PAPER *Special Section on Foundations of Computer Science - New Trends of Theory of Computation and Algorithm -*

Complexity of Critter Crunch

Tianfeng FENG[†a)], **Leonie RYVKIN**[††b)], **Jérôme URHAUSEN**[†††c)], ***Nonmembers***,
and **Giovanni VIGLIETTA**[†d)], ***Member***

SUMMARY We study the computational complexity of the puzzle game Critter Crunch, where the player has to rearrange Critters on a board in order to eliminate them all. Smaller Critters can be fed to larger Critters, and Critters will explode if they eat too much. Critters come in several different types, sizes, and colors. We prove the NP-hardness of levels that contain Blocker Critters, as well as levels where the player must clear the board in a given number of moves (i.e., "puzzle mode"). We also characterize the complexity of the game, as a function of the number of columns on the board, in two settings: (i) the setting where Critters may have several different colors, but only two possible sizes, and (ii) the setting where Critters come in all three sizes, but with no color variations. In both settings, the game is NP-hard for levels with exactly two columns, and solvable in linear time for levels with only one column or more than two columns.

key words: *Critter Crunch, puzzle game, computational complexity, NP-hard problem*

1. Introduction

Critter Crunch is a puzzle game developed by Capybara Games and released in 2008. It features a rectangular board containing several *Critters*, which come in different sizes (Small, Medium, and Large) and colors, and may have some special properties (e.g., the Blocker Critters). The player's goal is to clear the board of all Critters by picking some of them up and shooting them back into the grid: a smaller Critter is eaten when shot into a larger Critter; also, when a Critter has eaten two other Critters (or when it has eaten a single Critter that in turn has already eaten) it explodes, starting a "chain reaction" that blows up all adjacent Critters of similar size and color. When a Critter explodes, all the Critters below it climb up, possibly into the mouths of larger Critters above them, thus triggering *combo* explosions, and so on. A YouTube gameplay video can be found at [7].

In this paper we study the computational complexity of a version of Critter Crunch where the whole board is given to the player in advance, and the player has to decide if it is possible to clear it. That is, we study the decision problem whose input is a Critter Crunch board and the goal is to decide whether there is a sequence of moves that eliminates all Critters. In Sect. 2 we describe all the elements of Critter Crunch that are relevant to this paper, and we explain the rules of the game in greater detail.

In Sect. 3 we show that Critter Crunch is NP-hard if Blocker Critters are present on the board. The same reduction framework can also be used to prove the NP-hardness of levels with no special Critters but with a limit on the number of moves that the player can do (i.e., the so-called "puzzle mode").

The last two sections are devoted to Critter Crunch with no special Critters and with some limitations on sizes and colors: Sect. 4 studies the setting where Critters may have several different colors but only two sizes, and Sect. 5 studies the setting where Critters come in all three sizes but with no color variations. We prove that, in both settings, the game is NP-hard for levels with exactly two columns, and solvable in linear time for levels with only one column or more than two columns.

There is extensive literature on the computational complexity of video games and puzzles. Examples include Tetris, Candy Crush, Puzzle Bobble, Clickomania, and many others [1]–[6]. Critter Crunch is a puzzle game in the same vein as Tetris and Puzzle Bobble, and this paper is a contribution to this line of research.

2. Critter Crunch Gameplay and Definitions

Critter Crunch is played on a rectangular board whose columns are delimited by some vines, as illustrated in Fig. 1. The top part of the board contains some Critters, whereas the player controls an avatar below the board, called Biggs. Biggs can be moved left and right, and its long, sticky tongue can be used to pick up the bottom-most Critter of any column. Once Biggs has picked up a Critter, he can spit it out into any column. The Critter will hit the bottom-most Critter of that column, or stick to the top of the board if no Critter is present.

Critters can eat each other depending on their *sizes*: Medium Critters can eat Small Critters and can be eaten by Large Critters. However, Large Critters cannot directly eat Small Critters. If Biggs shoots a Small (resp. Medium) Critter into a Medium (resp. Large) Critter, the bigger Critter

Manuscript received March 25, 2021.
Manuscript revised July 24, 2021.
Manuscript publicized December 22, 2021.

[†] The authors are with School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Nomi-shi, 923–1292 Japan.

[††] The author is with Department of Mathematics, Ruhr University Bochum, Germany.

[†††] The author is with Department of Information and Computing Sciences, Utrecht University, Netherlands.

a) E-mail: ftflluy@jaist.ac.jp
b) E-mail: leonie.ryvkin@rub.de
c) E-mail: j.e.urhausen@uu.nl
d) E-mail: viglietta@gmail.com (Corresponding author)

DOI: 10.1587/transinf.2021FCP0008

will eat the smaller one. (Note that no Critter gets eaten if, for instance, a Large Critter is shot into a Medium Critter.)

When a Critter eats another Critter, it becomes *full*. If a full Critter eats one more Critter, or if any Critter eats a full Critter, the eating Critter *explodes*, and is eliminated from the board. Critters also come in different *colors*. Whenever a Critter explodes, all adjacent Critters of the same size and color explode with it (Critters can be adjacent only horizontally or vertically, but not diagonally). Thus, the explosion extends to a whole "connected component" of Critters of the same size and color.

After Critters have been eliminated by explosion, they leave some empty space on the board. Whenever a Critter has some empty space above, it climbs up along the vines. Specifically, if a location is empty, all the Critters below it move up simultaneously (see [7] at 2:54). When the upper Critter of such a "chain", say C_1, hits a Critter C_2 above it, C_1 is eaten by C_2 if and only if their sizes are compatible (i.e., if C_1 is Small and C_2 is Medium or if C_1 is Medium and C_2 is Large). If C_1 is eaten, all the Critters below it climb up, possibly being eaten, etc. This can give rise to *combo* explosions, i.e., explosions that are not directly caused by the player but happen as a consequence of other explosions.

There is another way a Critter can eat: say that a Small Critter has been fed to a Medium Critter (either because Biggs has shot the Small Critter into the Medium one or because the Small Critter has climbed up into the Medium one), and assume that there is a Large Critter immediately above the Medium one. Then, before the Medium Critter has the chance to explode, the Large Critter will eat it; and since the Medium Critter is full, the Large Critter will explode: this event is called a *food chain* (see [7] at 0:49). For example, referring to Fig. 1, if Biggs picks up the Small Critter in the third column and shoots it into the second column, he will trigger a food chain: the Medium Critter will eat the Small one, and it will be eaten by the purple Large Critter above it, which will explode (and no other Critter will explode, since there are no purple Large Critters adjacent to the exploding one).

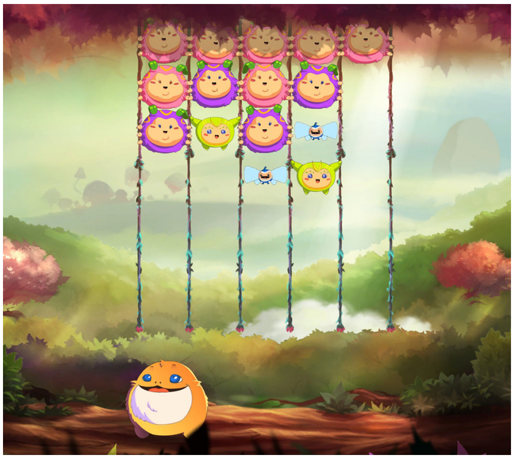

Fig. 1 A Critter Crunch screen

The Critters we have described so far are the *normal* ones. There are also some *special* Critters, which will not be relevant to this paper, except for one: the *Blocker Critter*. This Critter cannot be picked up by Biggs, it cannot eat, and it cannot be eaten. It only climbs up the vines as soon as it has some empty space above, like any normal Critter. The only way to eliminate a Blocker Critter is to remove all Critters below it and create an empty space directly above it: this is typically done by exploding the Critter above it.

As we have seen, every time the player picks up a Critter and shoots it back into the board (these two actions are collectively referred to as a *move*), a connected component of Critters may explode, and this first explosion may cause a chain of combo explosions. We are interested in analyzing the triplets of the form (s, m, l), indicating the amounts of Small, Medium, and Large Critters that can be eliminated in each explosion. More specifically, we count not only the Critters that actually explode, but also the Critters that have been eaten by the exploding ones. For instance, in a food chain we are eliminating one Small Critter, one Medium Critter, and $l \geq 1$ Large Critters that actually explode: this explosion is therefore represented by the triplet $(1, 1, l)$.

So, each player's move results in a (possibly empty) sequence of explosions

$$((s_1, m_1, l_1), (s_2, m_2, l_2), \ldots, (s_k, m_k, l_k)).$$

The explosion triplet (s_1, m_1, l_1) is called *initial*, while the

Table 1 Primitive explosions

Small	Medium	Large
2	≥ 1	
3	≥ 2	
	2	≥ 1
	3	≥ 2
1	1	≥ 1
1	2	≥ 1

Table 2 Combo-only explosions

Small	Medium	Large
2	1	≥ 1
2	2	1

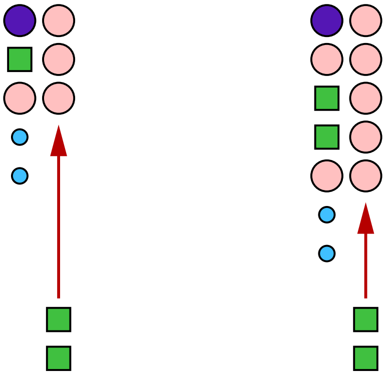

Fig. 2 Two combos that eliminate every Critter (squares represent Medium Critters). Both combos consist of a primitive explosion followed by a combo-only explosion.

others are *combo* triplets. The following lemma characterizes all such triplets. For a proof, refer to Appendix A.

