Towards an Algorithmic Guide to Spiral Galaxies

Guillaume Fertin, Shahrad Jamshidi, Christian Komusiewicz

Towards an Algorithmic Guide to Spiral Galaxies[☆]

Guillaume Fertin[a,*], Shahrad Jamshidi[a], Christian Komusiewicz[b,1,*]

[a] *Université de Nantes, LINA - UMR CNRS 6241, France.*
[b] *Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, D-10587 Berlin, Germany.*

Abstract

We study the one-player game Spiral Galaxies from an algorithmic viewpoint. Spiral Galaxies has been shown to be NP-hard [Friedman, 2002] more than a decade ago, but so far it seems that no one has dared exploring its algorithmic universe. We take this trip and visit some of its corners.

Keywords: NP-hard problems, fixed-parameter algorithms, one-player game

1. Orbital Launch

Spiral Galaxies (also called Tentai Show) is a one-player game, described as follows by the Help of its Linux version: "You have a rectangular grid containing a number of dots. Your aim is to draw edges along the grid lines which divide the rectangle into regions in such a way that every region is 180° rotationally symmetric, and contains exactly one dot which is located at its center of symmetry". Similarly to many other such puzzles (e.g., Sokoban, Sudoku), apart from being a discrete pastime in boring meetings, it is also a nice combinatorial and algorithmic problem that one might try to solve computationally. This trend has been developed, among others, by Demaine [1] who presented a survey over hardness results in games. For a more general introduction into combinatorial games, we also refer to the book by Hearn and Demaine [7].

Apart from the beauty (and fun!) aspect of such a study, this work is motivated by the fact that, while small to medium-size instances of Spiral Galaxies are still fun to solve, larger problems become just too hard. This can be frustrating for many players... and might even lead to fits of rage (something you may want to avoid in boring meetings). Hence, an automatic solver for Spiral Galaxies is highly desirable for these cases. In this paper, we thus visit the algorithmic universe of Spiral Galaxies. Our results are as follows.

☆A preliminary version of this article was presented at the *Seventh International conference on Fun with Algorithms (FUN '14)* held in July 2014 on Lipari Island, Sicily, Italy.

*Corresponding authors

[1]Supported by a post-doctorial grant funded by the Région Pays de la Loire and by the DAAD Procope program (project 55934856)

- In Section 2, we strengthen the previous NP-hardness result [6] for Spiral Galaxies by showing that it is NP-hard to decide whether there is a solution in which all galaxies have size at most seven.
- In Section 3, we present polynomial-time algorithms for several special cases of Spiral Galaxies such as: checking for solutions with galaxies of size at most two, checking for solutions that have only square galaxies, and checking for solutions that consist only of a certain type of trivially connected galaxies. We also provide exact exponential-time algorithms for the general case and for the case in which all galaxies are rectangles.
- In Section 4, we provide a polynomial-time algorithm for the case in which all galaxies are hole-free and the overall number of "galaxy corners" is bounded.

The latter two algorithms are fixed-parameter algorithms. Note that a problem with input size n is said to be *fixed-parameter tractable* with respect to a parameter k if it can be solved in $f(k) \cdot \text{poly}(n)$ time, where f is a computable function only depending on k. For an introduction to parameterized algorithmics refer to [2]. Before presenting our results we give a formal problem definition and introduce the notation used throughout the paper.

Problem Definition and Notation. We formalize the problem as follows. We have a two-dimensional universe U, where each field in U is described by the coordinate (i, j), where $i = 1, \ldots, N$ and $j = 1, \ldots, M$ for some $N, M \in \mathbb{N}$. We call two fields (i, j) and (i', j') *adjacent* if $|i-i'| = 1$ and $j-j' = 0$ or if $|j-j'| = 1$ and $i-i' = 0$. That is, a field is adjacent to the four fields that are directly left, right, above, and below this field. Let $n := |U|$ denote the number of fields. We furthermore are given a set G of k galaxy centers ('dots' in the description) along with the location $L : G \to \{1, 1.5, 2, 2.5, \ldots, N\} \times \{1, 1.5, 2, 2.5, \ldots, M\}$. We use the noninteger values to denote the case in which a galaxy center is located between two rows or columns. A galaxy center with noninteger coordinates is adjacent to all fields that can be obtained by rounding its location values. This is the input of Spiral Galaxies. For a galaxy center $g \in G$, we use $L_{\text{r}}(g)$ to denote the row coordinate of the center of g, and $L_{\text{c}}(g)$ to denote its column coordinate. We use the same notation to refer to the respective coordinates of a field f.

There are two natural ways of encoding the input of Spiral Galaxies. One is to present each field and each possible galaxy location as a position in a bit string of length $\Theta(n)$. Another way is to list the positions of the k galaxy centers and the dimensions of the universe. This representation has size $\Theta(k \cdot \log n)$ which is smaller than the first representation if $k = o(n/\log n)$. Hence, for our exact algorithms we assume that the input length is $\Theta(n)$ and for our fixed-parameter algorithms (which assume that k is small) we assume that the input length is $\Theta(k \cdot \log n)$.

A solution to Spiral Galaxies is given by assigning each field of U to some galaxy g. We describe this using the function $b : U \to G$. Before defining the

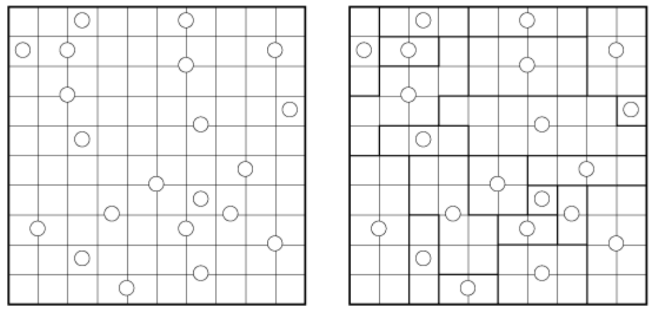

Figure 1: Two screenshots of a single SPIRAL GALAXIES scenario. Left: the field of squares U with the circular galaxy centers. Right: the corresponding solution. For example, $b(1,1) = g \in G$, where $L(g) = (2,1)$. Similarly, $b(1,5) = g' \in G$, where $L(g') = (1,6.5)$.

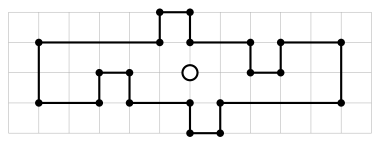

Figure 2: A galaxy with its corners marked as black dots.

properties of a solution, we give some further definitions. An *area* A is a subset of U. An area is called *connected* if between each field $(i,j) \in A$ and $(i',j') \in A$ there exists a path of adjacent fields that belong to A.

A total function $b : U \to G$ is a *solution* if it satisfies the following conditions:

- rotational symmetry, that is, if $b(i,j) = g$ then $b(i',j') = g$, where $i' = 2 \cdot (L_{\mathrm{r}}(g) - i) + i$ and $j' = 2 \cdot (L_{\mathrm{c}}(g) - j) + j$,
- connectivity, that is, $\{u \in U \mid b(u) = g\}$ is a connected area, and
- each galaxy center is contained in its galaxy, that is, $b(L_{\mathrm{r}}(g), L_{\mathrm{c}}(g)) = g$.

We call $(i',j') \in U$ the *g-twin* of (i,j). The notation is displayed in the screenshot in Figure 1.

A *corner* of an area A is a pair of noninteger coordinates (y,x) such that either one or three of the four neighboring fields $(\lceil y \rceil, \lceil x \rceil)$, $(\lceil y \rceil, \lfloor x \rfloor)$, $(\lfloor y \rfloor, \lceil x \rceil)$, and $(\lfloor y \rfloor, \lfloor x \rfloor)$ belongs to A (see Figure 2).

