How to find in polynomial time a well balanced orientation

S. Francfort

2002

Abstract

H. Gabow in [4] claim that A. Frank's proof [1] of the existence of an odd vertex pairing could be turned into an algorithm. The proof of the algorithm doesn't stand in his paper for a lack of space. So is the case of the proof of its polynomiality. In this paper, I follow Gabow's hint to present an algorithm which find an odd vertex pairing. Here is also written the proof of the algorithm and of its polynomiality.

1 Introduction

Given a graph $G = (V, E)$ and to vertices x and y in V, let $\lambda(x, y; G)$ denote the local edge connectivity from x to y in G, that is the number of edge-disjoint paths from x to y.
Let's call an orientation $D = \vec{G}$ **well balanced** if

$$\lambda(x, y; D) = \lfloor \lambda(x, y; G)/2 \rfloor \text{ holds for every } x, y \in V$$

Theorem 1 *(Nash-Williams) Every graph G has a well balanced orientation* ∎

Let's remark that any di-eulerian orientation of an eulerian graph is well balanced. Nash-Williams' idea is to make any given graph G eulerian by adding a set of edges matching the vertices of odd degree. Let's call such a set of edges M an **Odd Vertex Pairing** if

$$d_M(X) \leq b_G(X) \text{ holds for every } X \subseteq V$$

Theorem 2 *(Nash-Williams) Every graph G has an Odd Vertex Pairing.*∎

Nash-Williams' observation is: if M is an Odd Vertex Pairing of the graph G, any di-eulerian orientation of the eulerian graph $G + M$ is a Well Balanced Orientation of the graph G.

The purpose of this paper is to present an algorithm which find a well balanced orientation of a given graph G.
This algorithm by finding an Odd Vertex Pairing finds also a Well Balanced Orientation.

2 Preliminaries

We need some definitions:

Let $G = (V, E)$ be a graph.
If f is an integer valued function, we denote $\hat{f} = 2\lfloor f/2 \rfloor$.
Let's call $X \subseteq V$ trivial if $|X| \leq 1$ or $|(V - X)| \leq 1$, non trivial otherwise.
Let $d_G(X, Y)$ denote the number of edges between $X - Y$ and $Y - X$ in the graph G, and let $d_G(X)$ be $d_G(X, V - X)$. Let $\overline{d}_G(X, Y) = d_G(X, V - Y)$.
Let $R_G(X) = \max\{\lambda(x, y; G); x \in X \text{ and } y \in V - X\}$.
Let $b_G(X) = d_G(X) - \hat{R}_G(X)$.
For $e \in E$, let us call $G - e$ "$\hat{\lambda}_G$-connected" if $\lambda(x, y; G - e) \geq \hat{\lambda}(x, y; G) \forall x, y \in V$.
For $X \subseteq V$, let G/X be the graph where the vertices in X have been contracted into one vertex.
Splitting off a pair of edges $e = su$, $f = st$ means that we replace e and f by a new edge ut. The resulting graph is denoted: G^{ef}.
We say that X **separates** x and y, if $|X \cap \{x; y\}| = 1$. We call such an X **separating**.

And some properties:

Lemma 3 *For any $X, Y \subseteq V$, at least one of the following four inequalities holds.*

$$\hat{R}(X) \leq \hat{R}(X \cup Y) \text{ and } \hat{R}(Y) \leq \hat{R}(X \cap Y) \quad (1)$$
$$\hat{R}(Y) \leq \hat{R}(X \cup Y) \text{ and } \hat{R}(X) \leq \hat{R}(X \cap Y) \quad (2)$$
$$\hat{R}(X) \leq \hat{R}(X - Y) \text{ and } \hat{R}(Y) \leq \hat{R}(X - Y) \quad (3)$$
$$\hat{R}(Y) \leq \hat{R}(X - Y) \text{ and } \hat{R}(X) \leq \hat{R}(X - Y) \quad (4)$$

which implies:

Lemma 4 *For any $X, Y \subseteq V$, at least one of the following inequalities holds.*

$$\hat{R}(X) + \hat{R}(Y) \leq \hat{R}(X \cup Y) + \hat{R}(X \cap Y) \quad (5)$$
$$\hat{R}(X) + \hat{R}(Y) \leq \hat{R}(X - Y) + \hat{R}(Y - X) \quad (6)$$

moreover, if

$$R(X) \leq \min(R(Y), R(X \cap Y), R(X \cup Y)) \quad (7)$$

then (5) holds.