Lemma 1. *All possible initial triplets can be expressed as sums of* primitive *triplets, which are listed in Table 1. There are also some combo triplets that are not sums of primitive ones, and these are reported in Table 2.* □

In Fig. 2 we show how each type of combo-only explosion may occur. On the left we have the sequence $((0, 2, 4), (2, 1, 1))$, while on the right we have the sequence $((0, 2, 7), (2, 2, 1))$.

3. Basic NP-Hardness Results

In this section we prove the NP-hardness of Critter Crunch, assuming that the board can contain normal Critters of all sizes, as well as Blocker Critters. In Appendix B we will extend this result to boards with only normal Critters, but with a limit on the number of moves ("puzzle mode").

Theorem 1. *It is NP-hard to decide if it is possible to clear a board of Critter Crunch containing normal Critters as well as Blocker Critters.*

Proof. We give a reduction from the strongly NP-complete problem 3-Partition: $3n$ positive integers $a_1, a_2, \ldots, a_{3n}$ are given, and the goal is to decide if they can be partitioned in n sets of equal sum B. It is not restrictive to assume that $B/4 < a_i < B/2$ for all $1 \leq i \leq 3n$, so that every set of sum B must have exactly three elements. An example of such a reduction with $n = 3$ is shown in Fig. 3.

The $3n + n = 12$ Small Critters surrounded by a red rectangle are called the *reservoir*. Due to the presence of Blocker Critters in every column, the only relevant action the player can do is to pick up a Small Critter from the reservoir and shoot it into a Medium Critter: this will start a food chain that will explode some Large Critters. Hence, each Medium Critter that the player can directly shoot into is called *trigger*, and the number of Large Critters immediately above it is its *payload*. For instance, the rightmost trigger has a payload of $a_1 = 3$.

It is easy to see that the biggest connected component of Large Critters can be exploded only if the three Large Critters marked with a blue cross become aligned: then the player can shoot into the leftmost trigger and explode the rightmost crossed Large Critter, as well as all the Large Critters in the leftmost column and in the upper region of the board.

In order for this to happen, the central crossed Large Critter has to move up by two positions, and the rightmost crossed Large Critter has to move up by 17 positions. Since 17 is the combined payload of all triggers except the leftmost one, this is achieved only when the player has shot into all triggers except the leftmost one.

To move the central crossed Large Critter up by two positions, the player has to eliminate the two Large Critters above it. The topmost of these two Large Critter can be eliminated only by aligning it with the three Large Critters labeled "check 1", which in turn can be done only by shooting some targets between a_1 and a_9 with a combined payload of 5. After that, the second Large Critter can be eliminated only by aligning it with those labeled "check 2", which is done again by shooting targets with a combined

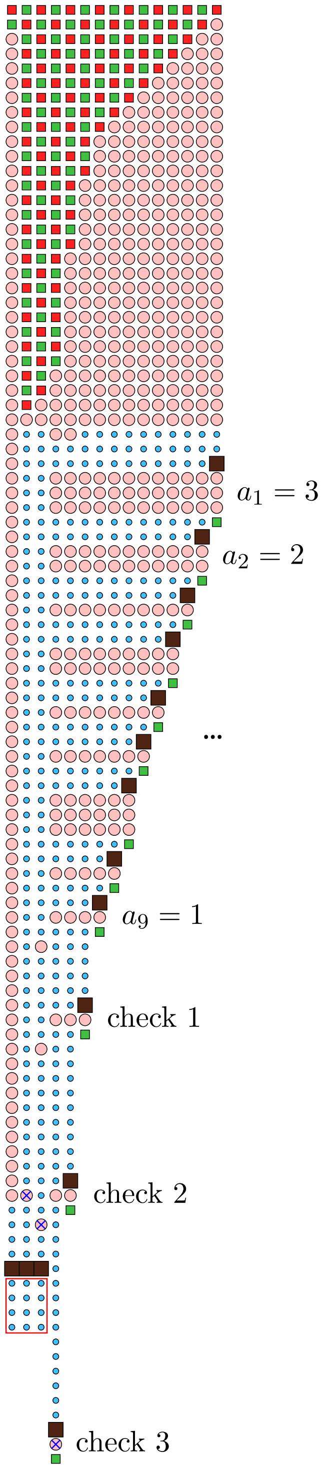

Fig. 3 Reduction from 3-Partition. Large brown squares represent Blocker Critters; smaller squares represent Medium Critters.

payload of 5. This is equivalent to partitioning the a_i's into sets each with sum 5.

When this is done, all the remaining Large Critters explode, and the Small Critters that are left move up into the Medium Critters in the top part of the board. These Medium Critters are exactly enough to eat all the Small ones and subsequently explode. Once no normal Critter is left, also the Blocker Critters are eliminated, and the board is cleared. □

Observe that, by a standard padding argument, the NP-hardness reduction in Theorem 1 can be extended to a proof of inapproximability. This is done by adding more rows of Large Critters to the upper part of the construction (right below the area with the Medium Critters). Indeed, these additional Large Critters can be eliminated if and only if the 3-Partition instance can be solved. Thus, if we place a polynomial number of additional rows of Large Critters, we still have a polynomial-time reduction from 3-Partition. The result is that the maximum number of Critters that can be eliminated from the board is NP-hard not only to compute exactly, but even to approximate within any polynomial factor.

Corollary 1. *For boards containing normal Critters and Blocker Critters, the maximum number of Critters that can be eliminated is NP-hard to approximate within any polynomial factor.* □

4. Levels with Critters of Only Two Sizes

In this section we study Critter Crunch levels containing only normal Critters, and of two sizes only. Without loss of generality, we may assume that the two sizes are Small and Medium: indeed, the case with Medium and Large Critters is symmetric, and any board with only Small and Large Critters is impossible to clear. We will characterize the complexity of the game as a function of the number of columns on the board: if the columns are only two, the game is NP-hard; in all other cases, it is decidable in linear time.

It is trivial to see that levels with a single column are decidable in linear time: the only move the player can do at any given point is to pick up the bottom-most Critter and shoot it back into place. If the Critter gets eaten, repeat the process until there are no Critters left; otherwise, report that the board cannot be cleared.

4.1 Boards with Three or More Columns

We will show how to decide in linear time if all Critters can be eliminated, provided that the board consists of three or more columns and sufficiently many rows.

Theorem 2. *If a board has at least three columns and it contains s Small Critters, m Medium Critters of c different colors, and no Large Critters, then all Critters can be eliminated if and only if $2c \leq s \leq 2m$.*

Proof sketch. The idea of our proof is that, given any configuration of Critters, we can rearrange them into a "canonical" configuration without causing any Critter to eat. From there, we can set up any desired primitive explosion, as long as the required number of Critters is present on the board. Note that we can clear the board if and only if we can do so by a sequence of primitive explosions: indeed, all combo-only explosions require Critters of all three sizes (cf. Table 2), but we are assuming to have only Small and Medium ones. So, to decide if a given board can be cleared, it is sufficient to count the number of Critters of each size and color, and check if these numbers satisfy a system of linear inequalities.

Proof. We call a configuration of Small and Medium Critters *canonical* if all the Medium Critters are in the first column and all the Small Critters are in the second column. We will now show that from any configuration we can reach a canonical one without feeding any Small Critter to a Medium one. This is done in steps:

1. If the bottom-most Critter of the first column is Medium, move it to the second column. Repeat until the first column is empty or its bottom-most Critter is Small.
2. Move the bottom-most Critter of the second column to the first column if it is Small or to the third column if it is Medium. Repeat until the second column is empty.
3. Repeat step 2 with all columns after the third one, emptying all of them into the first and third column.
4. Move the bottom-most Critter of the first column to the second column if it is Small or to the third column if it is Medium. Repeat until the first column is empty.
5. Move the bottom-most Critter of the third column to the second column if it is Small or to the first column if it is Medium. Repeat until the third column is empty.

Now we will show that, from a canonical configuration, we can set up any desired primitive explosion and restore a canonical configuration again, provided that we have enough Critters. Since there are no Large Critters, we have to consider only the first two rows of Table 1.

The first primitive explosion is of the form $(2, m, 0)$, with $m \geq 1$. We can set it up provided that there are at least two Small Critters and m Medium Critters of the same color. First we move the bottom-most Critter of the first column to the third column if it is of the desired color, or to the second column otherwise. We repeat this step until we have m Critters in the third column. Then we move all the Medium Critters from the second column back to the first column, and finally we shoot two Small Critters into the third column.

The second primitive explosion is of the form $(3, m, 0)$, with $m \geq 2$. Like before, we move $m - 1$ Medium Critters of the desired color to the third column. Then we shoot one Small Critter into the third column (thus feeding its bottom-most Critter, which becomes full). After that, we move one last Medium Critter of the same color to the third column, and we feed two Small Critters to it, triggering the explosion.

Since we can set up any sequence of primitive explosions, and no combo-only explosions are possible, deciding if a given board can be cleared reduces to counting the amount of Critters of every size and color and verify if these numbers satisfy some simple conditions. The details are given below.

Observe that at least two Small Critters must be used to trigger an explosion, and each explosion only affects Critters of one color; so, $2c \leq s$. Moreover, each Medium Critter can eat at most two Small ones; hence, $s \leq 2m$.