2. A "Binary Star"-like Hardness Proof

The original proof for showing the NP-hardness of SPIRAL GALAXIES uses a reduction from CIRCUIT SATISFIABILITY [6]. In this reduction, each gate is replaced by a gadget consisting of a constant number of galaxy centers. The SPIRAL GALAXIES instance then has a solution if and only if there is an assignment to the input variables of the circuit such that the output is true. The largest galaxy in such a solution has 22 fields. Thus, it is NP-hard to decide whether Spiral Galaxies has a solution with galaxies of size at most 22.

Our aim is to determine for which galaxy sizes the problem becomes tractable. As we will discuss in Section 3, it is trivial to decide whether there is a solution in which all galaxies have size at most two. To make progress towards a complexity dichotomy for Spiral Galaxies with respect to the maximum galaxy size we describe a new reduction. This reduction mostly follows the ideas of the original construction. Our main result is that is NP-hard to decide whether a Spiral Galaxies instance has a solution with galaxies of size at most seven.

The first difference to the old NP-hardness proof is that we do not reduce Circuit Satisfiability but the following, more constrained satisfiability problem.

> Positive Planar 1-in-3-SAT
> **Input:** A collection $\mathcal{C}$ of clauses, each containing three variables and a planar embedding of the clause-variable graph.
> **Question:** Is there an assignment to the variables such that each clause contains exactly one variable that is true?

Herein, the clause-variable graph has a vertex for each variable and each clause and an edge between a variable and a clause if the clause contains the variable. Positive Planar 1-in-3-SAT is NP-complete [9]; the embedding can be assumed to be rectilinear.

The general outline of the reduction of Positive Planar 1-in-3-SAT to Spiral Galaxies is as follows. We build an instance of Spiral Galaxies based on the planar embedding of the clause-variable graph. The instance will be composed of *variable gadgets* which will correspond to setting a variable to either true or false. Then, *wire gadgets* carry this truth value to the *clause gadgets*. The variable gadgets are placed at the position of the variables of the planar embedding of the clause-variable graph. The wire gadgets correspond to the edges of this embedding. Finally, the clause gadgets are placed at the positions of the clause vertices of the clause-variable graph. To ensure that the gadgets are far enough apart from each other we set the width and height of the universe to a sufficiently large (but polynomial in the size of the Positive Planar 1-in-3-SAT instance) value. Every field that does not belong to any of the shown gadgets will contain a galaxy center. This enforces that all the galaxies between the gadgets have size one and thus that the gadgets do not interfere with each other.

We first describe the *wire* gadgets galaxies that carry the information whether the variable connected to this wire is either true or false. The two wire configurations for a true and a false signal are shown in Figure 3. The top and bottom rows of the gadget have size-two galaxies, this is ensured by the fact that each field above and below the wire gadget contains a galaxy center. The signal will be carried either from left to right or from right to left. There are two inner rows which carry the signal and two outer rows which will "shield" the signal from the rest of the instance. The galaxy centers of the inner rows

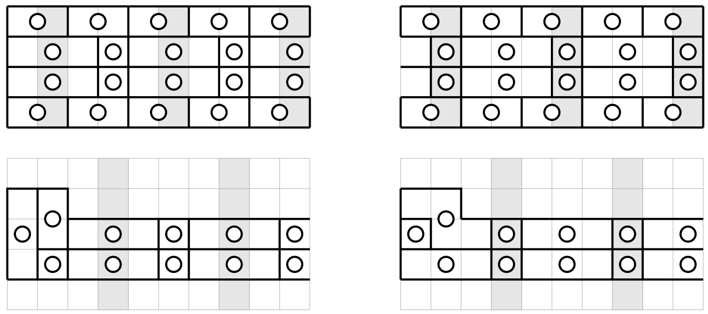

Figure 3: The wire gadget and the variable gadget for initializing a wire as true or false. The top part shows the two wire configurations for a true variable value (left) and a false variable value (right). Herein, columns i with $i \mod 4 = 0$ are drawn in gray. The bottom part shows the variable gadget for initializing a wire as either true (left) or false (right).

have even column coordinates. If the variable is true, then the galaxies where the column coordinate of the center is a multiple of four have size three and the other galaxies have size one. If the variable is false, then the configuration will be exactly inverted. Accordingly, we call a wire true or false. The subsequent statement follows immediately from the construction of the wire gadget.

Observation 1. *The output of a wire gadget is true if and only if its input is true.*

In the following gadget drawings, we will not draw the size-two galaxies above and below the wires in order to simplify the figures. At the position of the variable vertices in the clause-variable graph, we place a gadget which initializes the truth value of a wire as either true or false. This gadget is shown at the bottom of Figure 3. The wire gadget and the variable gadget are very similar to the original gadgets for these two purposes. The main difference is that in the original construction, the galaxies in the wires have height two and horizontal distance three. In our wires, the galaxy centers have height one and the horizontal distance between the centers is two.

Observation 2. *Using the variable gadget, each wire can be initialized to be either true or false.*

In order to realize a crossing-free embedding of the wires into the universe, we devise the two gadgets shown in Figure 4. The gadget for shifting a wire vertically is again roughly the same as the original gadget for this purpose. The gadget for reversing the horizontal direction of a wire signal is new: The original reduction featured a crossing gadget and thus the circuit can be drawn from left to right. In the planar clause-variable graph, however, there might be some variables which appear in a clause that is to the left and one clause that is to the right of this variable vertex. Hence, we need the ability to extend wires in any possible direction, which can be accomplished by combining the shifting and reversing gadget.

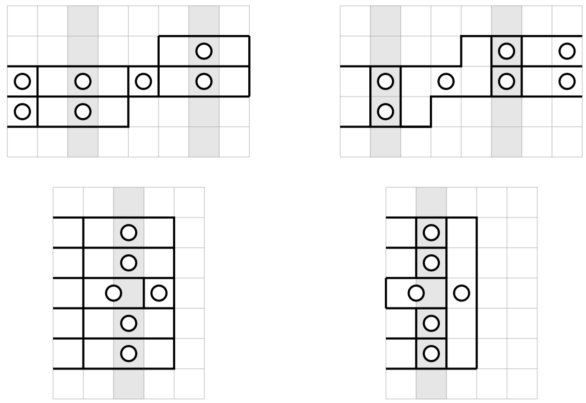

Figure 4: The gadgets for shifting a wire vertically (top) and for reversing the horizontal direction of a wire (bottom).

With the current gadgets we can build wires representing a variable and these wires can carry the values true and false in any direction. Since the variables of the Positive Planar 1-in-3-SAT instance may appear in an unbounded number of clauses, we need a gadget to duplicate the signal of a wire. This gadget is shown in Figure 5. It is the largest gadget of our construction and it also has the largest galaxy, namely a galaxy that has size seven if the input wire is false. Note that the gadget has two output wires. The bottom wire carries the same value as the input wire. The top wire has exactly the opposite truth value, that is, it is false if and only if the input wire is true. By a simple gadget, shown in Figure 6, we can negate the truth value of this top wire and obtain a wire that carries the same signal as the input wire.

So far, we are able to initialize variables, and to transport, duplicate, and negate their signal via wires. It remains to device a gadget for checking that each clause of the Positive Planar 1-in-3-SAT instance contains exactly one variable that is set to true. This is achieved by using two copies of the gadget shown in Figure 7. Consider a clause (x, y, z). Then in the construction we first move the wires of x and y next to each other and then place the first two galaxies of the first part of the clause gadget on a column that is a multiple of four. If both input wires are true, then, since the available area for the next two galaxy centers narrows down from four to two rows, there is no solution to the Spiral Galaxies instance. If one of the two input signals is true, then the signal carried by the output wire is true. Otherwise, if both input wires are false, then the output is false. Now, in the second part of the gadget we have the output wire of the first gadget as one input wire and the wire of z as the other input wire. If both input wires are true, then there is no solution. If exactly one of the two is true, then the output is true. Finally, if both are false signal, then the output is false.

