The Three-Color and Two-Color Tantrix^{TM} Rotation Puzzle Problems Are NP-Complete Via Parsimonious Reductions*

Dorothea Baumeister and Jörg Rothe

Institut für Informatik, Universität Düsseldorf, 40225 Düsseldorf, Germany

Abstract. Holzer and Holzer [7] proved that the Tantrix^{TM} rotation puzzle problem with four colors is NP-complete, and they showed that the infinite variant of this problem is undecidable. In this paper, we study the three-color and two-color Tantrix^{TM} rotation puzzle problems (3-TRP and 2-TRP) and their variants. Restricting the number of allowed colors to three (respectively, to two) reduces the set of available Tantrix^{TM} tiles from 56 to 14 (respectively, to 8). We prove that 3-TRP and 2-TRP are NP-complete, which answers a question raised by Holzer and Holzer [7] in the affirmative. Since our reductions are parsimonious, it follows that the problems Unique-3-TRP and Unique-2-TRP are DP-complete under randomized reductions. Finally, we prove that the infinite variants of 3-TRP and 2-TRP are undecidable.

1 Introduction

The puzzle game Tantrix^{TM}, invented by Mike McManaway in 1991, is a domino-like strategy game played with hexagonal tiles in the plane. Each tile contains three colored lines in different patterns (see Figure 1). We are here interested in the variant of the Tantrix^{TM} rotation puzzle game whose aim it is to match the line colors of the joint edges for each pair of adjacent tiles, just by rotating the tiles around their axes while their locations remain fixed. This paper continues the complexity-theoretic study of such problems that was initiated by Holzer and Holzer [7]. Other results on the complexity of domino-like strategy games can be found, e.g., in Grädel's work [6]. Tantrix^{TM} puzzles have also been studied with regard to evolutionary computation [4].

Holzer and Holzer [7] defined two decision problems associated with four-color Tantrix^{TM} rotation puzzles. The first problem's instances are restricted to a finite number of tiles, and the second problem's instances are allowed to have infinitely many tiles. They proved that the finite variant of this problem is NP-complete and that the infinite problem variant is undecidable. The constructions in [7] use tiles with four colors, just as the original Tantrix^{TM} tile set. Holzer and Holzer posed the question of whether the Tantrix^{TM} rotation puzzle problem remains NP-complete if restricted to only three colors, or if restricted to otherwise reduced tile sets.

* Full version: [2]; see also Baumeister's Master Thesis "Complexity of the Tantrix^{TM} Rotation Puzzle Problem," Universität Düsseldorf, September 2007. Supported in part by DFG grants RO 1202/9-3 and RO 1202/11-1 and the Humboldt Foundation's TransCoop program.

C. Martín-Vide, F. Otto, and H. Fernau (Eds.): LATA 2008, LNCS 5196, pp. 76–87, 2008.

Table 1. Overview of complexity and decidability results for k-TRP and its variants

k	k-TRP is	Parsimonious?	Unique-k-TRP is	Inf-k-TRP is
1	in P (trivial)		in P (trivial)	decidable (trivial)
2	NP-compl., Cor. 3	yes, Thm. 2	DP-$\leq^p_{ran}$-compl., Cor. 4	undecidable, Thm. 3
3	NP-compl., Cor. 1	yes, Thm. 1	DP-$\leq^p_{ran}$-compl., Cor. 4	undecidable, Thm. 3
4	NP-compl., see [7]	yes, see [1]	DP-$\leq^p_{ran}$-compl., see [1]	undecidable, see [7]

In this paper, we answer this question in the affirmative for the three-color and the two-color version of this problem. For $1 \leq k \leq 4$, Table 1 summarizes the results for k-TRP, the k-color Tantrix$^{\text{TM}}$ rotation puzzle problem, and its variants. (All problems are formally defined in Section 2.)

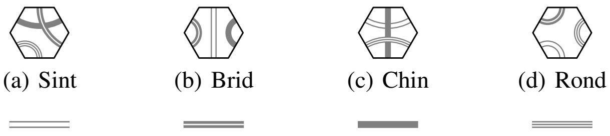
Since the four-color Tantrix$^{\text{TM}}$ tile set contains the three-color Tantrix$^{\text{TM}}$ tile set, our new complexity results for 3-TRP imply the previous results for 4-TRP (both its NP-completeness [7] and that satisfiability *parsimoniously* reduces to 4-TRP [1]). In contrast, the three-color Tantrix$^{\text{TM}}$ tile set does not contain the two-color Tantrix$^{\text{TM}}$ tile set (see Figure 2 in Section 2). Thus, 3-TRP does not straightforwardly inherit its hardness results from those of 2-TRP, which is why both reductions, the one to 3-TRP and the one to 2-TRP, have to be presented. Note that they each substantially differ—both regarding the subpuzzles constructed and regarding the arguments showing that the constructions are correct—from the previously known reductions [7,1], and we will explicitly illustrate the differences between our new and the original subpuzzles.

