Constant-Factor Approximation Algorithms for the Parity-Constrained Facility Location Problem

Kangsan Kim[1]
Devsisters Corp., Seoul, South Korea
kangsankim@devsisters.com

Yongho Shin
Department of Computer Science, Yonsei University, Seoul, South Korea
yshin@yonsei.ac.kr

Hyung-Chan An[2]
Department of Computer Science, Yonsei University, Seoul, South Korea
hyung-chan.an@yonsei.ac.kr

Abstract

Facility location is a prominent optimization problem that has inspired a large quantity of both theoretical and practical studies in combinatorial optimization. Although the problem has been investigated under various settings reflecting typical structures within the optimization problems of practical interest, little is known on how the problem behaves in conjunction with parity constraints. This shortfall of understanding was rather discouraging when we consider the central role of *parity* in the field of combinatorics.

In this paper, we present the first constant-factor approximation algorithm for the facility location problem with parity constraints. We are given as the input a metric on a set of *facilities* and *clients*, the opening cost of each facility, and the *parity requirement* – odd, even, or unconstrained – of every facility in this problem. The objective is to open a subset of facilities and assign every client to an open facility so as to minimize the sum of the total opening costs and the assignment distances, but subject to the condition that the number of clients assigned to each open facility must have the same parity as its requirement.

Although the unconstrained facility location problem as a relaxation for this parity-constrained generalization has unbounded gap, we demonstrate that it yields a structured solution whose parity violation can be corrected at small cost. This correction is prescribed by a T-join on an auxiliary graph constructed by the algorithm. This auxiliary graph does not satisfy the triangle inequality, but we show that a carefully chosen set of shortcutting operations leads to a cheap and *sparse* T-join. Finally, we bound the correction cost by exhibiting a combinatorial multi-step construction of an upper bound.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Theory of computation → Facility location and clustering

Keywords and phrases Facility location problems, approximation algorithms, clustering problems, parity constraints

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2020.21

Funding This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1C1C1008934). This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2016R1C1B1012910). This research was supported by the Yonsei University Research Fund of 2018-22-0093.

Acknowledgements We thank the anonymous reviewers of this paper for their helpful comments.

[1] Part of this research was conducted while K. Kim was at Yonsei University.
[2] Corresponding author. Department of Computer Science, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul 03722, South Korea.

31st International Symposium on Algorithms and Computation (ISAAC 2020).
Editors: Yixin Cao, Siu-Wing Cheng, and Minming Li; Article No. 21; pp. 21:1–21:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Parity plays a central role in a myriad of topics in combinatorics. This is quite natural that one would not need examples; yet, as a short sample of previous works, Schrijver and Seymour [34] for example studied the packing of odd paths, Everett et al. [13] and Kamiński & Nishimura [21] considered induced path parities in connection with the theory of perfect graphs, and Kakimura et al. [20] studied the packing of parity-constrained cycles intersecting a given vertex set. Naturally, there also exists a large volume of previous research that incorporates parity constraints into different combinatorial optimization problems. Submodular function minimization [15, 16], the minimum cut problem [31, 7], the shortest path problem (cf. [17]), and the connected subgraph problem [35, 9] are all examples of such problems. However, introducing parity constraints to combinatorial optimization problems usually results in a significant level of added complexity in their algorithms, and perhaps due to this, not all parity-constrained combinatorial optimization problems are as well studied as one would expect from the centrality of parity in this field.

The *facility location problem* is one of the prominent optimization problems that has guided a large volume of studies in both computer science and operations research (see, e.g., [5, 23, 27, 37]). In this problem, we are given as the input a set of *facilities* and a set of *clients*, along with the opening cost of each facility and the metric distance between every pair of facility and client. The goal of the problem is to choose a subset of facilities to *open* and a clustering that *assigns* every client to an open facility, so as to minimize the sum of the facility opening costs and the distance between each client and the facility it is assigned to. While the facility location problem also served as a test bed on which a variety of algorithmic theories were developed, another primary reason the problem has attracted the interests of many researchers is that it closely reflects the structure of optimization problems witnessed in practice. Precisely for this reason, facility location problems are studied in a wide variety of settings that better reflect typical constraints imposed on the problem, including the capacitated version [23, 22, 32, 6, 4] that places an upper bound on the number of clients assigned to an open facility, online and/or dynamic variants [10, 14, 12, 24], mobile facility location [3], planar versions [28], fusion with network design problems [29, 30, 1], and the lower-bounded version that imposes a lower bound on the number of clients assigned to an open facility [26]. Yet, in conjunction with parity constraints, it was not previously known how this problem behaves on the other hand.

This paper aims at filling this gap. In the *O-facility location problem*, a subset of facilities O is specified as part of the input in addition to the usual input for the unconstrained facility location problem. The goal of the problem is still to find a minimum-cost subset of open facilities with a clustering of the clients, but now we also need to ensure an additional constraint that the number of clients assigned to each open facility i must be odd if $i \in O$, and must be even if $i \notin O$. Note that this version of the problem definition is equivalent to a version in which we allow three types of parity constraints: unconstrained in addition to odd and even. (See Appendix A for a simple equivalence argument.) It is regrettable that we do not have a proper understanding of this generalized version to this date, especially when we consider its practical relevance. In many problems that seek an optimal clustering, we sometimes have a strong preference for either parity of the cluster sizes. For example, Ahamad and Ammar [2] demonstrate that the performance of a *distributed database system* (DDBS), measured by success rates and mean response times, depends on the parity of the number of storage sites. In fact, this preferred parity is determined as a function of the server failure rates and the ratio between the number of read and write transactions. Thus, the task of clustering a given set of storage sites into multiple instances of distributed database systems, which host different applications whose parameters vary, can be formulated as an O-facility location problem. Preference on a particular parity can also be witnessed in other distributed system design settings (see, e.g., [38]).

In this paper, we present the first constant-factor approximation algorithms for the parity-constrained facility location problem: first for the special case where every facility is even-constrained, and then for the general case. Let ρ_{FL} be the approximation ratio of an algorithm for the unconstrained facility location problem.

▶ **Theorem 1.** *There exists a randomized $2\rho_{\mathsf{FL}}$-approximation algorithm for the $\emptyset$-facility location problem.*

▶ **Theorem 2.** *There exists a $(3\rho_{\mathsf{FL}} + 2)$-approximation algorithm*[3] *for the O-facility location problem for arbitrary O.*

The difficulty of the classic unconstrained facility location problem lies in the fact that it is a "joint optimization" problem: it is trivial to find an optimal assignment when the set of open facilities is given, but the simultaneous optimization of the choice of open facilities along with the assignment makes the problem difficult and, in fact, NP-hard. The O-facility location problem is a generalization of the unconstrained facility location problem and therefore inherits this difficulty. Moreover, even when the set of open facilities is given, it is not as trivial to find an optimal assignment for this problem (although polynomial-time solvable).

In order to obtain good approximation algorithms for the unconstrained facility location problem, many algorithmic tools have been used. In particular, linear programming (LP) relaxations and methods based on them have been successful [36, 19, 18, 8, 25]. Unfortunately, however, the O-facility location problem does not appear directly amenable to LP-based techniques, and it is easy to show that the standard LP relaxation devised in the context of the unconstrained problem has an unbounded integrality gap, i.e., the LP optimum can be away from the true optimum by an arbitrarily large factor. In fact, even the integral optimum to the unconstrained instance obtained by dropping the parity constraints can be arbitrarily away from the true optimum.[4] Despite this gap, we will prove that the approximation algorithms for the unconstrained facility location problem can serve as a useful subroutine of an approximation algorithm for the parity-constrained generalization.

In Section 3, we present a "warm-up" approximation algorithm for the *all-even* case, i.e., $O = \emptyset$. The algorithm begins with finding a minimum perfect matching on the set of clients. Using the fact that every facility is assigned an even number of clients in an optimal solution, we can "shortcut" the optimal solution into a perfect matching, bounding the minimum cost of a perfect matching. We then reduce the given problem to an instance of the unconstrained facility location problem by designating one of the two matched clients as the representative, at the cost of a constant multiplicative factor in the approximation ratio.

