Efficient algorithms for the 2-gathering problem

Alon Shalita * *Uri Zwick* *

Abstract

Pebbles are placed on some vertices of a directed graph. Is it possible to move each pebble along at most one edge of the graph so that in the final configuration no pebble is left on its own? We give an $O(mn)$-time algorithm for solving this problem, which we call the *2-gathering* problem, where n is the number of vertices and m is the number of edges of the graph. If such a 2-gathering is not possible, the algorithm finds a solution that minimizes the number of solitary pebbles. The 2-gathering problem forms a non-trivial generalization of the non-bipartite matching problem and it is solved by extending the augmenting paths technique used to solve matching problems.

1 Introduction

A group of students is taking an algorithms course. To review the material they would like to form study groups. Each study group should comprise at least r students and should meet in one of several possible meeting places. Each student can conveniently reach only some of these meeting places. Is there a way of partitioning the students into study groups, and assign these study groups meeting places, so that each student can conveniently reach the meeting place of her group? It is easy to see that this problem is NP-complete when $r \geq 3$. (See, e.g., Armon [2].) For $r = 2$, the problem is equivalent to the 2-gathering problem mentioned in the abstract. We present an $O(mn)$-time algorithm for solving this problem.

Polynomial time algorithms for the 2-gathering problem can be obtained via reductions to the $\{K_2, K_3\}$-*packing* problem studied by Cornuéjols et al. [4] and by Hell and Kirkpatrick [10], to the *general factor* problem studied by Lovász [12, 13], Cornuéjols [3] and Sebő [15], and to *the simplex matching* problem studied by Anshelevich and Karagiozova [1]. These problems and their relation to the 2-gathering problem are described in the next section. Our algorithm for the 2-gathering problem is faster than the algorithms obtained via reductions by a factor of at least $\Omega(n^2)$.

*School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: `{shalitaa,zwick}@cs.tau.ac.il`. Research supported by BSF grant no. 2006261.

The 2-gathering problem is a non-trivial generalization of the non-bipartite matching problem. We solve it by using extensions of the augmenting paths technique used by Edmonds [5] to solve matching problems. Similar extensions were used before by Cornuéjols et al. [4], Hell and Kirkpatrick [10], Cornuéjols [3], and Anshelevich and Karagiozova [1]. They were mostly interested, however, in showing that the problems they study can be solved in polynomial time. We, on the other hand, obtain a very efficient algorithm whose complexity is not much worse than the complexity of algorithms for the non-bipartite matching problem.

2 2-gatherings and related problems

Let $G = (V, E)$ be a directed graph and let $S \subseteq V$ be a set of vertices each containing a single *pebble*. In various applications, pebbles may correspond to agents, clients, servers, terminals etc. Is there a *mapping* $M : S \to V$ such that for every $u \in S$ either $M(u) = u$ or $(u, M(u)) \in E$ and such that for every $v \in V$ we have $|M^{-1}(v)| \neq 1$, i.e., the set of pebbles mapped to v is *not* of size 1? This is exactly the *2-gathering* problem mentioned in the abstract.

The 2-gathering problem is a generalization of the non-bipartite matching problem. Indeed, given an undirected graph $G = (V, E)$ we can construct a directed graph $G' = (V \cup E, E')$ by replacing each undirected edge (u, v) in G by two directed edges $u \to uv$ and $uv \leftarrow v$ in G', where uv is a newly added vertex, and by placing pebbles on all original vertices of the graph. It is easy to check that G has a perfect matching if and only if G' has a 2-gathering.

Several more general versions of the 2-gathering problem can be easily reduced to the basic 2-gathering problem defined above. For example, we may assume that the edges of the graph $G = (V, E)$ have *lengths* associated with them, and that a pebble placed at a vertex $u \in S$ has a *travel budget* of $b(u)$. A pebble at $u \in S$ can then travel to any vertex $v \in V$ for which $\delta_G(u, v) \leq b(u)$, where $\delta_G(u, v)$ is the distance from u to v in G. Is there a mapping meeting these constraints under which each pebble ends up in a vertex with at least one other pebble? The 'budgeted' version of the problem can be reduced to the basic version of the problem by defining a graph $G' = (V, E')$ such that $(u, v) \in E'$ if and only if $u \in S$ and $\delta_G(u, v) \le b(u)$.

The main result of this paper is an $O(mn)$-time algorithm for the 2-gathering problem, where n is the number of vertices and m is the number of edges in the input graph. Our algorithm uses an approach similar to the one used to obtain a perfect matching in a non-bipartite graph. It starts with an arbitrary mapping M. The *deficiency* of a mapping M is defined to be the number of vertices $v \in V$ for which $|M^{-1}(v)| = 1$. We show that if the deficiency of M is not minimal, then there is always a relatively simple *augmenting structure* that can be used to augment M, i.e., decrease it deficiency. If a matching is not a maximum matching, then it can always be augmented using an *augmenting path.* In our case things are a bit more complicated. We need to use not only augmenting paths but also *augmenting cycles,* or a combinations of augmenting paths and cycles. We can, however, still reduce the task of finding an appropriate augmenting structure to the task of finding a conventional augmenting path in an appropriately constructed graph, with respect to a suitably constructed matching. A different graph, and a different matching, are used in each iteration of the algorithm. Our algorithm looks for at most $O(n)$ augmenting paths, each in a graph with $O(m)$ edges. As an augmenting path, if one exists, in a graph with $O(m)$ edges can be found in $O(m)$ time (Gabow and Tarjan [8]), the total running time of our algorithm is $O(mn)$.

The 2-gathering problem is related to a non-weighted version of the *simplex matching* problem studied recently by Anshelevich and Karagiozova [1]. The unweighted simplex matching problem is equivalent to the $\{K_2, K_3\}$-*packing* problem studied by Cornuéjols et al. [4] and by Hell and Kirkpatrick [10]. The input to the unweighted simplex matching problem is a hypergraph $H = (V, E)$ in which each (hyper-)edge $e \in E$ is of size 2 or 3 and if $e = \{u, v, w\} \in E$, then also $\{u, v\}, \{u, w\}, \{v, w\} \in E$. The goal is to a find a *perfect matching*, i.e., a disjoint collection of edges whose union is V. The input to the $\{K_2, K_3\}$-packing problem is an undirected graph $G = (V, E)$ and a subset T of triangles in G. The goal is to find a vertex disjoint collection of edges (K_2's) and triangles (K_3's) from T whose union is V. The $\{K_2, K_3\}$-packing problem on a graph $G = (V, E)$ is clearly equivalent to the unweighted simplex matching problem on the hypergraph $H = (V, E \cup T)$, where T is the set of allowed triangles in G. In the *weighted* simplex matching problem, each edge $e \in E$ has a nonnegative cost $c(e)$ associated with it such that if $\{u, v, w\} \in E$, then $c(u, v) + c(u, w) + c(v, w) \le 2c(u, v, w)$. The main result of [1] is a polynomial time algorithm for the weighted simplex matching. The running time of their algorithm on unweighted simplex matching problems is $O(n^3 m^2)$ (see Karagiozova [11], p. 51), where n is the number of vertices and m is the number of (hyper)-edges in H.