Since we provide *parsimonious* reductions from the satisfiability problem to 3-TRP and to 2-TRP, our reductions preserve the uniqueness of the solution. Thus, the unique variants of both 3-TRP and 2-TRP are DP-complete under polynomial-time randomized reductions, where DP is the class of differences of NP sets. We also prove that the infinite variants of 3-TRP and 2-TRP are undecidable, via a circuit construction similar to the one Holzer and Holzer [7] used to show that the infinite 4-TRP problem is undecidable.

2 Definitions and Notation

Complexity-Theoretic Notions and Notation: We assume that the reader is familiar with the standard notions of complexity theory, such as the complexity classes P (deterministic polynomial time) and NP (nondeterministic polynomial time). DP denotes the class of differences of any two NP sets [9].

Let Σ^* denote the set of strings over the alphabet $\Sigma = \{0,1\}$. Given any language $L \subseteq \Sigma^*$, $\|L\|$ denotes the number of elements in L. We consider both decision problems and function problems. The former are formalized as languages whose elements are those strings in Σ^* that encode the yes-instances of the problem at hand. Regarding the latter, we focus on the counting problems related to sets in NP. The counting version #A of an NP set A maps each instance x of A to the number of solutions of x. That is, counting problems are functions from Σ^* to $\mathbb{N}$. As an example, the counting version #SAT of SAT, the NP-complete satisfiability problem, asks how many satisfying assignments a given boolean formula has. Solutions of NP sets can be viewed as accepting paths of



(a) Sint (b) Brid (c) Chin (d) Rond

(e) red (f) yellow (g) blue (h) green

Fig. 1. Tantrix$^{\text{TM}}$ tile types and the encoding of Tantrix$^{\text{TM}}$ line colors

NP machines. Valiant [10] defined the function class #P to contain the functions that give the number of accepting paths of some NP machine. In particular, #SAT is in #P.

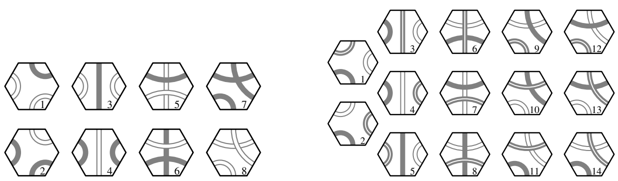
The complexity of two decision problems, A and B, will here be compared via the *polynomial-time many-one reducibility*: $A \leq_m^p B$ if there is a polynomial-time computable function f such that for each $x \in \Sigma^*$, $x \in A$ if and only if $f(x) \in B$. A set B is said to be NP-complete if B is in NP and every NP set $\leq_m^p$-reduces to B.

Many-one reductions do not always preserve the number of solutions. A reduction that does preserve the number of solutions is said to be *parsimonious*. Formally, if A and B are any two sets in NP, we say *A parsimoniously reduces to B* if there exists a polynomial-time computable function f such that for each $x \in \Sigma^*$, $\#A(x) = \#B(f(x))$.

Valiant and Vazirani [11] introduced the following type of *randomized polynomial-time many-one reducibility*: $A \leq_{ran}^p B$ if there exists a polynomial-time randomized algorithm F and a polynomial p such that for each $x \in \Sigma^*$, if $x \in A$ then $F(x) \in B$ with probability at least $1/p(|x|)$, and if $x \notin A$ then $F(x) \notin B$ with certainty. In particular, they proved that the unique version of the satisfiability problem, Unique-SAT, is DP-complete under randomized reductions.

Tile Sets, Color Sequences, and Orientations: The Tantrix$^{\text{TM}}$ rotation puzzle consists of four different kinds of hexagonal tiles, named *Sint*, *Brid*, *Chin*, and *Rond*. Each tile has three lines colored differently, where the three colors of a tile are chosen among four possible colors, see Figures 1(a)–(d). The original Tantrix$^{\text{TM}}$ colors are *red*, *yellow*, *blue*, and *green*, which we encode here as shown in Figures 1(e)–(h). The combination of four kinds of tiles having three out of four colors each gives a total of 56 different tiles.

Let C be the set that contains the four colors *red*, *yellow*, *blue*, and *green*. For each $i \in \{1,2,3,4\}$, let $C_i \subseteq C$ be some fixed subset of size i, and let T_i denote the set of Tantrix$^{\text{TM}}$ tiles available when the line colors for each tile are restricted to C_i. For example, T_4 is the original Tantrix$^{\text{TM}}$ tile set containing 56 tiles, and if C_3 contains, say, the three colors *red*, *yellow*, and *blue*, then tile set T_3 contains the 14 tiles shown in Figure 2(b).