This clean approach, however, crucially relies on the fact that every facility is even-constrained, and does not extend to the general case. This necessitates a different approach, which is presented in Section 4. The first step of our algorithm for the general case is to drop the parity constraints and solve this instance using an algorithm for the unconstrained

---

[3] A (somewhat loose) calculation of the approximation ratio gives 6.464.

[4] Consider an instance with two pairs of an even-constrained facility and a client, where the distance within each pair is zero and one across. Both opening costs are zeroes.

problem.[5] The unconstrained optimum is a lower bound on the true optimum, but as was noted earlier, it may be arbitrarily smaller. We show that the parity violation of the unconstrained *initial solution* can be repaired by performing a set of three types of operations: reassigning one client from an open facility to another, opening a new facility, and closing down an open facility after reassigning all its clients to another facility. It is easy to show that all three types of these operations are essential for repairing parity violation. Among them, the last type of operation is particularly costly as it involves the reassignment of multiple clients; however, we observe that it suffices to permit the third type of operation only when the facility is odd-constrained, and this allows such a set of operations to be encoded as a *sparse* T-join on an auxiliary graph (that unfortunately does not satisfy the triangle inequality). The auxiliary graph has three types of edges corresponding to the three types of operations. The key step of the analysis is then to show that the minimum cost of a T-join on the auxiliary graph is bounded by a combination of the initial solution (despite the gap) and an optimal solution.

Future Directions. One of the interesting questions that follow this paper is whether we can write an algorithmically useful LP relaxation for the O-facility location problem. As we could use an LP-based approximation algorithm for the unconstrained facility location problem as a subroutine of our algorithm, one could say that our algorithm can technically be an LP-based algorithm; this, of course, is not a satisfactory answer. Rather than having to solve an LP relaxation which itself is parameterized by a rounded integral solution to another relaxation (which is the case for our algorithm), it would be interesting to have a single relaxation that can be separated in polynomial time and solved to obtain an $O(1)$-approximate lower bound on the optimum. Recall that, for the minimum-cost T-join problem, an exact and polynomial-time separable relaxation exists [11].

Another intriguing future direction is in introducing parity constraints to further combinatorial optimization problems. A natural first target could be the k-median problem. As we noted earlier, there remain many parity-constrained optimization problems yet to be studied. We envision that a further understanding of our algorithm, particularly if we can positively answer our first open question, may lead to extending our knowledge to other parity-constrained optimization problems.

2 Preliminaries

Problem Definition. As the input of the *O-facility location problem*, we are given a set of *facilities* F, a set of *clients* D, *opening costs* $f : F \to \mathbb{Q}_{\geq 0}$, *assignment costs*[6] $c : \binom{F \cup D}{2} \to \mathbb{Q}_{\geq 0}$ satisfying the triangle inequality, and a set of facilities $O \subseteq F$ that, if open, are required to be assigned odd number of clients.

A feasible solution to the problem is given by a set of *open* facilities $S \subseteq F$ and an *assignment* of clients $\sigma : D \to S$ to the open facilities. In order for (S, σ) to be a feasible solution, it must satisfy the parity constraints: for all $i \in S \cap O$, $|\sigma^{-1}(i)|$ must be odd; for all $i \in S \cap \overline{O}$, $|\sigma^{-1}(i)|$ must be even. The objective is to find a feasible solution that minimizes the total solution cost, defined as $\sum_{i \in S} f(i) + \sum_{j \in D} c(\sigma(j), j)$.

[5] The analysis in Section 4 treats the algorithm for the unconstrained problem as a black box; using a bi-factor approximation algorithm will lead to a further improvement in the approximation ratio.

[6] Assignment costs are sometimes defined only between facilities and clients. In this "bipartite" case, the domain of c will be defined as $F \times D$ instead. These two definitions, however, are equivalent, since we can deduce inter-facility (and inter-client) distances by computing the metric closure of the given "bipartite" assignment cost.

An Equivalent Problem Definition. Alternatively, we can define our problem as taking a parity constraint function $\pi : F \to \{\mathsf{odd}, \mathsf{even}, \mathsf{unconstrained}\}$ instead of O. In this case, the parity constraint is redefined as follows: for each $i \in S$, $|\sigma^{-1}(i)|$ must be odd if $\pi(i) = \mathsf{odd}$ and even if $\pi(i) = \mathsf{even}$. If $\pi(i) = \mathsf{unconstrained}$, we do not impose any parity constraints on $|\sigma^{-1}(i)|$.

▶ **Observation 3.** *The two problem definitions are equivalent.*

Its proof is deferred to Appendix A. Note that this observation also shows the NP-hardness of the O-facility location problem.

Notation. To simplify the presentation, we introduce the following shorthands: for any $S \subseteq F$, let $f(S) := \sum_{i \in S} f(i)$, and for any $\sigma : D \to S$, let $c(\sigma) := \sum_{j \in D} c(\sigma(j), j)$. Using this notation, the objective function of the problem can be rewritten as $f(S) + c(\sigma)$. For $D' \subseteq D$ and $\sigma : D \to S$, let $\sigma|_{D'} : D' \to S$ denote the restriction of σ to D' as the new domain, i.e., $\sigma|_{D'}(j) = \sigma(j)$ for all $j \in D'$. Accordingly, $c(\sigma|_{D'})$ is defined as $c(\sigma|_{D'}) := \sum_{j \in D'} c(\sigma|_{D'}(j), j) = \sum_{j \in D'} c(\sigma(j), j)$. Let $\sigma^{-1} : S \to D$ denote the inverse function of σ, i.e., $\sigma^{-1}(i) := \{j \in D \mid \sigma(j) = i\}$. We will slightly abuse the notation by letting $\sigma^{-1}(i) := \emptyset$ for $i \in F \setminus S$.

Additional Definitions. Let $G = (V, E)$ be a graph. For $T \subseteq V$, we say $J \subseteq E$ is a *T-join* if, for every vertex $v \in V$, the number of edges in J that are incident with v is odd if and only if $v \in T$. Given a weighted graph, the minimum-cost T-join can be found in polynomial time [11] (see also [33]).

Given $T \subseteq V$, we say $U \subseteq V$ is *T-odd* if $|U \cap T|$ is odd, and $Y \subseteq E$ is a *T-join dominator* if, for every T-odd set $U \subseteq V$, there exists at least one edge in Y that has exactly one endpoint in U. Equivalently, $Y \subseteq E$ is a T-join dominator if and only if every connected component of the graph (V, Y) contains an even number of vertices from T.

▶ **Lemma 4** ([11, 33])**.** *Given a weighted graph $G = (V, E)$ with $T \subseteq V$ and a T-join dominator Y, the minimum cost of a T-join on G is no greater than the cost of Y.*

Given two sets P and Q, let $P \triangle Q$ denote the symmetric difference of the sets, i.e., $P \triangle Q := (P \setminus Q) \cup (Q \setminus P)$. Finally, let $\mathcal{A}_{\mathsf{FL}}$ denote a ρ_{FL}-approximation algorithm for the *unconstrained* facility location problem.

3 All-Even Case

In this section, we present a constant-factor approximation algorithm for a special case of the problem where the parity constraint of every facility is even, i.e., $O = \emptyset$. We assume that $|D|$ is even in what follows; otherwise, the instance is infeasible.

Our Algorithm. We first find a minimum-cost perfect matching $M^\star$ on D, using c as the cost function. For each $e \in M^\star$, we independently choose one of the two endpoints of e uniformly at random. Let j_e be the chosen client and $\widehat{j_e}$ be the remaining one. Let D' be the set of chosen clients, i.e., $D' = \{j_e \mid e \in M^\star\}$. We now construct an unconstrained facility location instance where the client set is replaced with D'. The rest of the input (F, c, and f) remains the same. We execute $\mathcal{A}_{\mathsf{FL}}$ on this instance; let $S_{\mathcal{I}} \subseteq F$ and $\sigma_{\mathcal{I}} : D' \to S_{\mathcal{I}}$ denote the solution returned by $\mathcal{A}_{\mathsf{FL}}$. We construct a solution S_{ALG} and $\sigma_{\mathsf{ALG}} : D \to S_{\mathsf{ALG}}$ to our problem as follows: we choose S_{ALG} simply as $S_{\mathcal{I}}$. For each remaining client $\widehat{j_e} \in D \setminus D'$, we assign $\widehat{j_e}$ to the same facility to which its pair is assigned, i.e.,

$$\sigma_{\mathsf{ALG}}(j) = \begin{cases} \sigma_{\mathcal{I}}(j), & \text{if } j \in D', \\ \sigma_{\mathcal{I}}(j_e), & \text{if } j = \widehat{j_e} \text{ for some } e \in M^\star. \end{cases}$$

Analysis. Now we show that this algorithm is a randomized $2\rho_{\mathsf{FL}}$-approximation algorithm for the problem. It is easy to see that our algorithm returns a feasible solution since every facility is assigned exactly twice the number of the clients it is assigned in $\sigma_{\mathcal{I}}$. Fix an arbitrary optimal solution to the original problem, and let $S_{\mathcal{O}} \subseteq F$ and $\sigma_{\mathcal{O}} : D \to S_{\mathcal{O}}$ denote this solution.