On the other hand, whenever $2c \leq s \leq 2m$, we can set up a primitive explosion of the form $(2, \geq 1, 0)$ or $(3, \geq 2, 0)$ that preserves both inequalities. For example, if $c = 3$, $s = 7$, $m = 8$, then we use three Small Critters to explode all the Medium Critters of the most numerous color, obtaining $c = 2$, $s = 4$, and $2 \leq m \leq 5$.

- Let s be odd and $2c+1 = s$. We thus have $2c < s < 2m$, so there is at least one color consisting of at least two Medium Critters. We explode all the Medium Critters of that color with three Small Critters. As this decreases s by three and c by one, $2c = s$ now holds. As we always have $c \leq m$, both inequalities still hold.
- Let s be odd and $2c + 1 < s$. The inequalities $2c + 3 \leq s < 2m$ follow. Again, we have at least one color consisting of at least two Medium Critters. We explode two Medium Critters of that color with three Small Critters. This decreases s by three, m by two, and either c does not change or it decreases by one. Due to $2c + 3 \leq s < 2m$ holding before the explosion, we now still have $2c \leq s \leq 2m$.
- Let s be even and $2c < s$. It follows that $c < m$, so we know that there is at least one color consisting of at least two Medium Critters. We now explode one Medium Critter of that color with two Small Critters. This reduces s by two, m by one, and c remains unchanged. After this explosion the inequalities still hold.
- Let s be even and $2c = s$. We explode all Medium Critters of one color using two Small Critters. As this decreases s by two and c by one, $2c = s$ still holds. As we always have $c \leq m$, both inequalities still hold. □

4.2 Boards with Only Two Columns

We will prove the NP-hardness of Critter Crunch with only Small and Medium Critters and boards of only two columns.

Theorem 3. *It is NP-hard to decide if it is possible to clear a board having exactly two columns and containing only Small Critters and Medium Critters.*

Proof sketch. Our reduction is from Vertex Cover: given a graph $G = (V, E)$ and an integer k, decide if there is a subset $U \subseteq V$ of exactly k vertices such that each edge in E has at least one endpoint in U.

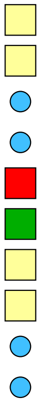

Fig. 4 Edge gadget

We use Medium Critters of $|V| + 2$ different colors: one color for each vertex, plus two "neutral" colors. To represent an edge (v_1, v_2), we use the edge gadget in Fig. 4. It consists of two Medium Critters of the first neutral color, followed by two Small Critters, then two Medium Critters of the colors corresponding to vertices v_1 and v_2 (red and green in the figure), and finally two Medium Critters of the first neutral color and two Small Critters.

Let us assume that an edge gadget is located at the bottom of the left column in a two-column board, and that the right column contains only Small Critters. The relevant property of the edge gadget is that it can be cleared only by "borrowing" four extra Small Critters from the right column to eliminate all of its Medium Critters. However, if one or both of its central Medium Critters are removed (i.e., the ones corresponding to v_1 and v_2), then no extra Critters are needed to clear the rest of the gadget: in this case, we say that the gadget is *satisfied*. Indeed, suppose that the red Critter is missing from Fig. 4. Then we can feed the two upper Small Critters to the green Critter: this connects the four Medium Critters of the neutral color, which can be eliminated using the two bottom Small Critters.

The full reduction is sketched in Fig. 5. The left column consists of an *offset* of several Medium Critters of the second neutral color (its purpose is to ensure the proper alignment when we move gadgets from left to right), followed by some *sink gadgets*: each of them is a sequence of $|V|$ Medium Critters, one of each vertex color. After that, we have an edge gadget for each edge in E, and then a *vertex gadget* for each vertex in V: a gadget for vertex v is simply a long-enough sequence of Medium Critters of the color corresponding to v, topped by two Small Critters. The right column only contains Small Critters.

The first thing the player must do is get rid of all the vertex gadgets. Roughly speaking, a vertex gadget can be eliminated in two ways: by using two Small Critters from the right column, or by using the two Small Critters on top of the vertex gadget itself. In the second case, the gadget's Medium Critters explode in the right column, and the length of the vertex gadget is such that it overlaps with all the sink gadgets as well as the edge gadgets (see the right side of

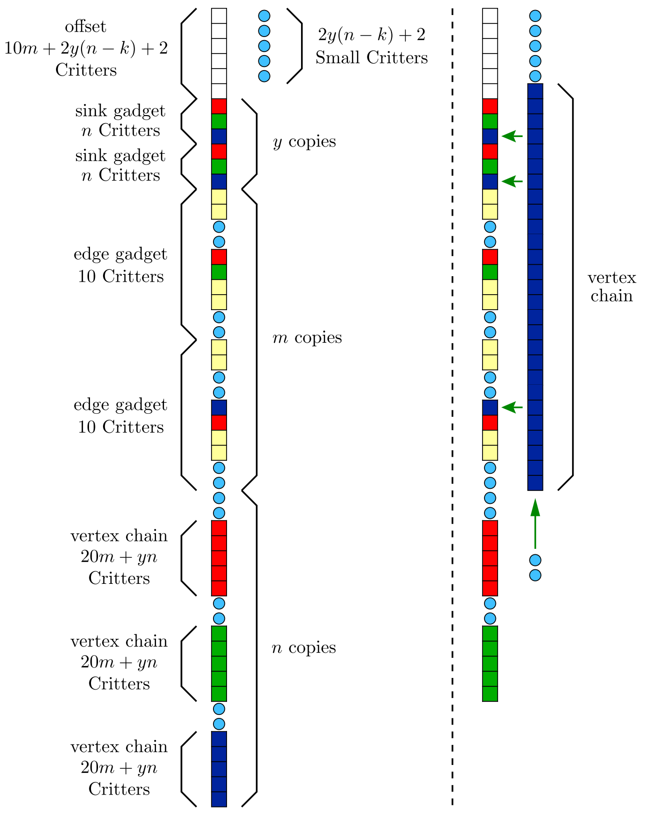

Fig. 5 Sketch of the reduction from Vertex Cover. The parameters n, m, k are given by the Vertex Cover instance, and $y = 5m + n$. Left: initial configuration. Right: selecting the blue vertex. The gadgets are not drawn to scale; in particular, all vertex chains on the left should be as long as the blue one on the right.

Fig. 5): in this case, we say that the vertex is *selected*.

Selecting a vertex v has two side effects: it removes one Medium Critter from each sink gadget, and it satisfies all the edge gadgets corresponding to edges incident to v.

After eliminating all the vertex gadgets, the player has to clear the edge gadgets, the sink gadgets, the offset, and the Small Critters in the right column, leaving no Critters on the board. However, if some edge gadgets are not satisfied, they must be cleared by borrowing some Small Critters from the right column, and in the end there will not be enough Small Critters to clear all the sink gadgets. Similarly, if the player selects a number of vertices other than k, the number of Medium Critters left in the sink gadgets and the number of Small Critters in the right column will not add up.

Note that the above argument depends on a careful selection of the number of sink gadgets and Small Critters in the right column. The details are given below.

Proof. We provide the missing details from the above sketch. The reduction from Vertex Cover takes as input a graph $G = (V, E)$, with $|V| = n$ and $|E| = m$, and an integer k.

Recall that the vertex gadgets encode each vertex $v_1, \dots, v_n$ through a long chain of Medium Critters of one color (we assign vertex v_i the color c_i), topped by two Small Critters. It is now possible to either feed two Small Critters from the right column to the bottom Critter to explode the chain, or move the whole chain aside to the right column and explode it there using the two Small Critters that were sitting on top of the chain. Exploding it in the right column encodes selecting the respective vertex for the vertex cover in G. Note that when exploding the chain in the left column, there are two Small Critters less in the right one. As two Small Critters were sitting on top of the chain and thus are now accessible, we simply put them to the right to access the next gadget's lowest Critter and have the correct number of Small Critters back at the right column.

When a vertex is chosen, i.e., the corresponding chain explodes in the right column, all incident edge gadgets lose the Medium Critter of the respective color. An edge gadget can be uncovered (all ten Critters are still present in the gadget), covered by one vertex (nine Critters remaining) or covered by both incident vertices (only eight Critters remaining).

For any uncovered edge, we need at least four Small Critters to explode the whole gadget (if playing "inefficiently", even five). The intended way to eliminate an uncovered edge gadget would be to explode the two bottom Medium Critters of the neutral color using the Small Critters at the bottom, then use four additional Small Critters from the right column to explode the colored Medium Critters, and finally explode the two neutral Medium Critters on top with the two remaining Small Critters.

Alternatively, we could potentially "connect" two neighboring edge gadgets E_1 and E_2, with E_1 located above E_2 in the left column, as follows: Let s_1 and s_2 be the two Small Critters at the bottom of E_1, with s_1 sitting above s_2. We feed s_2 to the highest Medium Critter of E_2 (after moving E_2 to the right column), and then we feed s_1 to the lowest Medium Critter of E_1. After moving E_2 back to the left column, our connected edge gadgets consist of two Medium Critters of the neutral color on top, followed by two Small Critters, the colored Medium Critters representing the first edge, one neutral Medium Critter, four full Medium Critters the lowest of which is colored, the second colored Medium Critter representing the second edge, and two neutral Medium Critters. Moving the two bottom Small Critters back to the left explodes the bottom Medium Critters immediately. Now we would need three Small Critters to explode the colored ones of E_2, then one Small Critter for the intermediate neutral Critters, and again four Small Critters for the colored Medium Critters of E_1. In total, we needed eight additional Small Critters to explode both edge gadgets. This argument is repeatable when "connecting" more than two edge gadgets, but the number of additional Small Critters needed in order to explode the chain of Medium Critters remains the same: four per uncovered edge gadget.

