The complexity of obtaining a distance-balanced graph

Sergio Cabello[*]

Faculty of Mathematics and Physics, University of Ljubljana, and
Institute of Mathematics, Physics and Mechanics
Jadranska 19, 1000 Ljubljana, Slovenia

sergio.cabello@fmf.uni-lj.si

Primož Lukšič[†]

Faculty of Mathematics and Physics, University of Ljubljana, and
Institute of Mathematics, Physics and Mechanics
Jadranska 19, 1000 Ljubljana, Slovenia

primoz.luksic@fmf.uni-lj.si

Abstract

An unweighted, connected graph is *distance-balanced* (also called *self-median*) if there exists a number d such that, for any vertex v, the sum of the distances from v to all other vertices is d. An unweighted connected graph is *strongly distance-balanced* (also called *distance-degree regular*) if there exist numbers $d_1, d_2, d_3, \ldots$ such that, for any vertex v, there are precisely d_k vertices at distance k from v.

We consider the following optimization problem: given a graph, add the minimum possible number of edges to obtain a (strongly) distance-balanced graph. We show that the problem is NP-hard for graphs of diameter three, thus answering the question posed by Jerebic et al. [Distance-balanced graphs; *Ann. Comb.* 2008]. In contrast, we show that the problem can be solved in polynomial time for graphs of diameter 2.

1 Introduction

For a graph G, let $d_G(u, v)$ denote the minimal path-length distance between vertices u and v of G. In this paper we restrict our attention to finite and *connected* graphs, and thus the

---

distances $d_G(\cdot, \cdot)$ are always finite. Let us introduce the notation $d_G(v) = \sum_{u \in V(G)} d_G(u, v)$ for the sum of the distances in G from v to all other vertices.

The *median* of a graph G is the set of all vertices v of G for which the value $d_G(v)$ is minimized. A graph is *self-median* if its median is the whole vertex set. Thus, a graph G is self-median if and only if the value $d_G(v)$ is constant over all vertices v of G. A seemingly unrelated concept is that of distance-balanced graphs, due to Jerebic et al. [8] (see also Handa [6]). A graph G is *distance-balanced* if for all edges uv of G it holds

$$|\{x \in V(G) \mid d_G(x, u) < d_G(x, v)\}| \; = \; |\{x \in V(G) \mid d_G(x, v) < d_G(x, u)\}|.$$

Balakrishnan et al. [2] noticed that a connected graph G is distance-balanced if and only if it is self-median. Thus, the concepts of distance-balanced and self-median are the same. (For non-connected graphs one has to look into each connected component separately. However, as we mentioned before, we will only consider *connected graphs* through the paper.) For the rest of this paper, we will use the term distance-balanced but in fact use the equivalent definition of self-median.

Distance-balanced graphs are relevant in the area of facility location problems [4] because the median of a graph comprises of vertices that have a minimal sum of distances to all other vertices. They are also useful in mathematical chemistry. For example, bipartite distance-balanced graphs have the maximal Szeged index among all graphs of the same size [1, 7], while distance-balanced graphs have the maximal revised Szeged index [1, 12].

A graph G is *distance-degree regular* (DDR) if for any integer k there is a number d_k such that

$$|\{x \in V(G) \mid d_G(v, x) = k\}| \; = \; d_k$$

for all vertices v of G. The concept is due to Bloom et al. [3]. Kutnar et al. [11] introduced the following notion: a graph G is *strongly distance-balanced* if for every edge uv of G and any integer k it holds

$$|\{x \in V(G) \mid k = d_G(x, u) = d_G(x, v) - 1\}| \; = \; |\{x \in V(G) \mid k = d_G(x, v) = d_G(x, u) - 1\}|.$$

It turns out [11] that a connected graph is strongly distance-balanced if and only if it is DDR. Thus, both concepts are actually the same (for connected graphs). For the rest of this paper, we will use the term strongly distance-balanced but in fact use the equivalent definition of DDR-graph. Note that a strongly distance-balanced graph is distance-balanced as well.