Proposition 5 *For $X, Y \subseteq V$,*

$$d_G(X) + d_G(Y) = d_G(X \cup Y) + d_G(X \cap Y) + 2d_G(X, Y) \quad (8)$$
$$d_G(X) + d_G(Y) = d_G(X - Y) + d_G(Y - X) + 2\overline{d}_G(X, Y) \quad (9)$$

3 Algorithm

The algorithm $\mathcal{M}$ takes a graph as input and computes an Odd Vertex Pairing.
Algorithm $\mathcal{M}(G)$:
INPUT: A graph G.
OUTPUT: An Odd Vertex Pairing M.

- END-TEST-STEP(G):
  - If $d_G(v)$ even $\forall v \in V$, then set $M = \emptyset$.
  - Else Call Delete-Edge-Step(G).
- DELETE-EDGE-STEP(G):
  - If there doesn't exist $u, v \in V$ s.t. $uv \in E$ and $d_G(u), d_G(v)$ odd then Call Split-Step-1(G).
  - Else set u, v s.t. $uv \in E$ and $d_G(u), d_G(v)$ odd
    - If $G - uv$ isn't $\hat{\lambda}_G$-connected then Call Contraction-Step-1(G).
    - Else set $M = \mathcal{M}(G - uv) + uv$.
- CONTRACTION-STEP-1(G):
  - Find x, y s.t. $\lambda(x, y; G - uv) < \hat{\lambda}(x, y; G)$.
  - Find $X \subset V$ s.t. X separates x and y and $d_G(X) = \lambda(x, y; G)$ by a max-flow algorithm .
    Note that X is non trivial and that $b_G(X) = 0$.
  - Set $M = \mathcal{M}(G/X) + \mathcal{M}(G/(V - X))$.
- SPLIT-STEP-1(G):
  - Let $S = \{v \in V; d_G(v) \neq 3\}$.
  - If $|S| \neq 1$ then Call Split-Step-2(G).
  - Else find $e = su, f = sv$ splittable and set $M = \mathcal{M}(G^{ef})$.
- SPLIT-STEP-2(G):
  - Find $s \in S$ s.t. $d_G(s)$ is minimum.
  - Compute $\lambda_s = \min\{\lambda(x, y; G); x, y \in S\}$.
  - If $\lambda_s \neq d_G(s)$ Call Contraction-Step-2(G).
  - Else find a pair $e = su, f = sv$ splittable and set $M = \mathcal{M}(G^{ef})$.
- CONTRACTION-STEP-2(G):
  - Find $x, y \in V$ s.t. $\lambda(x, y; G) = \lambda_s$.
  - Find $A \subset V$ s.t. A separates x and y, and $d(A) = \lambda(x, y; G)$ by a max-flow algorithm.
    Note that A is non trivial and that $b_G(A) \leq 1$.

- If $b(A) \neq 0$ then Call Contraction-Step-3(G, A).
- Else Set $M = \mathcal{M}(G/A) + \mathcal{M}(G/(V-A))$.

- CONTRACTION-STEP-3(G, A):
  - Let a (resp. $\overline{a}$) be the vertex appearing while contracting A (resp. $V - A$) into one vertex.
  - Let $e_1 = av_1$ in $\mathcal{M}(G/A)$ and $e_2 = \overline{a}v_2$ in $\mathcal{M}(G/(V-A))$.
  - Set $M = \mathcal{M}(G/A) + \mathcal{M}(G/(V-A)) - e_1 - e_2 + v_1v_2$.

4 Proof of the correctness of the algorithm

Claim 6 *Let G be a graph, and let $X \subset V$ be non trivial s.t. $b_G(X) = 0$. Let M_1 (resp. M_2) be an Odd Vertex Pairing of G/X (resp. of $G/(V-X)$) then $M_1 + M_2$ is an Odd Vertex Pairing of G.*

Proof: To prove this, we need to prove that $d_M(Y) \leq b_G(Y)\,\forall Y \subset V$. To see this, we may assume that of lemma 3(1) holds since otherwise we can replace Y by its complement and then 3(1) transforms into 3(4).
Using proposition 5, the fact that $\hat{R}$ is symmetric and that $b_G(X) = 0$ we get

$$d_M(Y) = d_M(X \cap Y) + d_M(X \cup Y) = d_{M(G/(V-X))}(X \cap Y) + d_{M(G/X)}(V - (X \cup Y))$$
$$\leq b_{G/(V-X)}(X \cap Y) + b_{G/X}(X \cup Y) \leq b_G(X \cap Y) + b_G(X \cup Y)$$
$$\leq b_G(X) + b_G(Y) \text{ as required.}$$

■

The proof of the correctness of the algorithm M is by induction on p with $p = |V| + |E|$.

