The Distance Matching Problem

Péter Madarasi
Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
madarasi@cs.elte.hu

Abstract: This paper introduces the \textit{d-distance matching problem}, in which we are given a bipartite graph $G = (S, T; E)$ with $S = \{s_1, \ldots, s_n\}$, a weight function on the edges and an integer $d \in \mathbb{Z}_+$. The goal is to find a maximum weight subset $M \subseteq E$ of the edges satisfying the following two conditions: i) the degree of every node of $S$ is at most one in $M$, ii) if $s_it, s_jt \in M$, then $|j - i| \geq d$. The question arises naturally, for example, in various scheduling problems.

We show that the problem is NP-complete in general and admits a simple 3-approximation. We give an FPT algorithm parameterized by $d$ and also settle the case when the size of $T$ is constant. From an approximability point of view, we show that the integrality gap of the natural integer programming model is at most $2 - \frac{1}{2d} - 1$, and give an LP-based approximation algorithm for the weighted case with the same guarantee. A combinatorial $(2 - \frac{1}{d})$-approximation algorithm is also presented. Several greedy approaches are considered, in particular, a local search algorithm that achieves an approximation ratio of $3/2 + \epsilon$ for any constant $\epsilon > 0$ in the unweighted case. The novel approaches used in the analysis of the integrality gap and the approximation ratio of locally optimal solutions might be of independent combinatorial interest.

Keywords: Distance matching, Parameterized algorithms, Approximation algorithms, Integrality gap

1 Introduction

In the \textit{perfect d-distance matching problem}, given are a bipartite graph $G = (S, T; E)$ with $S = \{s_1, \ldots, s_n\}, T = \{t_1, \ldots, t_k\}$, a weight function on the edges $w : E \to \mathbb{R}^+$ and an integer $d \in \mathbb{Z}_+$. The goal is to find a maximum weight subset $M \subseteq E$ of the edges such that the degree of every node of $S$ is one in $M$ and if $s_it, s_jt \in M$, then $|j - i| \geq d$. In the (non-perfect) \textit{d-distance matching problem}, some of the nodes of $S$ might remain uncovered. Note that the order of nodes in $S = \{s_1, \ldots, s_n\}$ affects the set of feasible d-distance matchings, but the order of $T = \{t_1, \ldots, t_k\}$ is indifferent. For example, Figure 1a is a feasible perfect 3-distance matching, but the example shown in Figure 1b is not, because edges $s_1t_2$ and $s_3t_2$ violate the 3-distance condition.

An application of this problem for $w \equiv 1$ is as follows. Imagine $n$ consecutive all-day events $s_1, \ldots, s_n$ each of which must be assigned one of $k$ watchmen $t_1, \ldots, t_k$. For each event $s_i$, a set of possible watchmen is given – those who are qualified to be on guard at event $s_i$. Appoint exactly one watchman to each of the events such that no watchman is assigned to more than one of any $d$ consecutive events, where $d \in \mathbb{Z}_+$ is given. In the weighted version of the problem, let $w_{s_it}$ denote the level of safety of event $s_i$ if watchman $t_j$ is on watch, and the objective is to maximize the level of overall safety.

As another application of the above question, consider $n$ items $s_1, \ldots, s_n$ one after another on a conveyor belt, and $k$ machines $t_1, \ldots, t_k$. Each item $s_i$ is to be processed on the conveyor belt by
one of the qualified machines $N(s_i) \subseteq \{t_1, \ldots, t_n\}$ such that if a machine processes item $s_i$, then it can not process the next $d-1$ items — because the conveyor belt is running.

Motivated by the first application, in the cyclic $d$-distance matching problem the nodes of $S$ are considered to be in cyclic order. The focus of this paper is on the above (perfect) $d$-distance matching problem, but some of the proposed approaches also apply for the cyclic case. In particular, the 3-approximation greedy algorithm achieves the same guarantee for the weighted cyclic case (see Section 3.3), and the $(3/2 + \epsilon)$-approximation algorithm for the unweighted case (Section 4.2).

Previous work Observe that in the special case $d = 1$, one gets the classic (perfect) bipartite matching problem. For $d = 1$, the problem reduces to the $b$-matching problem, and one can show that it is a special case of the circulation problem for $d = 2$. The perfect $d$-distance matching problem is a special case of the list coloring problem on interval graphs [5] and the frequency assignment problem [6].

Our results This paper settles the complexity of the distance matching problem, and gives an FPT algorithm parameterized by $d$. An efficient algorithm for constant $T$ is also given. We present an LP-based $(2 - \frac{1}{d+1})$-approximation algorithm for the weighted distance matching problem, which implies that the integrality gap of the natural IP model is at most $2 - \frac{1}{d+1}$. An interesting alternative proof for the integrality gap is also given. A combinatorial $(2 - \frac{1}{d})$-approximation algorithm is also described for the weighted case. One of the main contributions of the paper is a $(3/2 + \epsilon)$-approximation algorithm for the unweighted case for any constant $\epsilon > 0$ for the unweighted case. The proof is based on revealing the structure of locally optimal solutions recursively. Motivated by the second application above, we give a polynomial time algorithm to find a permutation of $S$ (i.e. the items on the conveyor belt) such that the weight of the optimal $d$-distance matching becomes as large as possible.

Notation Throughout the paper, assume that $G = (S, T; E)$ contains no loops or parallel edges, unless stated otherwise. Let $\Delta(v)$ and $N(v)$ denote the set of incident edges to node $v$ and the neighbors of $v$, respectively. For a subset $X \subseteq E$ of the edges, $N_X(v)$ denotes the neighbors of $v$ for edge set $X$. $\text{deg}(v)$ is the degree of node $v$. Let $L_d(s_i) = \{s_{\text{max}(i-d+1,1)}, \ldots, s_i\}$ and $R_d(s_i) = \{s_1, \ldots, s_{\text{min}(i+d-1,|S|)}\}$. The maximum of the empty set is $-\infty$ by definition. Given a function $f : A \rightarrow B$, both $f(a)$ and $f_a$ denote the value $f$ assigns to $a \in A$, and let $f(X) = \sum_{a \in X} f(a)$ for $X \subseteq A$. Let $\chi_Z$ denote the characteristic vector of set $Z$, i.e. $\chi_Z(y) = 1$ if $y \in Z$, and 0 otherwise. Occasionally, the braces around sets consisting of a single element are abandoned, e.g. $\chi_e = \chi_{\{e\}}$ for $e \in E$.

## 2 Complexity

This section settles the complexity of the $d$-distance matching problem. First, consider the following NP-complete problem.

**Lemma 1.** Given a bipartite graph $G = (S, T; E)$ and $S_1, S_2 \subseteq S$ s.t. $S_1 \cup S_2 = S$, it is NP-complete to decide if there exists $M \subseteq E$ for which $|M| = |S|$ and both $M \cap E_1$ and $M \cap E_2$ are
matchings, where $E_i$ denotes the edges induced by $T$ and $S_i$ for $i = 1, 2$. The problem remains NP-complete even if the maximum degree of the graph is at most 4.

Proof. Given are $X, Y, Z$ finite disjoint sets and a set of hyperedges $\mathcal{H} \subseteq X \times Y \times Z$, a subset of the hyperedges $F \subseteq \mathcal{H}$ is called 3-dimensional matching if $x_1 \neq x_2, y_1 \neq y_2$ and $z_1 \neq z_2$ for any two distinct triples $(x_1, y_1, z_1), (x_2, y_2, z_2) \in F$. Being one of Karp’s 21 NP-complete problems [1], it is NP complete to decide whether there exists a 3-dimensional matching $F \subseteq \mathcal{H}$ of size $|Z|$. In fact, the problem remains NP-complete even if no element of $X \cup Y \cup Z$ occurs in more than three triples in $\mathcal{H}$ [2, Page 221]. Without loss of generality, one might assume that $|X| = |Y| = |Z|$. Let $\mathcal{H}_z = \{e^z_1, \ldots, e^z_{k_z}\}$ denote the set of hyperedges incident to $z \in Z$, i.e. $\mathcal{H}_z = \mathcal{H} \cap (X \times Y \times \{z\})$ for each $z \in Z$. To reduce the 3-dimensional matching problem to the above problem, consider the following construction.

First define a bipartite graph $G = (S, T; E)$ where $S = X \cup (\mathcal{H} \setminus \{e^z_i : z \in Z\}) \cup Y$, $T = \mathcal{H}$ and $E$ is as follows. For each $s \in S \cap (X \cup Y)$, add an edge between $s$ and all the hyperedges $e \in T$ incident to $s$; and connect each $e^z_i \in S \cap \mathcal{H}$ to hyperedges $e^z_{i-1}, e^z_i \in T$ for each $z \in Z$ and $i = 2, \ldots, k_z$. Let $S_1 = S \setminus Y$ and $S_2 = S \setminus X$. Figure 2a and 2b show an instance of the 3-dimensional matching problem and the corresponding construction, respectively. Each hyperedge is represented by a unique line style, e.g. the dotted lines represent hyperedge $e = (x_2, y_1, z_1)$ on Figure 2a, and the dotted lines correspond to the same hyperedge $e$ on Figure 2b. Note that the edges represented by a straight line on Figure 2b do not represent hyperedges, but the edges between hyperedges. The highlighted edges on Figure 2a and 2b correspond to the same feasible 3-dimensional matching.