The unweighted simplex matching problem can be easily reduced to the 2-gathering problem. Given a hypergraph $H = (V, E)$, we construct a bipartite graph $G = (V \cup E, E')$ such that $(v, e) \in E'$ iff $v \in e$ and place pebbles on all original vertices of the graph. Clearly H has perfect matching iff G has a 2-gathering. Note that G has $O(m)$ vertices and edges. Using our 2-gathering algorithm, we can therefore solve the unweighted simplex matching problem in $O(m^2)$-time, improving on the $O(n^3 m^2)$ running time of [1].

Conversely, the 2-gathering problem can be reduced to the unweighted simplex matching problem. Given a directed graph $G = (V, E)$ and a subset $S \subseteq V$, construct a hypergraph $H = (S, E')$ such that $\{u, v, w\} \in E'$ if and only if $u, v, w \in S$ and $(u, w), (v, w) \in E$, or there exists a vertex $x \in V$ such that $(u, x), (v, x), (w, x) \in E$. Similarly, $\{u, v\} \in E'$ if and only if $u, v \in S$ and $(u, v) \in E$ or there exists $x \in V$ such that $(u, x), (v, x) \in E$. Clearly, G has a 2-gathering iff H has a perfect matching. Note, however, that E' may contain $\Omega(|S|^3)$ edges. If $|S| = \Omega(n)$, then running time of the algorithm of [1] on the instance produced may be $\Omega(n^9)$.

Anshelevich and Karagiozova [1] use their weighted simplex matching algorithm to solve the *terminal backup* problem. The input to this problem is a weighted undirected graph $G = (V, E)$ and a set of *terminals* $S \subseteq V$. The goal is to find a subset of edges $E' \subseteq E$ of minimal total weight such that in the subgraph $G' = (V, E')$ no connected component contains exactly one terminal. The terminal backup problem is equivalent to a weighted version of the 2-gathering problem in which the goal is to find a mapping $M : S \to V$ such that $|M^{-1}(v)| \ne 1$, for every $v \in V$, and such that $\sum_{u \in S} \delta_G(u, M(u))$ is minimized. Adapting our algorithm to the solution of the weighted 2-gathering problem is an interesting open problem.

The 2-gathering problem is also a special case of the *generalized factor* problem studied by Cornuéjols [3]. The input to the generalized factor problem is an undirected graph $G = (V, E)$ and a subset $B_v \subseteq \{0, 1, \ldots, d(v)\}$ for each $v \in V$, where $d(v)$ is the degree of v in G. The goal is to find a subgraph $H = (V, F)$ of G such that $d_H(v) \in B_v$ for each $v \in V$. The generalized factor problem can be solved in polynomial time if none of the sets B_v has a gap of size greater than 1, and the problem is NP-hard otherwise. (A set B has a gap of k at i if and only if $i, i+k+1 \in B$ but $i+1, \ldots, i+k \notin B$.)

It is easy to show, as we do in the next section,

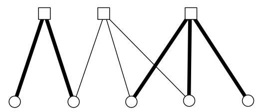

Figure 1: A 2-gathering in a bipartite graph.

that the 2-gathering problem is equivalent to a general factor problem in which for every $v \in V$, either $B_v = \{0, \ldots, d(v)\} - \{1\}$ or $B_v = \{1\}$. (Cornuéjols [3] refers to this problem as the *1-factor-antifactor* problem.) As none of these sets has a gap of size greater than 1, the resulting problem may be solved using the general algorithms provided by Cornuéjols [3] for the polynomial cases of the general factor problem. Cornuéjols [3] provides four algorithms that can be used to solve the 1-factor-antifactor problem. The complexity of the fastest two of these are $O(n^3 m)$ and $O(m^3)$, respectively, slower than our algorithm by a factor of at least $\Omega(n^2)$.

3 Augmenting paths, cycles and lassos

The 2-gathering problem on general directed graphs can be easily reduced to the 2-gathering problem on undirected *bipartite* graphs: Given a directed graph $G = (V, E)$ and a subset $S \subseteq V$, construct a bipartite graph $G' = (\bar{S}, V, E')$, where $\bar{S} = \{\bar{u} \mid u \in S\}$ and $\{\bar{u}, v\} \in E'$ if and only if $u = v$ or $(u, v) \in E$. (Here $\bar{u}$ is a *copy* of u.) It easy to see that there is a 2-gathering of S in G if and only if there is a 2-gathering of $\bar{S}$ in G'.

In the remainder of the paper we consider the bipartite version of 2-gathering problem. If $G = (S, T, E)$ is a bipartite graph, with $E \subseteq S \times T$, we call S the set of *sources*, and T the set of *targets*. A mapping $M : S \to T$ is then simply a subset $M \subseteq E$ such that $d_M(s) = 1$, for every $s \in S$, where $d_M(s)$ is the *degree* of s in the subgraph (S, T, M).

Definition 3.1. (Mappings and 2-gatherings) *Let $G = (S, T, E)$ be a bipartite graph. A subset $M \subseteq E$ for which $d_M(s) = 1$, for every source $s \in S$ is called a mapping. A mapping M is a* 2-gathering *if $d_M(t) \neq 1$, for every target $t \in T$. A target $t \in T$ for which $d_M(t) = 1$ is said to be* deficient. *We let $Def(M)$ denote the number of deficient targets under M. (For an example of a 2-gathering in a bipartite graph, see Fig. 1. Sources are represented by circles, targets by squares, and the 2-gathering edges are bold.)*

We say that target $t \in T$ is *odd* with respect to M if $d_M(t)$ is odd, and *even* if $d_M(t)$ is even.

In analogy with *alternating* paths used to augment matchings, we now define *altering* paths that would be used to augment mappings.

Definition 3.2. (Altering paths and cycles) *A simple path P from $t_1 \in T$ to $t_2 \in T$ is said to be an* altering path *with respect to M if for every source $s \in P$, among the two edges of the path touching s, one is from M while the other is not, i.e., $d_{M\cap P}(s) = 1$. Similarly, a simple cycle C is said to be to an* altering cycle *with respect to M if for every source $s \in P$ we have $d_{M\cap C}(s) = 1$.*

Note that unlike augmenting paths, altering paths and cycles with respect to a mapping M may contain two consecutive edges from M or two consecutive edges not from M. Such two edges, however, must share a target.