There is another way of playing with edge gadgets, and it involves the concept of *overfeeding*. By "overfeeding" we mean eliminating a set of Medium Critters with a single explosion by feeding them more than two Small Critters. This is done by feeding some Medium Critters once to make them full, and then causing another Medium Critter in the same connected component to explode by feeding it twice. So, in an uncovered edge gadget, we can potentially overfeed once by putting everything to the right and feeding one Small Critter to the topmost neutral Medium Critter. This observation will come into play later, when we will count how many Small Critters have to be used to eliminate all edge gadgets.

For a singly covered edge, we can put the five bottom Critters to the right, explode the remaining colored Medium Critter (now bottom-most in the right column) using the two Small Critters from the left, and then put everything back to the left, thus exploding the four neutral Medium Critters in one go. We can overfeed the top Critter again, and we can also first explode the two bottom neutral-color Critters, thus using up to three more Small Critters to explode the whole gadget.

When an edge is covered by both incident vertices, no colored Medium Critter is left. We have to explode the two neutral pairs separately. We can simply use the four Small Critters coming with them (intended way to play) or overfeed both pairs once, which uses up two additional Small Critters at most.

Recall that the sink gadget contains one Medium Critter of each vertex color. There are $y = 5m + n$ consecutive sink gadgets in the left column (the reason why we chose this value of y will be clear later). When a vertex is selected, i.e., the corresponding vertex chain is exploded in the right column, all Medium Critters of the same color in the sink gadgets explode with them. After exploding k vertex chains in that fashion, we have $y(n - k)$ Medium Critters of $n - k$ colors remaining in the sink gadgets. We will ensure to have exactly enough Small Critters initially in the right column to explode all of these Medium Critters plus the Medium Critters in the offset: this amounts to $2y(n - k) + 2$ Small Critters.

The offset gadget consists of as many Critters of the second neutral color as there are Small Critters beside it, and $10m$ more to ensure we can never align gadgets. After the sink gadget is cleared by exploding the remaining Medium Critters one by one, we need to have two additional Small Critters on the right to explode the neutral-colored Medium Critters. This part ensures that the alignment of the gadgets works properly when we move gadgets from the left to the right column.

To summarize:

- We need an offset of $10m+2y(n-k)+2$ neutral-colored Medium Critters in the left column and $2y(n - k) + 2$ Small Critters in the right column. The Small Critters are used to explode the remaining Critters in the sink plus the offset Critters.
- Below the offset Critters, there are y copies of the sink gadget, which consist of yn Critters in total.
- We have one edge gadget per edge, so in total $10m$ Critters for all edges.
- There is also one copy of each vertex chain: The chain needs to be long enough to align with all Critters in the sink and edges when moved over to the right column, so each vertex chain should consist of $20m + yn$ Medium Critters of the respective vertex color, plus 2 Small Critters on top of the chain.

Clearly, any solvable instance of Vertex Cover is mapped to a solvable instance of Critter Crunch. On the other hand, it remains to be shown that we can only solve our Critter Crunch instance if the underlying Vertex Cover instance has a solution. We will discuss the ways to "cheat" in detail, which will also provide a justification for the exact value we chose for $y = 5m + n$.

As described above, we are able to overfeed Medium Critters sometimes to lose some Small Critters. In total, we are able to get rid of at most $5m + n$ Small Critters if we explode all vertex chains in the left column. If we play like this, however, we can never clear the board, because we cannot reorder the Critters in any way to make larger groups explode and "save" Small Critters to even out our overfeeding.

However, we can explode more than k vertex chains in the right column to get rid of some Medium Critters in the sink. Per additional vertex chain exploded in the right column (corresponding to an additional vertex chosen), we lose y Medium Critters in the sink and thus need $2y$ Small Critters less to explode the remaining ones.

If we explode $k + 1$ vertex chains on the right, we have at most $m-(k+1)$ uncovered, $(k+1)$ singly covered, and $(k+1)/2$ doubly covered edges (note that these numbers do not occur simultaneously, but an upper bound is sufficient for our purposes). As mentioned, each uncovered edge can be overfed by at most 5 Small Critters, a singly covered edge by up to 3 Critters, and a doubly covered edge can be overfed at most twice. Additionally, we can overfeed each vertex chain once. We obtain the following number of Small Critters that can be eliminated by overfeeding:

$$5(m-(k+1))+3(k+1)+2(k+1)/2+n = 5m-(k+1)+n.$$

To ensure that exploding $k + 1$ or more vertex chains in the right column can never lead to an empty board, we need $2y > 5m - (k + 1) + n$. Since we chose $y = 5m + n$. Indeed, exploding one additional vertex chain on the right "frees" $2y$ Small Critters that are not necessary to clear the sink gadgets' Medium Critters of the corresponding color anymore. These would need to be cleared through overfeeding, which is impossible for our choice of y.

The discussion above shows that now the only way to eliminate all Critters is to explode exactly k vertex chains in the right column, otherwise the numbers of Small and Medium Critters never add up. Also, after exploding all vertex chains, we need at least one colored Medium Critter per edge gadget to have exploded (i.e., the corresponding edge must be incident to at least one of the vertices we chose), otherwise we cannot explode the respective edge gadget without using Small Critters from the right column. After successfully exploding all edges, we need exactly $2y(n - k)$ Small Critters to explode the remaining $y(n - k)$ Medium

Critters of the sink and another two Small Critters to explode the offset Critters of neutral color. Overfeeding or playing "inefficiently" is in no case beneficiary. □

5. Levels with Critters of Only One Color

In this section we study Critter Crunch levels containing only normal Critters where all Critters of the same size have the same color. As in Sect. 4, we will give a characterization of the game's complexity based on the number of columns on the board: if the columns are only two, the game is NP-hard; in all other cases, it is decidable in linear time. (Once again, for boards with a single column the analysis is trivial.)

5.1 Boards with Three or More Columns

Theorem 4. *If a board has at least three columns and it contains s Small Critters, m Medium Critters of only one color, and l Large Critters of only one color, then all Critters can be eliminated if and only if:*

$$\begin{aligned}
&2 \le m \le 2l && \text{if } s = 0,\\
&1 \le m \le 2l && \text{if } s = 1,\\
&(m = 1 \wedge l = 0) \vee (m = 2 \wedge l \ne 1) \vee m \ge 3 && \text{if } s = 2,\\
&(s = 2m \wedge l = 0) \vee s < 2m && \text{if } s \ge 3.
\end{aligned}$$

Proof sketch. We borrow our strategy from Sect. 4: if the board has at least three columns and enough rows, we can put the Critters in a "canonical" configuration, and show that from there we can set up any desired primitive explosion. This is sufficient because, as we will see, if the Large Critters are all of the same color, then all sequences of explosion triplets (possibly containing combo-only explosions) can be converted into sequences of primitive explosions. Note that this is not necessarily true if Large Critters come in different colors: Fig. 2 shows two examples of boards that can be cleared, but not by primitive explosions only.

In this section, a configuration of Critters is called *canonical* if all Critters of the same size are in the same column. It is not hard to put the Critters in a canonical configuration from any starting configuration without causing any Critter to eat: the idea is to repeatedly transfer all Critters from one column to two other columns depending on their sizes. First we put Small and Large Critters together and we isolate the Medium ones, and finally we separate the Small and Large Critters.

Then we can show that, from a canonical configuration, it is possible to set up any desired primitive explosion and restore a canonical configuration again. Firstly, we show that we can permute the columns of a canonical configuration in every possible way without causing any Critters to eat. Then, if we are planning to explode m Medium Critters, we can move all the Medium Critters to the first column and only m of them to the third column: thus, the explosion of these m Critters will not propagate to the remaining ones in the first column. By "juggling" Critters in this fashion, we can indeed set up any primitive explosion that involves exactly the number of Critters that we want.

Next we have to show that, if a board can be cleared at all, it can be cleared by primitive explosions only. For every possible sequence of explosions, we have to show that there exists a sequence of primitive explosions that eliminates the same total amount of Small, Medium, and Large Critters. For instance, the sequence $((1, 2, 3), (2, 2, 1))$, consisting of a primitive initial explosion followed by a combo-only explosion, is equivalent to the sequence $((1, 1, 4), (2, 3, 0))$, which consists of two primitive explosions.

As a consequence, deciding if a given board can be cleared reduces to counting the Critters of each size and verifying if the numbers satisfy some linear inequalities. The details are below.

Proof. To complete the proof sketched above, we will prove four facts: (i) from any starting configuration, we can put the Critters in a canonical configuration without causing any of them to eat; (ii) from a canonical configuration, we can set up any desired primitive explosion and restore a canonical configuration; (iii) any sequence of explosions involving Critters of the same color can be replaced by a sequence of primitive explosions involving the same amounts of Critters; (iv) the conditions given in Theorem 4 are necessary and sufficient for a level to be solvable.