Observation 3. *The clause gadget has as input three wires and outputs a true wire if and only if exactly one of the three input wires is true.*

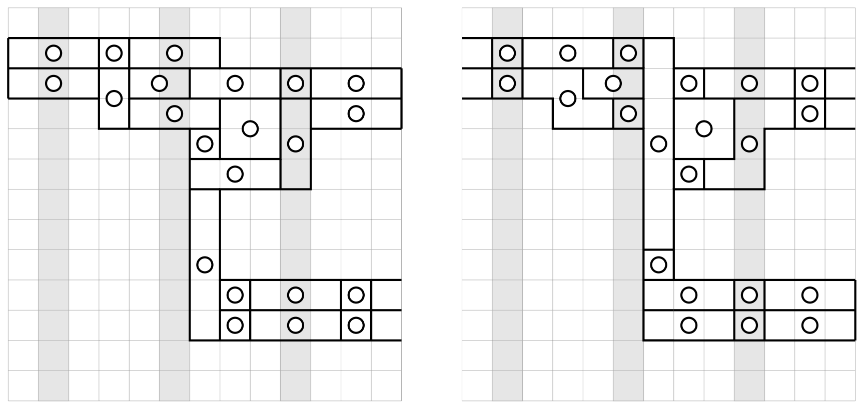

Figure 5: The gadget to duplicate a signal. The input wire is shown in the upper left part of the illustration. The left illustration corresponds to a true input and the right illustration corresponds to a false input.

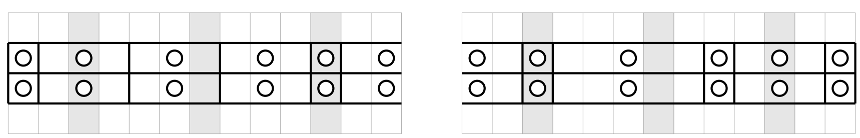

Figure 6: The gadget for negating the signal of the upper wire following the duplication gadget.

This almost completes the construction. The final check that needs to be performed is that the output wire of the second part of each clause gadget carries a true signal. This can be done by letting the wire "stop" after two galaxies that are narrow if the wire is true. That is, immediately after the two galaxy centers that are in an even column which is not a multiple of four we place two galaxy centers, one in each row. The construction can be performed in polynomial time. Altogether we arrive at the following.

Theorem 1. *Given an instance of* Spiral Galaxies *it is NP-complete to decide whether there is a solution in which every galaxy has size at most seven.*

Proof. We give a brief sketch of the correctness of the reduction.

If the Positive Planar 1-in-3-SAT instance is a yes-instance, then the Spiral Galaxies instance is also a yes-instance which can be seen as follows. Set the variable gadgets to the true configuration if the variable is true in the satisfying assignment for the Positive Planar 1-in-3-SAT instance. Otherwise, set them to the false configuration. Then, propagate the signal along the wire according to the configurations shown in the figures. When the wires meet the clause gadget, then exactly one of its three input wires is true since the assignment is satisfying. By the observation above, the output of the clause gadget will be true. This means there is a solution for all galaxies contained in

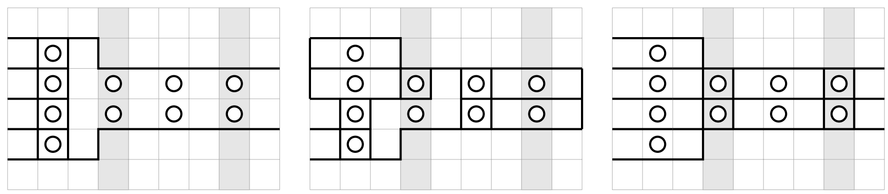

Figure 7: The gadget used for modeling the clauses. Left: If both input wires are true, then there is no solution. Middle: If exactly one input wire is true, then the output wire is true. Right: If both input wires are false, then the output wire is false.

the gadgets. All other galaxies have trivial solutions since they have either size one or two.

Conversely, if the Spiral Galaxies instance is a yes-instance, then the wires directly correspond to an assignment of the variables. Moreover, every clause gadget has a true output. By the above observation, this implies that the assignment is a satisfying one for the Positive Planar 1-in-3-SAT instance. □

In a solution for the constructed Spiral Galaxies instance, all except two possible galaxy shapes are rectangles. While we might replace the nonrectangular galaxy in the shifting gadget and in the duplication gadget, we do not know how to create the clause gadget and the variable gadget by using only rectangles.

3. A Nebula of Exact Algorithms

Although Spiral Galaxies is NP-hard even for small galaxies, one might still try to find algorithms that solve the problem. In this section, we will first focus on special cases of Spiral Galaxies, providing a polynomial-time algorithm for each of them. Then, we will focus on the general case, where three exact (thus exponential-time) algorithms will be given.

3.1. Polynomial-time Solvable Special Cases

In this section, we give several polynomial-time algorithms, for specific instances of Spiral Galaxies. The results mainly concern three types of constraints: on the sizes of the galaxies, on the shapes of the galaxies, and on the way galaxies are connected.

Before going further, we introduce some terminology: for any galaxy center g, we define the *kernel of* g, $\kappa(g)$, as the fields of U whose integer coordinates are obtained by rounding (from below and from above) the coordinates of g. In other words, if g has two (resp. one, zero) integer coordinates, then $\kappa(g)$ is of size one (resp. two, four). A rectangular galaxy with top-left corner f and galaxy center g is denoted by $R(f, g)$. In the particular case where the rectangle is actually a square, the notation becomes $S(f, g)$. Finally, let us recall that $n = |U|$ is the number of fields in the universe.

We begin by describing an easy result concerning small galaxies.

Proposition 1. *One can determine in* $\text{poly}(k \log n)$ *time whether an instance of* Spiral Galaxies *has a solution that contains only size-1 and size-2 galaxies.*

Proof. Any size-1 or size-2 galaxy of center g is exactly its kernel $\kappa(g)$. Thus any galaxy is completely determined by the location of its center. If we assume the input is correct (that is, no two kernels overlap), then we just need to check whether a field not belonging to a kernel exists in U. If this is the case, we have a no-instance. Otherwise, we have a yes-instance. The time complexity is in $\text{poly}(n)$. Since we can reject any instance such that $k < \frac{n}{2}$, we have $k = \Theta(n)$. Thus, $\text{poly}(n)$ is polynomial in the input size. □

The next three results, Theorems 2 to 4, are polynomial-time algorithms for restricted cases in which the shapes of the galaxies in a solution are constrained. These three results rely on the same generic idea. We iteratively fill the universe with galaxies. A field that is not yet assigned to any galaxy is called *free* in this process. Because of the shape constraint, it is always possible to find a free field f for which we can determine the unique galaxy center g it belongs to, together with the exact extent of the galaxy. If this process stops because no free field remains, it suffices to check whether the computed galaxies are pairwise nonoverlapping. Otherwise, at least one free field could not be assigned to any galaxy, and thus we are in presence of a no-instance.

Theorem 2. *One can determine in* $\text{poly}(k \log n)$ *time whether an instance of* Spiral Galaxies *has a solution that contains only square galaxies.*

Proof. First, if at least one of the galaxy centers has exactly one noninteger coordinate, then it cannot be the center of a square galaxy, and thus we are in presence of a no-instance. Otherwise, the algorithm iteratively constructs the square galaxies in the following manner: Find the leftmost free field f among the topmost ones and determine the unique galaxy center g to which f should belong. If, at a given iteration, no galaxy center is found for f, then we are in presence of a no-instance, and the algorithm stops. If no free field is left at the end of the process, it suffices to check whether all squares are pairwise nonoverlapping to determine whether this is a yes-instance.

