Balanced k-Colorings

Therese C. Biedl[c] Eowyn Čenek[c] Timothy M. Chan[c]
Erik D. Demaine[b,1] Martin L. Demaine[c] Rudolf Fleischer[a,*,2]
Ming-Wei Wang[c]

[a]*Hong Kong University of Science and Technology, Department of Computer Science, Clear Water Bay, Kowloon, Hong Kong*

[b]*MIT Laboratory of Computer Science, 200 Technology Square, Cambridge, Massachusetts 02139, USA*

[c]*University of Waterloo, Department of Computer Science, Waterloo, ON N2L 3G1, Canada*

Abstract

While discrepancy theory is normally only studied in the context of 2-colorings, we explore the problem of k-coloring, for $k \geq 2$, a set of vertices to minimize imbalance among a family of subsets of vertices. The *imbalance* is the maximum, over all subsets in the family, of the largest difference between the size of any two color classes in that subset. The *discrepancy* is the minimum possible imbalance. We show that the discrepancy is always at most $4d - 3$, where d (the "dimension") is the maximum number of subsets containing a common vertex. For 2-colorings, the bound on the discrepancy is at most $\max\{2d - 3, 2\}$. Finally, we prove that several restricted versions of computing the discrepancy are NP-complete.

Key words: Discrepancy, Balance Theorem, NP completeness

* Corresponding author
Email addresses: `biedl@uwaterloo.ca` (Therese C. Biedl), `ewcenek@uwaterloo.ca` (Eowyn Čenek), `tmchan@uwaterloo.ca` (Timothy M. Chan), `edemaine@mit.edu` (Erik D. Demaine), `mldemaine@uwaterloo.ca` (Martin L. Demaine), `rudolf@cs.ust.hk` (Rudolf Fleischer), `m2wang@uwaterloo.ca` (Ming-Wei Wang).

[1] The research was mainly done while the author was a PhD student at the University of Waterloo, Department of Computer Science

[2] The research was mainly done while the author was a visiting associate professor at the University of Waterloo, Department of Computer Science

1 Introduction

We begin with some basic notation and terminology. Let $\mathcal{L}$ be a family of nonempty subsets of a finite set P. We call the elements of P *vertices* and the elements of $\mathcal{L}$ *lines*. A vertex $v \in P$ *lies on a line* $\ell \in \mathcal{L}$ if $v \in \ell$. We denote the number of vertices on a line ℓ by $|\ell|$.

One topic in the area of combinatorial discrepancy theory [1,4,6,17] is the study of the minimum possible "imbalance" in a 2-coloring of the vertices. Formally, a 2-*coloring* is a function χ from the vertices in P to the two colors $-1, +1$ (note that we do not speak about coloring the nodes of a graph such that endpoints have different colors). The *imbalance* of χ is the maximum difference between the size of the two color classes considered for every line, i.e., $\max_{\ell \in \mathcal{L}} |\sum_{v \in \ell} \chi(v)|$. The *discrepancy* is the minimum possible imbalance over all 2-colorings; to avoid confusion, we call this standard notion the 2-*color discrepancy*.

In this paper we consider the following more general setting. A k-*coloring* of P is a mapping from the vertices in P to the k colors $1, \dots, k$. It is called c-*balanced* if for any line ℓ and any two colors i, j, $1 \le i, j \le k$, we have

$$|\#\{\text{vertices on } \ell \text{ colored } i\} - \#\{\text{vertices on } \ell \text{ colored } j\}| \le c.$$

We call c the *imbalance* of the coloring; it is a strong measure of additive error relative to the uniform distribution. The k-*color discrepancy* of a family $\mathcal{L}$ is the minimum possible imbalance over all k-colorings.

Ideally, we would hope for *perfectly balanced* colorings, i.e., with imbalance 0, but a necessary condition for the existence of a perfectly balanced coloring is that the number of vertices on each line is a multiple of k. Otherwise, the best we can hope for is an *almost-balanced* coloring which is a coloring with imbalance 1. Indeed, de Werra showed that an almost-balanced coloring (which he calls *equitable*) always exists if the incidence matrix of the lines is unimodular [8]. In general, however, this is not always possible.

Our work is strongly motivated by two previous results. In the context of 2-colorings, Beck and Fiala [5] gave an upper bound of $2d-1$ on the discrepancy where the *dimension* d (also often called the *maximum degree*) of a family $\mathcal{L}$ is the maximum number of lines passing through a common vertex.

On the other hand, Akiyama and Urrutia [2] studied k-colorings, for arbitrary k, for lines that form a two-dimensional grid and vertices that are a subset of the intersections. In this geometric setting (with $d = 2$), they showed that there is always an almost-balanced k-coloring. The same result for 2-colorings can also be derived using the algorithm by Šíma for table rounding [16] (where each grid point corresponds to a table entry of $\frac{1}{2}$). A more general table rounding problem with applications in digital image processing has recently been studied by Asano et al. [3]. Since finding a k-coloring of points on lines that form a two-dimensional grid can be reformulated as finding an edge k-coloring of a simple bipartite graph, the existence of almost-balanced k-colorings of 2-dimensional grid points was actually already proven much earlier by de Werra [8,9]. He also showed that any simple multigraph (not necessarily bipartite) has a 2-balanced edge k-coloring (see also [14]). Akiyama and Urrutia also showed that not all configurations of points in higher-dimensional grids have an almost-balanced k-coloring, but asked whether an $O(1)$-balanced k-coloring might be possible for such grids.