Observe that there exists a 3-dimensional matching $F$ of size $|Z|$ if and only if there exists $M \subseteq E$ for which $|M| = |S|$ and both $M \cap E_1$ and $M \cap E_2$ are matchings. Indeed, if there exists such an $M \subseteq E$, then $M$ maps $S \cap \mathcal{H}$ to $T$, therefore there exists a unique $e^z_i \in T \cap \mathcal{H}_z$ hyperedge for each $z \in Z$ that is not mapped to $S \cap \mathcal{H}$, but to exactly one element of $x \in X$ and exactly one element of $y \in Y$. These three edges correspond to the inclusion of hyperedge $(x, y, z)$. This way one obtains a 3-dimensional matching $F$ of size $|Z|$. On the other hand, if a 3-dimensional matching $F$ is given for which $|F| = |Z|$, then one might easily construct the desired $M \subseteq E$ as follows. For each $(x, y, z) \in F$, let $i$ be the unique index s.t. $e^z_i = (x, y, z)$ and let $M$ map $x \in S \cap X$ and $y \in S \cap Y$ to $e^z_i \in T$, and $S \cap \mathcal{H}_z$ to $(T \cap H_z) \setminus \{e^z_i\}$ (the latter mapping exists, because the induced subgraph consists of two disjoint paths of odd length). It is easy to see that $|M| = |S|$ and both $M \cap E_1$ and $M \cap E_2$ are matchings.

To complete the proof, observe that the maximum degree in $G$ is at most four if one starts with an instance of the 3-dimensional matching for which no element of $X \cup Y \cup Z$ occurs in more than three triples. Hence, the problem indeed remains NP-complete even if the maximum degree is 4.

In what follows, the previous problem is reduced to the $d$-distance matching problem, hence the hardness of the latter.

**Theorem 2.** It is NP-complete to decide if a graph has a perfect $d$-distance matching, even if the maximum degree of the graph is at most 4.

Proof. It suffices to reduce the above problem to the perfect $d$-distance matching problem. Let $G = (S, T; E)$; $S_1, S_2 \subseteq S$, $S_1 \cup S_2 = S$ be an instance of the above problem. Without loss of generality, one might assume that $S_1 \not\subseteq S_2$ and $S_2 \not\subseteq S_1$. To construct an instance $G' = (S', T'; E')$, $d \in \mathbb{N}$ of the perfect $d$-distance matching problem, let $G' = G$ and modify $G'$ as follows. Order the nodes of $S'$ s.t. the nodes of $S_1 \setminus S_2$, $S_1 \cap S_2$ and $S_2 \setminus S_1$ appear in this order (the order of the element inside the three sets is arbitrary). Insert $|S_1 \setminus S_2|$ and $|S_2 \setminus S_1|$ new nodes to $S'$ right after the last node covered by $S_1$ and right after the last node not covered by $S_2$, respectively. Finally, add $|S_1 \setminus S_2| + |S_2 \setminus S_1|$ new nodes to $T'$ and extend $E'$ with the edges of a perfect matching between the newly added nodes, and set $d = |S|$. Figure 3 illustrates the construction. The blank nodes are the ones added in the last step. The highlighted edges correspond to those on Figure 2b.
(a) An instance of the 3-dimensional matching problem.
(b) The corresponding instance of the problem stated in Claim 1, where \( S_1 = \{x_1, x_2, x_3, e_2, e_6\} \) and \( S_2 = \{e_2, e_6, y_1, y_2, y_3\} \).

Figure 2: Illustration of the proof of Claim 1. Each hyperedge is represented by a unique line style. The highlighted hyperedges on (a) and the highlighted edges on (b) correspond to the same feasible solution.

Figure 3: Illustration of the construction in the proof of Theorem 2 for the problem instance presented on Figure 2b. There exists a perfect 9-distance matching if and only if the problem given on Figure 2b has a feasible solution of size 9.

To complete the proof, observe that there exists a perfect \(|S|\)-distance matching in \( G' \) if and only if there exists \( M \subseteq E \) for which \(|M| = |S|\) and both \( M \cap E_1 \) and \( M \cap E_2 \) are matchings. Note that the maximum degree in \( G' \) is not larger than in \( G \), hence the problem remains hard even if the maximum degree is at most 4.

\[ \square \]

3 Weighted \( d \)-distance matching problem

This section presents various approaches to the weighted \( d \)-distance matching problem. Section 3.1 presents an FPT algorithm \([4]\) with parameter \( d \), while Section 3.2 settles the case when the size of \( T \) is constant. A simple greedy approach is presented in Section 3.3. Finally, Sections 3.4.1 and 3.4.2 are devoted to the investigation of the natural linear programming model.

3.1 FPT algorithm with parameter \( d \)

In what follows, an FPT algorithm with parameter \( d \) is presented for the weighted (perfect) \( d \)-distance matching problem. First observe that the weighted \( d \)-distance matching problem easily reduces to the perfect case by adding a new node \( t_s \) to \( T \) and a new edge \( st_s \) of weight zero for each \( s \in S \), therefore the algorithm is given only for the weighted perfect \( d \)-distance matching
problem. The next claim gives a way to reduce the problem so that it admits an efficient dynamic programming solution.

**Claim 3.** If \( \deg(s) \geq 2d \) for \( s \in S \), then one of the incident edges can be removed without changing the weight of the optimal \( d \)-distance matching.

**Proof.** Let \( st \) be a minimum weight edge incident to node \( s \). In order to prove that \( st \) can be removed, it suffices to show that there is a maximum weight \( d \)-distance matching that does not use edge \( st \). Given a \( d \)-distance matching \( M \) that contains edge \( st \), let \( Z \subseteq T \) denote the nodes that \( M \) assigns to \( L_d(s) \cup R_d(s) \). Since \( |Z| \leq 2d - 1 \), there exists a node \( t' \in N(s) \setminus Z \) for which \( w_{st} \leq w_{st'} \). To complete the proof, observe that \( M' = (M \cup \{st'\}) \setminus \{st\} \) is a \( d \)-distance matching of weight at least \( w(M) \), which does not contain edge \( st \). \( \square \)

Based on Claim 3, the problem can be reduced so that the degree of each node \( s \in S \) is at most \( 2d - 1 \). The reduction can be performed in \( O(m + n) \) steps if the edges are already sorted by their weights at each node \( s \in S \). In what follows, a dynamic programming approach is presented to solve the reduced problem in \( O(d^{d+1}n) \) steps.

For \( i \geq d \), let \( f(s_i, z_1, \ldots, z_d) \) denote the weight of the maximum weight \( d \)-distance matching if the problem is restricted to the first \( i \) nodes of \( S \) and \( s_{i-j+1} \) is assigned to its neighbor \( z_j \) for \( j = 1, \ldots, d \). Formally, \( f(s_i, z_1, \ldots, z_d) \) can be defined by the following recursive formula.

\[
f(s_i, z_1, \ldots, z_d) = \begin{cases} 
    w_{s_iz_1} + \max_{t \in \Delta(s_i-d)} f(s_{i-1}, z_2, \ldots, z_d, t) & \text{if } i > d \text{ and } z_1, \ldots, z_d \text{ are distinct} \\
    \sum_{j=1}^{d} w_{s_jz_{d-j+1}} & \text{if } i = d \text{ and } z_1, \ldots, z_d \text{ are distinct} \\
    -\infty & \text{otherwise},
\end{cases}
\]

where \( i \geq d, s_i \in S \) and \( z_j \in N(s_{i-j+1}) \) for \( j = 1, \ldots, d \).

The weight of the optimal \( d \)-distance matching is

\[
\max \{ f(s_n, z_1, \ldots, z_d) : z_j \in N(s_{n-j+1}) \text{ for } j = 1, \ldots, d \}.
\]

Observe that the number of subproblems is \( O(n(2d - 1)^d) \), since the degree of each \( s \in S \) is at most \( 2d - 1 \). Recursion 1 gives a way to compute \( f(s_i, z_1, \ldots, z_d) \) in \( O(d) \) steps if the subproblems are computed in appropriate order, i.e. the value \( f(s_{i-1}, z_1', \ldots, z_d') \) is available for all necessary \( z_1', \ldots, z_d' \in T \). Therefore the number of steps to compute all the subproblems is \( O(dn(2d - 1)^d) \), furthermore, the optimum value can be computed in \( O((2d - 1)^d) \) steps by (2). The overall running time of the algorithm is \( O(dn(2d - 1)^d + poly(|S| + |T|)) \).

### 3.2 Polynomial algorithm for constant \( |T| \)

If the size of \( T \) is constant, then consider the following subproblems. Let \( f(s_i, d_1, \ldots, d_{|T|}) \) denote the weight of the optimal perfect \( d \)-distance matching when the problem is restricted to \( s_1, \ldots, s_i \), and \( t_j \) can not be matched to nodes \( s_{i-d_j+1}, \ldots, s_i \) for \( j = 1, \ldots, |T| \). Formally, \( f(s_i, d_1, \ldots, d_{|T|}) \) can be defined as follows. If \( i \geq 2 \), then let

\[
f(s_i, d_1, \ldots, d_{|T|}) = \max_{t_j \in N(s_i) : d_j = 0} \{w_{s_it_j} + f(s_{i-1}, d_1-1, \ldots, d_{j-1} - 1, d - 1, d_{j+1} - 1, \ldots, d_{|T|} - 1)\},
\]

if \( i = 1 \), then let

\[
f(s_1, d_1, \ldots, d_{|T|}) = \max_{t_j \in N(s_1) : d_j = 0} w_{s_1t_j}.
\]

The weight of the optimal \( d \)-distance matching is given by

\[
\max_{t_i \in N(s_n)} f(s_{n-1}, 0, \ldots, 0, d - 1, 0, \ldots, 0).
\]
The number of subproblems to be solved is \( O(nd^{d/2}) \), each of which can be computed in \( O(|T|) \) steps by (3) and (4). Once all the subproblems are computed, it takes additional \( O(|T|) \) steps to compute the optimal value by (5). Hence the overall number of steps is \( O(n|T|d^{d/2}) \).

A similar approach settles the non-perfect case for constant \( |T| \), the details of which are left to the reader.

### 3.3 A greedy algorithm

This section describes a greedy method for the weighted \( d \)-distance matching problem, and proves that it is a 3-approximation algorithm.