▶ **Lemma 5.** *There exists a matching M whose cost is no greater than $c(\sigma_{\mathcal{O}})$.*

Proof. Let i be a facility in $S_{\mathcal{O}}$. Observe that i is assigned an even number of clients in the optimal solution: $|\sigma_{\mathcal{O}}^{-1}(i)|$ is even. We can thus find a matching M_i on $\sigma_{\mathcal{O}}^{-1}(i)$ by arbitrarily pairing them, and the cost of M_i is at most $\sum_{j\in\sigma_{\mathcal{O}}^{-1}(i)} c(i,j)$ since, for every $(j_1, j_2) \in M_i$, $c(j_1, j_2) \leq c(i, j_1) + c(i, j_2)$. Choose M as the union of M_i for all $i \in S_{\mathcal{O}}$. The lemma now follows from the fact that $\{\sigma_{\mathcal{O}}^{-1}(i)\}_{i\in S_{\mathcal{O}}}$ form a partition of D. ◀

▶ **Lemma 6.** $\mathbb{E}\left[c(\sigma_{\mathcal{I}}) + f(S_{\mathcal{I}})\right] \leq \rho_{\mathsf{FL}} \cdot \left[\frac{c(\sigma_{\mathcal{O}})}{2} + f(S_{\mathcal{O}})\right]$.

Proof. Observe that $(S_{\mathcal{O}}, \sigma_{\mathcal{O}}|_{D'})$ is a feasible solution to the unconstrained classic facility location instance. Let $(S_{\mathcal{O}'}, \sigma_{\mathcal{O}'})$ be an optimal solution to this instance, i.e., the unconstrained problem where the client set is replaced with D'. Then, we have

$$\mathbb{E}\left[c(\sigma_{\mathcal{O}'}) + f(S_{\mathcal{O}'})\right] \leq \mathbb{E}\left[c(\sigma_{\mathcal{O}}|_{D'}) + f(S_{\mathcal{O}})\right].$$

Since we constructed a perfect matching and chose one of the two endpoints of each edge in the matching uniformly at random, the marginal probability that a client is in D' is exactly $\frac{1}{2}$; thus we have

$$\mathbb{E}\left[c(\sigma_{\mathcal{O}}|_{D'}) + f(S_{\mathcal{O}})\right] = \sum_{j\in D} c(\sigma_{\mathcal{O}}(j), j) \cdot \Pr\left[j \in D'\right] + f(S_{\mathcal{O}}) = \frac{c(\sigma_{\mathcal{O}})}{2} + f(S_{\mathcal{O}}),$$

where the first equality follows from the linearity of expectation. The desired conclusion follows from the fact that $\mathcal{A}_{\mathsf{FL}}$ is a ρ_{FL}-approximation algorithm for the unconstrained facility location problem. ◀

▶ **Lemma 7.** $\mathbb{E}\left[c(\sigma_{\mathsf{ALG}}) + f(S_{\mathsf{ALG}})\right] \leq (\rho_{\mathsf{FL}} + 1) \cdot c(\sigma_{\mathcal{O}}) + 2\rho_{\mathsf{FL}} \cdot f(S_{\mathcal{O}})$.

Proof. Observe that we have $c(\sigma_{\mathsf{ALG}}(\widehat{j_e}), \widehat{j_e}) \leq c(\sigma_{\mathcal{I}}(j_e), j_e) + c(j_e, \widehat{j_e})$ for every client $\widehat{j_e} \in D \setminus D'$ from the triangle inequality, yielding

$$c(\sigma_{\mathsf{ALG}}|_{D\setminus D'}) \leq c(\sigma_{\mathcal{I}}) + c(M^\star)$$

since $M^\star$ is a perfect matching. We thus have

$$\begin{aligned}\mathbb{E}\left[c(\sigma_{\mathsf{ALG}}) + f(S_{\mathsf{ALG}})\right] &= \mathbb{E}\left[c(\sigma_{\mathsf{ALG}}|_{D'}) + c(\sigma_{\mathsf{ALG}}|_{D\setminus D'}) + f(S_{\mathsf{ALG}})\right] \\ &\leq \mathbb{E}\left[c(\sigma_{\mathcal{I}}) + c(\sigma_{\mathcal{I}}) + c(M^\star) + f(S_{\mathcal{I}})\right] \\ &\leq (\rho_{\mathsf{FL}} + 1) \cdot c(\sigma_{\mathcal{O}}) + 2\rho_{\mathsf{FL}} \cdot f(S_{\mathcal{O}}),\end{aligned}$$

where the last inequality follows from Lemmas 5 and 6. ◀

▶ **Theorem 1.** *There exists a randomized $2\rho_{\mathsf{FL}}$-approximation algorithm for the $\emptyset$-facility location problem.*

Proof. Immediate from Lemma 7. It is easy to observe that the algorithm runs in polynomial time. ◀

4 General Case

In this section, we consider the general O-facility location problem.

4.1 Our Algorithm

Outline. We start with a brief outline of our algorithm. As the first step of the algorithm, we execute $\mathcal{A}_{\mathsf{FL}}$ for the unconstrained facility location problem on the given input, but with its parity constraints dropped. Let $S_{\mathcal{I}} \subseteq F$ and $\sigma_{\mathcal{I}} : D \to S_{\mathcal{I}}$ be the algorithm's output. Note that $(S_{\mathcal{I}}, \sigma_{\mathcal{I}})$ may be infeasible since we dropped the parity constraints; though, our algorithm will use it as the "initial" solution and correct the parities at small cost.

The second step of our algorithm is to construct an auxiliary weighted graph G and a set of vertices $T \subseteq V(G)$. The construction is designed so that a T-join (almost) prescribes a way to correct the parities. Naturally, our algorithm will find a minimum-cost T-join.

Then the last step of our algorithm is to modify the initial solution as indicated by the minimum-cost T-join on the auxiliary graph. We first post-process the minimum-cost T-join we found to obtain a *sparse* T-join. We will show that this sparsified T-join specifies a "realizable" modification to the initial solution that restores the parity constraints.

In what follows, we describe the last two steps in more detail. As a final remark, we note that the running time of our algorithm depends on that of $\mathcal{A}_{\mathsf{FL}}$; given that LP-rounding algorithms [36, 8, 25] typically yield better approximation ratios for the unconstrained problem, the running time of our algorithm will be dominated by the LP solution time. However, one could instead use primal-dual algorithms [19, 18] if faster running time is preferred.

Construction of the Auxiliary Graph. We say a facility $i \in S_{\mathcal{I}}$ is *invalid* if its parity constraint is violated in the initial solution. Let S_{inv} denote the set of invalid facilities, i.e., $S_{\mathsf{inv}} := \{i \in S_{\mathcal{I}} \cap O \mid |\sigma_{\mathcal{I}}^{-1}(i)| \text{ is even}\} \cup \{i \in S_{\mathcal{I}} \cap \overline{O} \mid |\sigma_{\mathcal{I}}^{-1}(i)| \text{ is odd}\}$.

The vertex set of the auxiliary graph G is $F \cup \{z\}$ for an artificial vertex $z \notin F$. Let E be the edge set of the auxiliary graph and $\gamma : E \to \mathbb{Q}_{\geq 0}$ be the edge cost. The following are three types of edges that we create in G.

