Even Orientations with Forbidden Pairs and Demands

Sarah Cannon* Mashhood Ishaque† Csaba D. Tóth‡

Abstract. It is known that every multigraph with an even number of edges has an even orientation. We present efficient algorithms for constructing even orientations under certain additional constraints. We consider three constraints: (1) at every vertex of indegree 2, the two incoming edges are not from a given set of *forbidden* pairs; (2) at each vertex at most half of the incoming edges are parallel; (3) the indegree of each vertex is greater or equal than a specified *demand* value.

1 Introduction

An *orientation* of the edges of an undirected multigraph is an assignment of a direction to each edge in the graph. Orientations with various properties have a wide range of applications. For example, *unique sink* orientations of polytopes are used for modeling pivot rules in linear programming, *Pfaffian* orientations are used for counting perfect matchings in a graph [4]. It is well known that every connected multigraph with an even number of edges has an *even orientation*, in which every vertex has even indegree. It is easy to check (with a network flow algorithm) whether a multigraph has an orientation such that the indegree of every vertex meets some prescribed demand.

Results. We present algorithms that, for a multigraph $G = (V, E)$, construct an even orientation that satisfy some additional constraints, or report that no such orientation exists. We consider the following three constraints:

1. Given a set $F \subseteq \binom{E}{2}$ of disjoint pairs of adjacent edges (*forbidden* pairs), we require that whenever the indegree of a vertex is 2, the two incoming edges are not a pair in F (Section 2).
2. At every vertex, at most half of the incoming edges are parallel (Section 2).
3. Given a *demand* function $f : V \to \mathbb{N}$, the indegree of every vertex v is at least $f(v)$ (Section 3).

If G has n vertices and m edges, then our algorithms run in $O(m^{2.5})$ time in the RAM model for the first two constraints, and in $O(m^{1.5}+n(n+m))$ time for the third constraint. Our first two constraints are motivated by the *Extension Conjecture*, due to Aichholzer *et al.* [2]. It is conjectured that for every even set of disjoint line segments in the plane, there is a convex decomposition such that the associated dual multigraph G admits an even orientation with the property that whenever a vertex v of G has indegree 2, the two incoming edges of v do not correspond to the two endpoints of the same line segment.

2 Even Orientations with Forbidden Pairs

Let G be a connected even multigraph. Since G has an even number of edges, it has an even orientation [3]. Given a set $F \subseteq \binom{E}{2}$ of forbidden edge-pairs, we want to find an even orientation satisfying the additional constraint that if a node has indegree exactly 2 from edges e_1 and e_2, then $\{e_1, e_2\} \notin F$. Every edge $e \in G$ can be in at most one forbidden pair $\{e_1, e_2\}$. An even orientation under this constraint does not always exist: consider for example a graph with only two edges, which form a forbidden pair.

We present a polynomial time algorithm that either constructs such an orientation or reports that no such orientation exists. We first build a modified line graph G' for G, then search for a perfect matching in G'. If G' has a perfect matching, we can construct a desired even orientation for G. If G' has no perfect matching, then G has no desired even orientation.

Definition 1 *For a multigraph G, the* **line graph** *G' has a node for each edge in G and an edge between any two nodes whose corresponding edges in G are adjacent.*

For our purposes, we will define the modified line graph on G as follows.

Definition 2 *For a multigraph $G = (V, E)$ with forbidden edge pairs $F \subseteq \binom{E}{2}$, the* modified line graph *G' is the line graph of G omitting edges (e_1, e_2) for all $\{e_1, e_2\} \in F$.*

Theorem 1 *A multigraph G has an even orientation respecting forbidden pairs if and only if the modified line graph G' has a perfect matching.*

Proof. First, suppose that G' has a perfect matching M'. We construct an even orientation on G that respects forbidden pairs. For every edge $(e_1, e_2) \in M'$,

*Department of Mathematics, Tufts University, Medford, MA, sarah.cannon@tufts.edu

†Department of Computer Science, Tufts University, Medford, MA, mishaque@cs.tufts.edu

‡Department of Mathematics and Statistics, University of Calgary, Calgary, AB, Canada, cdtoth@ucalgary.ca

direct both e and e' towards one of their common endpoints in G. Since pairs of edges are directed towards each vertex of G, every vertex must have an even indegree. Since forbidden pairs $\{e_1, e_2\} \in F$ are not connected in G', they are not matched in M', and thus there will be no vertex in G with indegree 2 where the two incoming edges are a forbidden pair.