(a) Tantrix$^{\text{TM}}$ tile set T_2 (b) Tantrix$^{\text{TM}}$ tile set T_3

Fig. 2. Tantrix$^{\text{TM}}$ tile sets T_2 (for *red* and *blue*) and T_3 (for *red*, *yellow*, and *blue*)

For T_3 and T_4, we require the three lines on each tile to have distinct colors, as in the original Tantrix$^{\text{TM}}$ tile set. For T_1 and T_2, however, this is not possible, so we allow the same color being used for more than one of the three lines of any tile. Note that we care only about the sequence of colors on a tile, where we always use the clockwise direction to represent color sequences. However, since different types of tiles can yield the same color sequence, we will use just one such tile to represent the corresponding color sequence. For example, if C_2 contains, say, the two colors *red* and *blue*, then the color sequence *red-red-blue-blue-blue-blue* (which we abbreviate as `rrbbbb`) can be represented by a *Sint*, a *Brid*, or a *Rond* each having one short *red* arc and two *blue* additional lines, and we add only one such tile (say, the *Rond*) to the tile set T_2. That is, though there is some freedom in choosing a particular set of tiles, to be specific we fix the tile set T_2 shown in Figure 2(a). Thus, we have $\|T_1\| = 1$, $\|T_2\| = 8$, $\|T_3\| = 14$, and $\|T_4\| = 56$, regardless of which colors are chosen to be in C_i, $1 \leq i \leq 4$.

The six possible orientations for each tile in T_2 and in T_3, respectively, can be described by permuting the color sequences cyclically, and we omit the repetitions of color sequences (see the full version [2] for more details). For example, tile t_7 from T_2 has the same color sequence (namely, `bbbbbb`) in each of its six orientations. In Section 3, we will consider the counting versions of Tantrix$^{\text{TM}}$ rotation puzzle problems and will construct parsimonious reductions. When counting the solutions of Tantrix$^{\text{TM}}$ rotation puzzles, we will focus on color sequences only. That is, whenever some tile (such as t_7 from T_2) has distinct orientations with identical color sequences, we will count this as just one solution (and disregard such repetitions). In this sense, our reduction in the proof of Theorem 2 (which is presented in the full version [2]) will be parsimonious.

Definition of the Problems: We now recall some useful notation that Holzer and Holzer [7] introduced in order to formalize problems related to the Tantrix$^{\text{TM}}$ rotation puzzle. The instances of such problems are Tantrix$^{\text{TM}}$ tiles firmly arranged in the plane. To represent their positions, we use a two-dimensional hexagonal coordinate system, see [7] and also [2]. Let $T \in \{T_1, T_2, T_3, T_4\}$ be some tile set as defined above. Let $\mathscr{A} : \mathbb{Z}^2 \to T$ be a function mapping points in $\mathbb{Z}^2$ to tiles in T, i.e., $\mathscr{A}(x)$ is the type of the tile located at position x. Note that $\mathscr{A}$ is a partial function; throughout this paper (except in Theorem 3 and its proof), we restrict our problem instances to finitely many given tiles, and the regions of $\mathbb{Z}^2$ they cover may have holes (which is a difference to the original Tantrix$^{\text{TM}}$ game).

Define $shape(\mathscr{A})$ to be the set of points $x \in \mathbb{Z}^2$ for which $\mathscr{A}(x)$ is defined. For any two distinct points $x = (a,b)$ and $y = (c,d)$ in $\mathbb{Z}^2$, x and y are neighbors if and only if ($a = c$ and $|b - d| = 1$) or ($|a - c| = 1$ and $b = d$) or ($a - c = 1$ and $b - d = 1$) or ($a - c = -1$ and $b - d = -1$). For any two points x and y in $shape(\mathscr{A})$, $\mathscr{A}(x)$ and $\mathscr{A}(y)$ are said to be neighbors exactly if x and y are neighbors. For k chosen from $\{1,2,3,4\}$, define the following problem:

Name: k-Color Tantrix$^{\text{TM}}$ Rotation Puzzle (k-TRP, for short).
Instance: A finite shape function $\mathscr{A} : \mathbb{Z}^2 \to T_k$, encoded as a string in Σ^*.
Question: Is there a solution to the rotation puzzle defined by $\mathscr{A}$, i.e., does there exist a rotation of the given tiles in $shape(\mathscr{A})$ such that the colors of the lines of any two adjacent tiles match at their joint edge?