Correctness of the algorithm comes from the fact that since each galaxy is a square, knowing its top-left corner f and its center g completely determines the galaxy; moreover, if a square galaxy of center g contains f as its top-left field, then g lies below f and on the diagonal of U which contains f. Moreover, g is the closest galaxy center on this diagonal (choosing any other center on this diagonal leads to an invalid square containing at least two galaxy centers).

Now we bound the running time of the algorithm. First, the number of iterations is at most k (each iteration creates a new galaxy). In each iteration, the leftmost free field f among the topmost ones should be searched among $O(k)$ candidates: such a field is either to the right of the top-right corner, or below the bottom-left corner of a previously constructed square galaxy. Thus, it suffices to maintain a list of candidate free fields, removing from this list the field f each time it is chosen to create a new square galaxy, and storing the two new candidate fields each time a new square galaxy is found. Moreover, storing the coordinates of both f and g completely determines the square galaxy $S(f, g)$. Finally, determining whether two squares overlap can be done by considering their corners only. Hence, (i) finding the next free field f, (ii) finding the galaxy center g, and (iii) determining whether two galaxies overlap all come down to simple computations (additions, subtractions, comparisons) using the coordinates of the fields and the galaxy centers. Each individual computation takes time $O(\log n)$, and is performed among $O(k)$ objects ((cases (i) and (ii)) or among $O(k^2)$ objects (case (iii)). □

Theorem 3. *One can determine in* $\mathrm{poly}(k \log n)$ *time whether an instance of* Spiral Galaxies *has a solution that contains only horizontal rectangle galaxies, of height either one or two.*

Proof. First, note that for any g being the center of a horizontal height-one rectangle, $L_r(g)$ is an integer. On the other hand, for any g being the center of a horizontal height-two rectangle, $L_r(g)$ is not an integer. Now, the algorithm is similar to the one presented in the previous result: iteratively take the leftmost free field among the topmost ones, say f, and determine the unique galaxy center g that yields a galaxy which could contain f as its top-left corner. If the algorithm fails to find a rectangle for a given free field, then output no. Otherwise, when the algorithm stops, check that no two rectangles overlap.

Correctness of the above algorithm essentially follows from the fact that once f is fixed, there is a unique horizontal rectangle of height either one or two that has g as its center and f as its top-left field. First note that g necessarily lies on the right of f; besides, no two centers of height-one (resp. height-two) rectangles could simultaneously be considered (only the closest to f is relevant). Hence, the only remaining choice could occur between two "candidate centers" g_1 and g_2 yielding two different types of rectangles (one of height one, the other of height two).

We will now argue that it is sufficient to consider only the center whose column coordinate $L_c()$ is closest to the one of f. Indeed, suppose, without loss of generality, that $L_r(g_1)$ is an integer (thus $L_r(g_2)$ is not), that is, g_1 (resp. g_2) is the center of a height-one (resp. height-two) rectangle $R_1(f, g_1)$ (resp. $R_2(f, g_2)$). Thus, we have $L_r(f) = L_r(g_1)$ and $L_r(f) = L_r(g_2) - 0.5$. Now, let us compare $L_c(g_1)$ and $L_c(g_2)$. If $L_c(g_1) \geq L_c(g_2)$, then $R_1(f, g_1)$ would overlap $\kappa(g_2)$, and thus would not be valid. In other words, the only rectangle to consider is $R_2(f, g_2)$. Conversely, if $L_c(g_2) \geq L_c(g_1)$, $R_2(f, g_2)$ contains g_1, and the only rectangle to consider is $R_1(f, g_1)$. In summary, for each field f considered in the algorithm, g is uniquely determined to be the center g whose $L_r()$ coordinate is in $\{L_r(f), L_r(f) + 0.5\}$, and whose $L_c()$ coordinate is the closest to $L_c(f)$.

The time complexity follows from the proof of Theorem 2 above: finding the next free field, computing and storing the determined rectangles, and checking whether rectangles are pairwise nonoverlapping are performed in a similar way. Here again, we have at most k iterations of the algorithm, each of which only takes poly(k) operations, and each such operation can be performed in $O(\log n)$. Altogether, the running time of our algorithm is poly($k \log n$). □

Recall that the universe U consists of N rows and M columns. A *row-decomposition* of a solution of Spiral Galaxies is a partition $\{U_1, U_2, \ldots, U_p\}$ of U such that each U_i, $1 \leq i \leq p$, contains N_i consecutive rows and M columns, and each galaxy is completely contained in a single U_i. We say that a row decomposition of a solution is a *rectangle row-decomposition* if each U_i contains only rectangles of height N_i.

Theorem 4. *One can determine in* poly($k \log n$) *time whether an instance of* Spiral Galaxies *has a solution with a rectangle row-decomposition.*

Proof. The proof is very similar to the two previous theorems. The main difference here is a pre-processing step that computes the row-decomposition of U (if it exists), before applying the algorithm on each of the U_is. Finding the row decomposition of U consists of determining the values of the N_is, and thus the value of p. This is done as follows: set $i = 1$, start from the top-left field f of U, and search for a center g whose row coordinate $L_\mathrm{r}(g)$ is the closest to $L_\mathrm{r}(f)$. Set $N_i = 2 \cdot (L_\mathrm{r}(g) - L_\mathrm{r}(f) + 0.5)$, increase i by one, and iterate the process with the leftmost field f of row coordinate $L_\mathrm{r}(f) + N_i$. The list of the N_is is sufficient to determine the row decomposition of U. Now, it suffices to set $N_0 = 1$, and to consider each U_i, $1 \leq i \leq p$, separately. For each U_i, $1 \leq i \leq p$, we apply the following algorithm: take the leftmost free field f whose row coordinate is $r_i = N_{i-1} + 1$, and first check whether all galaxies centers whose row coordinates are in the interval $[r_i; N_i]$ have the same row coordinate, equal to $r_i + (N_i - r_i)/2$. If this is not the case, we have a no-instance. Otherwise, iteratively choose the center g whose column coordinate is the closest to the one of f, create the rectangle $R(f, g)$, mark all fields in $R(f, g)$ as occupied, and set f to be the leftmost free field of row coordinate r_i. If, for a given i, this process leaves free fields, or requires to create a rectangle that overpasses the rightmost column, we have a no-instance. Otherwise, we have a yes-instance.

Once the row-decomposition of U is computed, the algorithm that treats each U_i independently is clearly correct, because we assume that only rectangles of height N_i can exist. It thus remains to show that the row-decomposition we obtain is the correct one. Since we assume there exists a solution that has a rectangle row-decomposition, we have, by definition, that all galaxies of U_i have the same height. Thus their centers have to be on the same row, which means that one must choose a center g whose row coordinate is the closest to the one of f.

Concerning the time complexity, the pre-processing step that determines the row-decomposition of U can be achieved in poly($k \log n$) time, since it consists in $p \leq k$ iterations, each of which determines the unique candidate center g for the free field f, computes and stores N_i, and computes the next f. Thus, the same argument as in the two preceding proofs applies: each iteration consists in considering $O(k)$ objects (the galaxy centers), and apply on each of them a constant number of operations, each costing $O(\log n)$. Now, for each of the p

elements in $\{U_1, U_2 \dots U_p\}$, the algorithm is very similar to the previous ones, and for the same reasons can be achieved in $\text{poly}(k \log n)$ time. Altogether, we the algorithm has running time $\text{poly}(p \cdot k \log n) = \text{poly}(k \log n)$. $\square$

The next two results (Theorem 5 and Proposition 2) concern specific cases relying not on the shape of the galaxies, but on their "connectedness". Before we can state these results, we need some terminology: a galaxy with center g is said to be *trivially connected* if all its fields are either in $\kappa(g)$ or adjacent to a field in $\kappa(g)$, where $\kappa(g)$ is the kernel of g.