- (*reassign edges*) For each pair of distinct facilities $i, i' \in F$, we create an edge (i, i') in the auxiliary graph with cost $\gamma(i, i') := c(i, i')$.
- (*opening edges*) For each odd-constrained, initially closed facility $i \in O \setminus S_{\mathcal{I}}$, we create an edge (z, i) with cost $\gamma(z, i) := f(i)$.
- (*closing edges*) This last type of edges is created only if $|S_{\mathcal{I}}| \geq 2$ or $|\overline{O} \setminus S_{\mathcal{I}}| \geq 1$. For each odd-constrained, initially open facility $i \in O \cap S_{\mathcal{I}}$, we create an edge (z, i) with cost

$$\gamma(z, i) := \min \begin{cases} \min\limits_{i' \in S_{\mathcal{I}} \setminus \{i\}} \left[\left|\sigma_{\mathcal{I}}^{-1}(i)\right| \cdot c(i, i') \right] \\ \min\limits_{i' \in \overline{O} \setminus S_{\mathcal{I}}} \left[\left|\sigma_{\mathcal{I}}^{-1}(i)\right| \cdot c(i, i') + f(i') \right], \end{cases} \tag{1}$$

  where $\min \emptyset := +\infty$. Note that $\gamma(z, i)$ is finite since we have $|S_{\mathcal{I}}| \geq 2$ or $|\overline{O} \setminus S_{\mathcal{I}}| \geq 1$.

Finally, we choose $T = S_{\mathsf{inv}}$ if $|S_{\mathsf{inv}}|$ is even; we choose $T = S_{\mathsf{inv}} \cup \{z\}$ otherwise.

We remark that this auxiliary graph does not satisfy the triangle inequality, and hence a minimum-cost T-join may be cheaper than a minimum-cost perfect matching on the subgraph induced by T. For notational convenience, for a set $E' \subseteq E$, let $\gamma(E') := \sum_{e \in E'} \gamma(e)$.

Intuitively speaking, if the T-join chooses a *reassign edge* (i, i'), it is instructing us to reassign a client between i and i'; an *opening edge* (z, i) corresponds to opening an initially closed facility i; finally, a *closing edge* (z, i) corresponds to closing down an initially open facility i. Although we will formally describe this correction procedure later, here we introduce one more definition: if we decide to close down a facility i, we will need to reassign all clients that were previously assigned to i to some other facility. This facility is called the *substitute* of i. The substitutes are selected as the facilities that attain the minimum in (1). That is, for each closing edge (z, i), the substitute of i, denoted by $\phi(i)$, is given as follows (ties are broken arbitrarily when the arg min has more than one element):

$$\phi(i) \in \begin{cases} \operatorname*{arg\,min}_{i' \in S_{\mathcal{I}} \setminus \{i\}} \left[\left|\sigma_{\mathcal{I}}^{-1}(i)\right| \cdot c(i, i') \right], & \text{if } \gamma(z, i) = \min_{i' \in S_{\mathcal{I}} \setminus \{i\}} \left[\left|\sigma_{\mathcal{I}}^{-1}(i)\right| \cdot c(i, i') \right], \\ \operatorname*{arg\,min}_{i' \in \overline{O} \setminus S_{\mathcal{I}}} \left[\left|\sigma_{\mathcal{I}}^{-1}(i)\right| \cdot c(i, i') + f(i') \right], & \text{otherwise.} \end{cases} \tag{2}$$

Sparsifying the T-join. Given a minimum T-join J, we examine whether any of the following operations can be performed; if so, we perform the operation and repeat. We terminate when none of the operations can be applied any more.

(i) If $(i, i_1), (i, i_2) \in J$ for some $i, i_1, i_2 \in F$, remove (i, i_1) and (i, i_2) from J and add (i_1, i_2) instead.

(ii) If $(z, i) \in J$ for some open odd-constrained facility i and $(z, \phi(i)) \in J$, remove (z, i) and $(z, \phi(i))$ from J and add $(i, \phi(i))$ instead. (Note that the condition implies that (z, i) is a closing edge and $\phi(i)$ is, therefore, well-defined.)

(iii) If J contains a cycle, remove all edges on the cycle.

Parity Correction. The final step of the algorithm is to modify the initial solution as prescribed by the sparsified T-join J. The parity correction is performed in the following three substeps:

1. Firstly, for each opening edge $(z, i) \in J$, open i and remove (z, i) from J.
2. Secondly, for each reassign edge $(i_1, i_2) \in J$, reassign one arbitrary client from one of the two facilities to the other and remove (i_1, i_2) from J as follows:
   - if $(z, i_1) \in J$ (or $(z, i_2) \in J$), reassign *from* i_1 *to* i_2 (or from i_2 to i_1, respectively);
   - otherwise, at least one of these facilities is guaranteed to be currently assigned at least one client; reassign from that facility to the other.
3. Lastly, for each closing edge $(z, i) \in J$, close i and reassign all clients currently assigned to i to $\phi(i)$; if necessary, open $\phi(i)$. Remove (z, i) from J.

4.2 Analysis of the Sparsification and Parity Correction

In this section, we show that a sparsified T-join, unlike a general T-join, prescribes a cheap modification for correcting the invalid facilities in the initial solution. We first prove that the sparsification does not increase the cost of a T-join. We will slightly abuse the notation and treat a T-join J interchangeably as a graph $(F \cup \{z\}, J)$. For $x \in F \cup \{z\}$, let $\mathsf{deg}_J(x)$ denote the degree of x in such a graph J.

▶ **Lemma 8.** *The given sparsification procedure yields a T-join of no greater cost.*

Proof. It suffices to prove that each single operation produces a T-join of no greater cost, and the lemma follows from the induction on the number of operations. Observe that, for every vertex $x \in F \cup \{z\}$, the parity of $\mathsf{deg}_J(x)$ remains the same when we apply any of the three operations. It remains to show that all three operations never increase the cost of J.

Consider Operation (i) that replaces (i, i_1) and (i, i_2) with (i_1, i_2). The net increase in the cost is

$$\gamma(i_1, i_2) - [\gamma(i, i_1) + \gamma(i, i_2)] = c(i_1, i_2) - [c(i, i_1) + c(i, i_2)] \le 0,$$

where the inequality follows from the triangle inequality.

Operation (iii) does not increase the cost of J since all costs are nonnegative.

Now consider Operation (ii). We can assume without loss of generality that $\mathcal{A}_{\mathsf{FL}}$ returns a solution such that $i \in S_{\mathcal{I}}$ implies $\sigma_{\mathcal{I}}^{-1}(i) \neq \emptyset$. (Otherwise, we can simply exclude i from $S_{\mathcal{I}}$.) Since (z, i) is a closing edge, (1) and (2) imply that

$$\gamma(z, i) \ge \left|\sigma_{\mathcal{I}}^{-1}(i)\right| \cdot c(i, \phi(i)) \ge c(i, \phi(i)) = \gamma(i, \phi(i)).$$

The operation does not increase the cost of J since $\gamma(z, \phi(i)) \ge 0$. ◀

Following are the key sparsity observations we will use in the parity correction step. Let J denote the sparsified T-join on which no further operation was possible.

▶ **Observation 9.** *For all $i \in F$, i is adjacent in J with at most one vertex in F.*

Proof. If i were adjacent in J with more than one facility, say, i_1 and i_2, edges (i, i_1) and (i, i_2) would have been replaced with (i_1, i_2). ◀

▶ **Observation 10.** *For all edges $(i_1, i_2) \in J$ such that $i_1, i_2 \in F$, at least one of i_1 and i_2 belongs to $S_{\mathcal{I}}$, the set of initially open facilities.*

Proof. Suppose towards contradiction that $i_1, i_2 \in F \setminus S_{\mathcal{I}}$. Since $i_1 \notin T$, there exists some vertex x other than i_2 that is adjacent with i_1; we have $x = z$ from Observation 9. Likewise, we have $(z, i_2) \in J$, leading to contradiction since $\{(i_1, i_2), (z, i_1), (z, i_2)\}$ forms a cycle. ◀

▶ **Observation 11.** *If a closing edge (z, i) is in J, we have $(z, \phi(i)) \notin J$.*

Proof. Since $(z, i) \in J$ is a closing edge, we know i is an open odd-constrained facility. Suppose $(z, \phi(i)) \in J$. Then (z, i) and $(z, \phi(i))$ would have been replaced with $(i, \phi(i))$, leading to contradiction. ◀

We can now analyze the parity correction prescribed by J. Observations 9 and 10 show that, when we process a reassign edge, the facility that *gives* a client does have at least one client to give, and the facility that *receives* a client is open. Observation 11 proves that, when we process a closing edge (z, i), the substitute $\phi(i)$ is indeed open and never got closed by the algorithm. These arguments are formalized by the following lemma.