Let us start with (i). We say that two columns are *compatible* if at least one of them is empty or if the bottom-most Critter of the first column has a different size than the bottom-most Critter of the second column. We will obtain a canonical configuration in steps:

1. If the first two columns are not compatible, move the bottom-most Critter of the first column into the second column. Repeat until the first two columns are compatible.
2. Let us assign names to the first three columns. The third column will be called C. If the bottom-most Critter of the first or second column is Large, then that column is A and the other one is B. Otherwise, if the bottom-most Critter of the first or second column is Medium, then that column is B and the other one is A. Otherwise, the first column is A and the second is B. Note that, with this assignment, we can shoot a Small or Large Critter into column A or a Medium Critter into column B without causing any Critter to eat.
3. Move the bottom-most Critter of C to B if it is Medium or to A if it is Small or Large. Repeat until C is empty.
4. Repeat step 3 with all columns after the third one, emptying all of them into A and B.
5. Move the bottom-most Critter of B to C if it is Medium or to A if it is Small or Large. Repeat until B is empty. Now C contains only Medium Critters.
6. Move the bottom-most Critter of A to C if it is Medium or to B if it is Small or Large. Repeat until A is empty. Now C contains all the Medium Critters.
7. Move the bottom-most Critter of B to A if it is Small

or to C if it is Large. Repeat until B is empty. Now A contains all the Small Critters.

8. If the bottom-most Critter of C is Large, move it to B. Repeat until the bottom-most Critter of C is Medium or C is empty. Now the configuration is canonical.

To prove (ii), let us first show that we can arbitrarily permute the first three columns of a canonical configuration without causing any Critter to eat (and without using any column after the third one). To swap the Small and the Medium Critters, move all the Small ones under the Large ones, move the Medium ones to the empty column, and move the Small ones to the new empty column. To swap the Small and the Large Critters, move all the Large ones under the Medium ones, move the Small ones to the empty column, and move the Large ones to the new empty column. This is enough to obtain all permutations.

We will now show how to set up any primitive explosion and restore a canonical configuration:

- To set up $(2, m \geq 1, 0)$, start with the Medium Critters in the first column, the Large ones in the second, and the Small ones in the third. Move all the Small Critters to the second column, under the Large ones. Move m Medium Critters to the third column, feed two Small Critters to the last one, and move the rest of the Small Critters to the third column.
- To set up $(3, m \geq 2, 0)$, start with the Medium Critters in the first column, the Large ones in the second, and the Small ones in the third. Move all the Small Critters to the second column, under the Large ones. Move one Medium Critter to the third column, feed one Small Critter to it, move $m - 1$ Medium Critters to the third column, and feed two Small Critters to the last one. Move the rest of the Small Critters to the third column.
- To set up $(0, 2, l \geq 1)$, start with the Large Critters in the first column, the Small ones in the second, and the Medium ones in the third. Move all the Medium Critters to the second column, under the Small ones. Move l Large Critters to the third column, feed two Medium Critters to the last one, and move the rest of the Medium Critters to the third column.
- To set up $(0, 3, l \geq 2)$, start with the Large Critters in the first column, the Small ones in the second, and the Medium ones in the third. Move all the Medium Critters to the second column, under the Small ones. Move one Large Critter to the third column, feed one Medium Critter to it, move $l - 1$ Large Critters to the third column, and feed two Medium Critters to the last one. Move the rest of the Medium Critters to the third column.
- To set up $(1, 1, l \geq 1)$, start with the Large Critters in the first column, the Small ones in the second, and the Medium ones in the third. Move one Small Critter to the first column, and move all the Medium Critters to the second column, under the remaining Small ones. Feed the Small Critter in the first column to the last Medium Critter of the second column. This Medium Critter now becomes full. Move l Large Critters to the third column, feed the full Medium Critter to the last one, which causes the l Large Critters to explode. Finally, move the rest of the Medium Critters to the third column.
- To set up $(1, 2, l \geq 1)$, start with the Large Critters in the first column, the Small ones in the second, and the Medium ones in the third. Move one Small Critter to the first column, and move all the Medium Critters except one to the second column, under the remaining Small Critters. Feed the Small Critter in the first column to the last Medium Critter of the second column, and move the single Medium Critter from the third column to the second. Move l Large Critters to the third column, feed two Medium Critters (one empty and one full) to the last one, and move the rest of the Medium Critters to the third column.

We will now prove (iii). Consider a sequence of explosions, consisting of an initial one plus some combo ones. By Lemma 1, the initial explosion is equivalent to a sum of primitive explosions from Table 1, while each combo explosion is a sum of explosions from Table 1 and Table 2. So, the sequence of explosions can be decomposed into a sum of $p \geq 1$ primitive explosions plus a sum of q explosions of the form $(2, 1, l \geq 1)$ and $(2, 2, 1)$. Let us call *Medium explosion* an explosion of the form $(s, m \geq 1, 0)$, i.e., an explosion where only Medium Critters blow up. To prove (iii), it suffices to prove three facts: (a) if the p primitive explosions in the sum are all Medium explosions, then $q = 0$; (b) the sum of $(2, 2, 1)$ and any non-Medium primitive explosion is equivalent to a sum of primitive explosions, at least one of which is non-Medium; (c) a sum of any number of triplets of the form $(1, 2, l \geq 1)$ and any non-Medium primitive explosion is equivalent to a sum of primitive explosions. Indeed, if $q > 0$, the sum must contain at least one non-Medium primitive explosion, due to (a). Moreover, by (b), we can use such a non-Medium primitive explosion to eliminate every $(2, 2, 1)$ triplet from the sum while still retaining a non-Medium primitive explosion. Finally, by (c), we can use a single non-Medium primitive explosion to eliminate all triplets of the form $(1, 2, l \geq 1)$.

Let us prove (a). Assume for a contradiction that $q > 0$ and the p primitive explosions in the sum are all Medium explosions. Since a sum of Medium explosions is still a Medium explosion, we deduce that the initial explosion in our starting sequence was a Medium one. Since $q > 0$, this initial Medium explosion must have caused a combo explosion, meaning that $m > 0$ Medium Critters blew up in the initial explosion, and the empty space they left was occupied by at least one Critter C_1 below some of them, which climbed up into a larger Critter C_2 and got eaten. However, note that either C_1 or C_2 must be a Medium Critter, which by assumption has the same color as the m exploding Medium Critters. It follows that either C_1 or C_2 has exploded with the m Medium Critters, which is a contradiction.

To prove (b), let us consider all cases:

- $(2,2,1) + (0,2,l \geq 1) = (1,2,1) + (1,2,l)$.
- $(2,2,1) + (0,3,l \geq 2) = (0,2,1) + (1,1,1) + (1,2,l-1)$.
- $(2,2,1) + (1,1,l \geq 1) = (2,1,0) + (1,2,l+1)$.
- $(2,2,1) + (1,2,l \geq 1) = (3,2,0) + (0,2,l+1)$.

As for (c), we can reason as follows: suppose that we have a non-Medium primitive explosion of the form $(s,m,l \geq 1)$ and $k \geq 1$ explosions of the form $(2,1,l_i \geq 1)$. We can convert them into the primitive explosion $(s,m,l+\sum l_i)$ and k times $(2,1,0)$.

Finally, let us prove (iv). If $s = 0$ or if $l = 0$, then we are in the case of Theorem 2. If $s = 0 \wedge l = 0$ or $m = 0$, then no explosion is possible. So, in the following we assume $s \geq 1$, $m \geq 1$, and $l \geq 1$. What remains to be proven is that there is a solution except if one of the following holds:

$$s = 1 \wedge m > 2l, \tag{1}$$

$$s = 2 \wedge m = 2 \wedge l = 1, \tag{2}$$

$$s \geq 2m. \tag{3}$$

We start by proving that in those cases no solution exists. If Eq. (1) holds, there is no explosion that uses the Small Critter and helps to reduce the number of Medium Critters when compared to the Large Critters. Indeed, similar to Theorem 2, even when using the explosion $(0,2,1)$ until there are no more Large Critters, there will still be Medium Critters left that cannot be removed. For Eq. (2), a simple case analysis shows that it is impossible to remove all Critters. Finally, for Eq. (3), we again use Theorem 2, as Large Critters do not help when it comes to removing more Small Critters than Medium ones.

Assume now that none of the three equations holds. We make the following case distinction:

- Let $s < m$.
  - If $s = 1$, then we must have $m \leq 2l$. We feed the Small Critter to a Medium one and then apply Theorem 2 (the full Medium Critter will be the last to be fed to a Large one).
  - If $s \geq 2$, we start with the primitive explosion $(0,2,l)$, obtaining $(s, m' = m-2, 0)$. Since $m' \geq 1$, we have $2 \leq s \leq m-1 = m'+1 \leq 2m'$. So we can apply Theorem 2.
- Let $s = m$.
  - If $s = m = 1$, we use the primitive explosion $(1,1,l)$.
  - If $s = m = 2$, we must have $l \geq 2$. Then we use the primitive explosions $(1,1,1)$ and $(1,1,l-1)$.
  - If $s = m \geq 3$, we start with the primitive explosion $(1,1,l)$, obtaining $(s' = s-1, m' = m-1, 0)$. Since $2 \leq s' = m' \leq 2m'$, we can apply Theorem 2.
- Let $s > m$. Note that we must also have $s < 2m$, implying that $s \geq 3$. We start with the primitive explosion $(1,1,l)$, obtaining $(s' = s-1, m' = m-1, 0)$. Since $2 \leq s' = s-1 \leq 2m-2 = 2m'$, we can apply Theorem 2. □