**Algorithm 1**  
**Greedy**

Let \( e_1, \ldots, e_m \) be the edges in descending order by their weights.

\[
M := \emptyset
\]

for \( i = 1, \ldots, m \)

if \( M \cup \{ e_i \} \) is a feasible \( d \)-distance matching

\[
M := M \cup \{ e_i \}
\]

output \( M \)

**Theorem 4.** Greedy is a 3-approximation algorithm for the weighted \( d \)-distance matching problem.

**Proof.** Assume that Greedy returns edges \( f_1, \ldots, f_p \), and it selects them in this order. Let \( M_i \) denote a maximum weight \( d \)-distance matching that contains \( f_1, \ldots, f_i \), where \( 0 \leq i \leq p \), i.e. \( M_i = \arg \max \{ w(M) : f_1, \ldots, f_i \in M \text{ and } M \text{ is a } d \text{-distance matching} \} \). Furthermore, let \( \theta_i \) denote the weight of \( M_i \) for \( i = 0, \ldots, p \). Note that \( \theta_0 \) is the weight of the optimal \( d \)-distance matching and \( \theta_p \) is the weight of the matching Greedy returns. Observe that there exists edges \( e, e', e'' \in M_i \setminus \{ f_1, \ldots, f_i \} \) s.t. \( (M_i \setminus \{ e, e', e'' \}) \cup \{ f_{i+1} \} \) is a feasible \( d \)-distance matching, which contains edges \( f_1, \ldots, f_{i+1} \). By the greedy selection rule, \( w_e, w_{e'}, w_{e''} \leq w_{f_{i+1}} \), therefore one gets that

\[
\theta_{i+1} \geq \theta_i + w_{f_{i+1}} - w_e - w_{e'} - w_{e''} \geq \theta_i - 2w_{f_{i+1}}
\]

(6)

holds for all \( i = 0, \ldots, p-1 \). Simple inductive argument shows that (6) implies \( \theta_p \geq \theta_0 - 2 \sum_{i=1}^{p} w_{f_i} \), therefore \( 3 \theta_p \geq \theta_0 \) follows, which completes the proof.

The analysis is tight even for \( d = 2 \) and \( w \equiv 1 \) in the sense that Greedy might return only one edge, while the largest 2-distance matching consists of 3 edges, see Figure 4a for an example.

![Figure 4](image)

(a) For \( d = 2 \) and unit weights, Greedy might select edge \( s_2t_2 \) only, while the largest 2-distance matching is of cardinality 3.

(b) For \( d = 2 \) and unit weights, both S-Greedy and T-Greedy select edge \( s_1t_1 \) only, while the largest 2-distance matching is of cardinality 2.

**Remark 5.** The above proof shows that Greedy is a 3-approximation algorithm for the more general cyclic \( d \)-distance matching problem, in which the nodes of \( S \) are considered in cyclic order.
3.4 Linear programming

The following two sections prove that the integrality gap of the natural integer programming model is at most \(2 - \frac{1}{2d-1}\), and present an LP-based \((2 - \frac{1}{2d-1})\)-approximation algorithm for the weighted \(d\)-distance matching problem. First consider the relaxation of the natural 0–1 integer programming formulation of the weighted \(d\)-distance matching problem.

\[
\begin{align*}
\max & \sum_{st \in E} w_{st} x_{st} \\
\text{s.t.} \quad & x \in \mathbb{R}_+^E \\
& x_{st} \leq 1 \quad \forall s \in S \\
& \sum_{s' t \in E : s' \in R_d(s)} x_{s't} \leq 1 \quad \forall s \in S, t \in T
\end{align*}
\]

(LP1)

The relaxation of the 0–1 integer programming formulation of the weighted perfect \(d\)-distance matching problem is as follows.

\[
\begin{align*}
\max & \sum_{st \in E} w_{st} x_{st} \\
\text{s.t.} \quad & x \in \mathbb{R}_+^E \\
& x_{st} = 1 \quad \forall s \in S \\
& \sum_{s' t \in E : s' \in R_d(s)} x_{s't} \leq 1 \quad \forall s \in S, t \in T
\end{align*}
\]

(LP2)

3.4.1 Integrality gap

This section proves that the integrality gap of LP1 is at most \(2 - \frac{1}{2d-1}\), and proves the integrality of LP1 and LP2 in special cases. The former result also follows from the LP-based approximation algorithm described in Section 3.4.2. The following definition plays a central role both in the analysis of the integrality gap, and in the LP-based approximation algorithm presented in the next section.

**Definition 6.** Given a feasible solution \(x\) of LP1, an order of the edges \(e_1 = s_1 t_1, \ldots, e_m = s_m t_m\) is \(\theta\)-flat with respect to \(x\) if

\[
\xi_i + \bar{\xi}_i \leq \theta - x_{e_i},
\]

holds for each \(i = 1, \ldots, m\), where \(\xi_i = \sum \{x_{e_j} : j > i, e_j \in \Delta(s^i)\}\) and \(\bar{\xi}_i = \sum \{x_{e_j} : j > i, e_j \in \Delta(t^i), s' j \in L_d(s^i) \cup R_d(s^i)\}\).

That is, an order of the edges is \(\theta\)-flat if the sum of \(x\) on those edges among \(e_{i+1}, \ldots, e_m\) that are hit by an edge \(e_i\) is at most \(\theta - x_i\) for every \(i\).

**Lemma 7.** There exists an optimal solution \(x \in \mathbb{Q}^m\) of LP1 and an order \(e_1 = s_1 t_1, \ldots, e_m = s_m t_m\) of the edges that is \((2 - \frac{1}{2d-1})\)-flat with respect to \(x\).

*Proof.* Let \(E_s \subseteq \Delta(s)\) denote the first \(\min(2d - 1, \deg(s))\) largest weight edges incident to node \(s\) for each \(s \in S\). Let \(x\) be an optimal solution to LP1 for which \(\gamma(x) = \sum \{x_e : e \in E \setminus \bigcup_{s \in S} E_s\}\) is...
Algorithm 2 The ordering procedure for Lemma 7

Let $x$ be a given fractional solution to LP1, and $G = (S,T; E)$ a copy of the graph.

\[ j := 1 \]

for $i = 1, \ldots, n$ do

while $\deg(s_i) \neq 0$ do

Choose an edge $s_i t \in \Delta(s_i)$ for which $x_{s_i t}$ is as large as possible.

\[ e_j := s_i t \]

\[ j := j + 1 \]

\[ E := E \setminus \{s_i t\} \]

output $e_1, \ldots, e_m$

minimal. By contradiction, suppose that $\gamma(x) > 0$. By definition, $\gamma(x) > 0$ implies that there exists an edge $st \in E \setminus \bigcup_{k=1}^{m} E_k$ for which $x_{st} > 0$. There exists edge $st' \in E_k$ s.t. $x' = x - e_{st'k} + e_{st'}$ is feasible for sufficiently small $\epsilon > 0$, otherwise $x(\bigcup \{ \Delta(s') : s' \in L_d(s) \cup R_d(s) \}) \geq 2d - 1 + \epsilon$ would hold, which is not possible. Observe that $wx \leq wx'$ and $\gamma(x') < \gamma(x)$, contradicting the minimality of $\gamma(x)$. Therefore $\gamma(x) = 0$, meaning that $x_{se}$ is feasible for each $e \in E \setminus \bigcup_{S \in \mathcal{S}} E_s$. Hence one can restrict the edge set of the graph to $\bigcup_{S \in \mathcal{S}} E_s$ without change in the optimal objective value, which implies that there exists a rational optimal solution $x \in \mathbb{Q}^m$ of LP1 with $\gamma(x) = 0$.

Let $x$ be as above, and let $e_1 = s_1 t_1, \ldots, e_m = s_m t_m$ be the order of the edges given by Algorithm 2 for input $x$. To prove that this order is $(2 - \frac{1}{2d-1})$-flat with respect to $x$, let $\xi_i$ and $\xi_i^t (i = 1, \ldots, n)$ be as in Definition 6. First observe that $\xi_i \leq 1 - x_i$ holds for each $i = 1, \ldots, n$, because the algorithm places each edge $\bigcup_{j=1}^{i-1} \Delta(s_j)$ before $e_i$. Hence, to obtain (9), it suffices to prove that $\xi_i \leq 1 - \frac{1}{2d-1}$. For any node $s \in S$, if there exists an edge $st \in \Delta(s)$ for which $x_{st} \geq \frac{1}{2d-1}$, then $\xi_i \leq 1 - \frac{1}{2d-1}$ follows for each $e_i \in \Delta(s)$, since $x_{se} \geq \frac{1}{2d-1}$ holds for the first edge $e \in \Delta(s)$ selected by Algorithm 2. Otherwise, if there exists no edge $st \in \Delta(s)$ for which $x_{st} \geq \frac{1}{2d-1}$, then $\gamma(x(s)) < |E_s| \frac{1}{2d-1} \leq 1$, but then $x' = x + e_{st} - e_{st'}$ is feasible for some $st' \in E_s$ and sufficiently small $\epsilon > 0$ (because $x(\bigcup \{ \Delta(s') : s' \in L_d(s) \cup R_d(s) \}) < 2d - 1$) — contradicting the optimality of $x$. Therefore $\xi_i \leq 1 - \frac{1}{2d-1}$ follows for $i = 1, \ldots, n$, which means that the order of the edges is $(2 - \frac{1}{2d-1})$-flat. \hfill \Box

Theorem 8. The integrality gap of LP1 is at most $2 - \frac{1}{2d-1}$.