Lemma 3.1. *Let M be a mapping and let P be an altering path with respect to M. Then, $M \oplus P$ is also a mapping. Similarly, if C is an altering cycle with respect to M, then $M \oplus C$ is also a mapping.*

Proof. Every source $s \in P$ has one edge from M and one edge not from M adjacent to it in P. In $M \oplus P$ the role of these two edges is switched. The proof for cycles is identical. □

Lemma 3.2. *Let M_1 and M_2 be mappings. Then any path between two targets in $M_1 \oplus M_2$ is an altering path with respect to both M_1 and M_2. Similarly any cycle in $M_1 \oplus M_2$ is an altering cycle with respect to both M_1 and M_2.*

Proof. Since M_1 and M_2 are mappings, for any source $s \in S$ we have $d_{M_1}(s) = d_{M_2}(s) = 1$, and so $d_{M_1 \oplus M_2}(s) = 0$ or 2. In the later case, each of the edges adjacent to s belongs to a different mapping. □

Definition 3.3. (Even altering paths and cycles) *An altering path P with respect to M from $t_1 \in T$ to $t_2 \in T$ is said to be an* even altering path *if every target $t \in P - \{t_1, t_2\}$ is even with respect to M. (Recall that a target t is even with respect to M if and only if $d_M(t)$ is even.) An altering cycle C with respect to M is said to be an* even altering cycle passing through a target t *if $t \in C$, every other target $t' \in C - \{t\}$ is even, and the two edges of C touching t are not in M.*

Lemma 3.3. *Let M be a mapping and let P be an even altering path (with respect to M) from a deficient target $t_1 \in T$ to a target $t_2 \in T$. Then $Def(M \oplus P) \leq Def(M)$ and the inequality is strict if t_2 is not deficient with respect to $M \oplus P$. Similarly, if C is an even altering cycle (with respect to M) through t, then $Def(M \oplus C) \leq Def(M)$ and the inequality is strict if t is deficient in M.*

Proof. By Lemma 3.1, $M' = M \oplus P$ is a mapping. As t_1 and t_2 have only one path edge touching them, we have

$d_{M'}(t_1) \neq d_M(t_1)$ and $d_{M'}(t_2) \neq d_M(t_2)$. Any other target $t \in P - \{t_1, t_2\}$ has two path edges touching it and hence $d_{M'}(t) \equiv d_M(t) \pmod 2$. Since all targets in $P - \{t_1, t_2\}$ are even in M, they are all even in M', and hence none of them is deficient in M'. Thus, t_1 is deficient in M but not in M', while t_2 is the only target that may be deficient in M' but not in M. Thus, $Def(M') \leq Def(M)$. If t_2 is not deficient in M' then $Def(M') < Def(M)$.

If C is an even altering cycle through t, then as in the previous case $M' = M \oplus C$ is a mapping, and every target $t' \in C - \{t\}$ is even with respect to both M and M'. As the two edges of the cycle touching t are not in M, we get that $d_{M'}(t) = d_M(t) + 2 \neq 1$. No deficient targets are thus introduced and hence $Def(M') \leq Def(M)$. If t was deficient in M, we get that $Def(M') < Def(M)$. □

DEFINITION 3.4. (LASSOS) *A* lasso *is composed of an even altering path P from a target $t_1 \in T$ to a target $t_2 \in T$ and an even altering cycle C through t_2 such that $P \cap C = \{t_2\}$. We call t_2 the* base *or* end *of the lasso.*

We now describe five types of paths, cycles and lassos that can be used to *augment* a mapping, i.e., reduce its deficiency.

DEFINITION 3.5. (AUGMENTING STRUCTURES)

$\mathcal{P}_1$) *An even altering path P from a deficient target t_1 to a target t_2 with $d_M(t_2) \neq 0$ whose last edge is* not *in M is said to be a* $\mathcal{P}_1$-augmenting path.

$\mathcal{P}_2$) *An even altering path P from a deficient target t_1 to a target t_2 with $d_M(t_2) \neq 2$ whose last edge is in M is said to be a* $\mathcal{P}_2$-augmenting path.

$\mathcal{C}$) *An even altering cycle C through a deficient target t is said to be a* $\mathcal{C}$-augmenting cycle.

$\mathcal{L}_1$) *A lasso L with a base t_2 such that $d_M(t_2) = 0$ is said to be an* $\mathcal{L}_1$-augmenting lasso.

$\mathcal{L}_2$) *A lasso L with a base t_2 such that $d_M(t_2) = 2$, and such that the last edge of the lasso path is in M is said to be an* $\mathcal{L}_2$-augmenting lasso.

Examples of augmenting structures of these five types can be found in Figure 2. Bold gray edges in the figure represent edges of M that are not part of the augmenting structure.

A $\mathcal{P}$-augmenting path is a $\mathcal{P}_1$-augmenting path or a $\mathcal{P}_2$-augmenting path. Similarly, an $\mathcal{L}$-augmenting lasso is an $\mathcal{L}_1$-augmenting lasso or an $\mathcal{L}_2$-augmenting lasso.

LEMMA 3.4. *Let M be a mapping and let P be an altering path from a deficient target $t_1 \in T$ to a target $t_2 \in T$. If P is not an even altering path, then it contains a $\mathcal{P}$-augmenting path from t_1 to a target $t' \in T$ on P.*

Proof. If P is not an even altering path, then at least one of the intermediate targets on it is odd. Let $t' \in P$ be the first odd target encountered when P is traversed from t_1 to t_2. The subpath of P from t_1 to t' is an even altering path that ends with an odd target, so it is a $\mathcal{P}$-augmenting path. □

LEMMA 3.5. *Let M be a mapping, and let R be an augmenting structure of one of the types of Definition 3.5. Then, $Def(M \oplus R) < Def(M)$.*

Proof. Suppose that P is a $\mathcal{P}$-augmenting path. It is easy to check that t_2 is not deficient in $M \oplus P$ and thus by Lemma 3.3 we have $Def(M \oplus P) < Def(M)$. Similarly, if C is a $\mathcal{C}$-augmenting cycle through a target t, then t is not deficient in $M \oplus C$ and thus by Lemma 3.3 we have $Def(M \oplus C) < Def(M)$.

Suppose that L is an $\mathcal{L}$-augmenting lasso, composed of a path P and a cycle C with $P \cap C = \{t\}$. Then $Def(M \oplus P) = Def(M)$ and C is a $\mathcal{C}$-augmenting cycle through t with respect to $M \oplus P$. By the previous case, we get $Def(M \oplus L) = Def((M \oplus P) \oplus C) < Def(M \oplus P) = Def(M)$. □

We now show that a mapping is of minimum deficiency if and only if it admits no augmenting structure.

THEOREM 3.1. *A mapping M is a mapping of minimum deficiency if and only if there is no augmenting structure of type $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{C}$, $\mathcal{L}_1$ or $\mathcal{L}_2$ with respect to M.*

Proof. If R is an $\mathcal{R}$-augmenting structure, where $\mathcal{R} \in \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{L}_1, \mathcal{L}_2, \mathcal{C}\}$, then by Lemma 3.5 we have $Def(M \oplus R) < Def(M)$. We next show that if M is not a mapping of minimum deficiency, then it admits an $\mathcal{R}$-augmenting structure, where $\mathcal{R} \in \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{L}_1, \mathcal{L}_2, \mathcal{C}\}$.

Let M' be a mapping with $Def(M') < Def(M)$ for which $|M \cap M'|$ is maximal. Let $D = M \oplus M'$ and consider the subgraph (S, T, D). For a connected component Q of this subgraph, let $Def_Q(M)$ be the number of deficient targets in Q with respect to M. There must be at least one component Q for which $Def_Q(M') < Def_Q(M)$. By Lemma 3.2, every path or cycle in Q is altering with respect to M and M'.