In Section 2, we generalize the results of Beck and Fiala [5] and Akiyama and Urrutia [2] to k-colorings of an arbitrary family of lines. In particular, we settle the open questions posed by Akiyama and Urrutia within a constant factor. Specifically, our most general result states that the k-color discrepancy is at most $4d - 3$. Note that this bound is independent of the number of colors.

We can tighten the bound by 1 in the case that the number of vertices on each line is a multiple of k. For 2-colorings we can tighten the bound further to $\max\{2d - 3, 2\}$, improving by an additive constant the results of Beck and Fiala [5].[3] For $d = 2$, the bound of 2 is tight because the three vertices of a triangle have no almost-balanced 2-coloring. And in the special case of 2-dimensional geometric settings our proof can be strengthened to give the Akiyama and Urrutia result [2], i.e., we can prove there is always an almost-balanced 2-coloring.

In Section 4 we show that a simpler divide-and-conquer algorithm (which has apparently been known in the field of discrepancy theory [18]) but has never been published until recently [13]) computes k-colorings which are slightly less balanced than the colorings computed by the algorithm in Section 2. Both algorithms can be implemented efficiently in polynomial time. Recently, Doerr and Srivastav [13] used a randomized recursive approach to obtain colorings with low discrepancies.

Finally, in Section 5 we show that for $k \geq 2$ finding almost-balanced k-colorings is NP-complete for line families of dimension at least $\max\{3, k-1\}$. For $k = 2, 3$, this result even holds for the special case of various geometric settings. We suspect that finding an almost-balanced k-coloring is NP-complete for any $k \geq 2$ and $d \geq 3$.

An independent treatment of multicolor discrepancy theory was recently made

[3] We recently learned that this result was independently obtained by Bednarchak and Helm [7].

by Doerr and Srivastav [12]. This paper and our paper prove several different results; for example, Doerr and Srivastav generalize other theorems in 2-color discrepancy theory, and our NP-completeness results are new. The main overlap between the two papers is in our most general result, bounding the discrepancy by $4d-3$ (Theorem 1), which improves Theorem 7 in [12] by an additive constant. And very recently, Doerr [11] discovered graphs which have very different discrepancies in different numbers of colors.

A good survey on discrepancy theory as well as various other new results can be found in Doerr's PhD thesis [10].

2 The Balance Theorem

In this section we prove our main theorem which states that any set of vertices on a set of lines of dimension d can be k-colored such that the imbalance on each line is bounded by a constant only depending on the dimension d (and not on k, or on the number of vertices, or on the number of lines).

Theorem 1 (Balance Theorem) *Let $d \geq 2$ and $k \geq 2$. Let $\mathcal{L}$ be a set of lines of dimension d containing a set of vertices P. Then P has a $(4d-3)$-balanced k-coloring. The imbalance is at most $4d-4$ for all lines with a multiple of k many vertices.*

PROOF. This proof is an adaptation of the proof given by Beck and Fiala [5] (see also [1,17]) for 2-colorings.

With each vertex $v \in P$ we associate k variables $x_{v,1}, \ldots, x_{v,k}$ which will change in time. At all times all $x_{v,i}$ lie in the closed interval $[0,1]$. Initially, all $x_{v,i}$ are set to $\frac{1}{k}$. If $0 < x_{v,i} < 1$ then $x_{v,i}$ is called *floating*, so initially all variables are floating. If $x_{v,i} = 0$ or $x_{v,i} = 1$ then $x_{v,i}$ is called *fixed*. Once a variable is fixed, it can never change again. Eventually, all variables will be fixed.

At that time, the variables define a k-coloring of the vertices: vertex v is colored with color i if and only if $x_{v,i} = 1$ and $x_{v,j} = 0$ for $j \neq i$. To ensure that there is exactly one i with $x_{v,i} = 1$, we require that at all times

$$\sum_{i=1}^{k} x_{v,i} = 1 \qquad (C_v)$$

for all vertices v. We call these equations *color equations*.

For each line ℓ, we want to balance the colors. This can be expressed by k *balance equations* $E_{\ell,i}$, for $i = 1, \ldots, k$:

$$\sum_{v \in \ell} x_{v,i} = \frac{|\ell|}{k} \qquad (E_{\ell,i})$$

The color and balance equations form a system (LP) of linear equations in the variables $x_{v,i}$. Ideally, we would like to find an integer solution of (LP) which would then correspond to a perfectly balanced coloring. But such an integer solution does not always exist (if a line ℓ contains a number of vertices which is not a multiple of k then there can be no integer solution to the corresponding balance equations). So we allow some imbalance in the coloring by ignoring equations with only a 'few' floating variables. Setting these variables later to arbitrary values of 0 or 1 can only create a small imbalance.