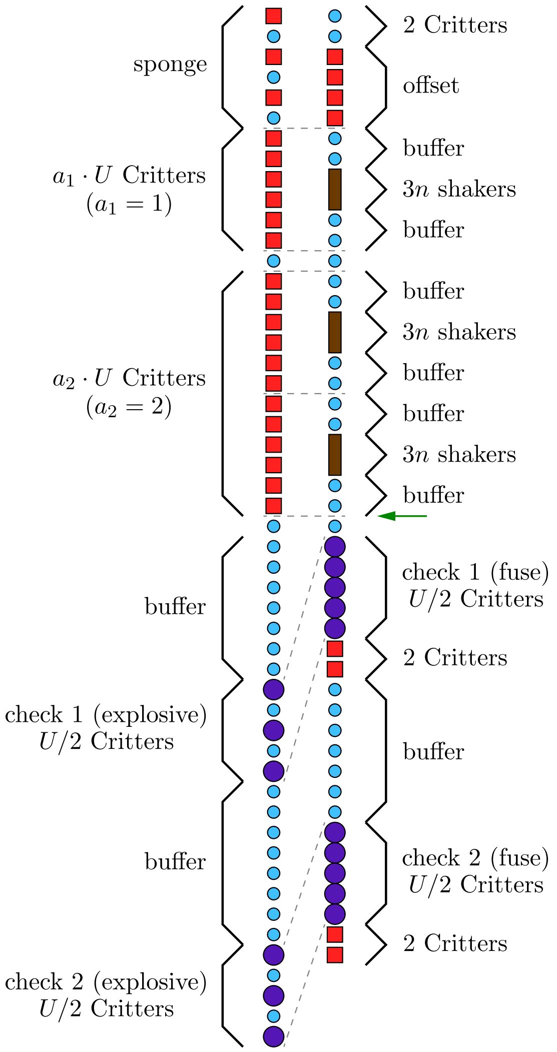

Fig. 6 Sketch of the reduction from 3-Partition (not to scale). Each brown rectangle represents $3n$ shaker gadgets. We have $U = 40Sn^2$, where $S = \sum a_i$. The green arrow marks the starting point.

5.2 Boards with Only Two Columns

We will prove the NP-hardness of Critter Crunch for boards of only two columns where there are no special Critters and no color variations.

Theorem 5. *It is NP-hard to decide if it is possible to clear a board having exactly two columns and containing only normal Critters with no color variations.*

Proof sketch. Our reduction is again from 3-Partition, where the input are $3n$ positive integers $a_1, a_2, \ldots, a_{3n}$, and the goal is to decide if they can be partitioned into n sets of equal sum. Let S be the sum of the a_i's, and let $B = S/n$. Recall that it is not restrictive to assume that $B/4 < a_i < B/2$ for all $1 \leq i \leq 3n$, which forces every set of sum B to have exactly three elements.

Our reduction is sketched in Fig. 6. In the top-left part we have the *sponge*, a long alternating sequence of Medium and Small Critters: its purpose is to eliminate all the Small Critters that will be left after the player is done solving the 3-Partition instance. The sponge is matched in length by the *offset*: a sequence of Medium Critters (topped by two Small Critters) in the right column, which gives the correct alignment to the gadgets that come next.

Below the sponge, we have $3n$ sequences of Medium Critters separated by single Small Critters: each of these sequences represents an a_i. The length of the ith sequence is $a_i \cdot U$, where $U = 40Sn^2$ is the length of a "unit". In the right column, corresponding to each unit, we have a sequence of $3n$ *shaker gadgets*, which will be described later (each sequence of $3n$ shaker gadgets is represented by a brown rectangle in Fig. 6). To fill the empty space around the shaker gadgets, we use *buffers* of Small Critters.

After that, we have n *check gadgets*: each of them has the purpose of verifying that the player has chosen a set of a_i's with the correct sum of B. A check gadget extends over both columns: in the left column it consists of Large and Small Critters, alternating each other in a sequence of length $U/2$; in the right column, it consists of a sequence of $U/2$ Large Critters followed by two Medium Critters. Check gadgets are separated by sequences of Small Critters: in the left column, each check gadget is topped by a *buffer* of $B \cdot U$ Small Critters; in the right column, check gadgets are separated by *buffers* of a variable number of Small Critters, which have the purpose of ensuring the proper alignment between gadgets.

The green arrow in Fig. 6 marks the "starting point", meaning that everything below it is initially in the left column. It should be noted that the configuration in Fig. 6 never actually occurs when the level is played (moving a pair of Medium Critters to the right column causes a fuse gadget to explode). The figure has been drawn this way to illustrate the alignment between explosive gadgets and fuse gadgets.

The proper way to play this level is to use the shaker gadgets to eliminate some a_i's (details on the shaker gadgets will be given later). Once the sum of the eliminated a_i's is exactly B, the check gadgets in the left column have moved up by $B \cdot U$, i.e., the size of the buffers above them. At this point, we can reach the two Medium Critters of the first check gadget and use them to explode the sequence of Large Critters in the right column. If everything was done correctly, the left and right parts of the first check gadget will be perfectly aligned, and all their Large Critters will be eliminated by the explosion. Then we can proceed to eliminate more a_i's, and so on. Once all the check gadgets have been properly eliminated, the only Critters left will be those in the buffers, in the sponge, and in the offset. The offset can consume at most two Small Critters (the ones at the top of the right column), and each Medium Critter in the sponge must consume exactly one Small Critter from the buffers; the length of the sponge is chosen in such a way that these numbers add up exactly.

Now we describe the shaker gadget, shown in Fig. 7. Its purpose is to give the player two options: either "skip" it to access some other gadgets above or below it, or destroy it. Two consecutive skips, one going up and one going down, are called a *shake*. As long as the shaker gadget is skipped, only its Large Critters explode, and this has no side effect on the a_i's in the left column. However, when the shaker gadget is destroyed, one of its Medium Critters eats the two bottom Small Critters, thus exploding and blowing up the a_i on its left. A shaker gadget should only allow n shakes, after which it is automatically destroyed: such is the number of shakes that the player needs to select a set of a_i's for n times. Figure 7 shows an example of a shaker gadget for $n = 3$, as well as its evolution after one and two shakes (after three shakes, no Critter is left).

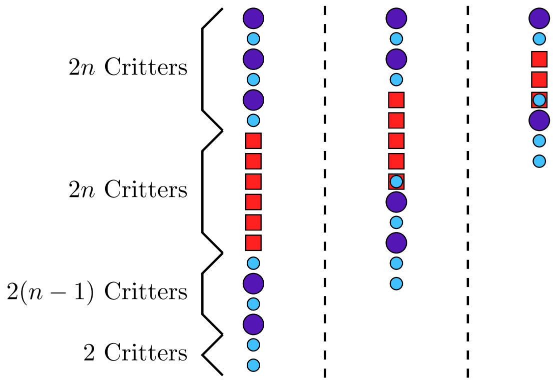

Fig. 7 Shaker gadget. Left: initial configuration for $n = 3$. Center: after one shake. Right: after two shakes.

Suppose that the player fails to solve the 3-Partition instance and blows up some a_i's with sum different than B. This means that the left part of the next check gadget moves up by the wrong multiple of U. Since the length of the left and right parts of the check gadget is $U/2$, it follows that they fail to overlap when the Large Critters on the right blow up. This leaves $U/4$ unexploded Large Critters on the left. The Medium Critters in the offset cannot be used to eliminate these Large Critters, because of the buffers of Small Critters between them. If the sponge were used to eliminate them, then there would not be enough Medium Critters to eat all the Small Critters in the buffers. So, the only way to eliminate these Large Critters is to align them with the shaker gadgets and blow them up with their Large Critters. However this is impossible, because we chose U so that $U/4$ is greater than the total number of Large Critters in the shaker gadgets. The missing details are given below.

Proof. We will provide additional details on the gadgets, starting with the shaker gadget.

As we mentioned, the intended way to use a shaker gadget is to explode it in the right column, next to an a_i-chain, in order to eliminate it: this corresponds to "selecting" that particular a_i as part of a triplet with sum B. In general, we want to blow up a triplet of shakers t_1, t_2, t_3, next to three a_i-chains we want to eliminate. To do so, we first move all shakers to the left column without exploding them. Then we move them back to the right column one by one; whenever we reach one of the three shakers t_i, we explode it in the right column. This procedure has two effects: the left column is reduced by $B \cdot U$ Medium Critters (assuming the three a_i's we selected have sum B), and the right column is reduced by a fixed number of Critters (additional details on this will be given later).

Hence, any shaker gadget should give us the option to "skip" it (by "shaking" it back and forth) for $n - 1$ times before it explodes. In other words, a shaker should be large enough to be moved to the left column and then back to the right column for exactly n times. Thus we construct a shaker gadget as follows: staring with a Large Critter, we alternate n Large Critters with the same number of Small Critters, followed by $2n$ Medium Critters. Below that we place, starting with a Small Critter another alternating chain of Small and Large Critters of total size $2n - 2$, $n - 1$ of each size, followed by another two Small Critters. The total size of a shaker gadget sums up to $h = 6n$. Note that the first shake decreases a shaker's total size by 4 Critters, while all following shakes (apart from the exploding last shake) decrease its size by 6. As mentioned, we place $3n$ shaker gadgets per unit length of Medium Critters in the a_i-chains. The buffer surrounding the $3n$ copies of the shaker gadget per unit is therefore of size $U - 3n \cdot h$.

The reason why we place $3n$ shaker gadgets per unit is because we need to use exactly $3n$ shaker gadgets overall to blow up all the a_i-chains. Eliminating one a_i-chain causes the chains below it to move up by a_i "units". So, even though we always select different a_i's, we may have to blow up shakers next to the same unit. This is why having $3n$ shaker gadgets corresponding to each unit ensures that at all times there is at least one shaker gadget aligned with any a_i-chain we would like to explode.

Below the a_i-chains are the check gadgets. Each one consists of the *explosive gadget*, followed (below all n explosive gadgets) by the *fuse gadget* (of which we also have n copies). The check ensures that we always have to explode three a_i-chains with values of total sum B, perform a check, and repeat until all a_i-chains are exploded and all checks have been performed successfully.