▶ **Lemma 12.** *The corrected solution is a feasible solution. Moreover, the correction cost is bounded by $\gamma(J)$ from above.*

Its full proof uses Observations 9, 10, and 11, which is deferred to Appendix A.

4.3 Bounding $\gamma(J)$

We show in this section that the cost of a minimum T-join in the auxiliary graph G is within a constant factor of the optimum. Here we fix an arbitrary optimal solution $S_{\mathcal{O}} \subseteq F$ and $\sigma_{\mathcal{O}} : D \to S_{\mathcal{O}}$; let $\mathsf{OPT} := f(S_{\mathcal{O}}) + c(\sigma_{\mathcal{O}})$ denote its value. In the interest of a simpler analysis, we will exhibit a T-join *dominator* $Y \subseteq E$ such that $\gamma(Y) \le O(1) \cdot \mathsf{OPT}$, rather than explicitly constructing a set of reassignment/opening/closing operations to restore the parity constraints.

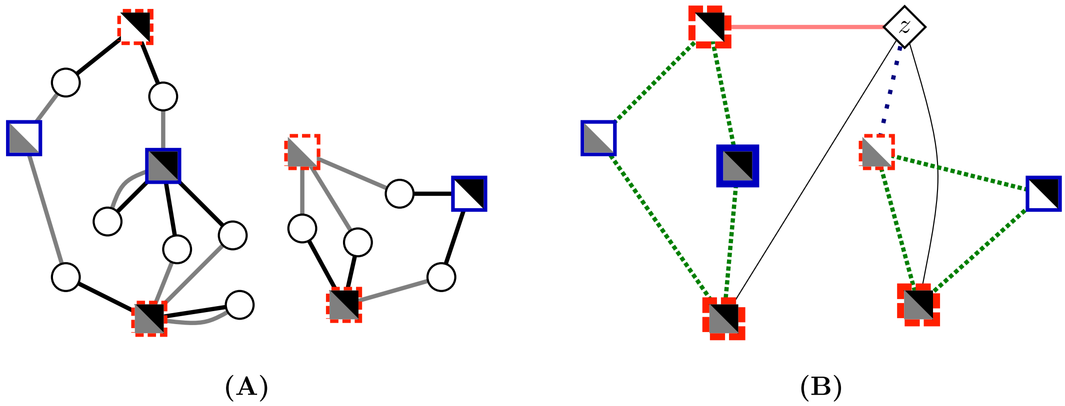

Figure 1 (A): An initial solution $(S_{\mathcal{I}}, \sigma_{\mathcal{I}})$ and an optimal solution $(S_{\mathcal{O}}, \sigma_{\mathcal{O}})$. A facility (and a client) is represented as a square (and a circle, respectively). Odd-constrained facilities have red dashed borders; even-constrained ones have navy-blue solid borders. The upper-right triangle is filled with black if the facility is open in the initial solution; the lower-left triangle is filled with gray if it is open in the optimal solution. Assignments in the initial solution are marked with black solid lines; assignments in the optimal solution are gray solid lines.
(B): A T-join dominator Y. If a facility i is in S_{inv}, it is marked with a thicker border. Every edge in Y_1 is drawn as a green densely-dotted line; Y_2 as a navy-blue loosely-dotted line; and Y_3 as a coral solid line. The remaining closing edges are drawn as black thin solid lines. (We omitted the remaining reassign edges.)

We construct Y as the union of three edge sets Y_1, Y_2, and Y_3, each of which is a subset of reassign edges, opening edges, and closing edges, respectively. Compared to the first two types of edges, closing edges are more expensive. (Recall that their costs contain a multiplicative factor given by the number of clients a facility is assigned in the initial solution.) Hence, in constructing Y_3, edges must be used much more sparingly compared to the first two sets. We will, however, show that a careful choice of closing edges ensures that the cost of Y_3 can be charged against $2 \cdot \mathsf{OPT}$, while maintaining the required cut connectivity for Y to be a T-join dominator. The three edge sets are defined as follows. (See Figure 1.)

- For each client $j \in D$, we add an edge to Y_1 between the facility j is assigned to in the initial solution and the facility j is assigned to in the fixed optimal solution. If both facilities are the same, we just ignore the client rather than creating a loop.
- Let Y_2 be the set of edges between z and each odd-constrained facility that is closed in the initial solution but open in the optimal solution.
- Let $\mathcal{C}$ be the set of connected components C in $(F \cup \{z\}, Y_1 \cup Y_2)$ such that C does not contain z and is T-odd, i.e.,

$$\mathcal{C} := \{C \mid C \text{ is a conn. comp. in } (F \cup \{z\}, Y_1 \cup Y_2),\ z \notin V(C), \text{ and } |V(C) \cap T| \text{ is odd}\},$$

  where $V(C)$ denotes the set of vertices in C. For each component $C \in \mathcal{C}$, pick an arbitrary odd-constrained facility $i^C \in V(C)$ that is open in the initial solution but closed in the optimal solution. We thus have $i^C \in V(C) \cap O \cap S_{\mathcal{I}} \setminus S_{\mathcal{O}}$. We now define Y_3 as the set of edges between z and i^C for all $C \in \mathcal{C}$.

The following observation holds since, if a client is assigned to different facilities in the initial and optimal solutions, we have an edge in Y_1 between these facilities, placing them in the same connected component in $\mathcal{C}$.

▶ **Observation 13.** *For each $C \in \mathcal{C}$, let $n_{\mathcal{I}}^C$ (and $n_{\mathcal{O}}^C$) be the number of clients assigned to a facility in $V(C)$ by the initial solution (and by the optimal solution, respectively). We then have $n_{\mathcal{I}}^C = n_{\mathcal{O}}^C$.*

In order to show that Y is a T-join dominator, it suffices to prove that i^C can be chosen for every connected component $C \in \mathcal{C}$. Suppose that i^C is well-defined for each $C \in \mathcal{C}$. Then, in the graph $(F \cup \{z\}, Y_1 \cup Y_2 \cup Y_3)$, every connected component that does not contain z must contain an even number of vertices in T, since otherwise, an edge in Y_3 would have connected this component to z. That is, the only connected component in $(F \cup \{z\}, Y_1 \cup Y_2 \cup Y_3)$ that may have an odd number of vertices in T is the one that contains z; however, since $|T|$ is even, this component also has even number of vertices in T. The conclusion now follows from the definition of a T-join dominator.

▶ **Lemma 14.** *For each connected component $C \in \mathcal{C}$, there exists an odd-constrained facility that is open in the initial solution but closed in the fixed optimal solution.*

Proof. From construction, for any facility i that is closed in both solutions, i cannot be incident with any edges in Y_1 or Y_2. Such facility i, therefore, forms a singleton connected component in $(F \cup \{z\}, Y_1 \cup Y_2)$ and we have $i \notin T$. Thus, for the T-odd component C, we have that every facility in $V(C)$ must be open in at least one of the two solutions.

Suppose towards contradiction that there does not exist an odd-constrained facility in $V(C)$ that is open in the initial solution but closed in the fixed optimal solution. That is, every facility $i \in V(C)$ that is open only in the initial solution is even-constrained.

If some facility $i \in V(C)$ is open only in the *optimal* solution, i cannot be odd-constrained: otherwise, the opening edge (z, i) would be in Y_2, contradicting $C \in \mathcal{C}$. So we now have that every facility $i \in V(C)$ that is open only in one of the two solutions is even-constrained. In other words, every facility in $V(C)$ is either even-constrained or open in both solutions. This, together with $n_{\mathcal{O}}^C = \sum_{i \in V(C)} |\sigma_{\mathcal{O}}^{-1}(i)|$ and the fact that the optimal solution satisfies all parity constraints, implies that the parity of $n_{\mathcal{O}}^C$ is the same as that of $|V(C) \cap O|$.