Clearly, 1-TRP can be solved trivially, so 1-TRP is in P. On the other hand, Holzer and Holzer [7] showed that 4-TRP is NP-complete and that the infinite variant of 4-TRP is undecidable. Baumeister and Rothe [1] investigated the counting and the unique variant of 4-TRP and, in particular, provided a parsimonious reduction from SAT to 4-TRP. In this paper, we study the three-color and two-color versions of this problem, 3-TRP and 2-TRP, and their counting, unique, and infinite variants.

Definition 1. *A* solution to a k-TRP instance $\mathscr{A}$ *specifies an orientation of each tile in shape*($\mathscr{A}$) *such that the colors of the lines of any two adjacent tiles match at their joint edge. Let* $\text{SOL}_{k\text{-TRP}}(\mathscr{A})$ *denote the set of solutions of* $\mathscr{A}$. *Define the* counting version *of* k-TRP *to be the function* $\#k$-TRP *mapping from* Σ^* *to* $\mathbb{N}$ *such that* $\#k\text{-TRP}(\mathscr{A}) = \|\text{SOL}_{k\text{-TRP}}(\mathscr{A})\|$. *Define the* unique version *of* k-TRP *as* $\text{Unique-}k\text{-TRP} = \{\mathscr{A} \mid \#k\text{-TRP}(\mathscr{A}) = 1\}$.

The above problems are defined for the case of finite problem instances. The infinite Tantrix$^{\text{TM}}$ rotation puzzle problem with k colors (Inf-k-TRP, for short) is defined exactly as k-TRP, the only difference being that the shape function $\mathscr{A}$ is not required to be finite and is represented by the encoding of a Turing machine computing $\mathscr{A} : \mathbb{Z}^2 \to T_k$.

3 Results

3.1 Parsimonious Reduction from SAT to 3-TRP

Theorem 1 below is the main result of this section. Notwithstanding that our proof follows the general approach of Holzer and Holzer [7], our specific construction and our proof of correctness will differ substantially from theirs. We will give a parsimonious reduction from SAT to 3-TRP. Let $\text{Circuit}_{\wedge,\neg}$-SAT denote the problem of deciding, given a boolean circuit c with AND and NOT gates only, whether or not there is a satisfying truth assignment to the input variables of c. The NP-completeness of $\text{Circuit}_{\wedge,\neg}$-SAT was shown by Cook [3], and it is easy to see that SAT parsimoniously reduces to $\text{Circuit}_{\wedge,\neg}$-SAT (see, e.g., [1]).

Theorem 1. SAT *parsimoniously reduces to* 3-TRP.

It is enough to show that $\text{Circuit}_{\wedge,\neg}$-SAT parsimoniously reduces to 3-TRP. The resulting 3-TRP instance simulates a boolean circuit with AND and NOT gates such that the number of solutions of the rotation puzzle equals the number of satisfying truth assignments to the variables of the circuit.

General remarks on our proof approach: The rotation puzzle to be constructed from a given circuit consists of different subpuzzles each using only three colors. The color *green* was employed by Holzer and Holzer [7] only to exclude certain rotations, so we choose to eliminate this color in our three-color rotation puzzle. Thus, letting C_3 contain the colors *blue*, *red*, and *yellow*, we have the tile set $T_3 = \{t_1, t_2, \dots, t_{14}\}$, where the enumeration of tiles corresponds to Figure 2(b). Furthermore, our construction will be parsimonious, i.e., there will be a one-to-one correspondence between the solutions of the given $\text{Circuit}_{\wedge,\neg}$-SAT instance and the solutions of the resulting rotation puzzle instance. Note that part of our work is already done, since some subpuzzles constructed in [1] use only three colors and they each have unique solutions. However, the remaining subpuzzles have to be either modified substantially or to be constructed completely differently, and the arguments of why our modified construction is correct differs considerably from previous work [7,1].

Since it is not so easy to exclude undesired rotations without having the color *green* available, it is useful to first analyze the 14 tiles in T_3. In the remainder of this proof, when showing that our construction is correct, our arguments will often be based on which substrings do or do not occur in the color sequences of certain tiles from T_3. (Note that the full version of this paper [2] has a table that shows which substrings of the form uv, where $u, v \in C_3$, occur in the color sequence of t_i in T_3, and this table may be looked up for convenience.)