We say a balance equation is *active* if it contains at least $2d$ floating variables; otherwise, it is *inactive*. And a color equation is *active* if it contains at least two floating variables; otherwise, it is *inactive*. Note that a color equation always contains either zero or at least two floating variables. We call the subsystem of active equations (LP_{act}), and we only require at all times that the active equations are satisfied. Since initially all variables are floating, (LP_{act}) initially consists of all balance equations with at least $2d$ variables.

Now suppose we have a solution of (LP_{act}) at a certain time. Let f be the number of floating variables at this time. Each vertex is contained in at most d lines. Therefore, each variable is contained in at most d balance equations. In particular, each floating variable is contained in at most d active balance equations. On the other hand, each active balance equation contains at least $2d$ floating variables, so the number of active balance equations is at most $\frac{f}{2}$. Moreover, each variable appears in exactly one color equation, whereas each active color equation contains at least two floating variables. Therefore, the number of active color equations is at most $\frac{f}{2}$. If (LP_{act}) is underdetermined, with more variables than equations, then we can move along a line of solutions of (LP_{act}) in Euclidean space. We do so until one of the floating variables reaches 0 or 1; then we stop. This fixes at least one previously floating variable. We continue this procedure until all of the $x_{v,i}$ are fixed, or until (LP_{act}) is not underdetermined. The latter can only happen if there are exactly $\frac{f}{2}$ active balance equations and $\frac{f}{2}$ active color equations. But then, none of these balance equations can have $2d + 1$ or more floating variables, and none of these color equations can have 3 or more floating variables (otherwise there would be strictly more than f floating variables, a contradiction). Thus, each active balance equation contains exactly $2d$ floating variables and each color equation contains exactly 2 floating variables. If we now round the two floating variables in each color equation to 0 and 1, respectively, we change the value of each balance equation by at most d. We call this the *rounding step*. This yields an imbalance of at most $2d$ (which is less than $4d-3$ for $d \geq 2$) for the lines corresponding to the active balance equations.

Since the final values of the $x_{v,i}$ still satisfy all the color equations (but not necessarily all the balance equations), we can read off a k-coloring of the vertices. We claim that this k-coloring is $(4d-3)$-balanced. Consider any balance equation $E_{\ell,i}$. At the first time it becomes inactive we have $\sum_{v\in\ell} x_{v,i} = \frac{|\ell|}{k}$, and at most $2d-1$ of the $x_{v,i}$ are floating. Later, each of these floating variables can change by less than 1 from that value to its final value. As such, $|\sum_{v\in\ell} x_{v,i} - \frac{|\ell|}{k}| < 2d-1$ for the final values of the $x_{v,i}$. Hence the imbalance is bounded from above by $2\cdot(2d-2) = 4d-4$ if $|\ell|$ is a multiple of k, and $2\cdot(2d-1)-1 = 4d-3$ if $|\ell|$ is not a multiple of k. $\square$

3 2-Colorings

If we want to find balanced 2-colorings we can improve Theorem 1 by approximately a factor of 2.

Theorem 2 (2-Color Balance Theorem) *Let $d \geq 2$. Let $\mathcal{L}$ be a set of lines of dimension d containing a set of vertices P.*

(a) If $d = 2$ then P has a 2-balanced 2-coloring.
(b) If $d \geq 3$ then P has a $2d-3$-balanced 2-coloring. If $d \geq 4$ then the imbalance is at most $2d-4$ for all lines with an even number of vertices.

PROOF. We adjust the proof of Theorem 1 in the following way. Since we try to find a 2-coloring we have only two variables $x_{v,1}$ and $x_{v,2}$ assigned to each vertex v. These two variables are related by $x_{v,2} = 1 - x_{v,1}$ by the color equation (C_v). Since it is impossible that only one of the two variables is floating, fixing one will automatically fix the other one. Therefore it is sufficient to consider only one of the two variables, $x_{v,1}$ for example, by replacing $x_{v,2}$ by $1 - x_{v,1}$ in all equations. When we do this substitution for all variables $x_{v,2}$ for all vertices v then for any line ℓ the balance equations $E_{\ell,1}$ and $E_{\ell,2}$ are identical. So we can remove all equations $E_{\ell,2}$ and only keep the equations $E_{\ell,1}$.

Since there are also no color equations left after the substitution we can strengthen the definition of active balance equations. We say a balance equation is *active* if it contains at least d floating variables. Then we proceed as in the proof of Theorem 1. Note that now (LP_{act}) is not underdetermined if each active balance equation contains exactly d floating variables, yielding an imbalance of at most d in the rounding step. $\square$

Part (a) also follows from the results of Werra et al. [8,14]. In the case of 2-dimensional geometric settings as studied by Akiyama and Urrutia [2] our proof can be adapted to show the existence of an almost-balanced 2-coloring (the rounding step never happens in this case because it can be shown that (LP_{act}) is always underdetermined, even if the number of equations equals the number of variables). We note however that both mentioned previous works give the same discrepancy bound for k-colorings for $k \geq 2$, whereas we only obtain them for 2-colorings.