The explosive gadget is as follows: we have D_1 Small Critters, followed by an alternating chain of C Large and C Small Critters, where $C = U/4$. After that we have another D_2 Small Critters, followed again by an alternating chain of C Large and C Small Critters, and so on. As we need to have n checks, in the end we have D_n Small Critters, followed by the last alternating chain of C Large and C Small Critters. We set $D_1 = BU - 3h - (3Sn - 3)4$ and, for $i \in \{2, \dots, n-1\}$,

$$D_i = D_{i-1} - 3(h - 4 - 6(i-2)) - 6(3Sn - 3i).$$

Lastly we set

$$\begin{aligned} D_n &= D_{n-1} - (3nS - 3(n-1))(h - 4 - 6(n-2)) \\ &= BU - 3nSh. \end{aligned}$$

The number D_1 is chosen such that the following holds: we force that three a_i-chains with a total value of B have exploded, i.e., the lowest Critters in the left column have moved up by BU. In order to explode three a_i-chains, we need to completely remove three shakers, hence the term $3h$. Furthermore, we moved each shaker back and forth exactly once (if we explode the topmost available shaker of each a_i and explode the a_i's of a triplet from top to bottom). The first time a shaker is moved back and forth, its size reduces by 4 and there are a total number of $3Sn$ shakers. Thus we get the term $(3Sn - 3)4$.

The definition of the other D_i's follows the same principle. Again 3 shaker gadgets have to be exploded, but now their size is smaller. The size of a gadget reduces by 4 for the first shake and by 6 for each following shake. The value D_n is special, because all shakers disappear completely during the last shake.

Below the explosive gadgets lie the n fuse gadgets of the check. Each gadget consists of n copies of the following: C Small Critters, below them are two Medium Critters, and then $2C - 1$ Large Critters. Note that when moving a fuse gadget to the right column the Large Critters explode. This cannot be avoided.

Let us summarize the intended way to play, assuming that we know a partition of the set $\{a_1, \dots, a_{3n}\}$ into triplets such that the sum of each triplet equals B. We take one such triplet $\{a_i, a_j, a_k\}$, with $i < j < k$. Let t be the topmost shaker in the topmost unit that is adjacent to the chain of Medium Critters representing a_i. We move all shakers from the right to the left column. Then we move as many shakers back to right column as we need to access a_i, i.e., all shakers above t. We explode t in the right column by moving it back and forth between the two columns. This removes the chain of Medium Critters representing a_i. We remove the Medium Critters associated with a_j and a_k in the same way. We move all shakers back to the right column to regain our "default position". Then we move the bottom-most fuse over to the right side, which causes it to explode. As we have $a_i + a_j + a_k = B$, the Large Critters from the fuse align perfectly with the Large Critters of the topmost explosive gadget, which means that the explosion removes all those Large Critters, which is a "successful" check.

This procedure is repeated for each other triplet. In the end, all shakers have exploded. The sponge and the offset gadget have not yet changed, but from the other gadgets only Small Critters remain. What remains are n Small Critters that used to separate the a_i-chains, $\sum_{i=1}^{n} D_i + nC$ Small Critters from the explosives, nC Small Critters from the fuse gadgets, and $S(U - 3nh)$ Small Critters from the buffers between the shakers. We feed one Small Critter to the offset gadget and then explode the offset gadget on the left column. Finally, we remove the Small Critters by exploding the Medium Critters from the sponge one after the other. Each Medium Critter eats the Small Critter below it plus one Small Critter from the pool of remaining Critters, so we need $K = (2C + 1)n + \sum_{i=1}^{n} D_i + S(U - 3nh)$ Medium Critters, which sets the total size of sponge and offset to $2K$. Note that, as $2Cn = 20Sn^3 > 18Sn^2 = S(3nh)$, we have $K > 2Cn + S(U - 3nh) > SU$.

We still need to prove that one cannot solve the Critter

Crunch instance without solving the underlying instance of 3-Partition. A player may try to "cheat" the game and clear the board without playing in the intended way. A first idea is to explode the offset before using the shakers to access the a_i-chains. As $K > SU$, after having exploded the offset, a fuse on the right can no longer be aligned with an explosive on the left. Furthermore, the sponge cannot be accessed without exploding all fuses. It follows that if the offset is exploded before all fuses are used, there will be Large Critters in the explosive gadgets that cannot be exploded.

Now, imagine that we chose (i.e., exploded) a number of a_i-chains whose sum is not B. The a_i's are all integers, and therefore the sum of the exploded a_i's differs at least by 1. It follows that the next fuse does not align with the respective explosives by a height of at least U. We can try to realign the Large Critters from the Fuse with the Large Critters from the explosives by reducing the size of some shakers by moving them back and forth multiple times.

At the beginning there are $3nS$ shakers, and each shaker has a starting size of h. Since $U = 40Sn^2 > 2\times 3nSh$, even if all shakers are exploded early, they can change the alignment between explosive gadgets and fuses by at most $U/2$. A single explosive check contains $U/2$ Critters, $U/4$ of which are Large. This means that if the sum of an exploded triplet of a_i's is not B, none of those $U/4$ Large Critters will be exploded by the corresponding fuse.

In order to remove those Large Critters and solve the instance, we have two options. Either we try and explode those Large Critters by aligning them with other Large Critters that will explode anyway, or we explode them using Medium Critters. The only Large Critters outside the check gadgets are the ones in the shakers. There are $2n + 1$ Large Critters in one shaker and at the start there are $3nS$ shakers. Since $U > 6nSh$ and $h = 6n$, we have $U/4 > 3nS(2n + 1) + 3nS$, and therefore at least $3nS$ Large Critters need to be removed using Medium Critters. However, there are exactly enough Medium Critters to remove all Small Critters, so no Medium Critter can be fed to a Larger Critter.

As discussed above, any solvable 3-Partition instance is mapped to an instance of Critter Crunch where the board can be emptied completely by playing the intended way (and only the intended way). An unsolvable instance leads to misaligned check gadgets, i.e., we get Large Critters that cannot be exploded. □

Acknowledgments

The authors are grateful to the anonymous reviewers for greatly improving the quality of this paper with helpful comments and suggestions. The third author is supported by the Netherlands Organization for Scientific Research (NWO) under project no. 612.001.651.

References

[1] A. Adler, E.D. Demaine, A. Hesterberg, Q. Liu, and M. Rudoy, "Clickomania is hard even with two colors and columns," The Mathematics of Various Entertaining Subjects (MOVES 2015), vol.2, pp.325–363, Princeton University Press, 2017.

[2] R. Breukelaar, E.D. Demaine, S. Hohenberger, H.J. Hoogeboom, W.A. Kosters, and D. Liben-Nowell, "Tetris is hard, even to approximate," International Journal of Computational Geometry & Applications, vol.14, no.01-02, pp.41–68, 2004.

[3] E.D. Demaine and S. Langerman, "Bust-a-Move/Puzzle Bobble is NP-complete," Proceedings of the 18th Japan Conference on Discrete and Computational Geometry and Graphs (JCDCGG 2015), pp.94–104, 2015.

[4] L. Gualà, S. Leucci, and E. Natale, "Bejeweled, Candy Crush and other match-three games are (NP-)hard," Proceedings of the 10th IEEE Conference on Computational Intelligence and Games (CIG 2014), pp.32:1–32:8, 2014.

[5] G. Kendall, A. Parkes, and K. Spoerer, "A survey of NP-complete puzzles," International Computer Games Association Journal, vol.31, no.1, pp.13–34, 2008.

[6] G. Viglietta, "Lemmings is PSPACE-complete," Theoretical Computer Science, vol.586, pp.120–134, 2015.

[7] Critter Crunch | Puzzle Guide | Levels 1–10. https://www.youtube.com/watch?v=LEfT3VHw8lc, 2013.

Appendix A: Primitive and Combo-Only Explosions

We will prove four facts: (i) Table 1 contains only feasible initial explosion triplets; (ii) every feasible initial explosion triplet can be expressed as a sum of triplets from Table 1; (iii) the triplets in Table 2 cannot be initial explosions; (iv) every feasible explosion triplet can be expressed as a sum of triplets from Table 1 and Table 2.

Let us prove (i) first. The triplet $(2, m \geq 1, 0)$ is obtained by shooting two Small Critters into an exposed Medium Critter of a connected component of m units. The triplet $(3, m \geq 2, 0)$ is obtained by having a connected component of m Medium Critters and shooting one Small Critter into one of them and two Small Critters into another. The triplets $(0, 2, l \geq 1)$ and $(0, 3, l \geq 2)$ are obtained in a similar way. The triplet $(1, 1, l \geq 1)$ can be triggered by feeding one Small Critter to a Medium Critter, and then shooting the Medium Critter into a Large Critter in a connected component of l units. Finally, the triplet $(1, 2, l \geq 1)$ is obtained by feeding one empty Medium Critter to a Large Critter in a connected component of l units, and then feeding a full Medium Critter to the same Large Critter.

We will now prove (iii). Consider the explosion triplet $(2, 1, l \geq 1)$. This triplet involves just one Medium Critter, which has to eat two Small Critters and at the same time be fed to a Large Critter without exploding. The only way this can happen as a result of a player's move is by triggering a food chain: a full Medium Critter must be sitting under an empty Large Critter, and the player must shoot a Small Critter into this Medium Critter. However, there is no way a full Medium Critter can be right under a Large Critter: if the Medium Critter had collided with the Large Critter, the Large Critter would have eaten it. So, the two Critters must have been adjacent since the beginning. But in this case, the Medium Critter could not be full, otherwise it would have been eaten by the Large Critter in a food chain at the time of eating its first Small Critter.