Let t_1 be a deficient target in Q with respect to M. If there exists a target $t_2 \in Q - \{t_1\}$ such that $d_M(t_2)$ is odd, then by Lemma 3.4, the path from t_1 to t_2 in Q, or a prefix of it, is a $\mathcal{P}$-augmenting path, and we are done. We may assume, therefore, that all targets in $Q - \{t_1\}$

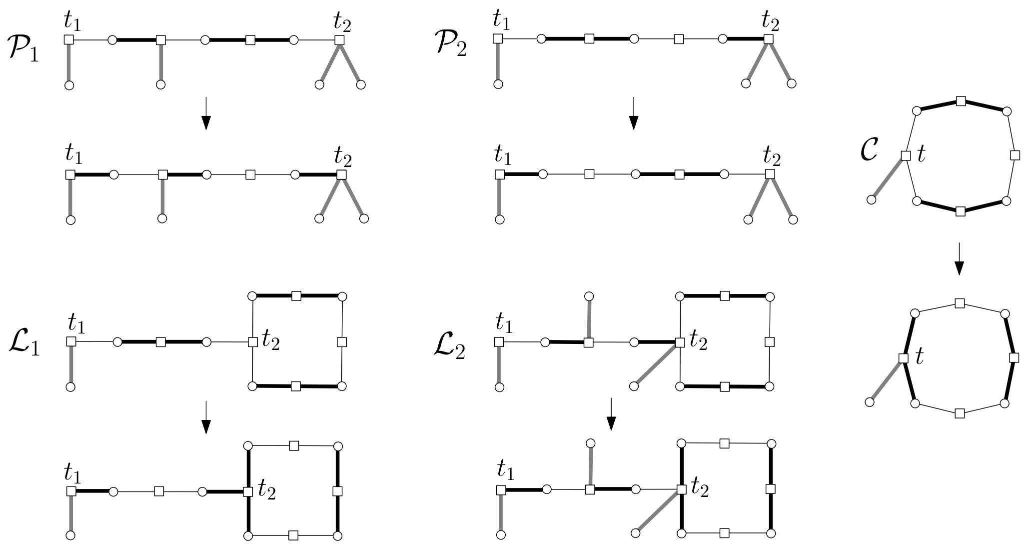

Figure 2: Five types of augmenting structures.

are even with respect to M and thus all paths from t_1 and cycles through t_1 in Q are even altering with respect to M. As t_1 is the only deficient target in Q with respect to M, we get that $Def_Q(M) = 1$ and $Def_Q(M') = 0$.

If there exists a target $t_2 \in Q - \{t_1\}$ with $d_Q(t_2) = d_{M \oplus M'}(t_2) = 1$, let P be a simple path in Q from t_1 to t_2. Let e be the last edge of P. Assume at first that $e \in M - M'$. Since t_2 is not deficient with respect to M, there must be another edge $e' \neq e$ adjacent to t_2 which belongs to $M \cap M'$. Since t_2 is not deficient with respect to M', there must be a third edge $e'' \neq e', e$ adjacent to t_2 which again belongs to $M \cap M'$. Thus $d_M(t_2) \geq 3$. If $e \in M' - M$, we get in a similar manner that $d_M(t_2) \geq 2$. In both cases we get that P is a $\mathcal{P}$-augmenting path, and we are again done.

We may assume, therefore, that for every target $t \in Q - \{t_1\}$ we have $d_Q(t) \geq 2$. For every source $s \in Q$ we have $d_Q(s) = 2$. Thus Q has at most one vertex of degree 1. It follows that Q is not a tree and it contains, therefore, a simple cycle C.

As $C \subseteq M \oplus M'$, the mapping $M' \oplus C$ has a larger intersection with M than M', i.e., $|(M' \oplus C) \cap M| > |M' \cap M|$. Hence, there must exist a target $t_0 \in C$ which is deficient with respect to $M' \oplus C$ but not with respect to M'. This can happen only if $d_{M'}(t_0) = 3$ and $d_{M' \cap C}(t_0) = 2$, i.e., both edges of C that touch t are in $M' - M$.

We next show that for every target $t \in C$ we have $d_{M' \cap C}(t) = 0$ or 2, and hence $d_{M \cap C}(t) = 0$ or 2. Assume, for the sake of contradiction, that there exists a target $t_2 \in C$ for which $d_{M' \cap C}(t_2) = 1$. As $d_{M'}(t_2) \geq d_{M' \cap C}(t_2) = 1$, and as no targets in Q are deficient with respect to M', we get that $d_{M'}(t_2) \geq 2$. Consider now the path P on C from t_2 to t_0 that starts with the edge of C that belongs to $M - M'$, and let $M'' = M' \oplus P$. We claim that $Def(M'') = Def(M') < Def(M)$. Indeed $d_{M''}(t_2) = d_{M'}(t_2) + 1 \geq 3$ and $d_{M''}(t_0) = d_{M'}(t_0) - 1 = 2$, while for every target $t \in C - \{t_0, t_2\}$ we have $d_{M''}(t) \equiv d_{M'}(t) \pmod 2$. As $|M'' \cap M| > |M' \cap M|$, this is a contradiction to the choice of M'.

Recall that $t_1 \in Q$ and $d_M(t_1) = 1$. If $t_1 \in C$, we must have $d_{M \cap C}(t_1) = 0$, as $d_{M \cap C}(t_1) = 2$ is impossible, and hence C is a $\mathcal{C}$-augmenting cycle passing through t_1. Assume, therefore, that $t_1 \notin C$. Let P be a simple path in Q from t_1 to a target t_2 on C such that $P \cap C = \{t_2\}$. (Note that the first encounter between a path and a cycle in Q must be at a target, as all sources are of degree 2 in Q.) Let e be the last edge on P. If $e \notin M$ and $d_M(t_2) \neq 0$, then P is a $\mathcal{P}_1$-augmenting path. If $d_M(t_2) = 0$, then $P \cup C$ is an $\mathcal{L}_1$-augmenting lasso. If $e \in M$ and $d_M(t_2) \neq 2$, then P is a $\mathcal{P}_2$-augmenting path. Finally, if $e \in M$ and $d_M(t_2) = 2$, then as $d_{M \cap C}(t_2) = 0$ or 2 and $e \in M - C$, we get that $d_{M \cap C}(t_2) = 0$ and $P \cup C$ is an $\mathcal{L}_2$-augmenting lasso. □

Finding augmenting lassos turns out to be a harder task than finding augmenting paths or cycles. The difficulty lies in the fact that the path and the cycle that comprise a lasso need to be disjoint. We next show that instead of looking for an augmenting lasso it is enough to look for a *deficiency transferring path* and then for an augmenting cycle.

Definition 3.6. (Deficiency transferring paths)

$\mathcal{P}'_1$) *An even altering path P from a deficient target t_1 to a target t_2 with $d_M(t_2) = 0$ is said to be a $\mathcal{P}'_1$-path.*

$\mathcal{P}_2'$) *An even altering path P from a deficient target t_1 to a target t_2 with $d_M(t_2) = 2$ whose last edge is in M is said to be a $\mathcal{P}_2'$-path.*

A $\mathcal{P}'$-path with respect to M is a $\mathcal{P}_1'$*-path or a* $\mathcal{P}_2'$*-path with respect to M.*

If P is a $\mathcal{P}'$-path with respect to M, then the deficiency of t_1 in M is replaced by the deficiency of t_2 in $M \oplus P$. The deficiency of all other targets does not change.