4 Alternative Approaches

It is tempting to try to use Theorem 2 to derive a greedy algorithm for k-colorings. We could start with a monochromatic coloring of all nodes, and then consecutively compute a good 2-coloring for all the nodes of a certain pair of two colors, until all the colors are sufficiently well distributed along the lines. The problem is that computing a good 2-coloring for one pair of colors can increase the imbalance of another color on some of the lines by $2d-3$ (or by 2, if $d=2$). It is not immediately clear which imbalance we can achieve by this greedy algorithm (and whether it would stop in finite time).

But we can also exploit Theorem 2 in another way. Instead of computing the colors of all vertices at the same time, we could first identify all vertices which should be colored with color 1, then discard these vertices and identify all vertices which should be colored with color 2, etc. In one step of this iteration, we only need to associate one variable x_v to each vertex v. Finally, when all variables are fixed, the vertices v with $x_v = 1$ belong to one color class; we then iterate on the set of all vertices w with $x_w = 0$. In this approach we do not need color equations, and we have just one *balance equation* (E_ℓ) for each line ℓ:

$$\sum_{v \in \ell} x_v = \alpha|\ell| \qquad (E_\ell)\ ,$$

where $\alpha = \frac{1}{k}$. A balance equation is *active* if it contains at least $d+1$ floating variables. As before, we conclude that at any time the system (LP_{act}) of active balance equations is underdetermined, so we can fix at least one floating variable. If all variables are fixed the number of variables with value 1 must be in the open interval $(\alpha|\ell| - d, \alpha|\ell| + d)$ for every line ℓ.

Iterating this procedure yields k color classes, but the imbalance could be as high as approximately $2d \cdot \ln k$. To get an imbalance independent of k, we can use divide-and-conquer instead of the iteration just described. If k is even we do one iteration step with α set to $\frac{1}{2}$, i.e., we search for a 2-coloring; we then recursively $\frac{k}{2}$-color the vertices of color 1 and $\frac{k}{2}$-color the remaining vertices. If k is odd we do one iteration step with α set to $\frac{k-1}{2k}$; we then recursively $\frac{k-1}{2}$-color the vertices of color 1 and $\frac{k+1}{2}$-color the remaining vertices. It can be shown that the total imbalance after the recursion will be a least $4d-1$ for sufficiently large k.

This bound is slightly worse than the bound in Theorem 1, but it has the advantage that it will adapt to any theorems proved about balanced 2-colorings. For example, it was conjectured that there always exists an $O(\sqrt{d})$-balanced 2-coloring [1,6]. If this were proved it would immediately imply the existence of an $O(\sqrt{d})$-balanced k-coloring.

5 NP-Completeness Results

Akiyama and Urrutia [2] showed that every set of points on the 2-dimensional rectangular grid has an almost-balanced k-coloring for $k \geq 2$, and there is an efficient algorithm to compute such a coloring. They also gave an example of points on a 3-dimensional grid that do not admit an almost-balanced coloring. We strengthen this result by showing that testing whether a set of vertices has an almost-balanced 2-coloring is NP-complete for line families of dimension $d \geq 3$.

Theorem 3 *Let $d \geq 3$. Let $\mathcal{L}$ be a set of lines of dimension d containing a set of vertices P. Then the problem to decide whether P has an almost-balanced 2-coloring is NP-complete.*

PROOF. Clearly, the problem is in NP, because given a 2-coloring, one can verify in polynomial time whether it is almost-balanced. In the following, we will therefore only show that the problem is NP-hard by reduction from Not-all-equal 3sat which is known to be NP-hard [15]. The problem Not-all-equal 3sat is the following: Given n Boolean variables $x_1, \ldots, x_n$ and m clauses $c_1, \ldots, c_m$ which each contain exactly three literals (i.e., variables or their negative), determine whether there exists an assignment of Boolean values to the variables such that for each clause at least one literal is true and at least one literal is false.

Assume an instance S of Not-all-equal 3sat is given. We want to construct a set of vertices P and a family $\mathcal{L}$ of lines containing the vertices P such that P can be almost-balanced 2-colored if and only if S has a solution. For ease of description, we will assume that the two colors are red and blue. For each clause c_j, we will have one line l_j that contains three vertices, one for each literal in c_j. The lines corresponding to different clauses use different vertices.

In any almost-balanced coloring of P at least one of the vertices on any line must be red and at least one of the vertices must be blue.