We are interested in the following optimization problem: given a graph, add the minimum possible number of edges to obtain a (strongly) distance-balanced graph. Since complete graphs are (strongly) distance-balanced, the problem always has a feasible solution. This problem is considered in Jerebic et al. [8], where it is mentioned that the computation seems quite hard. Cycles are distance-balanced graphs with the smallest possible number of edges. In [12] this optimization problem was considered for cycles with an added edge. The solutions were hard to compute even for small cycles, as there was no symmetry or common patterns in the solutions. The heuristics that were tried

to solve the problem did not return very good results either. Thus, it seemed that the problem must be hard.

We formulate the associated decision problems:

> DBEA (Distance-balanced edge addition)
> *Input:* Graph $G = (V, E)$ and an integer k.
> *Output:* Can we obtain a distance-balanced graph from G by adding at most k edges?

> STRONG-DBEA
> *Input:* Graph $G = (V, E)$ and an integer k.
> *Output:* Can we obtain a strongly distance-balanced graph from G by adding at most k edges?

In Section 2 we show that both problems are solvable in polynomial time for graphs with diameter 2. The algorithm is based in the theory of b-matchings and a simple characterization of (strongly) distance-balanced graphs of diameter 2. In contrast, we show in Section 3 that the problems DBEA and STRONG-DBEA are NP-complete for graphs of diameter 3, and thus also for general graphs. The proof is based on a delicate construction. The main obstacle is to have control over which edges may be added in an optimal solution.

2 Graphs of diameter two

In this section we will show that the problems DBEA and STRONG-DBEA are solvable in polynomial time for graphs with diameter 2. Let G be a graph with diameter 2. We have the following straightforward characterization.

Lemma 2.1. *If G is connected and has diameter 2, then the following statements are equivalent:*

(a) *G is distance-balanced;*

(b) *G is strongly distance-balanced;*

(c) *G is regular.*

Proof. Clearly, conditions (b) and (c) are equivalent for graphs of diameter 2. Since G has diameter two, for any vertex v we have

$$d_G(v) = \deg_G(v) + 2 \cdot (|V(G)| - \deg_G(v) - 1) = 2|V(G)| - \deg_G(v) - 2.$$

Thus, $d_G(v) = d_G(u)$ if and only if $\deg_G(v) = \deg_G(u)$, and the equivalence of (a) and (c) follows. $\square$

Theorem 2.2. *The distance-balanced edge addition problem* (DBEA) *and the strongly distance-balanced edge addition problem* (STRONG-DBEA) *can be solved in polynomial time for graphs of diameter* 2.

Proof. Because of Lemma 2.1, the problems DBEA and STRONG-DBEA are equivalent to the following problem: given a graph G and an integer k, is there a set F with at most k edges such that $G + F$ (G with the added edges F) is a regular graph? We next show that this problem can be solved in polynomial time.

Let $G = (V, E)$ be the given input graph and denote by Δ its maximum degree. Let $\bar{G} = (V, \bar{E})$ be its complement graph. The problem of deciding whether the graph $\bar{G}$ has a d-regular spanning subgraph can be solved in polynomial time because it is a special case of a b-matching; see for example Schrijver [14, Chapter 31] or Korte and Vygen [9, chapter 12]. Thus, for any given d we can decide in polynomial time if there exist a set of edges $F_d \subseteq \bar{E}$ whose removal makes $\bar{G}$ d-regular, which is equivalent to the graph $G + F_d$ being $(|V| - 1 - d)$-regular. Since such F_d has $|V|(|V| - d - 1)/2 - |E|$ edges, then $|F_i| > |F_j|$ whenever F_i and F_j exist and $i < j$. We can thus try each value d between 0 and $|V| - 1 - \Delta$ to find the maximum d^* for which there exist a set F_{d^*} such that $G + F_{d^*}$ is $(|V| - 1 - d^*)$-regular, and return "Yes" if and only if $|F_{d^*}| \leq k$. $\qquad\square$

3 Graphs of diameter three

In this section we will show that the problems DBEA and STRONG-DBEA are NP-complete for graphs with diameter 3. A *dominating set* in a graph G is a subset of vertices $U \subseteq V(G)$ such that each vertex of $V(G) \setminus U$ has at least one neighbor in U. Our reduction will be from the dominating set problem for cubic graphs.