LEMMA 3.6. *Let M_0 a mapping, let P be a $\mathcal{P}'$-path with respect to M_0 from t_1 to t_2, and let C be a $\mathcal{C}$-augmenting cycle with respect to $M_0 \oplus P$ through t_2. Then, for any $\mathcal{P}'$-path P' with respect to M_0 from t_1' to t_2 (t_1' might be equal to t_1) there is either a $\mathcal{C}$-augmenting cycle C' through t_2, or a $\mathcal{P}$-augmenting path P'' from t_1 to t_2, both with respect to $M_0 \oplus P'$.*

Proof. Let $M = M_0 \oplus P \oplus C$, $M' = M_0 \oplus P'$ and $D = M \oplus M' = P \oplus C \oplus C'$. By Lemma 3.2 all paths and cycles in D are altering with respect to $M' = M_0 \oplus P'$. Moreover, all targets on P, C or P', except t_2, and possibly t_1 (if $t_1 \neq t_1'$), are even with respect to $M' = M_0 \oplus P'$. Similarly, all vertices, except t_1, and possibly t_1' (if $t_1 \neq t_1'$), have even degrees in D. Let Q be the connected component of D that contains t_2.

If $t_1 \neq t_1'$ and $t_1 \in Q$, then the path in Q that connects t_1 and t_2 is a $\mathcal{P}$-augmenting path with respect to $M' = M_0 \oplus P'$ and we are done. We may assume, therefore, that $t_1 = t_1'$ or that $t_1 \notin Q$. If $t_1 = t_1'$, then all the vertices in Q have even degrees with respect to D. If $t_1 \neq t_1'$, then since t_1 and t_1' are the only odd degree vertices in D, we get that t_1 and t_1' must be in the same connected component of D. As $t_1 \notin Q$ we get that $t_1' \notin Q$ and again all degrees in Q are even. In both cases, Q is Eulerian.

Let p and p' be the edges of P and P', respectively, that touch t_2, and let c_1 and c_2 be the edges of C that touch t_2. Note that $c_1 \neq c_2$, but we may have $p = p'$, $p = c_i$, $p' = c_j$, etc. (See Fig. 3 for the different non-symmetric possibilities.) As P and P' are both $\mathcal{P}'$-paths with respect to M_0, we get that $p \in M_0$ if and only if $p' \in M_0$. Also $d_{M_0 \oplus P}(t_2) = d_{M_0 \oplus P'}(t_2) = 1$. As C is a $\mathcal{C}$-augmenting cycle through t_2 with respect to $M_0 \oplus P$, we get that $c_1, c_2 \notin M \oplus P$. Our goal is now to show that Q contains a $\mathcal{C}$-augmenting cycle through t_2 with respect to $M_0 \oplus P'$. We know that $d_D(t_2)$ is even. As p, p', c_1, c_2 are the only edge that can touch t_2 in Q, we get that $d_Q(t_2) = 0, 2$ or 4. (See Fig. 3.)

If $d_Q(t_2) = 4$, then p, p', c_1, c_2 are all distinct. As $d_{M_0 \oplus P'}(t_2) = 1$, at most one of these edges belongs to $M_0 \oplus P'$. Let C' be a simple cycle in Q that goes

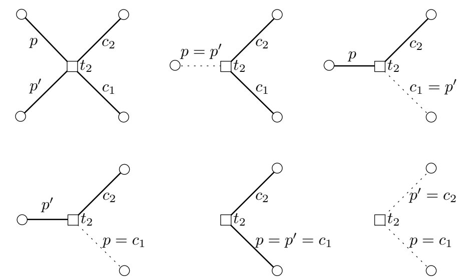

Figure 3: Edges touching t_2 in $P \oplus C \oplus P'$.

through t_2. (Such a cycle exists as Q is Eulerian.) If the two edges of C' touching t_2 are not in $M_0 \oplus P'$, then C' is a $\mathcal{C}$-augmenting cycle with respect to $M_0 \oplus P'$. Otherwise, let $Q' = Q - C'$. All degrees in Q' are again even and $d_{Q'}(t_2) = 2$. Let C'' be a simple cycle in Q' that goes through t_2. Since none of the edges touching t_2 in Q' is in $M_0 \oplus P'$, the cycle C'' is a $\mathcal{C}$-augmenting cycle with respect to $M_0 \oplus P'$. In both cases we are done.

Suppose now that $d_Q(t_2) = 2$. Let C' be a simple cycle in Q that goes through t_2. We show that both edges of Q, and hence of C', that touch t_2 are not in $M_0 \oplus P'$. It follows that C' is a $\mathcal{C}$-augmenting cycle with respect to $M_0 \oplus P'$. Indeed, if $c_i \in P \oplus C \oplus P'$, for $i \in \{1, 2\}$, then as $c_i \in C$ and $c_i \notin M_0 \oplus P$, we get that $c_i \notin M_0 \oplus P$'. Next, suppose that $p \in P \oplus C \oplus C'$ and that $d_Q(t_2) = 2$. That means that either $p \neq c_1, c_2$ and $p' = c_i$, for $i \in \{1, 2\}$, or $p = p' = c_i$, for $i \in \{1, 2\}$. Assume at first that $p \neq c_1, c_2$ and $p' = c_i$. As $c_i \notin M_0 \oplus P$ and $c_i \notin P$, we get that $p' = c_i \notin M_0$. Hence $p \notin M_0$. As $p \notin P'$, we get that $p \notin M_0 \oplus P'$. If $p = p' = c_i$, then as $c_i \notin M_0 \oplus P'$, $c_i = p \in P$ and $c_i = p' \in P$, we get that $p = p' = c_i \notin M_0 \oplus P'$. It remains to show that $p' \in P \oplus C \oplus C'$ implies $p' \notin M_0 \oplus P'$. The case $p = p' = c_i$ was already dealt with. Thus, we only need to consider the case $p' \neq c_1, c_2$ and $p = c_i$. As $c_i \notin M_0 \oplus P$ and $c_i = p \in P$, we get that $c_i = p \in M_0$ and hence $p' \in M_0$. As $p' \in P'$, we get that $p' \notin M_0 \oplus P'$, as required.

Finally, we show that the case $d_Q(t_2) = 0$ is impossible. For this to happen we need $p = c_1$ and $p' = c_2$ (or the symmetric case $p = c_2$ and $p' = c_1$). As $c_1, c_2 \notin M_0 \oplus P'$ and $c_1 = p \notin P'$ and $c_2 = p' \in P'$ we get that $c_1 = p \in M_0$ while $c_2 = p' \notin M_0$, a contradiction, as $p \in M_0$ if and only if $p' \in M_0$. □

4 The basic algorithm

Theorem 3.1 suggests the following natural algorithm for finding a mapping with a minimum number of deficient targets. Start with an arbitrary mapping M. Look for an augmenting structure of one of the five types considered above. If such an augmenting structure is found, use it to augment M, and repeat. Otherwise, M is an optimal mapping.