Now, by adding additional lines and vertices, we will ensure that two vertices representing the same literal must have the same color, and two vertices representing a literal and its negation, respectively, must have different color. Let x be a literal. Let $p_1, p_3, \ldots, p_{2s+1}$ be the vertices corresponding to the s occurrences of x in S, and let $p_{2s+2}, p_{2s+4}, \ldots, p_{2t}$ be the vertices corresponding to the t occurrences of $\bar{x}$ in S. Let p_{2j} be a set of new vertices, for $j = 1, \ldots, s$, and let p_{2j+1} be a set of new vertices, for $j = s+1, \ldots, t-1$. We create new lines ℓ_i, for $i = 1, \ldots, 2t-1$, where ℓ_i contains exactly the vertices p_i and p_{i+1}. Since these two vertices must be colored differently in any almost-balanced 2-coloring, the colors of the vertices $p_1, p_2, \ldots, p_{2s}, p_{2s+1}, p_{2s+2}, \ldots, p_{2t-1}, p_{2t}$ must alternate. In particular, all vertices representing x must have the same color, and all vertices representing $\bar{x}$ must have the other color.

This construction can be done in polynomial time. Note that each vertex is contained in at most three lines. Hence we have constructed a set $\mathcal{L}$ of dimension 3 (which could also be considered to be of any dimension $d \geq 3$). Following the construction, one immediately verifies that there exists an almost-balanced 2-coloring for this construction if and only if the instance S of NOT-ALL-EQUAL 3SAT has a solution. □

5.1 Geometric Settings

We just showed that the problem of finding an almost-balanced coloring is NP-complete for line families of dimension at least three. Now we want to strengthen this result to hold in geometric settings as well. In other words, we want to ensure that the vertices (called *points* in a geometric setting) are placed on grid-lines in dimension D, of which at most d intersect in one point. We show that the problem for the following geometric settings is also NP-complete:

- Finding an almost-balanced 2-coloring of points in the 2-dimensional rectangular grid with one set of diagonals, i.e., $D = 2$, $d = 3$, $k = 2$.
- Finding an almost-balanced 2-coloring of points in the 3-dimensional rectangular grid, i.e., $D = 3$, $d = 3$, $k = 2$.
- Finding an almost-balanced 3-coloring of points in the 2-dimensional rectangular grid with one set of diagonals, i.e., $D = 2$, $d = 3$, $k = 3$.

We suspect that finding an almost-balanced k-coloring is NP-complete in all geometric settings with $d \geq 3$, but leave this as an open problem. Note that an NP-completeness result for 2-colorings in some setting does not immediately imply NP-completeness for k-colorings, for $k \geq 3$, in the same setting, although we would intuitively expect that using more colors should make the problem more difficult.

5.1.1 NP-Completeness for 2-Colorings

Assume an instance S of Not-all-equal 3sat is given. We want to construct an instance S' of an almost-balanced 2-coloring problem on a grid such that S' has a solution if and only if S does.

The principal idea is the same as for the previous proof. As before for each clause c_j, we will have one grid line ℓ_j that contains three points, one for each literal in c_j. In any almost-balanced coloring at least one of these points must be red and at least one of these points must be blue.

To ensure that all points representing the same literal have the same color, we need to add more points than before in a geometric setting. More precisely, for each variable x_i, we will have one point p_i which can be freely colored with either red or blue. Point p_i will have the same color as a literal x_i, and the opposite color as a literal $\bar{x}_i$.

We give the construction first for $D = 2$ and $d = 3$, i.e., for the rectangular grid with one set of diagonals. As before, let s (t) be the number of times that x_i ($\bar{x}_i$) appears in a clause. We represent x_i by points that are arranged in a staircase with $s+t$ many steps. At the first s steps, we duplicate the left point one unit away in the diagonal, and at the t remaining steps, we duplicate the right point one unit away in the diagonal. The topmost point of the staircase is p_i. See Figure 1.

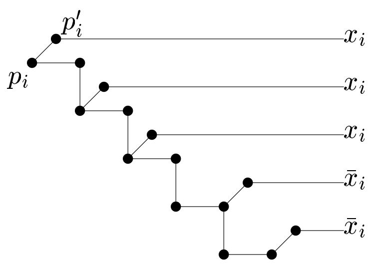

Fig. 1. Placing points for one variable x_i. In this example, $s = 3$ and $t = 2$.

Assume that we have an almost-balanced 2-coloring of this configuration, and that p_i is colored red. Then, assuming that no other points are added in the rows and columns containing the steps, one can argue that any left point of a step must be red and any right point of a step must be blue.

If p_i' is the point in the diagonal of p_i, then p_i' must be blue because p_i is red. At some point later, we will place exactly one more point p_i'' in the row of p_i'. This point then has to be red because p_i' is blue. Hence, point p_i'' has the same color as p_i in any almost-balanced coloring, and can be used to represent literal x_i in some clause.

Let q_i be one of the right points of a step that have another point q_i' in their diagonal. Then q_i must be blue, so q_i' must be red. At some point later, we will place exactly one more point q_i'' in the row of q_i'. This point then has to be blue because q_i' is red. Hence, point q_i'' has the opposite color as p_i in any almost-balanced coloring, and can be used to represent literal $\bar{x}_i$ in some clause.

We now have s rows to add a point for literal x_i, and t rows to add a point for literal $\bar{x}_i$.