Holzer and Holzer [7] consider a boolean circuit c on input variables $x_1, x_2, \ldots, x_n$ as a sequence $(\alpha_1, \alpha_2, \ldots, \alpha_m)$ of computation steps (or "instructions"), and we adopt this approach here. For the ith instruction, α_i, we have $\alpha_i = x_i$ if $1 \leq i \leq n$, and if $n+1 \leq i \leq m$ then we have either $\alpha_i = \mathrm{NOT}(j)$ or $\alpha_i = \mathrm{AND}(j,k)$, where $j \leq k < i$. Circuits are evaluated in the standard way. We will represent the truth value *true* by the color *blue* and the truth value *false* by the color *red* in our rotation puzzle. A technical difficulty in the construction results from the wire crossings that circuits can have. To construct rotation puzzles from *planar* circuits, Holzer and Holzer use McColl's planar "cross-over" circuit with AND and NOT gates to simulate such wire crossings [8], and in particular they employ Goldschlager's log-space transformation from general to planar circuits [5]. For the details of this transformation, we refer to [7].

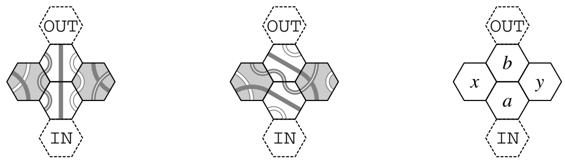
Holzer and Holzer's original subpuzzles [7] should be compared with those in our construction. To illustrate the differences between our new and these original subpuzzles, modified or inserted tiles in our new subpuzzles presented in this section will always be highlighted by having a grey background.

Wire subpuzzles: Wires of the circuit are simulated by the subpuzzles WIRE, MOVE, and COPY. We present only the WIRE here; see [2] for MOVE and COPY.

A vertical wire is represented by a WIRE subpuzzle, which is shown in Figure 3. The original WIRE subpuzzle from [7] does not contain *green* but it does not have a unique solution, while the WIRE subpuzzle from [1] ensures the uniqueness of the solution but is using a tile with a *green* line. In the original WIRE subpuzzle, both tiles, a and b, have two possible orientations for each input color. Inserting two new tiles at positions x and y (see Figure 3) makes the solution unique. If the input color is *blue*, tile x must contain one of the following color-sequence substrings for the edges joint with tiles b and a: `ry`, `rr`, `yy`, or `yr`. If the input color is *red*, x must contain one of these substrings: `bb`, `yb`, `yy`, or `by`. Tile t_{12} satisfies the conditions `yy` and `ry` for the input color *blue*, and the conditions `yb` and `yy` for the input color *red*.

The solution must now be fixed with tile y. The possible color-sequence substrings of y at the edges joint with a and b are `rr` and `ry` for the input color *blue*, and `yb` and `bb` for the input color *red*. Tile t_{13} has exactly one of these sequences for each input color. Thus, the solution for this subpuzzle contains only three colors and is unique.

Gate subpuzzles: The boolean gates AND and NOT are represented by the AND and NOT subpuzzles. Both the original four-color NOT subpuzzle from [7] and the modified four-color NOT subpuzzle from [1] use tiles with *green* lines to exclude certain



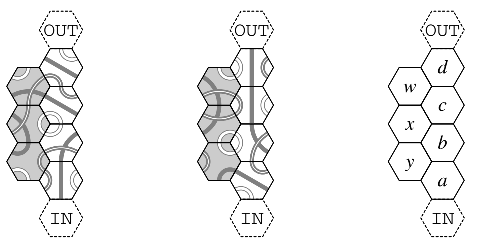
(a) In: *true* (b) In: *false* (c) Scheme

Fig. 3. Three-color WIRE subpuzzle

rotations. Our three-color NOT subpuzzle is shown in Figure 4. Tiles a, b, c, and d from the original NOT subpuzzle [7] remain unchanged. Tiles e, f, and g in this original NOT subpuzzle ensure that the output color will be correct, since the joint edge of e and b is always *red*. So for our new NOT subpuzzle in Figure 4, we have to show that the edge between tiles x and b is always *red*, and that we have unique solutions for both input colors.