> DOM3
> *Input:* Cubic graph G and an integer k.
> *Output:* Is there a dominating set $U \subseteq V(G)$ with at most k vertices?

The problem DOM3 is NP-complete [5, Update for the current printing], even when restricted to planar graphs [10]. It is clear that the problem keeps being NP-complete if we assume that $k > 1$. Furthermore, Reed's theorem [13] implies that a cubic graph G has a dominating set with at most $\frac{3}{8}|V(G)|$ vertices. Thus, we can restrict our attention to inputs (G, k) for DOM3 that satisfy

$$2 \leq k \leq \frac{3}{8}|V(G)|. \tag{1}$$

Let (G, k) be an input for DOM3 satisfying (1), and let us use $n = |V(G)|$. We define a graph $H = H(G, k)$ as the graph arising from the following construction; see Figure 1.

1. We start H as the graph G and a disjoint copy G' of G. We will use x' for the copy in G' of a vertex x of G.
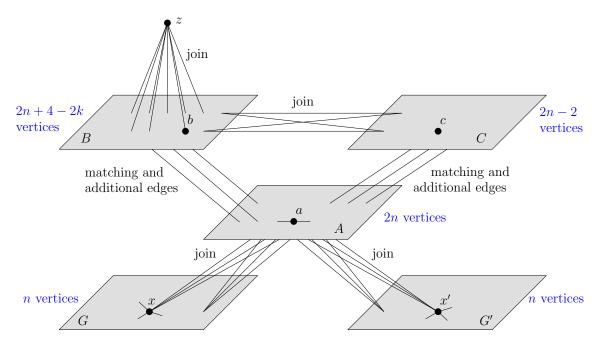
Figure 1: Graph $H = H(G, k)$ constructed as an input for DBEA.

2. We add to H a set $A = \{a_0, \ldots, a_{2n-1}\}$ of $2n$ vertices. We make a cycle on A putting the edges $a_i a_{i+1}$, where $0 \le i \le 2n - 1$ and indices are modulo $2n$.

3. We add to H a set $B = \{b_0, \ldots, b_{2n+3-2k}\}$ of $2n + 4 - 2k$ vertices. We make a 4-regular graph on B putting the edges $b_i b_{i+1}$ and $b_i b_{i+2}$, where $0 \le i \le 2n + 3 - 2k$ and indices are modulo $2n + 4 - 2k$.

4. We add to H a set $C = \{c_0, \ldots, c_{2n-3}\}$ of $2n - 2$ vertices. We make a (2k-1)-regular graph on C putting the edges $c_i c_{i+1}, c_i c_{i+2}, \ldots, c_i c_{i+k-1}$ and $c_i c_{i+n-1}$, where $0 \le i \le 2n - 3$ and indices are modulo $2n - 2$.

5. We add a vertex z to H and put edges between z and each vertex of B.

6. We put an edge between each vertex of B and C.

7. We put an edge between each vertex of A and $V(G) \cup V(G')$.

8. We make a maximal matching between A and C by putting the edges $a_i c_i$, where $0 \le i \le 2n - 3$. With this, each vertex of C has the same degree. To force that each vertex of A has the same degree, we remove the edge $c_0 c_1$, and add the edges $c_0 a_{2n-2}$ and $c_1 a_{2n-1}$. With this, each vertex of A is adjacent to some vertex of C, and vice versa.

9. Since $k \ge 2$ by (1), it holds $|A| \ge |B|$. We make a maximal matching between A and B by putting the edges $a_i b_i$, where $0 \le i \le 2n + 3 - 2k$. With this, each vertex of B has the same degree. To force that each vertex of A has the same degree, we

further make the following: for each even i between $t = 2n + 3 - 2k$ and $2n - 1$, we remove from H the edge $b_{i-t}b_{i-t+1}$, and add the edges $a_i b_{i-t}$ and $a_{i+1}b_{i-t+1}$. (Note that $i - t + 1$ is at most $2n - 1 - (2n + 3 - 2k) + 1 = 2k - 3 \leq 2n + 4 - 2k$ by (1), and thus the indices of b move in a valid range.) With this, each vertex of A is adjacent to some vertex of B, and vice versa. This finishes the construction of H.