Theorem 5. *One can determine in* $\text{poly}(k \log n)$ *time whether an instance of* Spiral Galaxies *has a solution such that all galaxies are trivially connected.*

Proof. First, notice that if, in the input instance, there exists a field not adjacent to any galaxy kernel $\kappa(g)$, then we have a no-instance. Otherwise, we build the graph $G_U = (V, E)$, whose vertex set is the set of fields f in U not belonging to a galaxy kernel, and where an edge connects two vertices u_f and u'_f (corresponding to fields f and f') whenever f and f' are g-twins for some center g *and* f and f' are adjacent to $\kappa(g)$. We then compute a maximum matching $\mathcal{M}$ in G_U. If $\mathcal{M}$ is a perfect matching, then output yes, otherwise output no. In case of a yes-instance, the edges of the matching $\mathcal{M}$ allow to infer the (trivially connected) galaxies of a solution.

Correctness of the above algorithm is implied by the following arguments: connectivity is ensured by the forced adjacency to $\kappa(g)$; symmetry of the galaxies is ensured by the definition of $E(G_U)$; galaxies are nonoverlapping because we ask for a matching in G_U; and finally, no field is left uncovered by either the pre-processing step or the request for a *perfect* matching.

Now let us analyze the time complexity of our algorithm. First, notice that since galaxies are all trivially connected, then they have constant size (because kernels have constant size). Thus, $k = \Theta(n)$ for yes-instances. After checking the value of k, we may thus assume that $\text{poly}(n)$ is polynomial in the input. The three main steps of our algorithm (pre-processing, construction of G_U and computation of a maximum matching in G_U) require to consider (almost) all fields of U. However, $\text{poly}(n)$ time suffices: G_U contains $O(n)$ vertices and $O(n)$ edges (since a field can be adjacent to at most 4 other fields, G_U is of maximum degree 4), and computing a maximum matching in such a graph can be done in time $O(n^{1.5})$ [3]. Altogether, we end up with a $\text{poly}(n)$ time algorithm. $\square$

Using the same ideas as in the previous result, but in a less constrained way, we are able to obtain a result for a new variant of Spiral Galaxies, in which galaxies may not necessarily be connected. Let us call this variant Possibly Disconnected Spiral Galaxies.

Proposition 2. Possibly Disconnected Spiral Galaxies *can be solved in* $\text{poly}(n)$ *time.*

PROOF. The proof is very similar to the one of Theorem 5 above: we first build the graph $G_U = (V, E)$, where V is the set of fields f in U not belonging to a galaxy kernel, and where an edge connects two vertices u_f and u'_f (corresponding to fields f and f') whenever f and f' are g-twins for some galaxy center g. Notice that here, we do not require that f and f' are adjacent to $\kappa(g)$ anymore; this is why the galaxies we will obtain may be disconnected. We then compute, as in the previous proof, a maximum matching $\mathcal{M}$ in G_U, and decide for a yes- or a no-instance depending on whether $\mathcal{M}$ is a perfect matching. Correctness and time complexity of the algorithm are straightforward from the previous proof.□

It should be noted that the above result relies on two strong hypotheses: First, POSSIBLY DISCONNECTED SPIRAL GALAXIES is a much less constrained version of SPIRAL GALAXIES where connectedness of the galaxies is no longer required, which means that the game is rather different. Second, as discussed in Section 1, the size of the input is $O(k \cdot \log n)$. Thus, the above result is of interest (because it leads to a polynomial-time algorithm) only if $k = \Omega(n/\log n)$.

3.2. Exact Algorithms

In this section, we first provide two exact exponential-time algorithms [5] for solving SPIRAL GALAXIES in the most general case. Though the running times of the two algorithms presented here are quite similar, we mention them both since they rely on two different viewpoints of the problem. We then focus on the case where any solution of SPIRAL GALAXIES contains only rectangular galaxies, and provide a fixed-parameter algorithm for the parameter number k of galaxies.

Proposition 3. SPIRAL GALAXIES *can be solved in* $4^n \cdot \text{poly}(n)$ *time.*

PROOF. Given an instance of SPIRAL GALAXIES, any solution can be interpreted as a two-dimensional map, where each galaxy g (and the fields it contains) is a region, and where two distinct regions are adjacent when they contain adjacent fields. The famous four color theorem (see e.g., [10]) tells us that such a map can be colored with at most four colors. The exact algorithm that follows from the above argument can be described easily as follows: generate every possible four-coloring $\mathcal{C}_\mathcal{U}$ of the fields of U; for each such $\mathcal{C}_\mathcal{U}$, check whether (a) each connected set of fields of the same color contains a unique galaxy center, and if so, whether (b) it is a valid galaxy, that is, it satisfies the symmetry condition. If this is the case, a solution has been found. If no coloring satisfies the two conditions (a) and (b), we have an instance without solution. The running time of the algorithm is straightforward: there are 4 colors and n fields to color. Hence the total number of colorings is 4^n; besides, for any given coloring $\mathcal{C}_\mathcal{U}$, checking whether conditions (a) and (b) hold can be done in poly(n) time. □

The time complexity can actually be slightly improved as shown in the following.

Theorem 6. SPIRAL GALAXIES *can be solved in* $\frac{4^n}{2^{N+M}} \cdot \text{poly}(n)$ *time.*

Proof. Recall that N is the number of rows and M is the number of columns in U, and thus $n = N \cdot M$. Take any solution to Spiral Galaxies, and consider two adjacent fields f and f'. They can be adjacent either horizontally or vertically. If f and f' belong to distinct galaxies, say g and g', we will say there exists a *border* between them; otherwise, the border does not exist. The algorithm is thus the following: generate all possibilities for borders between adjacent fields (i.e., existence or nonexistence) in U. For each such possibility, compute the maximal connected areas and check whether conditions (a) and (b) from proof of Proposition 3 hold. If this is the case, a solution has been found. If none of the tested possibilities yields a solution, we are in presence of a no-instance. The running time for this algorithm is thus $2^{n \cdot f} \cdot \text{poly}(n)$, where $n \cdot f$ is the number of neighboring fields in U. The neighboring fields amount to $(N-1)M$ horizontal ones, and $(M-1)N$ vertical ones; thus $n \cdot f = n - N - M$, which yields the claimed time complexity. □

Let Rectangular Spiral Galaxies denote the constrained version of Spiral Galaxies where a solution may contain only rectangular galaxies. We have the following result.

Theorem 7. Rectangular Spiral Galaxies *can be solved in* $k! \cdot \text{poly}(k \log n)$ *time.*

Proof. The idea here is very similar to the one used in proof of Theorem 2: at each iteration, we determine a free field f, determine the galaxy center g it will belong to, construct $R(f, g)$. The two main differences from the initial algorithm are the following: first, we look for rectangles instead of squares; however, once f and g are known, the galaxy $R(f, g)$ is completely determined, just as it was the case for $S(f, g)$. Second, for any iteration i of the algorithm, $1 \leq i \leq k$, there maybe up to $k - i + 1$ galaxy centers (thus this many rectangles) to consider; this is why we need to branch on all possible solutions. By the above discussion and proof of Theorem 2, it is easy to see that each iteration of the algorithm is achieved in $\text{poly}(k \cdot \log n)$. Due to the branching, up to $k!$ iterations may be performed. Altogether, we end up with the claimed complexity. □

4. Space is Big but May Have Few Corners

In this section, we present a fixed-parameter algorithm for the special case that the galaxies are hole-free and for the parameter overall number of galaxy corners ℓ. A galaxy is called *hole-free* if no other galaxy is completely contained in this galaxy. The hole-freeness property is not demanded by the original problem definition. Many real-world instances, however, fulfill this hole-freeness property.

Recall that $L(g)$ denotes the location of the center of galaxy g. By the following lemma, hole-freeness implies that each galaxy contains its center. Hence, our algorithm which constructs hole-free galaxies does not need to check this property explicitly.