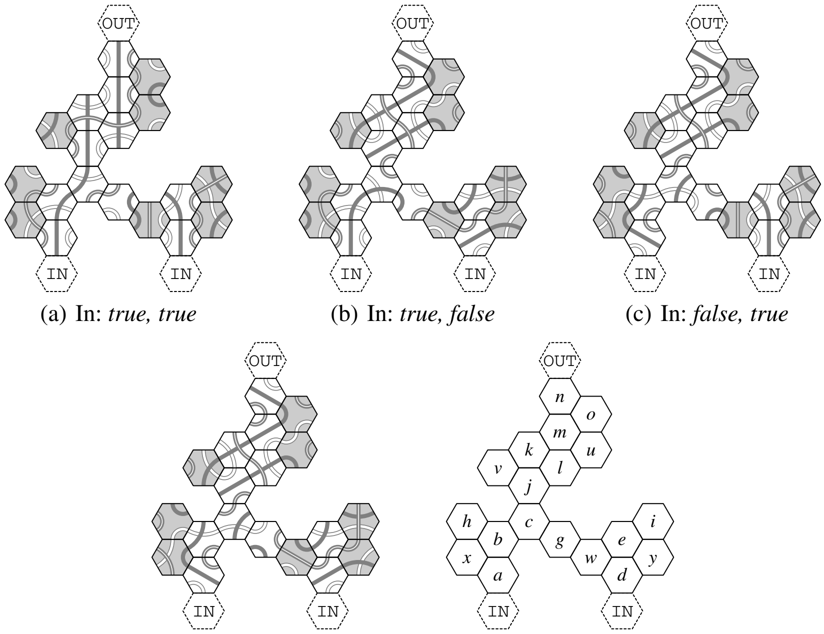
First, let the input color be *blue* and suppose for a contradiction that the joint edge of tiles b and x were *blue*. Then the joint edge of tiles b and c would be *yellow*. Since x is a tile of type t_{13} and so does not contain the color-sequence substring `bb`, the edge between tiles c and x must be *yellow*. But then the edges of tile w joint with tiles c and x must both be *blue*. This is not possible, however, because w (which is of type t_{10}) does not contain the color-sequence substring `bb`. So if the input color is *blue*, the orientation of tile b is fixed with *yellow* at the edge of b joint with tile y, and with *red* at the edges of b joint with tiles c and x. This already ensures that the output color will be *red*, because tiles c and d behave like a WIRE subpuzzle. Tile x does not contain the color-sequence substring `br`, so the orientation of tile c is also fixed with *blue* at the joint edge of tiles c and w. As a consequence, the joint edge of tiles w and d is *yellow*, and due to the fact that the joint edge of tiles w and x is also *yellow*, the orientation of w and d is fixed as well. Regarding tile a, the edge joint with tile y can be *yellow* or *red*, but tile x has *blue* at the edge joint with tile y, so the joint edge of tiles y and a is *yellow*, and the orientation of all tiles is fixed for the input color *blue*. The case of *red* being the input color can be handled analogously.

The most complicated figure is the AND subpuzzle. The original four-color version from [7] uses four tiles with *green* lines and the modified four-color AND subpuzzle



(a) In: *true* (b) In: *false* (c) Scheme

Fig. 4. Three-color NOT subpuzzle



(a) In: *true, true* (b) In: *true, false* (c) In: *false, true*

(d) In: *false, false* (e) Scheme

Fig. 5. Three-color AND subpuzzle

from [1] uses seven tiles with *green* lines. Figure 5 shows our new AND subpuzzle using only three colors and having unique solutions for all four possible combinations of input colors. To analyze this subpuzzle, we subdivide it into a lower and an upper part. The lower part ends with tile c and has four possible solutions (one for each combination of input colors), while the upper part, which begins with tile j, has only two possible solutions (one for each possible output color). The lower part can again be subdivided into three different parts.

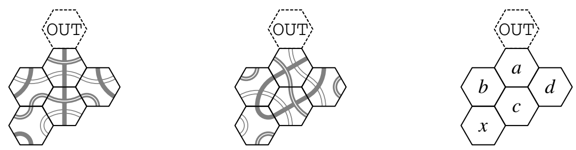
The lower left part contains the tiles a, b, x, and h. If the input color to this part is *blue* (see Figures 5(a) and 5(b)), the joint edge of tiles b and x is always *red*, and since tile x (which is of type t_{11}) does not contain the color-sequence substring `rr`, the orientation of tiles a and x is fixed. The orientation of tiles b and h is also fixed, since h (which is of type t_2) does not contain the color-sequence substring `by` but the color-sequence substring `yy` for the edges joint with tiles b and x. By similar arguments we obtain a unique solution for these tiles if the left input color is *red* (see Figures 5(c) and 5(d)). The connecting edge to the rest of the subpuzzle is the joint edge between tiles b and c, and tile b will have the same color at this edge as the left input color.

Tiles d, e, i, w, and y form the lower right part. If the input color to this part is *blue* (see Figures 5(a) and 5(c)), the joint edge of tiles d and y must be *yellow*, since tile y (which is of type t_9) does not contain the color-sequence substrings `rr` nor `ry` for the edges joint with tiles d and e. Thus the joint edge of tiles y and e must be *yellow*, since i (which is of type t_6) does not contain the color-sequence substring `bb` for the edges joint with tiles y and e. This implies that the tiles i and w also have a fixed orientation. If the input color to the lower right part is *red* (see Figures 5(b) and 5(d)), a unique solution is obtained by similar arguments. The connection of the lower right part to the rest of the subpuzzle is the edge between tiles w and g. If the right input color is *blue*, this edge will also be *blue*, and if the right input color is *red*, this edge will be *yellow*.