Proof. Let $\theta = 2 - \frac{1}{2d-1}$. By Lemma 7, there exists an $x \in \mathbb{Q}^m$ solution to LP1 and an order of the edges $e_1 = s_1 t_1, \ldots, e_m = s_m t_m$ that is $\theta$-flat with respect to $x$. First, it will be shown that there exist $d$-distance matchings $M_1, \ldots, M_q$ and coefficients $\lambda_1, \ldots, \lambda_q \in \mathbb{R}_+$ s.t. $\sum_{i=1}^{q} \lambda_i \chi_{M_i} = x$ and $\lambda := \sum_{i=1}^{q} \lambda_i \leq \theta$.

Let $K \in \mathbb{N}$ be the lowest common denominator of $\{x_e : e \in E\}$, and let $q = \lfloor K\theta \rfloor$. The main observation is that each edge $e \in E$ can be assigned a set of colors $C_e \subseteq \{1, \ldots, q\}$ s.t. each color class corresponds to a feasible $d$-distance matching and $|C_e| = K x_e$. To prove this, the edges are greedily colored one-by-one in order $e_m, \ldots, e_1$. By induction, assume that edges $e_{m-1}, \ldots, e_{i+1}$ already have their color sets. It suffices to assign a color set $C_{e_i}$ to edge $e_i$, which is of size $K x_{e_i}$ and distinct from both $A := \bigcup \{C_{e_j} : j > i, e_j \in \Delta(s_j)\}$ and $B := \bigcup \{C_{e_j} : j > i, e_j \in \Delta(t_j), s_j \in R_d(s_j) \cup L_d(s_j)\}$. Without loss of generality, assume that $x_{e_i} > 0$ (otherwise $C_{e_i} = \emptyset$). By (9), one gets that $|A \cup B| \leq |A| + |B| = K(\xi_i + \xi_i^t) \leq |K\theta| - K x_{e_i} = q - K x_{e_i}$, thus $|A \cup B| + K x_{e_i} \leq q$. That is, the number of free colors is at least $K x_{e_i}$, so let $C_{e_i}$ be any $K x_{e_i}$ colors in $\{1, \ldots, q\} \setminus (A \cup B)$.

Let the desired $d$-distance matching $M_i$ consist of the edges with color $i$ for $i = 1, \ldots, q$. Set $\lambda_i = \frac{1}{K}$ for all $i = 1, \ldots, q$, and observe that both $\sum_{i=1}^{q} \lambda_i \chi_{M_i} = x$ and $\sum_{i=1}^{q} \lambda_i = \sum_{i=1}^{q} \frac{1}{K} = \frac{q}{K} \leq \theta$ hold.
Figure 5: For \( w \equiv 1 \) and \( d = 5 \), \( x \equiv 1/2 \) is an optimal fractional solution of LP1, and the highlighted edges form an optimal 5-distance matching.

By contradiction, suppose that \( \lambda w(M_i) < w(M^*) \) for each \( i = 1, \ldots, q \), where \( M^* \) is an optimal distance matching. Observe that

\[
 w \sum_{i=1}^{q} \lambda_i \chi_{M_i} = \sum_{i=1}^{q} \lambda_i w(M_i) < \frac{1}{\lambda} w(M^*) \sum_{i=1}^{q} \lambda_i = w(M^*),
\]

that is, the LP optimum is strictly smaller than the IP optimum, which is a contradiction. Therefore, the largest weight \( d \)-distance matchings among \( M_1, \ldots, M_q \) are indeed \( \lambda \)-approximate. Since \( \lambda \leq 2 - \frac{1}{2d-1} \), the proof is complete.

Note that the above approach is algorithmic, but it is not necessarily polynomial. The next section presents a polynomial time method, and reproves that the integrality gap is at most \( 2 - \frac{1}{2d-1} \).

Remark 9. Figure 6 provides an example with the (largest known) integrality gap 6/5. Using this instance, one might easily derive an example (by adding two new nodes \( t_5 \) and \( t_6 \) to \( T \), and two new edges \( s_3t_5, s_6t_6 \) for which no perfect 5-distance matching exists, but there is a fractional perfect 5-distance matching — meaning that the integrality gap of LP2 is unbounded as it was expected due to the complexity of the problem.

In what follows, the integrality gaps of LP1 and LP2 are improved in special cases.

Theorem 10. If \( d = 1 \) or \( d = 2 \), then both LP1 and LP2 are integral.

Proof. Standard argument shows that the matrix of LP1 and LP2 is a network matrix \([3]\) for \( d = 1, 2 \), hence the theorem follows.

Note that the matrix of LP1 and LP2 is not totally unimodular for \( d \geq 3 \) if the input graph is the complete bipartite graph – the technical proof is omitted here. Having said that, the following theorem holds.

Theorem 11. If \( d = |T| \), then LP2 is integral.

Proof. Let \( A \) denote the matrix of LP2, and let \( \tilde{x} \) be an optimal integral solution. If \( \tilde{x} \) is not an optimal LP solution, then there is no complementary dual solution \( y \), therefore - by Farkas’ lemma - there exists \( z \in R^{|E|} \) for which

\[
\begin{align*}
qz &> 0 \quad & (11a) \\
Az &> 0 \quad & (11b) \\
\tilde{x}_e = 1 &\implies z_e \leq 0 & \forall e \in E \quad & (11c) \\
\tilde{x}_e = 0 &\implies z_e \geq 0 & \forall e \in E \quad & (11d)
\end{align*}
\]

Let \( z^j = (z_{s_1t_1}, z_{s_2t_2}, \ldots, z_{s_|S|t_1}) \). Observe that \( z^j = z^j_k \) for all \( j = 1, \ldots, m \) whenever \( i \equiv k \mod m \), which allows the simplification of 11a-11d. By introducing \( \hat{z}^k (k = 1, \ldots, m) \) new variables,
one for each remainder class (i.e. \( \hat{z}^k \) represents all variables \( \{ z^j : j \equiv k \mod n \} \)), one gets the following formulation.

\[
\begin{align*}
\max \quad & \hat{w} \cdot \hat{z} \\
\text{s.t.} \quad & \sum_{i=1}^{|S|} \hat{z}^i = 0 \quad \forall j = 1, \ldots, |T| \quad (12a) \\
& \sum_{j=1}^{|T|} \hat{z}^j = 0 \quad \forall i = 1, \ldots, |S| \quad (12b) \\
& \hat{x}_e = 1 \implies \hat{z}_e \leq 0 \quad \forall e \in E \quad (12c) \\
& \hat{x}_e = 0 \implies \hat{z}_e \geq 0 \quad \forall e \in E \quad (12d) \\
& -1 \leq z \leq 1 \\
\end{align*}
\]

where \( \hat{w}^j = \sum_{i=1}^n w^j_i \). Note that system 11a-11d has a feasible solution if and only if 12a-12e has one with positive objective value. As the optimal value of 12a-12e is finite and its matrix is totally unimodular, there is an integer solution \( \hat{z}^* \) for 12a-12e with positive objective value. This particular solution corresponds to a \( \hat{z}^* \) solution to 11a-11d with the same positive weight. But this means that \( \hat{x} + \hat{z}^* \) is an integer solution of LP2 and \( w \hat{x} < w(\hat{x} + \hat{z}^*) \), contradicting the fact that \( \hat{x} \) was an optimal integer solution.

\[\square\]

### 3.4.2 \((2 - \frac{1}{2\theta - 1})\)-approximation algorithm for the weighted \(d\)-distance matching

This section presents an “almost greedy” LP-based \((2 - \frac{1}{2\theta - 1})\)-approximation algorithm and proves that the integrality gap is at most \( \theta := 2 - \frac{1}{2\theta - 1} \).

**Algorithm 3** \(\theta\)-approximation algorithm for the weighted distance matching problem

1. Let \( e_1, \ldots, e_m \) be a \( \theta \)-flat order with respect to a solution \( x \) of LP1 (see Lemma 7).
2. procedure \textsc{WdmLpApx}(E, w)
3. \[ E := E \setminus \{ e \in E : w_e \leq 0 \} \]
4. if \( E = \emptyset \) then
5. \quad return \( \emptyset \)
6. Let \( st \) be the first edge according to the above order that appears in \( E \).
7. \( M' := \textsc{WdmLpApx}(E \setminus \{ st \}, w') \), where \( w' := w - w_{st} \chi_{\Delta (s) \cup \{ s' \in \Delta (t) : s' \in R_{d}(s) \}} \)
8. if \( M' \cup \{ st \} \) is a feasible \( d \)-distance matching then
9. \quad return \( M' \cup \{ st \} \)
10. else
11. \quad return \( M' \)

**Theorem 12.** Algorithm 3 is a \(\theta\)-approximation algorithm for the weighted \(d\)-distance matching problem if a \(\theta\)-flat order of the edges is given in the first step of the algorithm.

**Proof.** The proof is by induction on the number of edges. Let \( M \) denote the distance matching found by \textsc{WdmLpApx}(E, w), and let \( x \) be as defined in Algorithm 3. In the base case, if \( E = \emptyset \), then \( \theta w(M) \geq w x \) holds. Let \( st \in E \) be the first edge with respect to the order of the edges used by Algorithm 3. By induction, \( \theta w'(M') \geq w' x \) holds for \( M' = \textsc{WdmLpApx}(E \setminus \{ st \}, w') \), where \( w' := w - w_{st} \chi_{\Delta (s) \cup \{ s' \in \Delta (t) : s' \in R_{d}(s) \}} \). The key observation is that

\[
\theta (w - w')(M) \geq \theta w_{st} \geq (w - w') x \quad (13)
\]
follows by the definition of \( w' \) and the order of the edges. Hence, one gets that
\[
\theta w(M) = \theta(w - w')(M) + \theta w'(M) \geq (w - w')x + w'x = wx,
\]
where \( w'(M) = w'(M') \) because \( w_{st} = 0 \). Therefore \( M \) is indeed a \( \theta \)-approximate solution, which completes the proof.