Next suppose that G has an even orientation respecting forbidden pairs. We will show that G' has a perfect matching. Consider a node v in G. There are an even number of incoming edges at v. Partition the incoming edges at v into two classes of equal size such that the forbidden pairs are in different classes (recall that every edge participates in at most one forbidden pair). Match the incoming edges within the two classes arbitrarily. We have matched adjacent edges, but no forbidden pairs. After we have done this for all vertices of G, these pairs of edges in G correspond to a perfect matching of G'. □

2.1 Algorithm for Even Orientations with Forbidden Pairs

We use the following algorithm for constructing a desired even orientation or report that none exists.

Algorithm 1 EvenOrientationParallel(G)

```
Create the modified line graph G'.
Find a maximum matching M' on G'.
if M' is a perfect matching then
    Convert M' into an even orientation.
else
    Report no such orientation exists.
end if
```

For a multigraph $G = (V, E)$ with n vertices and m edges, the line-graph G' has m nodes and up to $O(m^2)$ edges. We use the Ford-Fulkerson max-flow algorithm [1] to find a maximum matching, it runs in $O(m^3)$ time (since maximum flow cannot exceed the number of edges here). Since G' is a unit-capacity graph, Dinic's blocking flow algorithm [1] gives a runtime of $O(m^{2.5})$ in the RAM model.

2.2 Parallel-Restricted Even Orientations

Let G be an even multigraph. We consider constraint (2), in which we look for an even orientation such that at every vertex at most half of the incoming edges are parallel—a *parallel-restricted even orientation* of G. Two edges in G are *parallel* if they join the same two nodes.

A parallel-restricted even orientation can be reduced to a special case of the even orientation with forbidden pairs described above. Let $F = \{\{e_1, e_2\} : e_1$ and e_2 are parallel$\}$. Even though the forbidden pairs in F are

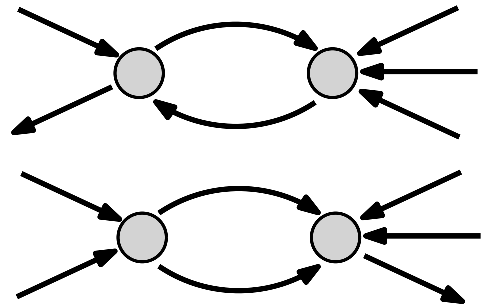

Figure 1: Two examples of valid orientations of parallel edges.

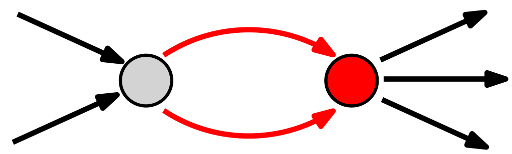

Figure 2: An invalid orientation: the red node has indegree two and those two edges are parallel.

not necessarily disjoint (for parallel edges of multiplicity 3 or higher), we can repeat the proof of Theorem 1 almost verbatim. In particular, if an even orientation under constraint (2) exists, then at each vertex of G, we can match the incoming edges such that no two parallel edges are matched. Since at least half of the incoming edges are not parallel, we can pair each parallel edge with a non-parallel one. Similarly if there is a perfect matching on the modified line graph corresponding to this set F, there exists a parallel-restricted even orientation on G: for each parallel edge contributing to the indegree of a node, there is a corresponding non-parallel edge.

3 Even Orientations with Lower Bounds on Indegree

Next we consider even orientations under constraint (3), involving lower bounds on the indegree of each vertex of G. Consider a multigraph G with a demand function $f : V(G) \to \mathbb{N}$ on the vertices. We wish to find an even orientation of the edges of G such that the indegree of every vertex v is at least $f(v)$, or report that there is no such orientation. Since we look for an orientation in which all nodes have even indegree, we can assume $f(v)$ is even for all v.

We now present an algorithm that does exactly this. First, we obtain an initial orientation on G, which meets the demands, but some vertices may have odd indegree. Then, we attempt to correct this initial orientation through augmenting paths (defined below), each of which changes the indegree of two odd nodes to even without violating the demands. We show that if we cannot decrease the number of odd nodes to zero through augmenting paths, then there is no desired even orientation.

Necessary Condition 1 *For every $k = 1, \ldots, n$, every k nodes $\{v_1, \ldots, v_k\} \subseteq V$ must jointly have at least $\sum_{i=1}^{k} f(v_i)$ incident edges.*

This necessary condition ensures there are enough edges incident on every subgraph of G to meet the demands placed on each node. We can use a max-flow algorithm on a modified graph to check this condition as follows.