The heart of the AND subpuzzle is its lower middle part, formed by the tiles c and g. The colors at the joint edge between tiles b and c and at the joint edge between tiles w and g determine the orientation of the tiles c and g uniquely for all four possible combinations of input colors. The output of this part is the color at the edge between c and j. If both input colors are *blue*, this edge will also be *blue*, and otherwise this edge will always be *yellow*.

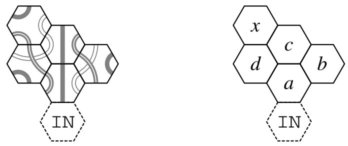
The output of the whole AND subpuzzle will be *red* if the edge between c and j is *yellow*, and if this edge is *blue* then the output of the whole subpuzzle will also be *blue*. If the input color for the upper part is *blue* (see Figure 5(a)), each of the tiles j, k, l, m, and n has a vertical *blue* line. Note that since the colors *red* and *yellow* are symmetrical in these tiles, we would have several possible solutions without tiles o, u, and v. However, tile v (which is of type t_9) contains neither `rr` nor `ry` for the edges joint with tiles k and j, so the orientation of the tiles j through n is fixed, except that tile n without tiles o and u would still have two possible orientations. Tile u (which is of type t_2) is fixed because of its color-sequence substring `yy` at the edges joint with l and m, so due to tiles o and u the only color possible at the edge between n and o is *yellow*, and we have a unique solution. If the input color for the upper part is *yellow* (see Figures 5(b)–(d)), we obtain unique solutions by similar arguments. Hence, this new AND subpuzzle uses only three colors and has unique solutions for each of the four possible combinations of input colors.

Input and output subpuzzles: The input variables of the boolean circuit are represented by the subpuzzle BOOL. Our new three-color BOOL subpuzzle is presented in Figure 6, and since it is completely different from the original four-color BOOL subpuzzle from [7], no tiles are marked here. The subpuzzle in Figure 6 has only two possible solutions, one with the output color *blue* (if the corresponding variable is *true*), and one with the output color *red* (if the corresponding variable is *false*). The original four-color BOOL subpuzzle from [7] contains tiles with *green* lines to exclude certain rotations. Our three-color BOOL subpuzzle does not contain any *green* lines, but it might not be that obvious that there are only two possible solutions, one for each output color. The proof can be found in the full version [2].



(a) Out: *true* (b) Out: *false* (c) Scheme

Fig. 6. Three-color BOOL subpuzzle



(a) TEST-true (b) Scheme

Fig. 7. Three-color TEST subpuzzle

Finally, a subpuzzle is needed to check whether or not the circuit evaluates to *true*. This is achieved by the subpuzzle TEST-true shown in Figure 7(a). It has only one valid solution, namely that its input color is *blue*. Just like the subpuzzle BOOL, the original four-color TEST-true subpuzzle from [7], which was not modified in [1], uses *green* lines to exclude certain rotations. Again, since the new TEST-true subpuzzle is completely different from the original subpuzzle, no tiles are marked here. Our argument of why this subpuzzle is correct can be found in the full version [2].

The shapes of the subpuzzles constructed above have changed slightly. However, by Holzer and Holzer's argument [7] about the minimal horizontal distance between two wires and/or gates being at least four, unintended interactions between the subpuzzles do not occur. This concludes the proof of Theorem 1. ❑

Corollary 1. 3-TRP *is* NP-*complete.*

Since the tile set T_3 is a subset of the tileset T_4, we have 3-TRP $\leq_m^p$ 4-TRP. Thus, the hardness results for 3-TRP and its variants proven in this paper immediately are inherited by 4-TRP and its variants, which provides an alternative proof of these hardness results for 4-TRP and its variants established in [7,1]. In particular, Corollary 2 follows from Theorem 1 and Corollary 1.

Corollary 2 ([7,1]). 4-TRP *is* NP-*complete, via a parsimonious reduction from* SAT.

3.2 Parsimonious Reduction from SAT to 2-TRP

In contrast to the above-mentioned fact that 3-TRP $\leq_m^p$ 4-TRP holds trivially, the reduction 2-TRP $\leq_m^p$ 3-TRP (which we will show to hold due to both problems being NP-complete, see Corollaries 1 and 3) is not immediatedly straightforward, since the tile set T_2 is not a subset of the tile set T_3 (recall Figure 2 in Section 2). In this section, we study 2-TRP and its variants. Our main result here is Theorem 2 below the proof of which can be found in the full version [2].