$\mathcal{P}_1$- and $\mathcal{P}_2$-augmenting paths, $\mathcal{C}$-augmenting cycles and $\mathcal{L}_2$-augmenting lassos, if they exist, can be found fairly easily. Finding $\mathcal{L}_1$-augmenting lassos, on the other hand, seems to be a harder task. We circumvent the need for finding augmenting lassos using Lemma 3.6. Suppose that L is an augmenting lasso through t_2. The lasso L is composed of a $\mathcal{P}'$-path P from t_1 to t_2, and a $\mathcal{C}$-augmenting cycle C through t_2. Instead of looking for the path P and the cycle C simultaneously, we first look for a $\mathcal{P}'$-path P' that ends at t_2, use it to modify the mapping, and then look for a $\mathcal{C}$-augmenting cycle C' through t_2, or a $\mathcal{P}$-augmenting path P'' that ends at t_2. By Lemma 3.6, the existence of L implies that for any $\mathcal{P}'$-path P' chosen, we are guaranteed to find either a $\mathcal{C}$-augmenting cycle C' or a $\mathcal{P}$-augmenting path P''. In both cases we can augment the mapping and proceed to the next iteration.

An optimal mapping would clearly be found after at most $|S| \le n$ augmentations. For each target $t_0 \in T$, we show below that we can find a $\mathcal{P}$-augmenting path ending at t_0, a $\mathcal{P}'$-path ending at t_0, or a $\mathcal{C}$-augmenting cycle passing through t_0, if they exist, in $O(m)$ time. As there are $|T| \le n$ targets, we can find an augmentation, if one exists, in $O(mn)$ time. The total running time of the algorithm would therefore be $O(mn^2)$. In the next section, we reduce the running time of the algorithm to $O(mn)$.

4.1 Finding augmenting structures Let $G = (S,T,E)$ be a bipartite graph and let $M \subseteq E$ be a mapping. Let $t_0 \in T$ be a specific target. Let $\mathcal{R} \in \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}'_1, \mathcal{P}'_2, \mathcal{C}\}$. To find an $\mathcal{R}$-structure ending at t_0, if one exists, we construct a new graph $G' = (V', E')$ and a matching $M' \subseteq E'$ such that there is an $\mathcal{R}$-structure ending at t_0 in G with respect to the *mapping* M, if and only if there is a (conventional) augmenting path ending at t_0 in G' with respect to the *matching* M'. Furthermore, given an augmenting path in G' ending at t, we can easily construct an $\mathcal{R}$-structure in G ending at t_0. The graph G' has $O(n)$ vertices and $O(m)$ edges. An augmenting path with respect to M' ending at t can be found in $O(m)$ time using an algorithm of Gabow and Tarjan [8].

The graph G' is obtained by replacing each target $t \in T$ by one of the *target gadgets* shown in Figure 4. Bold edges in the left hand side of each gadget correspond to edges of the mapping M while bold edges on the right correspond to edges of the newly constructed matching M'. Some targets and sources are removed from the graph. Some of the gadgets are similar to gadgets used by Gabow [7]. We next give a short description of each one of these gadgets:

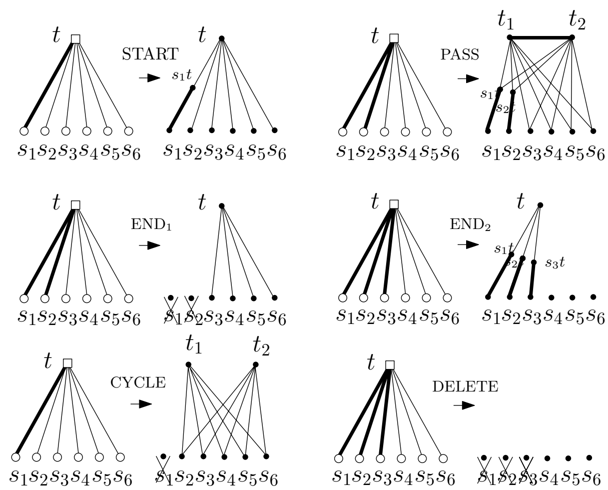

Figure 4: The target gadgets.

START: Applied to a deficient target, i.e., a target $t \in T$ with $d_M(t) = 1$. Suppose that $(s_1, t) \in M$. Introduce a new vertex s_1t and replace the edge (s_1, t) by the two edges (s_1, s_1t) and (s_1t, t). The edge (s_1, s_1t) is placed in M'. Other edges entering t are added to G' but not to the matching M'.

PASS: Applied to an even target, i.e., a target $t \in T$ with $d_M(t)$ even. The target t is replaced by two new vertices t_1 and t_2 and an edge (t_1, t_2) is added to both G' and M'. If $(s_i, t) \in M$, we again introduce a new vertex s_it and replace the edge (s_i, t) by the three edges (s_i, s_it), (s_it, t_1) and (s_it, t_2). The edge (s_i, s_it) is added to M'. If $(s_j, t) \notin M$, the edge (s_j, t) is replaced in G' by the two edges (s_j, t_1) and (s_j, t_2).

END$_1$: Applied to an arbitrary target t. Sources mapped to t are removed. Edges $(s, t) \notin M$ are retained.

END$_2$: Applied to a target t with $d_M(t) > 0$. Edges $(s_i, t) \in E$ are subdivided, as in START. Edges $(s_j, t) \notin M$ are removed.

CYCLE: Applied to a target t with $d_M(t) = 1$. The source mapped to t is removed from the graph. The target t in G is replaced by two vertices t_1 and t_2 in G'. Every edge (s, t) in G is replaced by two edges (s, t_1) and (s, t_2) in G'.

DELETE: Applied to a target t with $d_M(t)$ odd. Target t and all sources mapped to it are removed.

LEMMA 4.1. *Let G' and M' be the graph and the edge set obtained by applying a target replacement gadget, of any suitable kind, on each target of G. Then, M' is a matching in G' and all vertices in G' that correspond to sources of G are matched by M'.*

Proof. Let $s \in S$. As M is a mapping, there is a single target $t \in T$ such that $(s, t) \in M$. If s appears in G'

	t_0	$def(T) - \{t_0\}$	$odd_{\geq 3}(T) - \{t_0\}$	$even(T) - \{t_0\}$
$\mathcal{P}_1, \mathcal{P}'_1$	END_1	START	DELETE	PASS
$\mathcal{P}_2, \mathcal{P}'_2$	END_2	START	DELETE	PASS
$\mathcal{C}$	CYCLE	DELETE	DELETE	PASS

Table 1: The target replacement recipe.

then the edge (s, t), or an edge (s, st), appear in both G' and M'. No other edge of M' touches s. All other vertices of G' are vertices that belong to one of the gadgets and at most one edge of M' is adjacent to them. □

To find $\mathcal{R}$-structures, for $\mathcal{R} \in \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}'_1, \mathcal{P}'_2, \mathcal{C}\}$, we follow the recipe prescribed in Table 1. We use $def(T) = \{t \in T \mid d_M(t) = 1\}$ to denote the set of deficient targets, $odd(T) = \{t \in T \mid d_M(t) \geq 3\,,\, d_M(t)\,\text{is odd}\}$ to denote the set of non-deficient odd targets, and $even(T) = \{t \in T \mid d_M(t)\,\text{is even}\}$ to denote the set of even targets, all with respect to M.

For example, to find a $\mathcal{P}_1$-path or a $\mathcal{P}'_1$-path ending at a target t_0, we replace t_0 by an END_1 gadget, replace all deficient targets by a START gadget, replace all non-deficient odd targets by a DELETE gadget, and finally replace all even targets by PASS gadgets. Let G' be the resulting graph.