Theorem 12 also implies that the integrality gap of LP1 is at most \( \theta \). Note that if we have a \( \theta' \)-flat order of the edges in the first step of Algorithm 3, then it outputs a \( \theta' \)-approximate solution. Hence, if one proves that there always exist a \( \theta' \)-flat order of the edges for some \( \theta' < \theta \) (improving Lemma 7), then it automatically follows that the integrality gap is at most \( \theta' \).

### 3.5 A combinatorial \( (2 - \frac{1}{4}) \)-approximation algorithm

This section presents a \( (2 - \frac{1}{4}) \)-approximation algorithm for the weighted distance matching problem. Let \( k \in \{d - 1, \ldots, 3d - 3\} \) be such that \( 2d - 1 \) divides \(|S| + k\), and add \( k \) new dummy nodes \( s_{n+1}, \ldots, s_{n+k} \) to the end of \( S \) in this order. Let us consider the extended node set in cyclic order. Observe that the new cyclic problem is equivalent to the original one. Let \( H_j \) denote the graph induced by \( R_d(s_j) \cup T \), where \( R_d(s_j) \) is the set consisting of node \( s_j \) and the next \( d - 1 \) nodes on its right in the new cyclic problem. Let
\[
G_i = (S_i, T; E_i) = \bigcup_{j=0}^{2d-1} H_{i+j(2d-1)}
\]
for \( i = 1, \ldots, 2d - 1 \), where \( S_i \subseteq S \). For each \( i = 1, \ldots, 2d - 1 \), compute a maximum weigh matching \( M_i \) of \( G_i \) and let \( i^* = \arg \max \{w(M_i) : i = 1, \ldots, 2d - 1\} \). For example, consider the graph on Figure 7 with \( d = 3 \). The nodes of \( G_4 \) are highlighted on the figure and the edges of \( M_4 \) are the wavy ones (note that \( s_6, \ldots, s_{10} \) are the five dummy nodes).

**Theorem 13.** \( M_{i^*} \) is a feasible \( d \)-distance matching and it is \( (2 - \frac{1}{4}) \)-approximate.

**Proof.** Each node of \( S \) is covered by at most one edge of \( M_{i^*} \), as \( M_{i^*} \) is the union of matchings no two of which cover the same node of \( S \). If \( s_t, s_j \in M_{i^*} \), then \( s_t \) and \( s_j \) belong to two distinct \( M_k, M_l \) for some \( k, l \), hence \( |i - j| \geq d \) and the feasibility of \( M_{i^*} \) follows.

To show the approximation guarantee, let \( M^* \) be an optimal \( d \)-distance matching. For each node \( s \in S \), let \( \mu_s \in \mathbb{R}_+ \) denote the weigh of the edge covering \( s \) in \( M^* \), and zero if \( M^* \) does not cover \( s \). Note that \( \sum_{s \in S} \mu_s = w(M^*) \) by definition, and
\[
\sum_{s \in S_i} \mu_s \leq w(M_i)
\]
follows because \( \sum_{s \in S_i} \mu_s \) is the weight of a matching in \( G_i \). Observe that
\[
dw(M^*) = d \sum_{s \in S} \mu_s = \sum_{i=1}^{2d-1} \sum_{s \in S_i} \mu_s \leq \sum_{i=1}^{2d-1} w(M_i) \leq (2d - 1)w(M_{i^*}),
\]

Figure 6: For \( w \equiv 1 \) and \( d = 5 \), \( x = 1/2 \) is an optimal solution to LP1, and the highlighted edges form an optimal 5-distance matching, hence the integrality gap is \( 6/5 \).
where the first equation holds because \( \mu_s \) occurs exactly \( d \) times as a summand in \( \sum_{i=1}^{2d-1} \sum_{s \in S} \mu_s \) for all \( s \in S \), the first inequality follows from (15) and the last one holds because \( M^* \) is a largest weight matching among \( M_1, \ldots, M_{2d-1} \). One gets by (16) that \( w(M^*) \leq (2 - \frac{1}{d})w(M_i^*) \), which completes the proof.

The analysis is tight in the sense that, for every \( d \in \mathbb{Z}_+ \), there exists a graph \( G \) for which the algorithm returns a \( d \)-distance matching \( M \) for which \( w(M^*) = (2 - \frac{1}{d})w(M_i^*) \), where \( M^* \) is an optimal \( d \)-distance matching. Let \( S \) and \( T \) consist of \( 2d-1 \) and \( d \) nodes, respectively. Add edge \( s_it_i \) for \( i = 1, \ldots, d \), and edge \( s_{i+d}t_i \) for \( i = 1, \ldots, d-1 \). Note that the edge set is a feasible \( d \)-distance matching itself, and the above algorithm returns a matching that covers exactly \( d \) nodes of \( S \). Hence the approximation ratio of the found solution is \( \frac{2d-1}{d} \). Figure 7 shows the construction for \( d = 3 \), where \( s_6, \ldots, s_{10} \) are the dummy nodes.

4 Unweighted \( d \)-distance matching

First, two refined greedy approaches are considered, then the analysis of the approximation ratio of locally optimal solutions follows.

4.1 Greedy algorithms

This section describes two refined greedy algorithms for the unweighted \( d \)-distance matching problem, and proves that both of them achieve an approximation guarantee of 2.

Algorithm 4: S-Greedy

Let \( s_1, \ldots, s_n \) be the nodes of \( S \) in the given order

\[
M := \emptyset
\]

for \( i = 1, \ldots, n \) do

if \( M \cup \{s_it_i\} \) is feasible for some \( s_it_i \in \Delta(s_i) \) then

\[
\begin{align*}
\quad & j := \arg \min \{j : s_it_j \in \Delta(s_i) \text{ and } M \cup \{s_it_j\} \text{ is feasible} \} \\
\quad & M := M \cup \{s_it_j\}
\end{align*}
\]

output \( M \)

Theorem 14. S-Greedy is a 2-approximation algorithm for the unweighted \( d \)-distance matching problem.

Proof. Assume that S-Greedy returns edges \( f_1, \ldots, f_p \), and it selects them in this order. Let \( M_i \) and \( \theta_i \) be as above in the proof of Theorem 4, i.e. let \( M_i = \arg \max \{w(M) : f_i, \ldots, f_i \in \text{M and } M \text{ is a } d \text{-distance matching} \} \) and let \( \theta_i \) denote the weight of \( M_i \) for \( i = 0, \ldots, p \). Observe that there exists edges \( e, e' \in M_i \setminus \{f_i, \ldots, f_i\} \) s.t. \( (M_i \setminus \{e, e'\}) \cup \{f_i+1\} \) is a feasible \( d \)-distance matching containing edges \( f_1, \ldots, f_{i+1} \). By the greedy selection rule, one gets that

\[
\theta_{i+1} \geq \theta_i + 1 - 1 - 1 = \theta_i - 1
\]

(17)
holds for all \(i = 0, \ldots, p-1\). Straightforward inductive argument shows that (17) implies \(\theta_p \geq \theta_0 - p\), therefore \(2\theta_p \geq \theta_0\) follows, which completes the proof.

The analysis is tight in the sense that S-Greedy might return only one edge, while the largest 2-distance matching consists of 2 edges, see Figure 4b.

Algorithm 5. T-Greedy

Let \(s_1, \ldots, s_n\) be the nodes of \(S\) in the given order

\[
M := \emptyset \quad C := \emptyset
\]

\textbf{for} \(j = 1, \ldots, k\) \textbf{do}

\[
M_j := \emptyset \\
C_j := \emptyset \\
i := 1
\]

\textbf{while} \(i \leq n\) \textbf{do}

\[
\begin{align*}
\text{if } s_it_j &\in E \text{ and } s_i \notin C \\
M_j &:= M_j \cup \{s_it_j\} \\
C_j &:= C_j \cup \{s_i\} \\
i &:= i + d
\end{align*}
\]

\[
\text{else} \\
i &:= i + 1
\]

\[
M := M \cup M_j \\
C := C \cup C_j
\]

\textbf{output } M

Theorem 15. T-Greedy is a 2-approximation algorithm for the unweighted \(d\)-distance matching problem.

Proof. Let \(M_S\) and \(M_T\) denote the edge sets T-Greedy (Algorithm 5) and S-Greedy (Algorithm 4) outputs, respectively. It suffices to prove that \(M_S = M_T\). By contradiction, suppose that \(M_S \neq M_T\). Let \(s_i\) be the first node in \(S\) for which \(\Delta(s_i) \cap M_S \neq \Delta(s_i) \cap M_T\), and choose the edge \(s_it_j \in \Delta(s_i) \cap (M_S\Delta M_T)\) s.t. \(j\) is the smallest possible.

Case 1: \(s_it_j \in M_T \setminus M_S\). First, observe that \(M_T\) covers node \(s_i\), otherwise it would have included \(s_is_t\). Therefore, T-Greedy assigns node \(s_i\) to \(t_j\), where \(j' \neq j\). If \(j' < j\), then S-Greedy would have chosen edge \(s_it_{j'}\) instead of \(s_it_j\). If \(j' > j\), then T-Greedy would have included \(s_it_j\) instead of \(s_it_{j'}\) to \(M_T\).

Case 2: \(s_it_j \in M_T \setminus M_S\). Observe that \(M_S\) covers node \(s_i\), otherwise S-Greedy could have included edge \(s_is_t\). Therefore, S-Greedy assigns node \(s_i\) to \(t_{j'}\), where \(j' \neq j\). Similarly to the argument in Case 1, it is easy to see that neither \(j' < j\) nor \(j' > j\) is possible.

Figure 4b shows that the approximation ratio is tight.

4.2 Local search

This section investigates the approximation ratio of the so-called locally optimal solutions. First, consider the following notion, which plays a central role throughout the section.