First, we create a bipartite graph G' based on the graph G and the demand function $f(v)$. Let $V = \{v_1, v_2, \ldots, v_n\}$ be the set of nodes in G and let $E = \{e_1, e_2, \ldots, e_m\}$ be the set of edges. Create a node in G' for every element of $V \cup E$. Let a pair (v_i, e_j) be an edge of G' if an only if v_i is incident to e_j. Set the capacity of each edge (v_i, e_j) equal to 1. Add a source node s, and edges (s, v_i) for every node v_i, and set the capacity of each edge (s, v_i) equal to $f(v_i)$. Similarly, add a sink node t; and edges edge (e_j, t) for every node e_j, and set the capacity of each edge (e_j, t) equal to 1. Find the maximum flow in graph G'. If the out-flow of the source s is less than $\sum_{i=1}^{n} f(v_i)$, then the necessary condition is not met.

If our graph passes this test, then we can use the resulting maximum flow solution to start our search for a solution, if one exists. For each edge (v_i, e_j) that has non-zero flow in G', orient edge e_i in G towards the node v_i. Orient the remaining edges of G arbitrarily. Now we have an orientation on G that meets the demands, but some of nodes may have odd indegree. Since m is even, an even number of nodes have odd indegree. We can now try to restore an even orientation without reducing the indegree of any node v to less than $f(v)$ by searching for augmenting paths connecting two nodes of odd indegree.

Definition 3 *An* **augmenting path** *in G is an undirected path connecting two odd nodes with the additional property that reversing the orientation of every edge along that path will not decrease the indegree of any node v below $f(v)$.*

This means, for example, that if an augmenting path enters a node v with $f(v) = 2$ and indegree exactly two, then the path cannot exit that node along its other incoming edge. Figure 3 provides two examples of augmenting paths through nodes with demand 2. Figure 4 provides an example of an invalid augmenting path—reversing the orientation of every edge along this path would decrease the indegree of the middle node to zero. However, if a node has indegree 4 and demand 2, then an augmenting path can enter and leave the node along

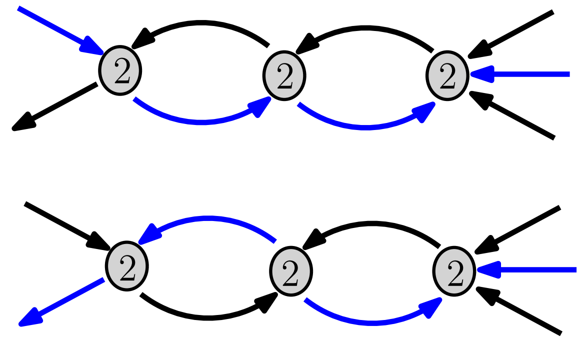

Figure 3: Augmenting paths: reversing the edges along the path retains the even orientation and the indegree remains at least 2. The labels in the nodes show the demands.

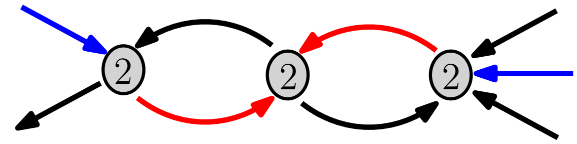

Figure 4: Invalid path: reversing the edges along the path, the indegree of a node becomes zero.

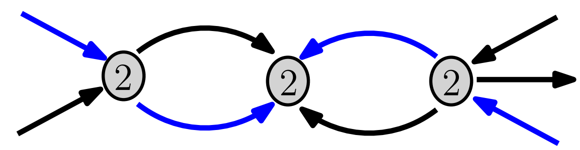

Figure 5: An augmenting path: a path is allowed to go through a node with indegree 4.

incoming edges (see Figure 5). Note also that an augmenting path does not have to be simple, it may have repeated vertices.

If we reverse the orientation of every edge along an augmenting path, this changes the indegree of the two odd nodes at each end by one, making these nodes even, while it does not change the parity of the indegree of any node along the path. Since this is an augmenting path, all demands $f(v)$ are still met after reversing the orientation of every edge along the path.

Theorem 2 *Let G be an even multigraph with a demand function $f : V \to \mathbb{N}$, and let O be the set of all orientations that meet the demands, and suppose $O \neq \emptyset$. Then an orientation $o \in O$ has the maximum number of even vertices if and only if there is no augmenting path.*

Proof. Let $o \in O$ be an orientation on G in which all demands are met. First, it is trivial that if o is an orientation that has the maximum number of even vertices, then there is no augmenting path in G, since if there is an augmenting path, the number of even indegree nodes can be increased by at least 2.