LEMMA 4.2. *Let $G = (S, T, E)$ be a bipartite graph, let M be a mapping of G, and let $t_0 \in T$. Let $\mathcal{R} \in \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}'_1, \mathcal{P}'_2, \mathcal{C}\}$ and let G' and M' be the graph and the matching obtained by replacing each target of G by a gadget, as prescribed by the corresponding row of Table 1. Then, there is a correspondence between $\mathcal{R}$-structures with respect to M that end at t_0 in G and augmenting paths that end at one of the vertices corresponding to t_0 in G'.*

Proof. We sketch the proof for $\mathcal{P}_1$-augmenting paths. The other cases are similar. Let $P = \langle t_1, s_1, t_2, \ldots, t_{k-1}, s_{k-1}, t_k = t_0\rangle$ be a $\mathcal{P}_1$-augmenting path ending at t_0. By definition $d_M(t_1) = 1$, and t_1 is therefore replaced by a START gadget, $d_M(t_2), \ldots, d_M(t_{k-1})$ are all even, and are therefore replaced by PASS gadgets. Finally t_k is replaced by an END_1 gadget. We can construct an augmenting path in G' with respect to M' from t_1 to t_k as follows. Each pair of edges $(s_{i-1}, t_i), (t_i, s_i)$, where $2 \leq i \leq k-1$, in P is replaced by an alternating path (of length 3,4 or 5) in the PASS gadget replacing t_i such that the first edge in this path is in M' if and only if $(s_{i-1}, t_i) \in M$. Similarly, the last edge is in M' if and only if $(t_i, s_i) \in M$. The edges (t_1, s_1) is replaced by the corresponding edge or pair of edges in the START gadget replacing t_1, and the edge (s_{k-1}, t_k) is replaced by the corresponding edge in the END_1 gadget replacing t_k.

Conversely, let P' be an augmenting path with respect to M' in G' ending at t_k. As the only other exposed vertices in G' are targets replaced by a START gadget, we get that P' must start at a target t_1 for which $d_M(t_1) = 1$. The only gadgets through which P' can pass are PASS gadgets that correspond to even targets with respect to M. A $\mathcal{P}_1$-augmenting path from t_1 to t_k can be obtained by collecting the sources and gadgets through which P' passes. □

4.2 The complete algorithm Given a mapping M, we first try to find a $\mathcal{P}_1$-augmenting path, a $\mathcal{P}_2$-augmenting path, or a $\mathcal{C}$-augmenting cycle. If such a path or a cycle is found, we of course use it to augment the mapping. To find such an augmenting path or cycle we do the following. For every target t_0 with $d_M(t_0) \geq 1$, we look for a $\mathcal{P}_1$-augmenting path ending at t_0. For every target t_0 with $d_M(t_0) \neq 2$, we look for a $\mathcal{P}_2$-augmenting path ending at t_0. For every deficient target t_0, i.e., $d_M(t_0) = 1$, we look for a $\mathcal{C}$-augmenting cycle through t_0. If no augmenting path or cycle is found, we proceed as follows. For every target t_0 with $d_M(t_0) = 0$, we look for a $\mathcal{P}'_1$-path ending at t_0. If such a path P is found, we let $M' \leftarrow M \oplus P$ (recall that $Def(M') = Def(M)$). We now look for an augmenting path or cycle with respect to M' that end at t_0. If found, we augment, otherwise, we revert back to M. Similarly, for every target t_0 with $d_M(t_0) = 2$, we look for a $\mathcal{P}'_2$-path ending at t_0, and then for an augmenting path or cycle ending at t_0. If no augmentation is found, the mapping M is optimal.

THEOREM 4.1. *The algorithm described finds an optimal mapping in $O(mn^2)$ time.*

5 Speeding up the algorithm

The running time of the algorithm described above is $O(mn^2)$ as in each iteration we have to examine all targets and look for augmenting structures ending at them. One way to reduce the running time of the algorithm to $O(mn)$ would be to devise an $O(m)$-time algorithm for finding an augmenting structure that ends at an arbitrary target. Unfortunately, we do not know how to do that. A second way of reducing the running time of the algorithm to $O(mn)$ would be to show that if augmenting structures ending at a given target t_0 were sought at a given iteration, then no matter whether they were found or not, there is no need to look for augmenting structures ending at t_0 again at subsequent iterations. Unfortunately, this is generally not the case. We show, however, that this claim is true if the mappings we work with are *maximum even mappings.*

DEFINITION 5.1. (MAXIMUM EVEN MAPPINGS) *Let $G = (S, T, E)$ be a bipartite graph. A mapping $M \subseteq E$ is a* maximum even mapping *if it has the maximum number of even targets among all mappings of G.*

Using the framework developed in the previous section, we can obtain an $O(mn)$-time algorithm for finding a maximum even mapping of a given graph. The proof of this claim is given in Section 6.

Our more efficient algorithm for finding an optimal mapping starts with a maximum even mapping. We show that a mapping obtained by augmenting a maximum even mapping remains a maximum even mapping.

LEMMA 5.1. *Let M be a maximum even mapping of G and let P be a $\mathcal{P}$-augmenting path or a $\mathcal{P}'$-path with respect to M from t_1 to t_2. Then $M' = M \oplus P$ is also a maximum even mapping of G. Furthermore, $d_M(t_1)$ is odd and $d_M(t_2)$ is even, while $d_{M'}(t_1)$ is even and $d_{M'}(t_2)$ is odd. Similarly, if C is a $\mathcal{C}$-augmenting cycle through t_2, then $M' = M \oplus C$ is also a maximum even mapping of G and both $d_M(t_2) = 1$ and $d_{M'}(t_2) = 3$ are odd. In both cases, the parities of all targets $t \neq t_1, t_2$ are unchanged, and if $d_M(t)$ is odd, then $d_M(t) = d_{M'}(t)$.*

Proof. Omitted due to lack of space. □

In the sequel, we refer to a pair (P, C) such that P is a $\mathcal{P}'$-path with respect to M that ends in a target t_2 and C is a $\mathcal{C}$-augmenting cycle through t_2 with respect to $M \oplus P$ as a $\mathcal{P}'\mathcal{C}$-augmenting pair with respect to M. To unify the terminology used, we also allow ourselves to say that the $\mathcal{C}$-augmenting cycle C and the pair (P, C) *end* at t_2.

As an immediate corollary of Lemma 5.1, we get that there is no need to look for augmenting paths that end at *odd* targets, as such augmenting paths cannot exist when M is a maximum even mapping. Furthermore, if a $\mathcal{P}$-augmenting path, or a $\mathcal{C}$-augmenting cycle, or a $\mathcal{P}'\mathcal{C}$-augmenting pair ending at a target t_2 is found, then after the augmentation we have $d_{M'}(t_2) \geq 3$ and $d_{M'}(t_2)$ is odd, and t_2 would never serve as the end of an augmentation structure.

We next show that if the search for an augmenting structure ending at a target t_2 *fails*, then there is again no need to look for augmenting structures ending at t_2 at subsequent iterations.