Definition 16. Given an edge \(e^* \in E\), let \(\mathcal{H}(e^*, M) \subseteq M\) denote the inclusion-wise minimal subset of \(M\) for which \(M \setminus \mathcal{H}(e^*, M) \cup \{e^*\}\) is feasible \(d\)-distance matching. An edge \(e^*\) hits \(e \in M\) if \(e \in \mathcal{H}(e^*, M)\).

Definition 17. Given an edge set \(X \subseteq E\), let \(\mathcal{H}(X, M) \subseteq M\) denote the set of edges hit by at least one edge in \(X\), i.e. let \(\mathcal{H}(X, M) = \bigcup_{e^* \in X} \mathcal{H}(e^*, M)\).
Definition 18. A $d$-distance matching $M$ is $l$-locally optimal if for each $X \subseteq E \setminus M, |X| \leq l$ there exists no $Y \subseteq M$ s.t. $|Y| < l$ and $(M \setminus Y) \cup X$ is $d$-distance matching. In other words, $M$ is $l$-locally optimal if there exists no $X \subseteq E \setminus M$ s.t. $l \geq |X| > |H(X, M)|$. Similarly, $M$ is $l$-locally optimal with respect to $M^*$ if $M$ is $l$-locally optimal in $G' = (S, T; M \cup M^*)$, where $M^*$ is $l$-locally optimal.

Note that a $d$-distance matching $M$ is 1-locally optimal if and only if there exists a permutation of $E$ s.t. Greedy outputs $M$ for $w \equiv 1$. In what follows, an upper bound $g_l$ is shown on the approximation ratio of $l$-locally optimal solutions for each $l \geq 1$, where $g_l$ is defined by the following recursion.

$$g_l = \begin{cases} 3, & \text{if } l = 1 \\ 2, & \text{if } l = 2 \\ \frac{4g_{l-2} - 3}{2g_{l-2} - 1}, & \text{if } l \geq 3. \end{cases}$$  \hspace{1cm} (18)

For $l = 1, 2, 3, 4$, the statement can be proved by a simple argument, given below. However, this approach does not seem to work in the general case. The proof of the general case, which is much more involved and quite esoteric, is given after the following theorem.

Theorem 19. If $M, M^*$ are $(R, d)$-distance matchings s.t. $M$ is $l$-locally optimal with respect to $M^*$, then the approximation ratio $|M^*|/|M|$ is at most $g_l$, where $l = 1, \ldots, 4$ and $g_l$ is as defined above.

Proof. Let $M_i^* = \{ e^* \in M^* : |H(e^*, M)| = i \}$ for $i = 0, \ldots, 3$. Note that $M_0, M_1, M_2, M_3$ is a partition of $M^*$, and $M_0 = \emptyset$ since each edge of $M^*$ hits at least one edges of $M$ if $l \geq 1$. Since each edge $e \in M$ can be hit by at most three edges of $M^*$, one gets that

$$3|M| \geq \sum_{e^* \in M^*} |H_+(e^*, M)| = |M_1^*| + 2|M_2^*| + 3|M_3^*|. \hspace{1cm} (19)$$

Case $l = 1$.

It easily follows from (19) that

$$|M^*| = |M_1^*| + |M_2^*| + |M_3^*| \leq |M_1^*| + 2|M_2^*| + 3|M_3^*| \leq 3|M|. \hspace{1cm} (20)$$

Case $l = 2$.

$$2|M^*| = 2(|M_1^*| + |M_2^*| + |M_3^*|) \leq |M_1^*| + |M_1^*| + 2|M_2^*| + 3|M_3^*| \leq |M_1^*| + 3|M| \leq 4|M|, \hspace{1cm} (21)$$

where the second inequality follows from (19) and the third one holds because $M$ is 2-locally optimal with respect to $M^*$.

Case $l = 3$.

$$5|M^*| = 5(|M_1^*| + |M_2^*| + |M_3^*|) = 2(|M_1^*| + 2|M_2^*| + 3|M_3^*|) + 3|M_1^*| + |M_2^*| - |M_3^*| \leq 6|M| + 3|M^*_1| + |M^*_2| - |M^*_3| \leq 6|M| + 3|M_1^*| + |M_2^*| \leq 9|M|, \hspace{1cm} (22)$$

where the last inequality holds by the following claim.

Claim 20. $|M_2^*| \leq 3(|M| - |M_1|)$ if $M$ is $3$-locally optimal with respect to $M^*$.

Proof. It suffices to show that there exist $d$-distance matchings $\bar{M}, \bar{M}^*$ s.t. 1) $|\bar{M}| = |M| - |M_1^*|$, 2) $|\bar{M}^*| = |M_2^*|$, and 3) $\bar{M}$ is 1-locally optimal with respect to $\bar{M}^*$. Indeed, $|\bar{M}^*| \leq 3|M|$ holds, from which one obtains the inequality to be proved by substituting 1) and 2).

Let $\bar{M} = M \setminus H(M_1^*, M)$ and $\bar{M}^* = M_2^*$. Clearly, both 1) and 2) hold. By contradiction, suppose that 3) does not hold, that is, there exists $e_1^* \in \bar{M}^*$ s.t. $\bar{M} \cup \{ e_1^* \}$ is feasible $d$-distance matching. By definition, $e_1^* \in M_2^*$, therefore $e_1^*$ hits exactly two edges $e_1, e_2$ in $M$. Neither $e_1$, nor $e_2$ are in $\bar{M}$, thus $e_1, e_2 \in H(M_1^*, M)$, that is $e_1$ is hit by an edge $e_{j+1}^* \in M_2^*$ for $j = 1, 2$. Note that $e_{1}^*, e_{2}^*, e_{3}^*$ are pairwise distinct edges, and $\bar{H}(\{ e_{1}^*, e_{2}^*, e_{3}^* \}, M) = \{ e_1, e_2 \}$, contradicting that $M$ is 3-locally optimal. \qed
Case $l = 4$.

\[6|M^*| = 6(|M_1^*| + |M_2^*| + |M_3^*)| = 2(|M_1^*| + 2|M_2^*| + 3|M_3^*|) + 4|M_1^*| + 2|M_2^*|\]
\[\leq 6|M| + 4|M_1^*| + 2|M_2^*| \leq 10|M|, \quad (23)\]

where the first inequality holds by (19), the last one by the following claim.

**Claim 21.** $2|M_2^*| \leq 4(|M| - |M_1|)$ if $M$ is 4-locally optimal with respect to $M^*$.

**Proof.** It suffices to show that there exist $d$-distance matchings $\tilde{M}, M^*$ s.t. 1) $|\tilde{M}| = |M| - |M_1^*|$, 2) $|M^*| = |M_2^*|$, and 3) $M$ is 2-locally optimal with respect to $M^*$. Indeed, $|M^*| \leq 2|\tilde{M}|$ holds, from which one obtains the inequality to be proved by substituting 1) and 2).

Let $\tilde{M} = M \setminus H(M_1^*, M)$ and $M^* = M_2^*$. Similarly to the proof of Claim 20, one might show that 3) holds, hence the desired inequality follows. \qed

This concludes the proof of the theorem. \qed

It is worth noting that the proof for $l = 3, 4$ refers inductively to the case $l = 2$, which is quite unexpected. The same idea does not seem to work for $l = 5$. Based on cases $l = 1, 2, 3, 4$, one gains the following analogous computation.

\[13|M^*| = 13(|M_1^*| + |M_2^*| + |M_3^*|) = 4(|M_1^*| + 2|M_2^*| + 3|M_3^*|) + 9|M_1^*| + 5|M_2^*| + |M_3^*|\]
\[\leq 12|M| + 9|M_1^*| + 5|M_2^*| + |M_3^*| \leq 21|M|, \quad (24)\]

where the last inequality requires that $5|M_2^*| + |M_3^*| \leq 9(|M| - |M_1^*|)$. However, the latter inequality does not admit a constructive argument similarly to the cases $l = 3, 4$ (see Claim 20 and 21). To overcome this complication, consider the following extended problem setting, which surprisingly does admit a constructive argument.

**Definition 22.** Let $R$ be a set of (parallel) loops on the nodes of $S$. A subset $M \subseteq E \cup R$ is **$(R,d)$-distance matching** if it is the union of a $d$-distance matching and $R$.

Consider the following extension of Definition 16.

**Definition 23.** Given an $(R,d)$-distance matching $M$ and an edge $sv \in (S \times T) \cup R$, let

\[H_+(sv, M) = \begin{cases} H(sv, M \setminus R) \cup \{e \in R : e \text{ is incident to node } s\}, & \text{if } sv \in S \times T \\ sv, & \text{if } sv \in R. \end{cases} \]

In other words, each $st \in E$ hits the edges of $H(st, M)$ and all the loops incident to node $s$, while each loop hits only itself.

**Definition 24.** Given an edge set $X \subseteq E$, let $H_+(X, M) = \bigcup_{e \in X} H(e, M)$.

Using $H_+$, a natural definition of $l$-locally optimality is as follows.

**Definition 25.** An $(R,d)$-distance matching $M$ is **$l$-locally optimal** if there exists no $X \subseteq E \setminus M$ s.t. $l \geq |X| > |H_+(X, M)|$. Similarly, $M$ is **$l$-locally optimal with respect to $M^*$** if there exists no $X \subseteq M^* \setminus M$ s.t. $l \geq |X| > |H_+(X, M)|$, where $M^*$ in an $(R,d)$-distance matching.

Note that each definition reduces to its original counterpart if $R = \emptyset$. Therefore, it suffices to show that $q_l$ is an upper bound on the approximation ratio of $(R,l)$-locally optimal solutions.