We combine these constructions for $x_1, \ldots, x_n$ in such a way that no two points for two different variables share a grid line. For example, this can be done by placing them along the other diagonal, as illustrated in Figure 2.

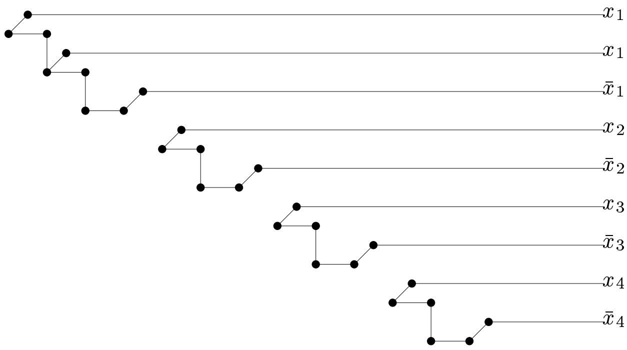

Fig. 2. Placing all variables.

Finally, we add points in m extra columns, one for each clause. If c is the column for clause c_j, then for each literal l of c_j we put a point into one of the rows that correspond to this literal. Furthermore, we choose the row in such a way that for each row there is exactly one point in one of the columns for the clauses.

Also, care must be taken that the columns for the clauses are far enough apart such that no two points in two different columns could possibly be on one diagonal. This can be achieved by spacing the columns at least $6m + 1$ units aparts. See Figure 3.

Following the construction, one immediately verifies that there exists an almost-balanced coloring for this construction if and only if the instance of Not-all-equal 3sat has a solution. This proves the following theorem.

Theorem 4 *The problem of finding an almost-balanced 2-coloring in a two-dimensional rectangular grid with one set of diagonals is NP-complete.* □

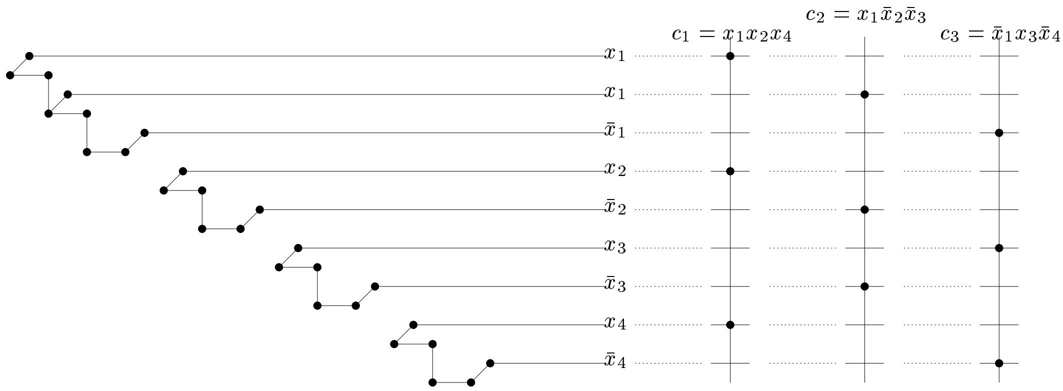

Fig. 3. Adding columns for the clauses. For readability the figure is not to scale.

The construction for 2-colorings of points in the three-dimensional rectangular grid is identical. The only change is that Figure 3 should now be interpreted as a three-dimensional picture, with two planes depth. All points for clauses are added in the second (deeper) of these planes. Thus, we obtain:

Theorem 5 *The problem of finding an almost-balanced* 2*-coloring in a three-dimensional rectangular grid is NP-complete.* □

Note one interesting feature: we "barely" use the third dimension, because we actually only use two parallel planes. Nevertheless, having two planes instead of one makes the problem NP-complete.

5.1.2 NP-Completeness for 3-Colorings

Now we show that the problem does not become easier if we allow one more color. More precisely, the problem of finding an almost-balanced 3-coloring in a two-dimensional rectangular grid with one set of diagonals is also NP-complete.

We use almost the same reduction as in the previous section. The main difference is that having two points per grid line is not sufficient because this does not enforce any color. Therefore, for every grid line that contains two points in the above construction, we will add a third point. By adding even more points, we force these third points all to have the same color, say white. Hence, the two original points have exactly as much color choice as before, which means that the same reduction applies. For each column of a clause (which are the only grid lines containing three points) we add one point that also must be white. The remaining three points in such a column must all be red or blue (because their row contains now two more points, one of which is forced to be white), so as before, we must have at least one red and at least one blue point per column for a clause.

The precise addition of points works as follows. Assume that in the above construction, we have v grid lines that are vertical or diagonal and contain at least two points, and we have h horizontal grid lines that contain exactly two points.

Imagine placing v vertical diamonds, one attached to each other. See the left picture in Figure 4. Assume we have an almost-balanced 3-coloring of this construction. Let p be a tip of one diamond. The two points at the middle of this diamond both share a grid line with p, and because every grid line contains at most three points, they must have different colors from p. But then the other tip of the same color, which also shares grid lines with these points, must have the same color as p. Hence, all the tips of all diamonds have the same color.