We begin with the *End-Test-Step*:
If $p = 0$, then $\mathcal{M}(G)$ returns $M = \emptyset$ by the End-Test-Step.
If $p > 0$ then if $d(v)$ is even for all $v \in V, \mathcal{M}(G)$ returns $M = \emptyset$ by the End-Test-Step.
Therefore, $\exists u, v \in V$, s.t. $d(u)$ and $d(v)$ are odd.

We are now in the *Delete-Edge-Step*:
Now, if there exists such u and v and uv is an edge, there are two cases:
First case: $G - uv$ is $\hat{\lambda}_G$-connected. In this case, by induction on p, $\mathcal{M}(G - uv)$ is an Odd Vertex Pairing of $G - uv$. And we claim that $\mathcal{M}(G - uv) + uv$ is an Odd Vertex Pairing of G.
To prove this, we can notice that $\hat{\lambda}(x, y; G - uv) = \hat{\lambda}(x, y; G)\,\forall x, y \in V$ and thus, $b_{G-uv}(X) = b_G(X) - 1$ (resp. $b_{G-uv}(X) = b_G(X)$) $\forall X \subset V$ separating (resp. non separating) u and v, and this is what we wanted.

The Second case begins in the *Contraction-Step-1*:
$G-uv$ is not $\hat{\lambda}_G$-connected. In this case, we claim that we can find a non trivial set $X \subset V$ s.t. $b_G(X)=0$. Let $x,y \in V$ s.t. $\hat{\lambda}(x,y;G-uv) < \hat{\lambda}(x,y;G)$. Then $\lambda(x,y;G-uv) = \lambda(x,y;G)-1$ and $\lambda(x,y;G)$ is even. By Menger's theorem there is an $X \subset V$ for which $d_G(X) = \lambda(x,y;G)$.
By taking the complement if necessary, we may assume that $|X| \leq |V-X|$. Now, by $b_G(X)=0$ we get $d_G(X)$ even. But $d_G(u)$ and $d_G(v)$ are odd and then $|X| \geq 2$ and $|(V-X)| \geq 2$ that is exactly: X non trivial.
Again, by induction on p, $\mathcal{M}(G/X)$ and $\mathcal{M}(G/(V-X))$ are Odd Vertex Pairings of G/X and $G/(V-X)$. With the claim 6, we can conclude that $\mathcal{M}(G/X)+\mathcal{M}(G/(V-X))$ is an Odd Vertex Pairing of G.
So, we can assume that there aren't any edges between two vertices of odd degree.

Remark (FRANK): $d_M(X) \leq b_G(X)\, \forall X \subset V$ essential implies $d_M(X) \leq b_G(X)\, \forall X \subset V$.

This is now the *Split-Step-1*:
Let S be the set of vertices v with $d_G(v) \neq 3$, and s be the vertex of minimum degree of S.
If $|S|=1$ then, s is joined to any other vertices of G by three parallel edges. So $d_G(s)$ is even and by Mader's theorem, there exist a pair $e=su, f=sv$ splittable. By induction on p, $\mathcal{M}(G^{ef})$ is an Odd Vertex Pairing of G^{ef}. We claim that it is also an Odd Vertex Pairing of G. Just remark that since $d_{G^{ef}} \leq d_G$ then $b_{G^{ef}} \leq b_G$ for any essential sets.
Hence M is an Odd Vertex Pairing for G as well.

At this point begins the *Split-Step-2*:
Therefore, $|S|>1$. Let $\lambda_s = \min\{\lambda(x,y;G); x,y \in S\}$. Clearly, $\lambda_s \leq d_G(s)$.
There are once again two cases.
First case: $\lambda_s = d_G(s)$. By Mader's theorem there exist a pair $e=su, f=sv$ splittable and we claim that $\mathcal{M}(G^{ef})$ is an Odd Vertex Pairing of G. Just remark that since $d_{G^{ef}} \leq d_G$ then $b_{G^{ef}} \leq b_G$ for any essential sets.
Hence M is an Odd Vertex Pairing for G as well.