On the other hand, since $V(C)$ contains invalid facilities, we have that the parity of $n_{\mathcal{I}}^C$ is equal to that of $|V(C) \cap O| + |V(C) \cap S_{\mathsf{inv}}|$. From Observation 13, this implies that $|V(C) \cap S_{\mathsf{inv}}|$ is even, contradicting the fact that C is T-odd. ◀

To argue that the closing edge (z, i^C) indeed exists in G, we need to verify that $|S_{\mathcal{I}}| \geq 2$ or $|\overline{O} \setminus S_{\mathcal{I}}| \geq 1$. Note that $|S_{\mathcal{I}}| \geq 1$ as long as $D \neq \emptyset$.

▶ **Lemma 15.** *If $|S_{\mathcal{I}}| = 1$ and $\overline{O} \setminus S_{\mathcal{I}} = \emptyset$, we have $\mathcal{C} = \emptyset$.*

Proof. Suppose towards contradiction that $|S_{\mathcal{I}}| = 1$, $\overline{O} \setminus S_{\mathcal{I}} = \emptyset$, and $C \in \mathcal{C}$. Since $|V(C) \cap T|$ is odd, we can choose some $i \in V(C) \cap T$. Moreover, we have $i \in T \setminus \{z\} = S_{\mathsf{inv}} \subseteq S_{\mathcal{I}}$ since $z \notin V(C)$.

Now we claim that $V(C) = \{i\}$. Suppose not. Let i' be an arbitrary facility in $V(C) \setminus \{i\}$. Since $S_{\mathcal{I}} = \{i\}$, we have $i' \notin S_{\mathcal{I}}$. This, together with $\overline{O} \setminus S_{\mathcal{I}} = \emptyset$, yields $i' \in O$. Since C does not contain z, we have $(z, i') \notin Y_2$, showing $i' \notin S_{\mathcal{O}}$. Recall from the proof of Lemma 14 that a facility in $V(C)$ cannot be closed in both solutions.

Therefore, i must be open in the optimal solution because the clients assigned to i in $S_{\mathcal{I}}$ can only be assigned to i in $S_{\mathcal{O}}$. Since $V(C)$ does not contain any facility that is closed in the optimal solution, we cannot choose i^C, contradicting Lemma 14. ◀

Now we bound the cost of Y.

▶ **Observation 16.** *We have* $\gamma(Y_1) \le c(\sigma_{\mathcal{I}}) + c(\sigma_{\mathcal{O}})$.

Proof. By the definition of Y_1, for every edge $(i, i') \in Y_1$, there exists a client j such that $\sigma_{\mathcal{I}}(j) = i$ and $\sigma_{\mathcal{O}}(j) = i'$, yielding that $\gamma(i, i') = c(i, i') \le c(i, j) + c(i', j)$. We thus have

$$\gamma(Y_1) = \sum_{(i,i') \in Y_1} \gamma(i, i') \le \sum_{\substack{i \in F, \\ j \in \sigma_{\mathcal{I}}^{-1}(i)}} c(i, j) + \sum_{\substack{i' \in F, \\ j \in \sigma_{\mathcal{O}}^{-1}(i')}} c(i', j) = c(\sigma_{\mathcal{I}}) + c(\sigma_{\mathcal{O}}).$$ ◀

▶ **Observation 17.** *We have* $\gamma(Y_2) \le f(S_{\mathcal{O}})$.

Proof. Note that the cost of each edge $(z, i) \in Y_2$ is the opening cost of i and we add (z, i) to Y_2 only if $i \in S_{\mathcal{O}} \setminus S_{\mathcal{I}}$. ◀

▶ **Lemma 18.** *We have* $\gamma(Y_3) \le c(\sigma_{\mathcal{I}}) + c(\sigma_{\mathcal{O}}) + f(S_{\mathcal{O}})$.

Proof. Let C be an arbitrary connected component in $\mathcal{C}$. By Lemma 14, i^C is well-defined. Let $B^C \subseteq D$ be the set of the clients that are assigned to i^C in the initial solution, i.e., $B^C := \sigma_{\mathcal{I}}^{-1}(i^C)$.

We claim that B^C can be partitioned into two sets B^C_{open} and B^C_{ec} where the former is a set of clients assigned in the optimal solution to a facility which is also open in the initial solution and the latter is to a facility which is even-constrained and closed in the initial solution. For each $j \in B^C$, consider $\sigma_{\mathcal{O}}(j) =: i' \in S_{\mathcal{O}}$. Note that $i' \in V(C)$ from the construction of Y_1. If i' is open in the initial solution, since i^C is closed in the optimal solution, we have $i' \in S_{\mathcal{I}} \setminus \{i^C\}$. Otherwise, since i' is open in the optimal solution but (z, i') was not chosen in Y_2, it must be the case that i' is even-constrained. Thus, to reiterate,

$$B^C_{\mathsf{open}} := \left\{ j \mid \sigma_{\mathcal{I}}(j) = i^C \text{ and } \sigma_{\mathcal{O}}(j) \in (S_{\mathcal{I}} \setminus \{i^C\}) \cap V(C) \right\};$$
$$B^C_{\mathsf{ec}} := \left\{ j \mid \sigma_{\mathcal{I}}(j) = i^C \text{ and } \sigma_{\mathcal{O}}(j) \in (\overline{O} \setminus S_{\mathcal{I}}) \cap V(C) \right\}.$$

Let $\lambda := \left|B^C_{\mathsf{open}}\right| / \left|B^C\right|$. Since B^C_{open} and B^C_{ec} forms a partition of B^C, we have $0 \le \lambda \le 1$ and $\left|B^C_{\mathsf{ec}}\right| / \left|B^C\right| = 1 - \lambda$.

Now we bound $\gamma(z, i^C)$ from above using the assignment costs (in both solutions) of the clients in B^C, along with the opening costs of $V(C) \cap S_{\mathcal{O}}$.

Assume for now that $\lambda > 0$. Then we have

$$\begin{aligned} c(\sigma_{\mathcal{I}}|_{B^C_{\mathsf{open}}}) + c(\sigma_{\mathcal{O}}|_{B^C_{\mathsf{open}}}) &= \sum_{j \in B^C_{\mathsf{open}}} c(i^C, j) + \sum_{j \in B^C_{\mathsf{open}}} c\left(\sigma_{\mathcal{O}}(j), j\right) \\ &\ge \sum_{j \in B^C_{\mathsf{open}}} c\left(i^C, \sigma_{\mathcal{O}}(j)\right) \\ &\ge \left|B^C_{\mathsf{open}}\right| \cdot \min_{i' \in S_{\mathcal{I}} \setminus \{i^C\}} c(i^C, i') \\ &= \lambda \cdot \min_{i' \in S_{\mathcal{I}} \setminus \{i^C\}} \left[\left|\sigma_{\mathcal{I}}^{-1}(i^C)\right| \cdot c(i^C, i')\right], \end{aligned} \tag{3}$$

where the first inequality holds due to the triangle inequality and the second inequality is from the fact that, for every $j \in B^C_{\mathsf{open}}$, we have $\sigma_{\mathcal{O}}(j) \in S_{\mathcal{I}} \setminus \{i^C\}$. Note that the above trivially holds if $\lambda = 0$.