Consider now the explosion triplet $(2, 2, 1)$. Two Medium Critters have to be fed to a Large Critter, but they must have eaten two Small Critters as well. Suppose the two Medium Critters eat one Small Critter each. Then, the first Medium Critter to be fed to the Large Critter would cause it to explode immediately, which means it would not eat the second Medium Critter. It follows that one of the two Medium Critters eats both Small Critters and gets eaten by the Large Critter before exploding, which cannot happen as an initial explosion, as already proved in the previous paragraph.

To prove (iv), consider a generic explosion triplet (s, m, l). Note that every Critter can eat at most two other (possibly full) Critters. So, we have the inequalities $s \leq 2m$ and $m \leq 2l$. Suppose the Critters that actually explode are Medium Critters: this means that $l = 0$ and $s \geq 2$ (we need at least two Small Critters to trigger an explosion of Medium Critters). We can prove by induction on s that the triplet $(s \geq 2, m \geq \lceil s/2 \rceil, 0)$ can be expressed as a sum of the two first primitive triplets of Table 1. For $s = 2$ and $s = 3$, we have exactly the first and second primitive triplets, respectively. Suppose now that $s \geq 4$ and that our claim is true for $s - 2$. Then we can subtract the primitive triplet $(2, 1, 0)$ (first row of Table 1) to obtain the triplet $(s - 2, m \geq \lceil s/2 \rceil - 1, 0)$. If $s = 2s'$ is even, we have $m \geq \lceil s/2 \rceil - 1 = s' - 1 = \lceil (s - 2)/2 \rceil$, so the new triplet satisfies the inductive hypothesis. If $s = 2s' + 1$ is odd, we have $m \geq \lceil s/2 \rceil - 1 = s' = \lceil (s - 2)/2 \rceil$, and again the new triplet satisfies the inductive hypothesis.

Let us now assume that the explosion involves Large Critters, and therefore $l \geq 1$. If $s = 0$, the same argument used in the previous paragraph proves that our explosion triplet can be obtained as a sum of the third and fourth primitive triplets of Table 1. So, let us assume that $s \geq 1$, and therefore also $m \geq 1$, and let us prove our claim by induction on s: specifically, we will prove that, if $1 \leq s \leq 2m$, $1 \leq m \leq 2l$, and $l \geq 1$, then the explosion triplet (s, m, l) can be expressed as a sum of triplets from Table 1 and Table 2.

Assume first that $s = 1$. If $m = 1$ or $m = 2$, we have the fifth or the sixth primitive triplet of Table 1, so let us assume $m \geq 3$, and therefore $l \geq 2$. Now we can subtract the primitive triplet $(1, 2, 1)$ (sixth row of Table 1) to obtain the triplet $(0, m' = m-2, l' = l-1)$, with $1 \leq m' \leq 2l'$ and $l' \geq 1$. As we have already seen, this triplet can be expressed as a sum of the third and fourth primitive triplets of Table 1.

Let us assume that $s = 2$. If $m = 1$, we have the first combo-only triplet of Table 2; if $m = 2$ and $l = 1$, we have the second combo-only triplet of Table 2. Let us assume now that $m = 2$ and $l \geq 2$. The triplet $(2, 2, l \geq 2)$ is the sum of the two triplets $(1, 1, 1)$ and $(1, 1, l - 1)$, both of which are in the fifth row of Table 1. Assume now that $m \geq 3$. We can subtract the first primitive triplet of Table 1, $(2, 1, 0)$, to obtain the triplet $(0, m' = m - 1, l)$, where $2 \leq m' \leq 2l - 1 < 2l$. As we have already proved above, this triplet can be expressed as a sum of the third and fourth primitive triplets of Table 1.

Finally, assume that $s \geq 3$, which implies that $m \geq 2$, and assume that the inductive hypothesis holds for $s - 2$. We can subtract the primitive triplet $(2, 1, 0)$ (first row of Table 1) to obtain the triplet $(s' = s - 2, m' = m - 1, l)$, with $1 \leq s' \leq 2m - 2 = 2m'$ and $1 \leq m' \leq 2l - 1 < 2l$. By the inductive hypothesis, this triplet can be expressed as a sum of triplets from Table 1 and Table 2.

Finally, let us prove (ii). Recall that initial explosions are those that are a direct consequence of a player's move (as opposed to combo explosions, which are a consequence of other explosions). Suppose that a move causes $m > 0$ Medium Critters and $l = 0$ Large Critters to explode. As we have already proved for (iv), this explosion is a sum of the first and second primitive explosions of Table 1. Suppose now that a move causes $m = 0$ Medium Critters and $l > 0$ Large Critters to explode. If the Medium Critters that are fed to these Large Critters have never eaten, then we can repeat the previous argument to conclude that our explosion triplet can be expressed as a sum of the third and fourth primitive triplets of Table 1.

So, assume that the explosion triplet is of the form (s, m, l), with $s, l \geq 1$. Since Large Critters only explode when one of them is fed some Medium Critters, then $m \geq 1$ as well. Let $l = l_0 + l_1 + 1$, where l_0 is the number of Large Critters that have never eaten, and l_1 is the number of Large Critters that have eaten exactly one empty Medium Critter. The "+1" accounts for a single Large Critter that has been fed either one full Medium Critter or one empty Medium Critter followed by one full Medium Critter: this is the move that triggered the explosion (note that we do not have to consider the case where the Large Critter has eaten two empty Medium Critters, because this would imply $s = 0$). Thus, we have either $m = l_1 + 1$ or $m = l_1 + 2$: in the first case, we have a triplet of the form $(1, m \geq 1, l \geq m)$, and in the second case we have $(1, m \geq 2, l \geq m - 1)$. All triplets of the first form are already covered by triplets of the second form, except for the case $m = 1$: this case yields triplets of the form $(1, 1, l \geq 1)$, which is the fifth primitive triplet of Table 1. Let us now consider the triplets of the second form: $(1, m \geq 2, l \geq m - 1)$. The case $m = 2$ is given by the sixth primitive triplet of Table 2, so let us assume that $m \geq 3$, implying that $l \geq 2$. In this case, we can subtract the triplet $(1, 1, 1)$ (fifth row of Table 1) to obtain a triplet of the form $(0, m' = m - 1, l' = l - 1)$. Since $2 \leq m' = m - 1 \leq l = l' + 1 \leq 2l'$, as seen for (iv), this triplet can be expressed as a sum of the third and fourth primitive triplets of Table 1.

Appendix B: NP-Hardness of Puzzle Mode

Using the same framework as in Sect. 3, we can show that "puzzle mode" with only normal Critters is NP-hard; an example is shown in Fig. A· 1. Here we have a limit of $4n$ moves; this is exactly the number of Small Critters in the reservoir. What forces us to "play by the rules" of 3-Partition is the fact that we have $4n$ rows of Small Critters just above the reservoir, and $4n$ more rows of Small Critters

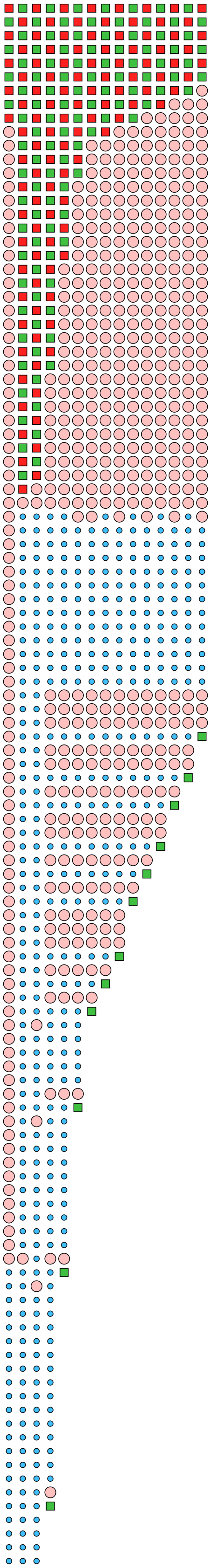

Fig. A· 1 Reduction from 3-Partition to Critter Crunch levels where a limit on the number of moves is given (puzzle mode). In this example the player can perform at most 12 moves.

just below the upper part of the board, under the Large Critters. This means that we do not have enough moves to "dig" through these Small Critters and blow up the big connected component of Large Critters. Similarly, rearranging Critters around the triggers does not help: our ultimate goal is still to align the bottom-most Large Critters of the second, third, and fourth columns. No matter how we rearrange Critters in the columns after the fourth one, we still end up blowing up segments of Large Critters in the fourth column that correspond to the original payloads. Hence, the only way to clear the board is to solve the 3-Partition instance.

Tianfeng Feng is a PhD student at the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST). She is a member of Uehara Laboratory.

Leonie Ryvkin née Lange, received her Master's degree in Mathematics in 2017 from the Ruhr University Bochum. She is now a PhD student in the Lehrstuhl Mathematik & Informatik at the Ruhr University Bochum, supervised by Maike Buchin.

Jérôme Urhausen received his Master's degree in Computer Science from the Karlsruhe Institute of Technology (DE) in 2017. He is now a PHD student at Utrecht University (NL) under the supervision of Marc van Kreveld, Maarten Löffler and Frank Staals.

Giovanni Viglietta received his PhD in Computer Science from the University of Pisa in 2013. He is now Assistant Professor at the Japan Advanced Institute of Science and Technology (JAIST).