**Theorem 26.** If $M, M^*$ are $(R,d)$-distance matchings s.t. $M$ is $l$-locally optimal with respect to $M^*$, then the approximation ratio $|M^*|/|M|$ is at most $q_l$, where $l \geq 1$ and $q_l$ is as defined above.
Proof. Let $M^*_i = \{e^* \in M^* : |\mathcal{H}_+(e^*, M)| = i\}$ for $i \in \mathbb{N}$, and let $M^*_+ = \bigcup_{k=1}^{\infty} M^*_k$. Note that $M_0, M_1, \ldots$ is a partition of $M^*$, for which $R \subseteq M^*_1$ by definition, and $M^*_0 = \emptyset$ since each edge of $M^*$ hits at least one edge of $M$ if $l \geq 1$. Observe that each edge $e \in M$ can be hit by at most three edges of $M^*$, therefore

$$3|M| \geq \sum_{e^* \in M^*} |\mathcal{H}_+(e^*, M)| = \sum_{k=1}^{\infty} k|M^*_k|.$$  \hfill (25)

The proof is by induction on $l$. The argument for $l = 1, 2$ is analogous to that in the proof of Claim 19.

Case 1: $l = 1$.
It easily follows from (25) that

$$|M^*| = \sum_{k=1}^{\infty} |M^*_k| \leq \sum_{k=1}^{\infty} k|M^*_k| \leq 3|M|.$$  \hfill (26)

Case 2: $l = 2$.

$$2|M^*| = 2 \sum_{k=1}^{\infty} |M^*_k| \leq |M^*_1| + \sum_{k=3}^{\infty} k|M^*_k| \leq |M^*_1| + 3|M| \leq 4|M|,$$  \hfill (27)

where the second inequality follows from (25) and the third one holds because $M$ is 2-locally optimal with respect to $M^*$.

Case 3: $l \geq 3$.
First, introduce the notation $\alpha(M, M^*) = \sum_{k=3}^{\infty} (k-2)|M^*_k|$. In the following computation, inequality (25) is forced with an appropriate coefficient so that the rest admits the application of case $l - 2$ (see Claim 27).

$$(2\varrho_{l-2} - 1)|M^*| = (2\varrho_{l-2} - 1) \sum_{k=1}^{\infty} |M^*_k| = (\varrho_{l-2} - 1) \sum_{k=1}^{\infty} k|M^*_k| + \sum_{k=1}^{\infty} ((k-1) - (k - 2)\varrho_{l-2})|M^*_k|$$

$$\leq 3(\varrho_{l-2} - 1)|M| + \sum_{k=1}^{\infty} ((k-1) - (k - 2)\varrho_{l-2})|M^*_k|$$

$$= 3(\varrho_{l-2} - 1)|M| + \varrho_{l-2} |M^*_1| + |M^*_2| + \sum_{k=3}^{\infty} (k-1)|M^*_k| - \varrho_{l-2} \sum_{k=3}^{\infty} (k - 2)|M^*_k|$$

$$= 3(\varrho_{l-2} - 1)|M| + \varrho_{l-2} |M^*_1| + |M^*_2| + \sum_{k=3}^{\infty} |M^*_k| + \alpha(M, M^*) - \varrho_{l-2} \alpha(M, M^*)$$

$$= 3(\varrho_{l-2} - 1)|M| + \varrho_{l-2} |M^*_1| + |M^*_2| + |M^*_3| + \alpha(M, M^*) - \varrho_{l-2} \alpha(M, M^*)$$

$$\leq 3(\varrho_{l-2} - 1)|M| + \varrho_{l-2} |M| = (4\varrho_{l-2} - 3)|M|,$$  \hfill (28)

where the first inequality holds by (25), the second one by the following claim.

Claim 27. If $l \geq 3$ and $M, M^*, \alpha(M, M^*)$ are as above, then

$$|M^*_2| + \alpha(M, M^*) \leq \varrho_{l-2}(|M| - |M^*_1| + \alpha(M, M^*))$$  \hfill (29)

Proof. It suffices to show that if $M$ is $l$-locally optimal with respect to $M^*$, then there exist $\tilde{M}$, $\tilde{M}^*$ and $\tilde{R}$ s.t. 1) $\tilde{M}$ and $\tilde{M}^*$ are $(\tilde{R}, d)$-distance matchings 2) $\tilde{M} = |M| - |M^*_1| + \alpha(M, M^*)$ 3) $|\tilde{M}^*| = |M^*_2| + \alpha(M, M^*)$ 4) $|\tilde{R}| = \alpha(M, M^*)$ and 5) $\tilde{M}$ is $(l-2)$-locally optimal with respect to $\tilde{M}^*$. Indeed, $|\tilde{M}^*| \leq \varrho_{l-2}|M|$ holds by induction, from which one obtains (29) by substituting 2) and 3).
Let \( \tilde{R} = \bigcup_{s^* \in M^*_2} \{|H_+(s^*t^*, M)| - 2 \text{ parallel loops incident to } s\} \), \( \tilde{M} = M \setminus H_+(M^*_1, M) \cup \tilde{R} \) and \( \tilde{M}^* = M^*_2 \cup \tilde{R} \). It is easy to see that \( \tilde{M}, \tilde{M}^* \) and \( \tilde{R} \) fulfills 1)-4). By contradiction, suppose that 5) does not hold, that is, there exists \( Z \subseteq \tilde{M}^* : l - 2 \geq |Z| > |H_+(Z, \tilde{M})| \). Assume that the instance of the problem at hand is minimal in the sense that \( |M| + |M^*| + |\tilde{M}| + |\tilde{M}^*| + |Z| \) is minimal. First, various useful properties of minimal problem instances are derived. Note that \( |Z| = |H_+(Z, M)| + 1 \) follows, otherwise \(|Z| > |H_+(Z, M)| + 1 \) and therefore one could have removed an arbitrary edge from \( Z \).

Observe that if an edge \( e \in H_+(Z, \tilde{M}) \) were hit by a sole edge \( e^* \in Z \), then \( l - 2 \geq |Z \setminus \{ e^* \}| > |H_+(Z \setminus \{ e^* \}, \tilde{M})| \) would hold, i.e. one could have left \( e^* \) from \( Z \). Therefore, each edge \( e \in H_+(Z, \tilde{M}) \) is hit by at least two edges of \( Z \). This also implies that \( s^*t^* \in Z \) if and only if \( \{ e \in \tilde{R} : e \) is incident to \( s^* \} \subseteq Z \). Using this, \( Z = \tilde{M}^* \) follows, because removing all edges \( \tilde{M}^* \setminus Z \) from \( M^* \) and all those loops from \( R \) that are incident to the removed edges, one obtains a smaller instance (where \( \alpha(M, M^*), \tilde{R}, \tilde{M} \) and \( \tilde{M}^* \) need to be adjusted appropriately after the edge-removal), which satisfies 1)-4) but 5) (given the indirect assumption). Clearly, \( M \) remains \( l \)-locally optimal with respect to \( M^* \) after the edge-removal. So, one can assume that \( Z = \tilde{M}^* \).

A minimal instance also fulfills that there exists no edges \( e \in M \) and \( e^* \in M^*_1 \) s.t. \( e \in H_+(e^*, M) \setminus H_+(M^*_1, M) \), otherwise the removal of \( e \) and \( e^* \) results in a smaller instance satisfying 1)-4) but 5) (given the indirect assumption). Observe that after the removal, \( M \) remains \( l \)-locally optimal with respect to \( M^* \), because there exists no \( X \subseteq M^* \setminus \{ e^* \} \) s.t. \( e \in H_+(X, M) \) (since \( e \in H_+(e^*, M) \setminus H_+(M^*_2, M) \)), therefore if the new instance were not \( l \)-locally optimal, then the original instance would not have been either. So, one can assume that \( H_+(M^*_1, M) \setminus H_+(M^*_2, M) = \emptyset \).

On the one hand,

\[
|H_+(M^*, M)| = |H_+(M^*_1, M)| + |H_+(M^*_2, M) \setminus H_+(M^*_1, M)| + |H_+(M^*_1, M) \setminus H_+(M^*, M)|
\]

\[
= |H_+(M^*_1, M)| + |H_+(M^*_2, M) \setminus H_+(M^*_1, M)| = |M^*_1| + |H_+(M^*_2, M) \setminus H_+(M^*_1, M)|
\]

\[
= |M^*_1| + |H_+(Z, M) \setminus \tilde{R}| = |M^*_1| + |Z| - 1 - |\tilde{R}| = |M^*_1| + |M^*| - 1 - |\tilde{R}|
\]

\[
= |M^*_1| + |M^*_2| + |\tilde{R}| - 1 - |\tilde{R}| = |M^*| - 1, \quad (30)
\]

on the other hand

\[
|M^*| = |M^*_2| + |M^*_1| = |M^*_2| + |H_+(M^*_1, M)| \leq |M^*_2| + \sum_{e^* \in M^*_2} \big|H_+(e^*, M) \cap H_+(M^*_1, M)\big|
\]

\[
\leq |M^*_2| + \sum_{e^* \in M^*_2} \big|H_+(e^*, M) \setminus H_+(M^*_1, M)\big|
\]

\[
= |M^*_2| + \sum_{e^* \in M^*_2} \big|H_+(e^*, M) - 2|H_+(M^*_2, M) \setminus H_+(M^*_1, M)|\big|
\]

\[
= |M^*_2| + \sum_{e^* \in M^*_2} \big|H_+(e^*, M) - 2|H_+(M^*_2, M) \setminus H_+(M^*_1, M)|\big|
\]

\[
= |M^*_2| + 2|M^*_2| + |\tilde{R}| - 2(|M^*_2| - 1) = |M^*_2| + |\tilde{R}| + 2 = |M^*| + 2 = |Z| + 2 \leq l, \quad (31)
\]

where the second inequality holds by the following computation.

\[
2|H_+(M^*_2, M) \setminus H_+(M^*_1, M)| = 2|H_+(M^*_2, M) \setminus H_+(M^*_1, M)| = 2|H_+(M^*_2, \tilde{M} \setminus \tilde{R})|
\]

\[
= 2|H_+M^*, \tilde{M} \setminus \tilde{R})| = 2|H_+(Z, \tilde{M} \setminus \tilde{R})| \leq \sum_{e^* \in Z} |H_+(e^*, \tilde{M} \setminus \tilde{R})|
\]

\[
= \sum_{e^* \in M^*_2} |H_+(e^*, M \setminus H_+(M^*_1, M))| = \sum_{e^* \in M^*_2} |H_+(e^*, M) \setminus H_+(M^*_1, M)|, \quad (32)
\]
where the inequality holds because each edge of $H_+(Z,\tilde{M})$ is hit at least twice by $Z$. Combining (30) and (31), one obtains that $|H_+(M^*, M)| < |M^*| \leq l$, which contradicts that $M$ is $l$-locally optimal with respect to $M^*$, and proves the claim.