LEMMA 5.2. *Let M be a maximum even mapping of G and let R be an $\mathcal{R}$-structure ending at t_2, where $\mathcal{R} \in \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}'_1, \mathcal{P}'_2, \mathcal{C}\}$. If there is a $\mathcal{P}$-augmenting path P, or a $\mathcal{C}$-augmenting cycle C, or a $\mathcal{P}'\mathcal{C}$-augmenting pair (P, C) with respect to $M \oplus R$ ending at $t'_2 \neq t_2$, then there is a $\mathcal{P}$-augmenting path P', or a $\mathcal{C}$-augmenting cycle C', or a $\mathcal{P}'\mathcal{C}$-augmenting pair (P', C') with respect to M ending at t'_2.*

Proof. Omitted due to lack of space. □

Thus, if we start with a maximum even mapping, we only need to look once at augmenting structures ending at a given target. As a consequence, we get

THEOREM 5.1. *The algorithm described finds an optimal mapping in $O(mn)$ time.*

6 Maximum even mappings

In this section we describe an $O(mn)$-time algorithm for finding a maximum even mapping of a bipartite graph $G = (S, T, E)$, completing the description of the $O(mn)$-time algorithm for finding optimal 2-gatherings given in the previous section. Maximum even mappings are also interesting in their own right as they correspond, as we show below, to the problem of finding a maximum matching in a graph represented as a union of cliques.

Let V be a vertex set, and let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a collection of subsets of V. Construct a graph $G(\mathcal{C}) = (V, E)$ such that $(u, v) \in E$ if and only if there exists $C_i \in \mathcal{C}$ such that $u, v \in \mathcal{C}$. How efficiently can we find a maximum matching in G? If $|C_i| = \ell_i$, then G contains $O(\sum_{i=1}^{k} \ell_i^2)$ edges and a maximum matching in G can therefore be found in $O(n^{1/2} \sum_{i=1}^{k} \ell_i^2)$ time ([14],[16],[9]). Via a reduction to the problem of computing a maximum even matching in a suitably constructed graph, it is possible to solve the problem in $O(n \sum_{i=1}^{k} \ell_i)$ time, which is faster in many situations.

Given a collection $\mathcal{C} = \{C_1, \ldots, C_k\}$ of subsets of V, construct a bipartite graph $G'(\mathcal{C}) = (V, \mathcal{C}, E')$, where $(u, C_i) \in E'$ if and only if $u \in C_i$. It is not difficult to see that $G(\mathcal{C})$ has a matching with at most k unmatched vertices if and only if $G'(\mathcal{C})$ has a mapping with at most k odd targets. We omit the straightforward proof. We mention in passing that a somewhat related problem of finding maximum matchings in bipartite graphs represented as a union of bi-cliques is considered in [6].

The $O(mn)$-time algorithm for finding a maximum even matching is based on the following analogues of

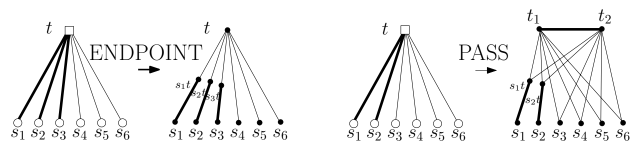

Figure 5: Target gadgets for maximum even mappings

Lemma 3.5 and Theorem 3.1. The proofs in this section are omitted due to lack of space.

LEMMA 6.1. *Let M be a mapping and let P be an even altering path between two odd targets with respect to M. Then, $M \oplus P$ is a mapping with more even targets.*

THEOREM 6.1. *A mapping M is maximum even mapping if and only if there is no even altering path between two odd targets with respect to M.*

This suggests the following natural algorithm for finding a maximum even mapping of G. Start with any mapping. While there exists an even altering path between two odd targets with respect to the current mapping, use it to improve the mapping. We next claim that such an improving path, if one exists, can be found in $O(m)$ time.

The search for improving paths is similar to the search for augmenting structures conducted in Section 4. Only two gadgets, ENDPOINT and PASS gadgets, shown in Fig. 5, are needed this time.

The ENDPOINT gadget is an extension of the START gadget for cases where the number of sources mapped to t is odd (but not necessarily one). The PASS gadget is identical to the PASS gadget used in Section 4.

To find an even altering path between two odd targets with respect to M in G, we construct a new graph G' and a matching M' in G' by replacing every odd target in G by an ENDPOINT gadget, and every even target in G by a PASS gadget.

LEMMA 6.2. *There is an even altering path between two odd targets with respect to M in G if and only if there is an augmenting path P with respect to M' in G'.*

As the graph G' contains only $O(m)$ edges, an augmenting path in G', and thus an improving path in G, if one exists, can be found in $O(m)$ time ([8]). The whole algorithm for finding maximum even mappings runs, therefore, in $O(m|S|) = O(mn)$ time.

7 Concluding remarks

We presented an $O(mn)$-time algorithm for the 2-gathering problem. It is an interesting open problem whether the running time of our algorithm could be improved to $O(m|S|)$,where S is the set of sources, or equivalently the set of vertices that contain pebbles, or even to $O(m|S|^{1/2})$, which would then match the complexity of the fastest known matching algorithms (see [14],[16],[9]). It is also an interesting open problem whether our algorithm could be extended to handle the *weighted* version of the 2-gathering problem.

Acknowledgments

The authors would like the thank Amitai Armon for helpful discussions.

References

[1] E. Anshelevich and A. Karagiozova. Terminal backup, 3D matching, and covering cubic graphs. In *Proc. of 39th STOC*, pages 391–400, 2007.

[2] A. Armon. On min-max r-gatherings. In *Proc. of the 5th WAOA*, pages 128–141, 2007.

[3] G. Cornuéjols. General factors of graphs. *Journal of Combinatorial Theory, Series B*, 45(2):185–198, 1988.

[4] G. Cornuéjols, D. Hartvigsen, and W. Pulleyblank. Packing subraphs in a graph. *Operations Research Letters*, 1(1):139–143, 1982.

[5] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[6] T. Feder and R. Motwani. Clique partitions, graph compression and speeding-up algorithms. *Journal of Computer and System Sciences*, 51:261–272, 1995.

[7] H.N. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proc. of 15th STOC*, pages 448–456, 1983.

[8] H.N. Gabow and R.E. Tarjan. A linear-time algorithm for a special case of disjoint set union. *Journal of Computer and System Sciences*, 30(2):209–221, 1985.

[9] H.N. Gabow and R.E. Tarjan. Faster scaling algorithms for general graph matching problems. *Journal of the ACM*, 38(4):815–853, 1991.

[10] P. Hell and D.G. Kirkpatrick. Packings by cliques and by finite families of graphs. *Discrete Mathematics*, 49(1):45–59, 1984.

[11] A. Karagiozova. *Aspects of Network Design*. PhD thesis, Princeton University, 2007.

[12] L. Lovász. Subgraphs with prescribed valencies. *Journal of Combinatorial Theory*, 8:391–416, 1970.

[13] L. Lovász. The factorization of graphs. II. *Acta Math. Acad. Sci. Hungar.*, 23:223–246, 1972.

[14] S. Micali and V.V. Vazirani. An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs. In *Proc. of 21st FOCS*, pages 17–27, 1980.

[15] A. Sebő. General antifactors of graphs. *Journal of Combinatorial Theory. Series B*, 58(2):174–184, 1993.

[16] V.V. Vazirani. A theory of alternating paths and blossoms for proving correctness of the $O(\sqrt{V}E)$ general graph maximum matching algorithm. *Combinatorica*, 14(1):71–109, 1994.