We next discuss basic properties of the graph H. The graph H has

$$n + n + 2n + (2n + 4 - 2k) + (2n - 2) + 1 = 8n - 2k + 3$$

vertices. The degrees of the vertices in H are:

$$\deg_H(v) = \begin{cases} 2n + 4 & \text{if } v \in A \cup B \cup C; \\ 2n + 3 & \text{if } v \in V(G) \cup V(G'); \\ 2n - 2k + 4 & \text{if } v = z. \end{cases}$$

We have the following key property in H:

The distance in H from z to vertices in $V(G) \cup V(G')$ is exactly 3. Any other distance in H is at most 2.

As an example, let us see that this property holds for a vertex $a \in A$: it is adjacent to any vertex $x \in V(G)$ or $x' \in V(G')$, it is at distance at most 2 to any other vertex $a' \in A$ through any vertex $x \in V(G)$, it is at distance at most 2 to any vertex $b \in B$ through a vertex $c \in C$, it is at distance at most 2 to any vertex $c \in C$ through a vertex $b \in B$, and it is at distance 2 to vertex z through a vertex $b \in B$. Similar arguments work for the other cases.

A simple calculation shows that the sum of distances from vertices are:

$$d_H(v) = \begin{cases} 14n - 4k & \text{if } v \in A \cup B \cup C; \\ 14n - 4k + 2 & \text{if } v \in V(G) \cup V(G'); \\ 16n - 2k & \text{if } v = z. \end{cases}$$

We can now prove the following characterization.

Lemma 3.1. *Let G be a cubic planar graph with n vertices and k an integer satisfying (1). The graph G has a dominating set of size at most k if and only if the graph H can be made distance-balanced with the addition of at most $n + k$ edges.*

Proof. Assume that G has a dominating set of size at most k, and let U be a dominating set of size exactly k. Consider the graph obtained from H by adding the edges zu and zu' for each $u \in U$, as well as the edges vv' for each $v \in V(G) \setminus U$. The resulting graph is $(2n + 4)$-regular: the degree of each vertex $x \in V(G)$ and $x' \in V(G')$ is increased by one, the degree of z is increased by $2k$, and the other degrees are untouched. Furthermore, the diameter of the resulting graph is 2: the distance from z to a vertex $u \in U$ is 1 and the

distance from z to vertex $x \in V(G) \setminus U$ is 2 through the vertex $u \in U$ that dominates x. Since the resulting graph is regular and has diameter 2, it is distance-balanced by Lemma 2.1. Together, we added $2k + (n - k) = n + k$ edges, which proves the forward direction of the Lemma.

The rest of the proof is devoted to the other direction of the Lemma. Assume that there is a set F with at most $n + k$ edges whose addition to H gives a distance-balanced graph $H + F$. Let U be the set of vertices in $V(G) \cup V(G')$ that are adjacent to z through an edge of F:

$$U = \{v \in V(G) \cup V(G') \mid vz \in F\}.$$

Let $X = (V(G) \cup V(G')) \setminus U$ be the set of edges from $V(G) \cup V(G')$ not adjacent to z through F. We will show the following:

> *Structural Claim.* U has exactly $2k$ vertices, the restriction of F to X is a matching, and no edge of F connects U to X.

We first argue that no edge from F is incident to $A \cup B \cup C$. Let $\deg_F(v)$ be the number of edges from F adjacent to a vertex v. For any vertex $v \in A \cup B \cup C$ we have

$$d_{H+F}(v) = d_H(v) - \deg_F(v) = 14n - 4k - \deg_F(v)$$

because in H any vertex is at distance at most 2 from v. Thus, if F is incident to some vertex of $A \cup B \cup C$, then it must be incident to all vertices of $A \cup B \cup C$. However, this is not possible because by (1) there are not enough edges in F:

$$|F| \leq n + k < 3n - k < \frac{(2n) + (2n + 4 - 2k) + (2n - 2)}{2} = \frac{|A| + |B| + |C|}{2}.$$

Therefore, the edges in F are incident to $\{z\} \cup V(G) \cup V(G')$ and disjoint from $A \cup B \cup C$.