Now suppose for the moment that $\lambda < 1$. Let S^C_{ec} be the set of the facilities that are assigned a client from B^C_{ec} in the optimal solution, i.e., $S^C_{\mathsf{ec}} := \left\{\sigma_{\mathcal{O}}(j) \mid j \in B^C_{\mathsf{ec}}\right\}$. Let i_{ec} be the closest facility from i^C in S^C_{ec}. We then have the following:

$$\begin{aligned}
c(\sigma_{\mathcal{I}}|_{B^C_{\mathsf{ec}}}) + c(\sigma_{\mathcal{O}}|_{B^C_{\mathsf{ec}}}) + f(S^C_{\mathsf{ec}}) &= \sum_{j \in B^C_{\mathsf{ec}}} c(i^C, j) + \sum_{j \in B^C_{\mathsf{ec}}} c\left(\sigma_{\mathcal{O}}(j), j\right) + \sum_{i \in S^C_{\mathsf{ec}}} f(i) \\
&\geq \sum_{j \in B^C_{\mathsf{ec}}} c\left(i^C, \sigma_{\mathcal{O}}(j)\right) + \sum_{i \in S^C_{\mathsf{ec}}} f(i) \\
&\geq \left|B^C_{\mathsf{ec}}\right| \cdot c(i^C, i_{\mathsf{ec}}) + f(i_{\mathsf{ec}}) \\
&\geq (1-\lambda) \cdot \left[\left|\sigma_{\mathcal{I}}^{-1}(i^C)\right| \cdot c(i^C, i_{\mathsf{ec}}) + f(i_{\mathsf{ec}})\right] \\
&\geq (1-\lambda) \cdot \min_{i' \in \overline{O} \setminus S_{\mathcal{I}}} \left[\left|\sigma_{\mathcal{I}}^{-1}(i^C)\right| \cdot c(i^C, i') + f(i')\right]. \qquad (4)
\end{aligned}$$

Again, the above trivially holds when $\lambda = 1$. Combining (3) and (4) yields

$$c(\sigma_{\mathcal{I}}|_{B^C}) + c(\sigma_{\mathcal{O}}|_{B^C}) + f(S^C_{\mathsf{ec}}) \geq \gamma(z, i^C).$$

It is noteworthy that B^C's for $C \in \mathcal{C}$ are mutually disjoint since a client can be assigned to exactly one facility. We can also observe that S^C_{ec}'s are mutually disjoint because each facility belongs to at most one connected component in $\mathcal{C}$. We thus have

$$\gamma(Y_3) = \sum_{C \in \mathcal{C}} \gamma(z, i^C) \leq \sum_{C \in \mathcal{C}} \left(c(\sigma_{\mathcal{I}}|_{B^C}) + c(\sigma_{\mathcal{O}}|_{B^C}) + f(S^C_{\mathsf{ec}})\right) \leq c(\sigma_{\mathcal{I}}) + c(\sigma_{\mathcal{O}}) + f(S_{\mathcal{O}}). \blacktriangleleft$$

▶ **Lemma 19.** *There exists a T-join J in G whose cost is at most $2 \cdot (c(\sigma_{\mathcal{I}}) + c(\sigma_{\mathcal{O}}) + f(S_{\mathcal{O}}))$.*[7]

Proof. Immediate from Lemmas 4, 14 and 18, and Observations 16 and 17. ◀

We can now prove our main theorem.

▶ **Theorem 2.** *There exists a $(3\rho_{\mathsf{FL}} + 2)$-approximation algorithm for the O-facility location problem.*

Proof. Immediate from Lemmas 12 and 19. Note that our algorithm returns a feasible solution of cost at most

$$(c(\sigma_{\mathcal{I}}) + f(S_{\mathcal{I}})) + 2 \cdot (c(\sigma_{\mathcal{I}}) + c(\sigma_{\mathcal{O}}) + f(S_{\mathcal{O}})) \leq (3\rho_{\mathsf{FL}} + 2) \cdot \mathsf{OPT},$$

since $c(\sigma_{\mathcal{I}}) + f(S_{\mathcal{I}}) \leq \rho_{\mathsf{FL}} \cdot \mathsf{OPT}$. It can be easily verified that the algorithm runs in polynomial time. ◀

References

1 Anna Adamaszek, Antonios Antoniadis, Amit Kumar, and Tobias Mömke. Approximating airports and railways. In Rolf Niedermeier and Brigitte Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:13, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.STACS.2018.5`.

2 Mustaque Ahamad and Mostafa H. Ammar. Performance characterization of quorum-consensus algorithms for replicated data. *IEEE Transactions on Software Engineering*, 15(4):492–496, 1989.

[7] We remark that a more careful analysis yields a tighter bound of $2c(\sigma_{\mathcal{I}}) + 2c(\sigma_{\mathcal{O}}) + f(S_{\mathcal{O}})$, although we present the current proof in favor of simplicity.

3 Sara Ahmadian, Zachary Friggstad, and Chaitanya Swamy. Local-search based approximation algorithms for mobile facility location problems. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1607–1621, 2013. URL: `http://dl.acm.org/citation.cfm?id=2627817.2627932`.

4 Hyung-Chan An, Mohit Singh, and Ola Svensson. LP-based algorithms for capacitated facility location. *SIAM Journal on Computing*, 46(1):272–306, 2017.

5 Michel L. Balinski. On finding integer solutions to linear programs. Technical report, Mathematica Princeton NJ, 1964.

6 Manisha Bansal, Naveen Garg, and Neelima Gupta. A 5-approximation for capacitated facility location. In *European Symposium on Algorithms (ESA)*, pages 133–144, 2012.

7 András A. Benczúr and Ottilia Fülöp. Fast algorithms for even/odd minimum cuts and generalizations. In *European Symposium on Algorithms (ESA)*, pages 88–99, 2000.

8 Jaroslaw Byrka. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 29–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

9 Joseph Cheriyan, Zachary Friggstad, and Zhihan Gao. Approximating minimum-cost connected T-joins. *Algorithmica*, 72(1):126–147, 2015. `doi:10.1007/s00453-013-9850-8`.

10 Marek Cygan, Artur Czumaj, Marcin Mucha, and Piotr Sankowski. Online facility location with deletions. In *European Symposium on Algorithms (ESA)*, pages 21:1–21:15, 2018.

11 Jack Edmonds and Ellis Johnson. Matching, Euler tours and the Chinese postman. *Mathematical Programming*, 5:88–124, 1973. `doi:10.1007/BF01580113`.

12 David Eisenstat, Claire Mathieu, and Nicolas Schabanel. Facility location in evolving metrics. In *Automata, Languages, and Programming*, pages 459–470, 2014.

13 Hazel Everett, Celina M.H. de Figueiredo, Cláudia Linhares-Sales, Frédéric Maffray, Oscar Porto, and Bruce A. Reed. Path parity and perfection. *Discrete Mathematics*, 165:233–252, 1997.

14 Gramoz Goranci, Monika Henzinger, and Dariusz Leniowski. A tree structure for dynamic facility location. In *European Symposium on Algorithms (ESA)*, pages 39:1–39:13, 2018.

15 Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

16 Martin Grötschel, László Lovász, and Alexander Schrijver. Corrigendum to our paper "the ellipsoid method and its consequences in combinatorial optimization". *Combinatorica*, 4(4):291–295, 1984.

17 Martin Grötschel and William R. Pulleyblank. Weakly bipartite graphs and the max-cut problem. *Operations Research Letters*, 1(1):23–27, 1981.

18 Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In *ACM Symposium on Theory of Computing (STOC)*, pages 731–740, 2002.

19 Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM*, 48(2):274–296, 2001.

20 Naonori Kakimura, Ken-ichi Kawarabayashi, and Yusuke Kobayashi. Erdős-Pósa property and its algorithmic applications — parity constraints, subset feedback set, and subset packing. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1726–1736, 2012. URL: `http://dl.acm.org/citation.cfm?id=2095116.2095253`.

21 Marcin Kaminski and Naomi Nishimura. Finding an induced path of given parity in planar graphs in polynomial time. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 656–670, 2012. `doi:10.1137/1.9781611973099.55`.

22 Madhukar R. Korupolu, C.Greg Plaxton, and Rajmohan Rajaraman. Analysis of a local search heuristic for facility location problems. *Journal of Algorithms*, 37(1):146–188, 2000. `doi:10.1006/jagm.2000.1100`.

23 Alfred A. Kuehn and Michael J. Hamburger. A heuristic program for locating warehouses. *Management Science*, 9(4):643–666, 1963. URL: `http://www.jstor.org/stable/2627368`.

24 Christiane Lammersen and Christian Sohler. Facility location in dynamic geometric data streams. In *European Symposium on Algorithms (ESA)*, pages 660–671, 2008.

25 Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. In *Automata, Languages and Programming*, pages 77–88, 2011.

26 Shi Li. On facility location with general lower bounds. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2279–2290, 2019.