Lemma 1. *A nonempty area that is connected, hole-free and symmetric with respect to a location $L(g)$ contains all fields that are adjacent to $L(g)$.*

Proof. If the area is a rectangle or shaped like a cross, then the claim holds trivially. Otherwise, assume without loss of generality, that the area contains a field f_1 which is at least as high as $L(g)$ and to the left of $L(g)$. By the symmetry property, the area also contains a field f_2 that is at most as high as $L(g)$ and to the right of $L(g)$. Since the area is connected, the two fields are connected by a path of other fields of the area. For each field of this path, its g-twin, however, also belongs to g. Consequently, there are two paths from f_1 to f_2 that enclose $L(g)$. Since the area is hole-free, this implies that all fields that are adjacent to $L(g)$ belong to the area. □

It is relatively straightforward to decide in $n^{f(\ell)}$ time whether there is a solution with ℓ corners: Guess the exact position and orientation of each corner, then connect the corners accordingly and finally check whether this gives a solution. The running time follows from the fact that for each corner we have to consider $\text{poly}(n)$ choices and that all steps after the guessing can be easily performed in polynomial time.

We now describe an algorithm that can find solutions with at most ℓ corners in $f(\ell) \cdot \text{poly}(\log n)$ time. The outline of the algorithm is as follows. First, we show how to represent each spiral galaxy as a tiling of $O(\ell)$ rectangles. Then, we present an integer linear program (ILP) with $f(\ell)$ many variables which, using a known result on the running time of bounded variable ILPs [8] implies the claimed running time.

Consider any galaxy. Our aim is to represent the galaxy as a tiling of few rectangles. The first step is to divide the galaxy into three parts which will allow us to naturally capture the symmetry condition when defining the rectangle tiling. While doing so, we aim to keep the number of corners low.

Lemma 2. *The fields of every hole-free galaxy g with ℓ corners can be colored with at most three colors black, red, and blue such that*

- *if $L(g)$ has integer coordinates, then the field containing $L(g)$ is black, otherwise no fields are black,*
- *the red area has at most $\ell + 2$ corners,*
- *the g-twin of every red field is blue.*

Proof. Let $L(g) = (y, x)$ be the location of the galaxy center. We discuss only the case in which y is noninteger (the case in which x is noninteger is symmetrical), or the case in which both x and y are integers (see Figure 8).

If y is noninteger, then color $(\lceil y \rceil, \lceil x \rceil)$ red. Then check whether there is an uncolored field that belongs to the galaxy and is to the left or to the right of a red field. If this is the case, color it also red. When there is no such field, then note that the row of red fields is a separator of the galaxy. Color the g-twins of this line of red fields (which is the line below) blue. Now color all uncolored

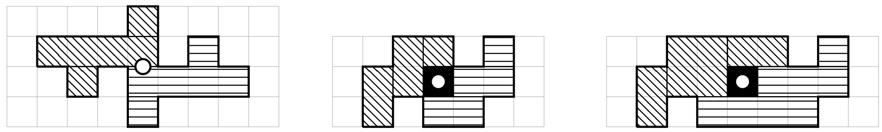

Figure 8: Examples of some galaxies and the coloring as provided by the algorithm in the proof of Lemma 2 (herein, the red fields are hatched diagonally and the blue fields horizontally). Left: $L(g)$ has noninteger coordinates, middle: $L(g)$ is a separator, right: $L(g)$ has 8 neighbors.

fields that can reach blue fields only via red fields with red and all other fields blue. The resulting coloring clearly fulfills the first and the last condition of the lemma. It remains to show the number of corners. At most $\ell/2$ corners of the red area are also corners in g. In addition, there are at most two corners between red and blue fields (recall that the separator is a straight line of fields). Since g has at least four corners, the number of corners in the red fields is at most $\ell/2 + 2 \leq \ell$.

If y is integer, then color the field that contains y black. If the galaxy contains no further fields, then the lemma holds. Otherwise, distinguish two further cases.

Case 1: Removing the center from the galaxy cuts the area in at least two connected components. In this case, pick one of these components, color it red and color the fields of its g-twins blue. Then pick, if it exists, another uncolored connected component and color it red and its g-twins blue. No further uncolored connected components exist. The red connected components has again at most $\ell/2$ corners that are corners in g. Furthermore, there are at most four corners between red fields and the black field. Hence, the number of corners in the areas defined by the red fields is at most $\ell/2 + 4 \leq \ell + 2$.

Case 2: Otherwise. In this case, the galaxy center has eight neighbors. Color the field to the left of the center red and call it the current field. While the current field has a left neighbor that is part of g, color this field red and make it the current field. Next, make the field to the top-left of the galaxy the current field and color it red. Now, while the current field has a right neighbor that belongs to g, color it red and make it the current field. Now, the red fields are a separator. Color all fields that can reach the center only via red fields also with red. After this, color all g-twins of the red fields blue.

Again, the area defined by the red fields has at most $\ell/2$ corners that are corners in g plus four corners on the border to the blue or black fields. The overall number of corners is thus at most $\ell/2 + 4 \leq \ell + 2$. □

Note that this is the only part of the algorithm where we assume hole-freeness of the galaxy. In order to drop this demand, one approach could be to find a different way of separating the red area from the blue area while still maintaining the bound on the number of corners of the red area.

In order to formulate the ILP we will now model each galaxy as a union of rectangles. The following lemma is a straightforward corollary of [4, Theorem 1].

Lemma 3. *An orthogonal polygon with ℓ corners can be partitioned into at most ℓ nonoverlapping rectangles.*

Definition 1. *A* rectangle representation *of a galaxy g is a set of rectangles $\mathcal{R}_g = \{R_1, R_2, \ldots, R_q\}$ such that:*

- *the union of all R_i is exactly g,*
- *R_i and R_j do not overlap if $i \neq j$,*
- *for each R_i, there is exactly one rectangle $R_j \in \mathcal{R}_g$ such that the four corners of R_j are exactly the g-twins of the four corners of R_i.*

For a galaxy representation, we use $s(R_i)$ to denote the rectangle that is symmetric to R_i.

Lemma 4. *A galaxy g with at most ℓ corners has a rectangle representation with $O(\ell)$ rectangles.*

Proof. By Lemma 2 we can partition g into three areas that have $O(\ell)$ corners altogether. Now, for the red part we choose some partition into $O(\ell)$ rectangles which exists by Lemma 3. For the blue part choose the symmetric (with respect to $L(g)$) partition into rectangles. The black part, if it exists, is a rectangle so add the corresponding rectangle in this case. Clearly, the union of the rectangles is g, the rectangles do not overlap and, by the choice of the partition of the blue part, there is for each red rectangle R_i a symmetric blue counterpart R_j. The black rectangle R, if it exists, consists just of one field whose center is $L(g)$, so the corners of R are symmetric to themselves. □

We now use the rectangle representation to fix the main structure of a putative solution. A *layout* of a solution is a structure consisting of the following parts:

- For each galaxy g, we fix a set of rectangle identifiers $R_1^g, \ldots, R_q^g$, $q = O(\ell)$.
- For each rectangle R_i^g, we fix $s(R_i^g)$ (note that if $L(g)$ has integer coordinates, $R_i^g = s(R_i^g)$ is possible).
- For each pair of rectangles R_i^g and $R_j^{g'}$, we fix whether R_i^g is above, below, to the left, or to the right of $R_j^{g'}$ (at least one of the four must be the case).
- For each pair of rectangles R_i^g and $R_j^{g'}$, we fix whether they are adjacent or not, and if this is the case, we fix the "extent" of the adjacency. For example, if R_i^g is above $R_j^{g'}$, then we fix whether the left side of $R_j^{g'}$ is at least as far to the left as the left side of R_i^g or not, and similarly, whether the right side of $R_j^{g'}$ is at least as far to the right as R_i^g or not.
- For each rectangle, we fix whether it is adjacent to the left, right, top, or bottom limit of the universe.