Now add h horizontal diamonds, scaled such that they do not share any grid lines with the vertical diamonds, except at the attachment point. Again this construction has an almost-balanced 3-coloring, and all tips of all diamonds have the same color. See Figure 4.

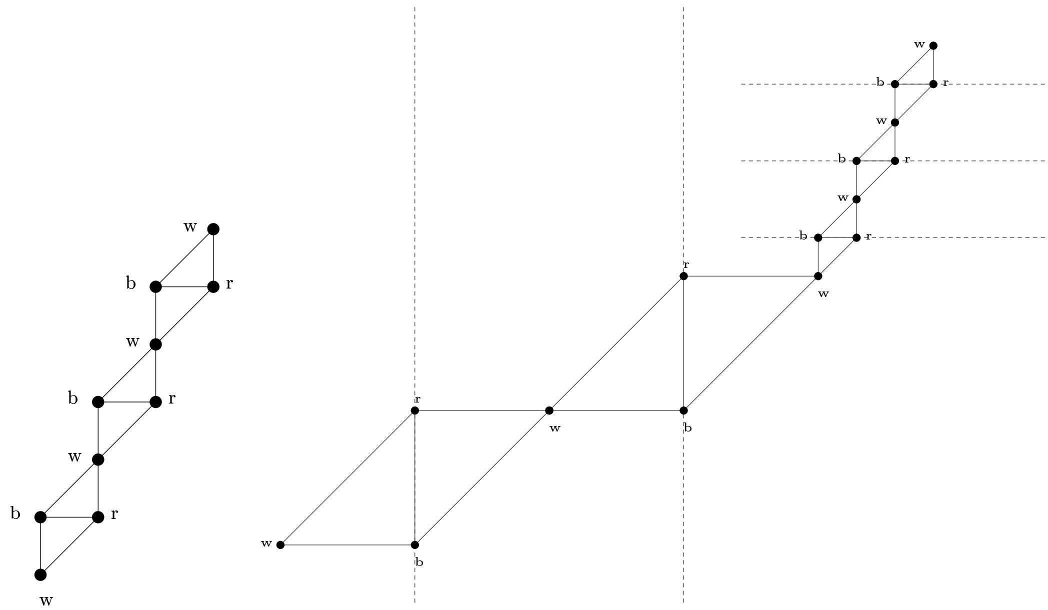

Fig. 4. The construction of a color splitter. We show here $v = 3$ and $h = 2$, though normally these numbers would be bigger.

Assume that all tips of diamonds are colored white. In this construction, which we call a *color splitter*, there are then v rows and h columns that contain exactly one red and one blue point. These rows/columns are indicated with dashed lines in Figure 4. Hence, if we place a third point in these rows/columns, then that point must be colored white.

All that remains to do is to place the color splitter in such a way that all these extra points can at the same time be in some grid line of the original construction. This can be done as follows:

Start with the construction of the previous section. Extend all lines that contain at least two points infinitely. Place the color splitter such that it is below and to the right of any of the intersection points of these infinite lines. Now, for every horizontal infinite line from the original construction, choose one of the h columns of the color splitter, and place a point at their intersection. Similarly, for any vertical infinite line or any diagonal infinite line of the original construction, choose one of the v rows of the color splitter and place a point at their intersection. See Figure 5.

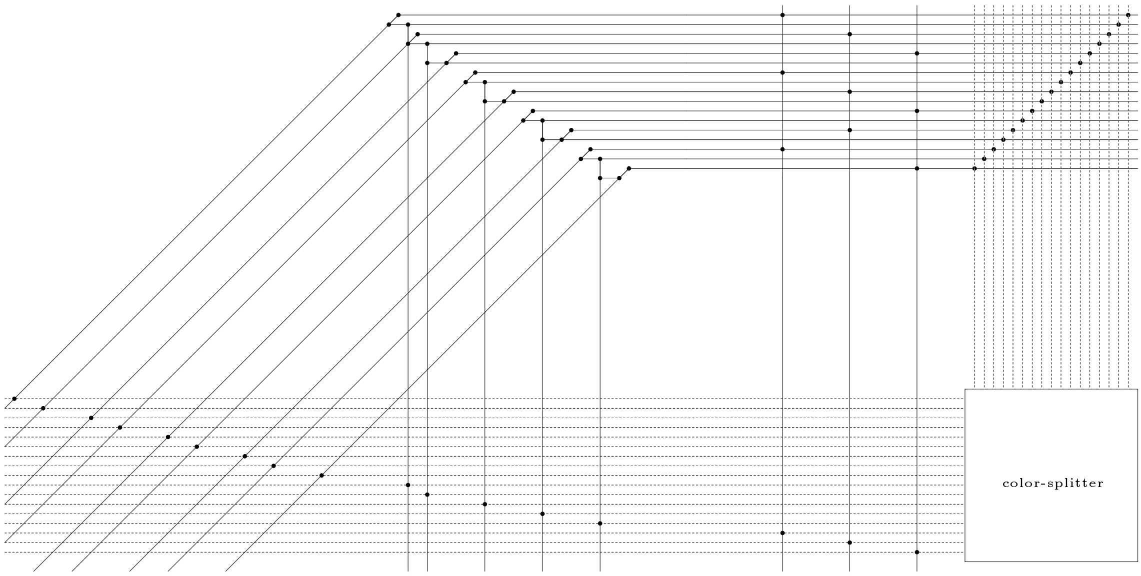

Fig. 5. Combining the construction for the 2-coloring with a color splitter.