27 Alan S. Manne. Plant location under economies-of-scale—decentralization and computation. *Management Science*, 11(2):213–235, 1964. URL: `http://www.jstor.org/stable/2627598`.

28 Dániel Marx and Michał Pilipczuk. Optimal parameterized algorithms for planar facility location problems using Voronoi diagrams. In *European Symposium on Algorithms (ESA)*, pages 865–877, 2015.

29 Jens Maßberg and Jens Vygen. Approximation algorithms for a facility location problem with service capacities. *ACM Trans. Algorithms*, 4(4):50:1–50:15, 2008. `doi:10.1145/1383369.1383381`.

30 Jannik Matuschke, Andreas Bley, and Benjamin Müller. Approximation algorithms for facility location with capacitated and length-bounded tree connections. In Hans L. Bodlaender and Giuseppe F. Italiano, editors, *Algorithms – ESA 2013*, pages 707–718, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

31 Manfred W. Padberg and M. R. Rao. Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research*, 7(1):67–80, 1982.

32 Martin Pál, Éva Tardos, and Tom Wexler. Facility location with nonuniform hard capacities. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 329–338, 2001. `doi:10.1109/SFCS.2001.959907`.

33 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*. Springer Science & Business Media, 2003.

34 Alexander Schrijver and Paul Seymour. Packing odd paths. *Journal of Combinatorial Theory*, 62:280–288, 1994.

35 András Sebő. Eight-fifth approximation for the path TSP. In *Integer Programming and Combinatorial Optimization (IPCO)*, pages 362–374, 2013.

36 David B. Shmoys and Karen I. Aardal. *Approximation algorithms for facility location problems*. Utrecht University: Information and Computing Sciences, 1997.

37 John F. Stollsteimer. A working model for plant numbers and locations. *Journal of Farm Economics*, 45(3):631–645, 1963. URL: `http://www.jstor.org/stable/1235442`.

38 The Apache Software Foundation. Apache ZooKeeper. `http://zookeeper.apache.org/`.

A Deferred Proofs

This appendix presents the deferred proofs.

▶ **Observation 3.** *The two problem definitions are equivalent.*

Proof. Given an O-facility location problem instance, simply by defining the parity constraint π as

$$\pi(i) := \begin{cases} \mathsf{odd}, & \text{if } i \in O, \\ \mathsf{even}, & \text{otherwise,} \end{cases}$$

we arrive at an equivalent instance of the second form, i.e., the form where unconstrained facilities are allowed. Now suppose that we are given an instance of the second form. We create two copies of every facility i such that $\pi(i) = \mathsf{unconstrained}$ and put exactly one of these two copies into O, in addition to the facilities i with $\pi(i) = \mathsf{odd}$. Since all opening costs are nonnegative, we can assume without loss of generality that an optimal solution to the new instance will open at most one copy of the duplicates. This shows the equivalence of the two problem definitions. ◀

▶ **Lemma 12.** *The corrected solution is a feasible solution. Moreover, the correction cost is bounded by $\gamma(J)$ from above.*

Proof. We say a facility i has the *incorrect* parity in a solution (S, σ) if the parity constraint of the facility is violated in the "current" solution. (This definition differs from the *invalid* facilities, which are *fixed* as the facilities with the incorrect parities in the initial solution.) We show the feasibility of the corrected solution by establishing invariants throughout the parity correction procedure. We will modify the procedure so that it now modifies T in addition to J in each iteration. The invariants are the following:

- J is a T-join in the auxiliary graph, and
- $T \setminus \{z\}$ is exactly the set of facilities having the incorrect parities.

Observe that $|J|$ decreases by one in each iteration; the corrected solution is, therefore, feasible since the empty set of edges is an $\emptyset$-join. The correction cost will be bounded by showing that the cost incurred in each iteration can be covered by the cost of the corresponding edge removed from J. Recall that we start with the sparsified T-join J where $T \setminus \{z\} = S_{\mathsf{inv}}$; it is clear that both invariants initially hold.

Now we start erasing the edges from J. Here we remark that, given a T-join J and an edge $(i_1, i_2) \in J$, $J \setminus \{(i_1, i_2)\}$ is a $T \triangle \{i_1, i_2\}$-join since the degrees of i_1 and i_2 decrease by one.

Let us consider the first substep. For each opening edge $(z, i) \in J$, we open facility i in the solution, remove (z, i) from J, and update $T \leftarrow T \triangle \{z, i\}$. It can be easily seen that J is still a T-join. By the construction of the auxiliary graph, facility i is closed in the initial solution and hence $i \notin T$ at the beginning of this iteration. Moreover, since i is odd-constrained at the same time, after opening i, this facility enters the set of facilities having the incorrect parities, establishing the second invariant. Observe that the cost for opening i can be covered by $\gamma(z, i)$. At this point after completing the first substep, any facility i with $\mathsf{deg}_J(i) \neq 0$ is currently open. To observe this fact, consider the initial sparse T-join J *before* the first substep. If any $i \notin S_{\mathcal{I}}$ has a nonzero degree, the degree must be at least two since $i \notin T$. This implies (from Observation 9) that (z, i) must have been in J, which causes i to be opened during the first substep.

Next, for each reassign edge $(i_1, i_2) \in J$, we transfer a client j from one of the two facilities to the other. We then remove $(i_1, i_2) \in J$ and update $T \leftarrow T \triangle \{i_1, i_2\}$. By Observation 10, at least one of i_1 and i_2 was open in $S_{\mathcal{I}}$. Recall that we can assume without loss of generality that $\mathcal{A}_{\mathsf{FL}}$ returns a solution such that $i \in S_{\mathcal{I}}$ implies $\sigma_{\mathcal{I}}^{-1}(i) \neq \emptyset$. Thus, we can always choose a client j to be reassigned. (Note that Observation 9 shows that every facility can participate in at most one reassignment.) Since exactly one client is reassigned, the parities of i_1 and i_2 are flipped and the second invariant is maintained. Note that the cost of reassigning j from, say, i_1 to i_2 is $-c(i_1, j) + c(i_2, j) \leq c(i_1, i_2) = \gamma(i_1, i_2)$ from the triangle inequality. If the reassignment is from i_2 to i_1, the symmetric argument holds.

Let us now consider the last substep where we handle closing edges. For each closing edge $(z, i) \in J$, we close i and reassign all the clients currently assigned to i to its substitute $\phi(i)$. We then remove (z, i) from J and update $T \leftarrow T \triangle \{z, i\}$.

Consider the time point right before closing i. We claim that the number of clients assigned to i is even. Due to the construction of the auxiliary graph, we have $i \in O$. Since we have already processed (and removed) all reassign edges, i is adjacent with only z at the moment and thus is in T. This, from the induction hypothesis, implies that i has the incorrect parity, i.e., i is assigned even number of clients. Therefore, reassigning all the clients assigned to i to $\phi(i)$ would not change the parity of $\phi(i)$. (Note that, if $\phi(i)$ was closed at the beginning, then $\phi(i) \in \overline{O} \setminus S_{\mathcal{I}} \subseteq \overline{O}$.) With the fact that i becomes closed at this iteration, this shows that both invariants hold.

We finally verify that the correction cost here is no more than $\gamma(z, i)$. Recall that we close facility i and reassign every client j assigned to i to $\phi(i)$. Thus, the change of the assignment cost for each j is exactly $-c(i, j) + c(\phi(i), j)$. As argued above, by the triangle inequality, we know that this value can be bounded by $c(i, \phi(i))$ from above. We can thus see that the total assignment cost may increase by at most $|\sigma_{\mathcal{I}}^{-1}(i)| \cdot c(i, \phi(i))$ since the number of clients assigned to i does not increase during the previous substeps. (During the second substep, when we process (i_1, i_2) and find that one of the facilities, say i_1, is adjacent with z in J, we reassigned a client *from* i_1 *to* i_2.) If $\phi(i)$ was closed at the beginning of the correction, we may need to open $\phi(i)$, but $\gamma(z, i)$ already pays for it. If $\phi(i)$ was open, by Observation 11, we know $\phi(i)$ will never be closed. These together imply that the correction cost is no greater than $\gamma(z, i)$. ◀