Clearly, the number of layouts is bounded by a function of ℓ since the number of rectangles to consider in any solution is $O(\ell)$. The main structure of the algorithm is now as follows. Try all possible layouts. For each layout, first filter "bad" layouts that do not guarantee that the galaxies are connected, that the galaxies are hole-free, or that they cover the whole universe. Then, create an ILP with $O(\ell)$ variables and solve it. If it has a feasible solution, then use this solution to construct a solution of Spiral Galaxies.

We now describe how to filter bad layouts. For each galaxy g, create a graph whose vertices are the rectangles R_i^g. Make two vertices adjacent in this graph if the corresponding rectangles are fixed to be adjacent by the layout. Reject the layout if the resulting graph is not connected for some galaxy.

Now create a graph whose vertices are the galaxies. In this graph, make two galaxies g and g' adjacent if there is a pair of rectangles R_i^g and $R_j^{g'}$ that are fixed to be adjacent by the layout. Furthermore, add one vertex that represents the limits of U and make it adjacent to each galaxy g that has a rectangle R_i^g that is fixed to be adjacent to the respective limit. Now, reject the layout if there is a galaxy g that is a cut-vertex in this graph, that is, there is a pair of galaxies $g' \neq g$ and $g'' \neq g$ such that all paths between g' and g'' contain g. Finally, consider each rectangle R_i of the layout that is adjacent to at least one other rectangle. Assume without loss of generality, that the bottom fields of R_i are adjacent to the rectangles $Q_i^1, \dots, Q_i^q$, which are ordered such that

- the left border of Q_i^1 is fixed to be at least as far to the left as the left border of R_i and there is no other rectangle Q_i^j for which this holds,
- the right border of Q_i^q is fixed to be at least as far to the right as the right border of R_i and there is no other rectangle Q_i^j for which this holds, and
- for $i' > 1$, $Q_i^{i'}$ is fixed to be adjacent and to be to the right of $Q_i^{i'-1}$.

If such an order does not exist, then reject the layout. Otherwise, build the ILP formulation. We only need to check for feasibility, hence there will be no objective function that we need to maximize.

For each rectangle R_i in the layout, we introduce four variables: x_i^1, y_i^1, x_i^2, and y_i^2, where (y_i^1, x_i^1) shall be the top-left field of R_i and (y_i^2, x_i^2) shall be the bottom-right field of R_i. No further variables are introduced and the number of variables thus is $O(\ell)$. We now introduce inequality constraints that guarantee that the ILP solution gives a solution to Spiral Galaxies. First, we constrain all coordinates to be in the universe:

$$\forall x_i^j : 1 \leq x_i^j \leq M, \tag{1}$$

$$\forall y_i^j : 1 \leq y_i^j \leq N. \tag{2}$$

The second set of constraints forces all rectangles to be nonempty and guarantees that the coordinate pair (y_i^1, x_i^1) is indeed the top-left field:

$$\forall x_i^1 : x_i^1 - x_i^2 > 0, \tag{3}$$

$$\forall y_i^1 : y_i^1 - y_i^2 > 0. \tag{4}$$

Now we introduce constraints for rectangle pairs to force that the rectangles do not overlap, that adjacencies are preserved as fixed in the layout, and that each galaxy is symmetric. Herein, we describe only the case in which the rectangle R_i is above R_j, all other cases can be obtained by rotating the universe. First, we guarantee that the rectangles do not overlap:

$$y_i^2 - y_j^1 > 0. \tag{5}$$

If R_i and R_j are fixed to be the corresponding rectangles in the two symmetric parts of a galaxy g, that is, $R_i = s(R_j)$, then assume without loss of generality, that the right border of R_i is not to the left of the right border of R_j. Then, we add the constraints

$$2 \cdot L_{\mathrm{c}}(g) - x_i^1 - x_j^2 = 0, \tag{6}$$
$$2 \cdot L_{\mathrm{r}}(g) - y_i^1 - y_j^2 = 0, \tag{7}$$
$$2 \cdot L_{\mathrm{c}}(g) - x_i^2 - x_j^1 = 0, \tag{8}$$
$$2 \cdot L_{\mathrm{r}}(g) - y_i^2 - y_j^1 = 0. \tag{9}$$

Now, we add the constraints concerning adjacent rectangles. If R_i and R_j are fixed to be adjacent, then, since R_i is above R_j, we add the constraint

$$y_i^2 - y_j^1 = 1. \tag{10}$$

If we fix the left border of R_j to be at least as far to the left as the left border of R_i, then we add the constraint

$$x_i^1 - x_j^1 \leq 0. \tag{11}$$

We add similar constraints for the right borders of R_i and R_j (according to whether or not we have fixed R_i to extend further to the right than R_j).

Finally, we add the following constraints for the rectangles that are adjacent to the limit of the universe:

$$y_i^1 = 1 \qquad \text{if } R_i \text{ is adjacent to the top limit of } U, \tag{12}$$
$$y_i^2 = N \qquad \text{if } R_i \text{ is adjacent to the bottom limit of } U, \tag{13}$$
$$x_i^1 = 1 \qquad \text{if } R_i \text{ is adjacent to the left limit of } U, \tag{14}$$
$$x_i^2 = M \qquad \text{if } R_i \text{ is adjacent to the right limit of } U. \tag{15}$$

Lemma 5. *If the ILP as constructed above has a feasible solution, then the* Spiral Galaxies *instance is a yes-instance.*

Proof. First, we show that the rectangles that are fixed to make up a galaxy g create an area that fulfills the properties of a galaxy. By the filtering step and by Constraint 10, the area created by the rectangles is connected. Furthermore, every field in this area is contained in a rectangle. By Constraints 6–9 there is a rectangle whose corners are g-twins of the rectangle containing this point.

Hence, the g-twin of the each field is also contained in g. Now, by the filtering step before the ILP construction, the galaxy is also hole-free and therefore it fulfills all properties of a galaxy.

It remains to show the global property that the rectangles form a partition of the universe. By Constraint 5, the rectangles do not overlap, so it remains to show that the rectangles cover the universe. Assume that this is not the case. Then there is some field not covered by any rectangle but adjacent to some rectangle R_i. Assume without loss of generality that this field is below R_i. Clearly, R_i is not fixed to be adjacent to the bottom border by the layout. By the filtering step, the field R_i thus has at least one neighboring rectangle that is fixed to be below R_i and extend further at least as far to the left than R_i. Similarly, such a rectangle exists for the right border of R_i. Furthermore, between these two rectangles the filtering step guarantees that there is a chain of horizontally adjacent rectangles that are below R_i and adjacent to R_i. Hence, every field below R_i is covered by some rectangle. □

The converse of the above statement is also true in the sense, that if there is a solution to Spiral Galaxies, then there is one layout which passes the filtering and whose constructed ILP has a feasible solution. The existence of this layout is guaranteed by Lemma 3 plus the fact that the filtering step does not remove this layout. The feasible ILP solution is then obtained by simply plugging in the coordinates of the rectangles. Altogether we thus obtain the following.

Theorem 8. *In $f(\ell)\cdot\text{poly}(n)$ time one can determine whether* Spiral Galaxies *has a solution which has at most ℓ corners and in which every galaxy is hole-free.*

Proof. The correctness follows from Lemma 5 and the discussion above. It remains to show the running time. The number of layouts to consider is bounded by a function in ℓ since the number of galaxies is $O(\ell)$ and the number of rectangles that we need to consider for each galaxy is also $O(\ell)$. Hence, the number of objects for which all different possibilities of "applying some fix" only depends on ℓ. Therefore, the number of ILPs to solve is also bounded by a function of ℓ. Since each ILP has $O(\ell)$ variables and $\text{poly}(\ell)$ many constraints it can be solved in $f(\ell)\cdot\text{poly}(\log n)$ time [8]. All other steps can be performed in polynomial time. □

It would be nice to obtain a fixed-parameter algorithm for the smaller parameter k. A natural approach would be to show that the solution has only $f(k)$ corners. Indeed, we tried to show that this is the case. However, even for four galaxies only, this is not true.