Since any vertex is at distance at most 2 from a vertex v of $A \cup B \cup C$, and F is not incident to $A \cup B \cup C$, it follows that adding the edges F preserves the sum of distances from v. Therefore we have $d_{H+F}(v) = 14n - 4k$ for any vertex v of $A \cup B \cup C$. Since $H + F$ is distance-balanced, adding the edges F of H must then decrease the value $d_H(z)$ by $2n + 2k$, and the values $d_H(x)$ and $d_H(x')$ by 2 for each $x \in V(G)$.

The sum of distances from a vertex $x \in V(G)$ can only decrease by 2 if some edge of F is incident to x, as only the vertex z is at distance larger than 2 from x. The same holds for any vertex $x' \in V(G')$. Thus, F must be incident to each vertex of $V(G) \cup V(G')$.

Adding F to H, the distance from z to any vertex of U decreases by 2, while the distance from z to any vertex of X can decrease at most by 1. Thus

$$d_{H+F}(z) \geq d_H(z) - (2 \cdot |U| + |X|) = 14n - 2k - |U|,$$

where we have used $|X| = 2n - |U|$. Since $H + F$ is distance-balanced, it must be $d_{H+F}(z) = 14n - 4k$, and therefore $|U| \geq 2k$. If $|U| > 2k$, then it cannot be that F is adjacent to each vertex of $V(G) \cup V(G')$ because there are not enough edges left:

$$\frac{|X|}{2} = n - \frac{|U|}{2} = (n + k) - k - \frac{|U|}{2} > |F| - |U|.$$

Therefore $|U| = 2k$, which implies $|X| = 2(n - k)$. Furthermore, each of the $2(n - k)$ vertices of X must be incident to some of the $n - k$ edges of F that are not incident to z, and thus the restriction of F to X must be a matching. In particular $|F| = n + k$, and no edge of F connects U to X. This finishes the proof of the Structural Claim.

Let x be a vertex from X. Since $\deg_F(x) = 1$ and x is not adjacent to z, the sum of distances from x can decrease by 2 only if $d_{H+F}(x, z) = 2$, which is equivalent to x being adjacent in $H + F$ to a vertex of U. Since F does not have edges connecting U to X by the Structural Claim, then x must be adjacent to U in the graph H. We conclude that U dominates any vertex of X in $G \cup G'$. In particular $V(G) \cap U$ is a dominating set of G and $V(G') \cap U$ is a dominating set of G'. Since $|U| = 2k$, the size of one of these dominating sets is bounded by k. However, since G' is a disjoint copy of G, there exists a dominating set of size at most k in G. □

We can now prove the main theorem.

Theorem 3.2. *The distance-balanced edge addition problem* (DBEA) *is NP-complete, even for graphs of diameter* 3.

Proof. First, we prove that DBEA belongs to the complexity class NP. Using a breadth-first search we can compute the distances between all vertices in polynomial time, and thus we can decide in polynomial time if adding a guessed set of edges gives a distance-balanced graph.

To show that DBEA is NP-complete we use a reduction from the problem DOM3: given an input (G, k) for DOM3, we construct the graph $H = H(G, k)$ in polynomial time, and consider the instance $(H, n + k)$ for DBEA. The answer to input $(H, n + k)$ for DBEA is the same as the answer to input (G, k) for DOM3 because of Lemma 3.1. Since the graph H has diameter 3, the reduction shows NP-hardness for graphs of diameter 3. □

The same reduction works for STRONG-DBEA.

Corollary 3.3. *The strongly distance-balanced edge addition problem* (STRONG-DBEA) *is NP-complete, even for graphs of diameter* 3.

Proof. The problem STRONG-DBEA is in NP: after adding a guessed set of edges in the solution, we can compute the distance degrees of all vertices in polynomial time and decide if the graph is strongly distance-balanced by checking that the graph is DDR.