All these added points must be white. Hence, any of the grid lines of the original construction that contained two points before now must color these two points with red and blue. Hence, adding the third color does not give us any additional freedom, and the problem remains NP-complete.

Theorem 6 *The problem of finding an almost-balanced* 3*-coloring in a two-dimensional rectangular grid with one set of diagonals is NP-complete.* □

We leave as an open problem whether finding an almost-balanced coloring is NP-hard for $k \geq 4$ colors on a rectangular grid with one set of diagonals. We would expect that the answer to this problem is yes, at least if we also increase d (the number of grid lines that are allowed to cross in one point). Observe that it would be enough to find a color splitter for $k \geq 4$ colors; if this exists, then the problem becomes NP-hard for k colors with a similar technique as in the previous section. We note however, that the construction of the color-splitter can be generalized to non-geometric settings with $d \geq \max\{3, k-1\}$

so we have NP-completeness for these cases as well.

Acknowledgements

We thank Torben Hagerup for pointing out the related work by Šíma [16], and Gerhard Woeginger for bringing the work by Bednarchak and Helm [7] to our attention. We also thank Joel Spencer, Benjamin Doerr, and an unknown referee for their helpful comments.

References

[1] Noga Alon and Joel H. Spencer. *The Probabilistic Method.* Wiley, New York, 1992. Chapter 12, pages 185–196.

[2] Jin Akiyama and Jorge Urrutia. A note on balanced colourings for lattice points. *Discrete Mathematics*, 83(1):123–126, 1990.

[3] Tetsuo Asano, Tomomi Matsui, and Takeshi Tokuyama. On the complexities of the optimal rounding problems of sequences and matrices. In *Proceedings of the 7th Scandinavian Workshop on Algorithm Theory (SWAT'00)*, Bergen, Norway, July 2000. To appear.

[4] József Beck. Some results and problems in "combinatorial discrepancy theory". In *Topics in Classical Number Theory: Proceedings of the International Conference on Number Theory*, pages 203–218, Budapest, Hungary, July 1981. Appeared in *Colloquia Mathematica Societatis János Bolyai*, volume 34, 1994.

[5] József Beck and Tibor Fiala. "Integer-making" theorems. *Discrete Applied Mathematics*, 3:1–8, 1981.

[6] József Beck and Vera T. Sós. Discrepancy theory. In *Handbook of Combinatorics*, volume 2, pages 1405–1446. Elsevier, Amsterdam, 1995.

[7] Debe Bednarchak and Martin Helm. A note on the Beck-Fiala theorem. *Combinatorica*, 17(1):147–149, 1997.

[8] Dominique de Werra. Equitable colorations of graphs. *R.I.R.O.*, R-3, pages 3–8, 1971.

[9] Dominique de Werra. Some results in chromatic scheduling. *Zeitschrift für Operations Research*, 18: 167–175, 1974.

[10] Benjamin Doerr. Multi-Color Discrepancies. Dissertation. Christian-Albrechts-Universität zu Kiel, 2000.
Url: `http://www.numerik.uni-kiel.de/~bed/wiss.html/diss.ps.gz`

[11] Benjamin Doerr. Discrepancy in different numbers of colors. *Discrete Mathematics*, accepted for publication.

[12] Benjamin Doerr and Anand Srivastav. Approximation of multi-color discrepancy. In D. Hochbaum, K. Jansen, J. D. P. Rolim, and A. Sinclair, editors, *Randomization, Approximation, and Combinatorial Optimization: Algorithms and Techniques*, volume 1671 of *Lecture Notes in Computer Science*, pages 39–50, 1999.

[13] Benjamin Doerr and Anand Srivastav. Recursive randomized coloring beats fair dice random colorings. In A. Ferreira and H. Reichel, editors, *Proceedings of the 18th Annual symposium on Theoretical Aspects of Computer Science (STACS 2001)*, volume 2010 of *Lecture Notes in Computer Science*, pages 183–194, 2001.

[14] Anthony J. W. Hilton and Dominique de Werra. A sufficient condition for equitable edge-colourings of simple graphs. *Discrete Mathematics*, 128(1-3): 179–201, 1994.

[15] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 216–226, San Diego, California, May 1978.

[16] Jiří Šíma. Table rounding problem. *Comput. Artificial Intelligence*, 18(3): 175–189, 1999.

[17] Joel Spencer. Geometric discrepancy theory. *Contemporary Mathematics*, 223, 1999.

[18] Joel Spencer. Personal communication. 2000.