Theorem 2. SAT *parsimoniously reduces to* 2-TRP.

Corollary 3. 2-TRP *is* NP-*complete.*

3.3 Unique and Infinite Variants of 3-TRP and 2-TRP

Parsimonious reductions preserve the number of solutions and, in particular, the uniqueness of solutions. Thus, Theorems 1 and 2 imply Corollary 4 below that also employs

Valiant and Vazirani's results on the DP-hardness of Unique-SAT under $\leq_{ran}^{p}$-reductions (which were defined in Section 2). The proof of Corollary 4 follows the lines of the proof of [1, Theorem 6], which states the analogous result for Unique-4-TRP in place of Unique-3-TRP and Unique-2-TRP.

Corollary 4

1. Unique-SAT *parsimoniously reduces to* Unique-3-TRP *and* Unique-2-TRP.
2. Unique-3-TRP *and* Unique-2-TRP *are* DP-*complete under* $\leq_{ran}^{p}$*-reductions.*

Holzer and Holzer [7] proved that Inf-4-TRP, the infinite Tantrix$^{\mathrm{TM}}$ rotation puzzle problem with four colors, is undecidable, via a reduction from (the complement of) the empty-word problem for Turing machines. The proof of Theorem 3 below, which can be found in the full version [2], uses essentially the same argument but is based on our modified three-color and two-color constructions.

Theorem 3. *Both* Inf-2-TRP *and* Inf-3-TRP *are undecidable.*

4 Conclusions

This paper studied the three-color and two-color Tantrix$^{\mathrm{TM}}$ rotation puzzle problems, 3-TRP and 2-TRP, and their unique and infinite variants. Our main contribution is that both 3-TRP and 2-TRP are NP-complete via a parsimonious reduction from SAT, which in particular solves a question raised by Holzer and Holzer [7]. Since restricting the number of colors to three and two, respectively, drastically reduces the number of Tantrix$^{\mathrm{TM}}$ tiles available, our constructions as well as our correctness arguments substantially differ from those in [7,1]. Table 1 in Section 1 shows that our results give a complete picture of the complexity of k-TRP, $1 \leq k \leq 4$. An interesting question still remaining open is whether the analogs of k-TRP *without holes* still are NP-complete.

Acknowledgments. We are grateful to Markus Holzer and Piotr Faliszewski for inspiring discussions on Tantrix$^{\mathrm{TM}}$ rotation puzzles, and we thank Thomas Baumeister for his help with writing a program for checking the correctness of our constructions and for producing reasonably small figures. We thank the anonymous LATA 2008 referees for helpful comments, and in particular the referee who let us know that he or she has also written a program for verifying the correctness of our constructions.

References

1. Baumeister, D., Rothe, J.: Satisfiability parsimoniously reduces to the Tantrix$^{\mathrm{TM}}$ rotation puzzle problem. In: Durand-Lose, J., Margenstern, M. (eds.) MCU 2007. LNCS, vol. 4664, pp. 134–145. Springer, Heidelberg (2007)
2. Baumeister, D., Rothe, J.: The three-color and two-color Tantrix$^{\mathrm{TM}}$ rotation puzzle problems are NP-complete via parsimonious reductions. Technical Report cs.CC/0711.1827, ACM Computing Research Repository (CoRR) (November 2007)
3. Cook, S.: The complexity of theorem-proving procedures. In: Proceedings of the 3rd ACM Symposium on Theory of Computing, pp. 151–158. ACM Press, New York (1971)

4. Downing, K.: Tantrix: A minute to learn, 100 (genetic algorithm) generations to master. Genetic Programming and Evolvable Machines 6(4), 381–406 (2005)
5. Goldschlager, L.: The monotone and planar circuit value problems are log space complete for P. SIGACT News 9(2), 25–29 (1977)
6. Grädel, E.: Domino games and complexity. SIAM Journal on Computing 19(5), 787–804 (1990)
7. Holzer, M., Holzer, W.: Tantrix^{TM} rotation puzzles are intractable. Discrete Applied Mathematics 144(3), 345–358 (2004)
8. McColl, W.: Planar crossovers. IEEE Transactions on Computers C-30(3), 223–225 (1981)
9. Papadimitriou, C., Yannakakis, M.: The complexity of facets (and some facets of complexity). Journal of Computer and System Sciences 28(2), 244–259 (1984)
10. Valiant, L.: The complexity of computing the permanent. Theoretical Computer Science 8(2), 189–201 (1979)
11. Valiant, L., Vazirani, V.: NP is as easy as detecting unique solutions. Theoretical Computer Science 47, 85–93 (1986)