This case is the *Contraction-Step-2*'s case:
Second case: $\lambda_s < d(s)(*)$. Let x and y be such that $\lambda(x,y;G) = \lambda_s(**)$, and let $A \subset V$ separating x and y be such that $d_G(A) = \lambda(x,y;G)(***)$. (*), (**) and (***) implies that A is non trivial.
If $b_G(A)=0$ then, by induction on p, $\mathcal{M}(G/A)$ and $\mathcal{M}(G/(V-A))$ are Odd Vertex Pairings of G/A and $G/(V-A)$. With the claim 6, we can conclude that $\mathcal{M}(G/A)+\mathcal{M}(G/(V-A))$ is an Odd Vertex Pairing of G.
So we can assume that $b_G(A) \geq 1$, but we can see that

$$R_G(A) \geq \lambda(x,y;G) = d_G(A) \geq R_G(A) \text{ and hence } d_G(A) = R_G(A)$$

And hence: $b_G(A) \leq 1$.

We are now in the *Contraction-Step-3*:
This leads us to the last case where $b_G(A) = 1$.

By induction on p, $\mathcal{M}(G/A)$ and $\mathcal{M}(G/(V-A))$ are Odd Vertex Pairings of G/A and $G/(V-A)$. Using the same notation as in the algorithm, we claim that $\mathcal{M}(G/A) + \mathcal{M}(G/(V-A)) - e_1 - e_2 + v_1v_2$ is an Odd Vertex Pairing of G. As we remarked, it suffices to show that $d_M(X) \leq b_G(X)\, \forall X$ essential.
By replacing X with its complement if necessary, we may assume that $d_M(A, X) = 0$. Because X is essential and by the fact that at this point, each edge has at least one end vertex of even degree, $S \cap A \cap X \neq \emptyset$ and $S \cap (V - (A \cup X)) \neq \emptyset$. From the choice of A it also follows that $R(A) \leq R(Z)$ holds for every set $Z \subset V$ separating two elements of S.
Hence (7) follows and therefore (5) holds for A, X.
Using proposition 5 and that $\hat{R}$ is symmetric we get:

$$d_M(X) = d_M(A \cap X) + d_M(A \cup X) - d_M(A) + 2d_M(A, X)$$

$$= d_M(A \cap X) + d_M(A \cup X) - 1$$

$$= d_{M(G/(V-X))}(A \cap X) + d_{M(G/X)}(V - (A \cup X)) - 1$$

$$\leq b_{G/(V-X)}(A \cap X) + b_{G/X}(A \cup X) - 1$$

$$\leq b_G(A \cap X) + b_G(A \cup X) - 1$$

$$\leq b_G(A) + b_G(X) - 1 = b_G(X)$$

This ends the proof. ■

5 Proof of the polynomiality of the algorithm

Remark first that each step of the algorithm $\mathcal{M}$ may contain tests which stop this step to call another step. The steps that are not stopped always end with a recursive call to the algorithm $\mathcal{M}$.
The recursive calls may be double when the step calls $\mathcal{M}$ on two contracted graphs, like Contraction-Step-1, Contraction-Step-2 and Contraction-Step-3.
These may also be simple, when the step calls $\mathcal{M}$ on only one graph which has got one edge less, like in Delete-Edge-Step, Split-Step-1, or Split-Step-2.
Remark also that each recursive call to $\mathcal{M}$ goes straightforward to another double or simple recursive call to $\mathcal{M}$.
Thus, it is sufficient, to obtain an upper bound on the number of elementary computations, to multiply the number of recursive calls to $\mathcal{M}$ by the maximum number of elementary computations in one call to $\mathcal{M}$.

DEFINITION: Let $\mathcal{F}$ be a family of sets on the same ground set E.
Let's call $\mathcal{F}$ Laminar if and only if either $X \cap Y = \emptyset$ or $X \subset Y$ or $Y \subset X\, \forall X, Y \in$

$\mathcal{F}$.