Proposition 4. *For every $\ell \in \mathbb{N}$, there are yes-instances of* Spiral Galaxies *with four galaxies such that every solution has more than ℓ corners.*

Proof. Let $x > \ell+2$ and consider the following instance with $N = M = 2x+1$ and four galaxies $\alpha, \beta, \gamma, \delta$. In the instance, α will be very large, γ and δ will

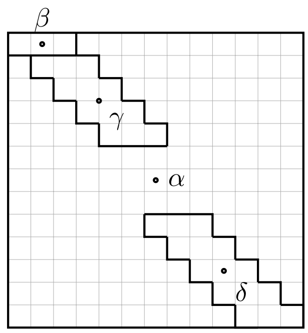

Figure 9: An instance of Spiral Galaxies with four galaxies and many corners.

be small, and β will be tiny. To denote the positioning of the centers of the small galaxies, we introduce another variable y which is some even integer such that $\ell \leq y < x$. The four galaxy centers are as follows.

- $L(\alpha) = (x+1, x+1)$,
- $L(\beta) = (1, 2)$,
- $L(\gamma) = ((y+3)/2, y/2 + 5/2)$, and
- $L(\delta) = (2x + 1 - y/2, 2x - y/2)$.

A sketch of the instance and of a solution for one set of values of x and y is given in Figure 9.

First, observe that by the choice of y, the galaxies β and γ live in the upper-left quadrant, that is, the set of fields (i, j) with $i, j \leq x+1$, and galaxy δ lives in the lower-right quadrant. Therefore, all α-twins of fields of β and γ must belong to δ. Hence, the shape of δ is a union of the shapes of β and γ. Further, galaxy β has its center on the last row of U, so it must be flat (have height one) and, since its center is on the second column, it can be either a galaxy containing just the center or a flat galaxy of width three.

We now show that the solution must essentially look like the one shown in Figure 9. The center of β does not belong to α, so its α-twin which is $(2x+1, 2x)$, belongs to δ. But then the δ-twin of $(2x+1, 2x)$ belongs to δ as well. This field is $(2x-y+1, 2x-y)$. Now, the α-twin of this field is $(y+1, y+2)$ and it is again not in galaxy α, so it must be in galaxy γ (recall that β is flat). Now the main difference between γ and δ that creates the many corners is the following. The height difference for γ-twins is odd since its center sits at noninteger coordinates. The height difference for δ-twins, however, is even since δ's center sits at integer coordinates. Moreover, the height of γ is exactly y since it cannot reach fields above $(y+1, y+2)$ (their α-twins are unreachable for δ). Therefore, galaxy γ has at least one field in each row from row $y+1$ until row 2. Similarly, galaxy δ has at least one field in each row from row 1 until row $y+1$. Note that for galaxy δ the formula $(i+1, i)$ defines a straight diagonal line between the three fields that are already assumed to be in δ. We now show a statement which implies that the fields of δ must be close to this main diagonal.

> *Claim.* For $1 < i \leq y+1$ and $1 \leq j < 2x-i$, if $(2x+1-i, 2x-i-j)$ belongs to δ, then $(2x+2-i, 2x+1-i-j)$ belongs to δ.

If $(2x+1-i, 2x-i-j)$ belongs to δ, then its α-twin $(i+2, i+3+j)$ belongs to γ (since $i > 1$ it cannot belong to β). Therefore, the γ-twin of this field also belongs to γ. This field is $(y+1-i, y+2-i-j)$. Again, the α-twin of this field belongs to δ. This field is $(2x-y+i, 2x-1-y+i+j)$. Finally, the δ-twin of this field belongs to δ. This field is $(2x+2-i, 2x+1-i+j)$ which proves the claim.

Now, we know that $(2x+1, 2x+2)$ cannot belong to δ as it is outside of the universe. Hence, the maximum deviation of δ from the main diagonal to the right is one, which implies that the maximum deviation of δ from the main diagonal to the left is also one. We also know that in row $2x-y$, the field that is on the main diagonal belongs to δ. By the above claim the complete main diagonal thus also belongs to δ. Since the galaxy is connected, this implies that either in row $2x-y$ or in row $2x-y+1$ the right or left neighbor of the main diagonal also belongs to δ. Again by the above claim this implies that either the left or the right neighbor of the main diagonal is part of δ for all rows $i \geq 2x-y+1$. By the symmetry of the galaxy we then obtain that the left and right neighbor of the main diagonal have to belong to δ for all rows of δ.

Hence, each of the $y+1$ rows of δ, i.e., rows $2x-y$ to $2x+1$, has two corners. Therefore, the overall number of corners is larger than ℓ. It is easy to verify, that there is indeed a solution for all x and y as chosen above: the galaxy δ is as described, the galaxy β is a flat galaxy with width 3, the galaxy γ is the set of remaining α-twins of the fields that are in δ, and all other fields are in α. □

5. Satellite Ideas

This work is the first exploration of the algorithmic space of Spiral Galaxies. We have identified several polynomial-time solvable special cases such as the problem of checking for a solution with trivially connected components. We also provided exact algorithms for the general problem that have a running time which is exponential either in the universe size or in the number of galaxy corners. On the negative side, we showed that it is NP-hard to check for a solution containing only galaxies of size at most seven.

As wide areas of the algorithmic space of Spiral Galaxies remain unexplored, we conclude with several open problems. First, is the problem of finding a solution with galaxies of size at most three polynomial-time solvable? Similarly, is the problem of finding a solution with galaxies of size at most six polynomial-time solvable? An answer to these questions would bring us closer to a complexity dichotomy with respect to the galaxy size. Further, is Rectangular Spiral Galaxies solvable in polynomial time? Finally, is Spiral Galaxies fixed-parameter tractable with respect to the number k of galaxies?

Currently, we do not have a conjecture on any of these questions. We do have a proof that the answer is not 42 but we defer it to follow-up work.

Acknowledgments. We thank the anonymous referees of *FUN 2014* for several comments improving the presentation of this paper, Matúš Mihalák for fruitful discussions during the conference, and Hélène Farine for her help with the section titles.

[1] E. D. Demaine. Playing games with algorithms: Algorithmic combinatorial game theory. In *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science (MFCS '01)*, volume 2136 of *LNCS*, pages 18–32. Springer, 2001. Available at `http://arxiv.org/abs/cs.CC/0106019`.

[2] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized complexity.* Undergraduate Texts in Computer Science. Springer, 2013.

[3] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[4] L. Ferrari, P. Sankar, and J. Sklansky. Minimal rectangular partitions of digitized blobs. *Computer Vision, Graphics, and Image Processing*, 28(1): 58–71, 1984.

[5] F. V. Fomin and D. Kratsch. *Exact exponential algorithms.* Springer, 2010.

[6] E. Friedman. Spiral galaxies puzzles are NP-complete. Available at `http://www2.stetson.edu/~efriedma/papers/spiral.pdf`, 2002.

[7] R. A. Hearn and E. D. Demaine. *Games, puzzles, and computation.* AK Peters, Limited, 2009.

[8] R. Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.

[9] W. Mulzer and G. Rote. Minimum-weight triangulation is NP-hard. *Journal of the ACM*, 55(2), 2008.

[10] N. Robertson, D. P. Sanders, P. D. Seymour, and R. Thomas. The four-colour theorem. *Journal of Combinatorial Theory, Series B*, 70(1):2–44, 1997.