Suppose o is not an orientation with the maximum number of even vertices; we will prove that there is an augmenting path in G. For any two orientations $o, o' \in O$, let $D(o, o')$ be the difference multi-graph of all edges that have opposite orientations in o and o'. If o is not an orientation with the maximum number of even vertices, then there is some other orientation in O with more even vertices. Let $o' \in O$ be an orientation that has more even vertices than o and the difference $D = D(o, o')$ is minimal. Let $V^+ \subset V$ be the set of vertices that are odd in o and even in o', and let $V^- \subset V$ be those that are even in o and odd in o'. Clearly, we have $|V^+| \geq |V^-| + 2$. Clearly, only the nodes in $V^+ \cup V^-$ have odd degree in D. Observe that D must be connected by the minimality of D.

Let v_0 be an arbitrary vertex in V^+. We construct an augmenting path in D starting from v_0, by appending new edges to a path incrementally. Let (v_0, v_1) be an arbitrary edge in D incident to v_0, and let our initial path be $\pi_0 = (v_0, v_1)$. In a general step, we have a path $\pi_i = (v_0, \ldots, v_i)$ in D such that if we reverse the orientation along π_i (from o to o'), then every internal vertex of π_i still meets their demand, and the last vertex v_i has indegree at least $f(v_i) - 1$. If we reach another vertex $v_i \in V^+$, $v_i \neq v_0$, then π_i is an augmenting path. We cannot reach a vertex $v_i \in V^-$, since $D \setminus \pi_i$ would be a strictly smaller difference graph between o and an even orientation $o'' \in O$, contradicting the minimality of D. Hence we may assume that the vertices $v_1, v_2, \ldots, v_i$ have even degrees in D.

It is enough to show that we can append a new edge to π_i while maintaining the above properties. Vertex v_i is the endpoint of path π_i, and so it is incident to an odd number of edges of π_i. Hence it is incident to an odd number of edges in $D \setminus \pi_i$. By induction, if we reverse the orientation along all edges of π_i (from o to o'), then the in-degree of v_i is at least $f(v_i) - 1$. Recall that if we reverse the orientation of *all* edges in D, then the in-degree of v_i is still at least $f(v_i)$. Therefore, there must an edge $e \in D \setminus \pi_i$ incident to v_i such that by reversing the orientation of $\pi_i \cup \{e\}$, the indegree of v_i remains at least $f(v_i)$. We append this edge e to π_i to construct path π_{i+1}. This completes one step of the induction, and the proof of the theorem. □

3.1 Algorithm for Even Orientations with Lower Bound Demands

We now present our algorithm for finding an even orientation on G that meets the demands $f(v)$.

As described above, run a network flow algorithm on the modified graph G' to check for the necessary condition. If there is a feasible flow meeting the demands $f(v)$, use the resulting flow to determine an initial orientation on G. In this initial orientation, all demands $f(v)$ are met, but there may be an even number of nodes with odd indegree.

Algorithm 2 EvenOrientationLowerBound(G)

```
Check for the necessary condition using max flow.
if there exists a flow meeting demands then
  Create an initial orientation of G based on the flow.
  for all nodes of odd indegree do
    Find an augmenting path to another odd node.
    Reverse every edge along the path.
  end for
end if
```

Pick an arbitrary node with odd indegree; search for an augmenting path originating at this node using breadth-first-search. Reverse the orientation of every edge along this path. After each breadth-first-search we either reduce the number of odd-indegree nodes by 2 or conclude that there is no such augmenting path. If there is no node with odd indegree which is the starting point of an augmenting path, then the current orientation has the maximum number of even vertices among all orientations that meet the demands. As there are still odd nodes in this orientation, there is no even orientation on G respecting the lower bound demands given by $f(v)$.

For the graph G with n vertices and m edges, the graph G' has $O(m + n)$ nodes and $O(m)$ edges. The graph G' can be made unit-capacity by introducing $\sum_v f(v) = O(m)$ auxiliary nodes, thus using the Dinic's blocking flows algorithm we can find the maximum flow in $O(m\sqrt{m + n})$ time. To find the augmenting paths, we perform at most n breadth-first searches in $O(n(m + n))$ time. Thus the total time complexity of the algorithm is $O(m^{1.5} + n(n+m))$ in the RAM model.

References

[1] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, February 1993.

[2] Aichholzer, O., Bereg, S., Dumitrescu, A., García, A., Huemer, C., Hurtado, F., Kano, M., Márquez, A., Rappaport, D., Smorodinsky, S., Souvaine, D., Urrutia, J., and Wood, D. R. Compatible geometric matchings. *Comput. Geom. Theory Appl. 42* (2009), 617–626.

[3] Frank, A., Jordan, T., and Szigeti, Z. An orientation theorem with parity conditions. *Discrete Appl. Math. 115*, 1-3 (2001), 37–47.

[4] Lovász, L., and Plummer, M. D. *Matching Theory.* AMS Chelsea, 2009.