For showing hardness, we argue the following variation of Lemma 3.1: G has a dominating set of size at most k if and only if the graph H can be made *strongly* distance-balanced with the addition of at most $n + k$ edges. By Lemma 2.1 a graph of diameter 2 is distance-balanced if and only if it is strongly distance-balanced. It follows that the proof of the forward implication in Lemma 3.1 goes untouched for this case. For the reverse implication, we use the following property in the proof of Lemma 3.1: If $H + F$ is distance-balanced, then U must be a dominating set in $G \cup G'$. This structure implies that in the graph $H + F$ the vertex z is at distance at most 2 from any vertex of $V(G) \cup V(G')$, and thus $H + F$ has diameter 2. With this variation of Lemma 3.1 and Lemma 2.1, the reduction used in the proof of Theorem 3.2 applies to STRONG-DBEA as well. □

4   Conclusions

We have shown that there is dichotomy for the problem DBEA: it is polynomially solvable for graphs of diameter 2 but NP-hard for graphs of diameter 3. It seems that our technique does not show that DBEA is NP-hard for other natural families of graphs or related problems. We pose the following questions:

- Is DBEA NP-hard for graphs of diameter 4? More generally, we conjecture that, for every constant $k \neq 1, 2$, the problem DBEA is NP-hard for graphs of diameter k.

- Is DBEA NP-hard for trees?

- It is easy to see that DBEA can be solved in time $n^{O(k)}$ by trying all ($\leq k$)-tuples of edges. Is DBEA fixed-parameter tractable with respect to the number of edges that are added? That is, is there an algorithm that solves DBEA in time $f(k)n^c$ for some function f and constant c?

- In the problem DBEA we add edges. One could consider the version where edges are removed or where edges are removed and added to obtain a distance-balanced graph, while minimizing the number of edges in the symmetric difference between the input graph and the distance-balanced graph. Are these problems NP-hard? Note that if we do not insist on the resulting graph being connected, then the concepts of distance-balanced and self-median are different, and give rise to different problems.

References

[1] M. Aouchiche and P. Hansen. On a conjecture about the Szeged index. *European J. Combin.*, 31(7):1662–1666, 2010.

[2] K. Balakrishnan, M. Changat, I. Peterin, S. Špacapan, P. Šparl, and A. R. Subhamathi. Strongly distance-balanced graphs and graph products. *European J. Combin.*, 30(5):1048–1053, 2009.

[3] G. S. Bloom, L. V. Quintas, and J. W. Kennedy. Distance degree regular graphs. In *The theory and applications of graphs (Kalamazoo, Mich., 1980)*, pages 95–108. Wiley, New York, 1981.

[4] N. Christofides. *Graph theory.* Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1975. An algorithmic approach, Computer Science and Applied Mathematics.

[5] M. R. Garey and D. S. Johnson. *Computers and intractability.* W. H. Freeman and Co., San Francisco, Calif., 1979. A guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences.

[6] K. Handa. Bipartite graphs with balanced (a, b)-partitions. *Ars Combin.*, 51:113–119, 1999.

[7] A. Ilić, S. Klavžar, and M. Milanović. On distance-balanced graphs. *European J. Combin.*, 31(3):733–737, 2010.

[8] J. Jerebic, S. Klavžar, and D. F. Rall. Distance-balanced graphs. *Ann. Comb.*, 12(1):71–79, 2008.

[9] B. Korte and J. Vygen. *Combinatorial optimization*, volume 21 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, fourth edition, 2008. Theory and algorithms.

[10] J. Kratochvíl and M. Křivánek. On the computational complexity of codes in graphs. In *Mathematical foundations of computer science, 1988 (Carlsbad, 1988)*, volume 324 of *Lecture Notes in Comput. Sci.*, pages 396–404. Springer, Berlin, 1988.

[11] K. Kutnar, A. Malnič, D. Marušič, and Š. Miklavič. Distance-balanced graphs: symmetry conditions. *Discrete Math.*, 306(16):1881–1894, 2006.

[12] P. Lukšič. *Growth in graphs*. PhD thesis, University of Ljubljana, Faculty of Computer and Information Science, 2009.

[13] B. Reed. Paths, stars and the number three. *Combin. Probab. Comput.*, 5(3):277–295, 1996.

[14] A. Schrijver. *Combinatorial optimization. Polyhedra and efficiency. Vol. A*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2003. Paths, flows, matchings, Chapters 1–38.