. A. C. Schultz, L. E. Parker, and F. E. Schneider, editors. *Multi-Robot Systems: From Swarms to Intelligent Automata*. Springer, 2003.