By Claim 27, inequality (28) follows, meaning that the desired recursion (18) gives a valid upper bound on the approximation ratio of the $l$-locally optimal solutions.

**Corollary 28.** The approximation ratio of $l$-locally optimal $d$-distance matchings is at most $\varrho_l$, where $\varrho_l$ is as defined above.

**Proof.** Let $M^*$ denote an optimal $d$-distance matching. By definition, $M$ is $l$-locally optimal with respect to $M^*$, therefore $M$ is $(\emptyset, l)$-locally optimal with respect to $M^*$. By Theorem 26, one gets that $|M^*|/|M| \leq \varrho_l$, which completes the proof.

**Corollary 29.** For any constant $\epsilon > 0$, there exist a polynomial algorithm for the unweighted $d$-distance matching problem that achieves an approximation guarantee of $3/2 + \epsilon$.

**Proof.** By Corollary 28, the approximation ratio of $l$-locally optimal solutions is at most $\varrho_l$. One might easily show that $\lim_{l \to \infty} \varrho_l = 3/2$. Hence for any $\epsilon > 0$, there exists $l_0 \in \mathbb{N}$ s.t. $\varrho_l \leq 3/2 + \epsilon$. To complete the proof, observe that $l_0$ is independent from the problem size, therefore one can compute an $l_0$-locally optimal solution in polynomial time.

**Remark 30.** Figure 4a, 8 and 9 show that the upper bound on the approximation ratio of $l$-locally optimal solutions given by Theorem 26 is tight for $l = 1, 2$ and $3$, respectively. It remains open whether the analysis is tight for $l \geq 4$.

**Remark 31.** Similar proof shows that for any constant $\epsilon > 0$, the above local-search algorithm is a $(3/2 + \epsilon)$-approximation algorithm for the unweighted cyclic $d$-distance matching problem.
5 Regular distance matching

The following theorem is a straightforward generalization of the well known result that every regular bipartite graph has a perfect matching.

**Definition 32.** An instance of the d-distance matching problem is r-regular if \( \deg(s) = r \) for each \( s \in S \) and the number of edges between \( t \) and \( R_d(s_i) \) is \( r \) for each \( t \in T \) and \( s_i = \{s_1, \ldots, s_{n-d+1}\} \).

**Theorem 33.** If a problem instance is r-regular, then there exists a perfect d-distance matching.

**Proof.** There exists a perfect matching between \( \{s_1, \ldots, s_d\} \) and \( T \), because the induced graph is r-regular. By induction, assume that \( M \) already covers \( \{s_1, \ldots, s_{i-1}\} \), where \( i - 1 \geq d \). Let \( t \) denote the node that \( M \) maps to \( s_{i-d+1} \). If \( s_t \not\in E \), then the number of edges between \( t \) and \( L_d(s_i) = r - 1 \), meaning that the instance at hand is not r-regular, hence \( s_t \in E \). Therefore \( M \cup \{s_t\} \) is feasible for the first \( i \) nodes of \( S \), hence the claim follows.

6 Optimal permutation of \( S \)

This section investigates a slightly different problem, motivated by the second application presented in Section 1. It is natural to ask whether we can find a permutation of \( S \) (i.e. the items on the conveyor belt, see Section 1) that maximizes the weight of the maximum weight d-distance matching. The proof of the next theorem provides a polynomial time algorithm to solve this problem. We say that a triple \( y \in \mathbb{N}^{S \cup T}, z \in \mathbb{N}^{E}, v \in \mathbb{N}^T \) is a \( u \)-cover of \( G = (S, T; E) \) for \( u \in \mathbb{N} \), if \( y_s + z_t + v_t \geq u_t \) for all \( s \in E \) and \( v_t + y_t \geq u \) for all \( t \in T \).

**Theorem 34.** The maximum weight of a d-distance matching under all permutations is equal to the minimum of \( \{yb + zv + d(\frac{n}{d})u : y \in \mathbb{N}^{S \cup T}, z \in \mathbb{N}^E, v \in \mathbb{N}^T \} \), where \( b \in \mathbb{N}^{S \cup T} \) is such that \( b_s = 1 \) for \( s \in S \) and \( b_t = \lceil n/d \rceil \) for \( t \in T \).

**Proof.** Let \( n = |S| \) and let \( M \subseteq E \) be a maximum weight edge set s.t. \( \deg_M(s) \leq 1 \) for all \( s \in S \), \( \deg_M(t) \leq \lceil n/d \rceil \) and the number of nodes \( t \in T \) for which \( \deg_M(t) = \lceil n/d \rceil \) is at most \( n - \lfloor n/d \rfloor d \). Such an edge set \( M \) can be found in polynomial time by a reduction to the maximum cost circulation problem.

It is easy to see that \( w(M) \geq W \). To show that \( w(M) = W \), it suffices to construct a permutation of \( S \) under which \( M \) is a d-distance matching. Let \( S_1, \ldots, S_{k+1} \) be a partition of \( S \) s.t. \( k = \lfloor n/d \rfloor \), \( |S_i| = d \) for \( i = 1, \ldots, k \) and \( M \) induces a (not necessarily perfect) matching between \( T \) and \( S_{k+1} \) covering each node \( t \in T \) that has degree \( \lfloor n/d \rfloor + 1 \). Note that \( |S_{k+1}| = n - \lfloor n/d \rfloor d < d \).

Let \( \alpha \) denote the number of edge pairs \( s, t \in M \) s.t. \( s, s' \in S_i \) for some \( i = 1, \ldots, k + 1 \). If \( \alpha = 0 \), then \( M \) is a d-distance matching with respect to the order given by the concatenation of \( S_1, \ldots, S_{k+1} \) if the nodes of each \( S_i \) are in appropriate order. Otherwise, let \( i \) be an index for which there exists \( s, s' \in M \) s.t. \( s, s' \in S_i \) and \( |N_M(S_i)| \) is as small as possible (\( \alpha > 0 \) implies that at least one such index exists). There exists index \( j \in 1, \ldots, k \) s.t. \( N_M(t) \cap S_j = \emptyset \), and one can easily show that there is a node \( s'' \in S_j \) for which \( N_M(s'') \not\subseteq N_M(S_i) \) or \( N_M(s'') = \emptyset \). By setting \( S_1 = S_1 + s'' - s \) and \( S_j = S_j + s - s'' \), \( \alpha \) decreases by one, hence the algorithm terminates in polynomial time. One can easily derive the min-max formula using LP-duality or the max-flow min-cut theorem.

**Remark 35.** A similar approach solves the analogue problem for the perfect d-distance matching problem. In this case, one should look for an edge set \( M \) for which \( \deg_M(s) = 1 \) (instead of \( \deg_M(s) \leq 1 \)) and repeat the proof of Theorem 34.
7 Conclusion

This paper introduced the $d$-distance matching problem. We proved that the problem is NP-complete in general and admits a 3-approximation. We gave an FPT algorithm parameterized by $d$ and also settled the case when the size of $T$ is constant. The integrality gap of the natural integer programming model is shown to be at most $2 - \frac{1}{d+1}$, and gave an LP-based approximation algorithm for the weighted case with the same guarantee. Using alternative approach, we described a combinatorial $(2 - \frac{1}{d})$-approximation algorithm. Several greedy approaches, including a local search algorithm, were presented. The latter method achieves an approximation ratio of $3/2 + \epsilon$ for any constant $\epsilon > 0$ in the unweighted case. We also gave an algorithm to find a permutation that maximizes the weight of the optimal distance matching.

The novel approaches used in the analysis of the integrality gap and the approximation ratio of locally optimal solutions might be of independent combinatorial interest. The problem itself has various generalizations (degree bound on the nodes, cyclic version of the problem, etc.), which are subjects for further research.

Acknowledgement

The project has been supported by the ÚNKP-19-3 New National Excellence Program of the Ministry for Innovation and Technology. The author is grateful to Kristóf Bérczi and Alpár Jüttner for their valuable comments that greatly improved the manuscript. The author is indebted to an anonymous reviewer of an earlier version of the manuscript for suggesting an approach which led to the algorithm described in Section 3.5.

References

[1] Karp, R.M.: Reducibility among Combinatorial Problems, Complexity of Computer Computations (1972) p85-103.
[2] Garey, M.R., Johnson, D. S.: Computers and Intractability: A Guide to the Theory of NP-Completeness (Freeman, San Francisco, 1979).
[3] Frank, A.: Connections in Combinatorial Optimization, Oxford University Press (2011).
[4] Downey, R.G., Fellows, M.R.: Parameterized Complexity, Springer Publishing Company, Incorporated (2012).
[5] Zeitlhofer, T., Wess B.: List-coloring of interval graphs with application to register assignment for heterogeneous register-set architectures, Signal Processing, (2003) 83/7, p1411-1425.
[6] Aardal, K.I., van Hoesel, S.P.M., Koster, A.M.C.A. et al.: Models and solution techniques for frequency assignment problems, Annals of Operations Research (2007) 153, p79-129.