Proposition 7 *Let $\mathcal{F}$ be Laminar on the ground set E. Then $|\mathcal{F}| \leq 2|E|$.*

proof:
The proof is by induction on $|V|$. We can assume that $|V| \geq 2$ and $V \in \mathcal{F}$.
Let U be the inclusionwise maximal set in $\mathcal{F}$ with $|U| \geq 2$. Reseting $\mathcal{F}$ to $\mathcal{F} - \{v; v \in U\}$, and identifying all elements in U, $|\mathcal{F}|$ decreases by at most $|U|$ and $|V|$ by $|U| - 1$. Since $|V| \geq 2(|U| - 1)$ as $(|U| \geq 2)$, induction gives the required inequality.■

Claim 8 *The number of double recursive calls to $\mathcal{M}$ is bounded by $2n$.*

Proof: Set $\mathcal{X} = \{X \subset V(G); \mathcal{M}(G/X)$ is called in the algorithm$\}$.
Notice that $|\mathcal{X}|$ is the number of double recursive calls.
Observe that since each X in $\mathcal{X}$ contains contracted vertices, we can expand them in order to make each X in $\mathcal{X}$ be sets on the same ground set $V(G)$. Let's call those sets $\overline{X}$ and $\overline{\mathcal{X}}$. Clearly $|\overline{\mathcal{X}}| = |\mathcal{X}|$ and since $\overline{\mathcal{X}}$ is a Laminar family we can conclude the proof.■

Claim 9 *The number of recursive calls to $\mathcal{M}$ is bounded by $2n + m$.*

Proof: Remark that each simple recursive call decreases the number of edges by one. This implies that the number of simple calls is bounded by m. It is easy to see that the total number of calls is bounded by the sum of the number of simple calls and the number of double calls and this suffices to prove the claim.■

Claim 10 *The number of elementary computations from the begining of the End-Test-Step to any recursive call to $\mathcal{M}$ is polynomial.*

Proof: It's easy to see that each step can be computed in polynomial time.
It's also straightforward to verify that the number of steps explored from the begining of the End-Test-Step to any recursive call to $\mathcal{M}$ is bounded by 6 (cf. figure). We can thus conclude the proof.■

Hence, we can affirm that an Odd Vertex Pairing can be found in polynomial time by the algorithm $\mathcal{M}$.

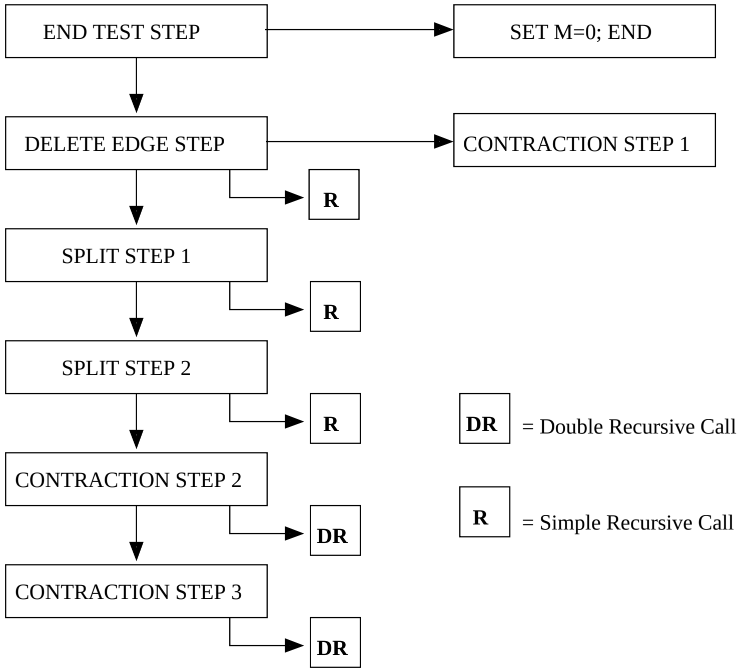

Figure 1: Sketch of the algorithm $\mathcal{M}$

References

[1] A. Frank, *Applicatons of submodular functions,* in: Surveys in Combinatorics, London Mathematical Society Lecture Note Series 187, Cambridge Univ. Press, (Ed. K. Walker) 1993, pp. 85-136

[2] C.St.J.A. Nash-Williams, *On orientations, connectivity and odd vertex pairings in finite graphs,* Canad. J. Math. 12, 1960, pp.555-567

[3] C.St.J.A. Nash-Williams, *Well-balanced orientations of finite graphs and unobstrusive odd-vertex-pairings,* in: Recent Progress of Combinatorics, 1969, pp.133-149

[4] H.N. Gabow, *Efficient Splitting Off Algorithms for Graphs,* in: STOC 1994, pp.696-705