AN l_∞ BALANCING OF A WEIGHTED DIRECTED GRAPH

Hans Schneider and Michael H. Schneider

1 Introduction

A problem that occurs frequently in economics, urban planning, image reconstruction, and statistics is to adjust the entries of a large matrix so that they satisfy prior linear restrictions on the entries. We have shown in [10] and [9] that important instances of these problems can be posed as

Problem 1 *Given a weighted, directed graph, (X,U,g), find arc weights, $f_u, u \in U$ that are "close" to the original weights and satisfy a given set of restrictions on the entries.*

For example, in [9] we studied the problem of finding vertex weights π_x for which $\pi_x g_u \pi_y^{-1}$, $u = (x,y)$, is a circulation in the underlying graph, (X,U). We will refer to adjustments of this form as *Scaling* the data. In [9] we discussed the relationship between this scaling problem and general equilibrium modeling and analyzed a simple-iterative algorithm for finding the π's.

The circulation conditions are linear restrictions requiring that at every vertex, x, the l_1 norm of the vector of weights on arcs directed out of x equals the l_1 norm of the vector of weights on arcs directed into x. A related problem of scaling the data so that these vectors have equal l_2 norms occurs in pre-conditioning of square matrices to reduce round-off error in computing eigenvalues [8]. The generalization to requiring equality of arbitrary l_p norms for $1 \leq p < \infty$ can be reduced to the l_1 balancing problem. The case of requiring equality in the l_∞ norm produces a different problem which, apparently, cannot be solved efficiently using the techniques described in [10] and [9].

In this paper we are interested in showing that the arc weights of G can be adjusted so that every strong component is *balanced* using a definition based on the l_∞ norm and on a stronger notion of circulation. The term *balanced* appears in many contexts in graph theory, optimization, and matrix theory. Our definition of *balanced* is related to matrix balancing as described in [2,10]

In the case of l_1-balancing, it is easy to see that balancing every vertex of G implies that G is also balanced at *cutsets*. That is, if the arc-weights of G form a circulation, then for any subset of the vertices A, the sum of the weights on arcs directed out of A equals the sum of the weights on arcs directed into A. The corresponding statement is not true with respect to the l_∞ norm. This stronger circulation condition based on cutsets is the appropriate definition in the l_∞ case. That is, we define a graph to be l_∞-*balanced* if for every subset of the vertices A the maximum weight on arcs directed out of A equals the maximum weight on arcs directed into A. We call graphs with this property *balanced* and delete the l_∞ prefix.

We describe an algorithm for finding additive adjustments to the original arc-weights so that the resulting graph is balanced. Specifically, given arbitrary arcs weights, $g_u, u \in U$, we want to find vertex weights $\pi_x, x \in X$, such that the weight function $f_u = \pi_x + g_u - \pi_y$, $u = (x, y) \in U$, is balanced. (We show in Section (3) that this problem has an equivalent multiplicative form.) We show that if a graph is balanced, then every arc of G must be contained in a strong component. Thus, we consider the problem of finding vertex weights $\pi_x, x \in X$, for which every strong component of the reweighted graph (X, U, f) is balanced. We show that the function f_u is uniquely determined on every strong component.

The principal subroutine used in our algorithm is a variant of Karp's [7] algorithm for finding maximum-mean cycles in a weighted, directed graph (see also [5,6]). Given an arbitrary graph G, our algorithm constructs a sequence of graphs

$$G = H^0 \to H^1 \to H^2 \to \cdots \to H^k,$$

where H^{i+1} is constructed from H^i by contracting a maximum-mean cycle and deleting any resulting loops. The final term of this sequence is the acyclic graph formed by contracting every strong component of G to a point and deleting any resulting loops. At each iteration of the algorithm, we generate a set of vertex weights σ^i for H^i corresponding to a maximum-mean cycle of H^i. At the conclusion of the algorithm, the sum of the weights σ^i computed at each iteration is the desired set of vertex weights which balances every strong component of G.

We show that the number of times the maximum-cycle mean subroutine is used is bounded by $2n$, where n is the number of vertices in G. Since the running time for the minimum cycle-mean algorithm is $O(nm)$, where m is the number of arcs in G, the running time for our algorithm is $O(n^2m)$.

2 Notation and Definitions

We want to clarify some basic graph definitions used throughout the paper. Let (X, U) be a (directed) graph with vertex set X and arc set U. We allow (X, U) to have loops and parallel arcs. We will use the notation $u = (x, y)$ to refer to an arc $u \in U$ with initial vertex x and final vertex y. (There may be more than one such arc u.) A *walk of length k* is a sequence of arcs (possibly empty) $\nu = (u_1, u_2, u_3, \ldots, u_k)$ in which the terminal endpoint of u_i is the initial endpoint of u_{i+1}. That is, a *walk* is directed and may contain repeated arcs (or vertices). A *cycle* is an walk $\nu = (u_1, u_2, \ldots, u_k)$ with no repeated vertices in which the initial endpoint of u_1 is the final endpoint of u_k. The number of arcs in a walk (or cycle) μ is denoted by $|\mu|$, and the set of all cycles of (X, U) is denoted by Φ.

We follow the convention that the maximization over an empty set is defined to be $-\infty$. Let S be a non-empty set and let a_s be an arbitrary extended real-valued function on S (i.e., the values of ∞ and $-\infty$ are permitted). The equation

$$\min_{s \in S} a_s = -\infty$$

means that the value $-\infty$ is attained at some $s \in S$. For any finite real r, $\infty + r = \infty$ and $-\infty + r = -\infty$. The operation of $-\infty + \infty$ is defined to be ∞. This is a notational convenience, since it only appears in minimization expressions, and when it occurs there is always a term with value strictly less than ∞. Operations involving ∞ and $-\infty$ occur when we consider maximum weight paths of a fixed length ending at a vertex x.

A *weighted graph* $G = (X, U, g)$ is a graph (X, U) together with a real-valued function g defined on U; g_u is called the *weight of u*. For a walk ν in (X, U, g), the *weight of ν*, g_ν, is defined by

$$g_\nu = \sum_{u \in \nu} g_u.$$

For a cycle μ in G, the *cycle mean of μ* is defined by

$$\textit{cycle mean of } \mu = \frac{1}{|\mu|} \sum_{u \in \mu} g_u.$$

A cycle $\mu \in \Phi$ is a *maximum-mean cycle* if

$$\frac{1}{|\mu|} \sum_{u \in \mu} g_u \geq \frac{1}{|\mu'|} \sum_{u \in \mu'} g_u, \quad \text{for any } \mu' \in \Phi.$$

Also,

$$\lambda(G) = \max_{\mu \in \Phi} \left\{ \frac{1}{|\mu|} \sum_{u \in \mu} g_u \right\}$$

is called the *maximum-cycle mean* for G. We will usually delete the dependence on G. Note, $\lambda(G) = -\infty$ if and only if G is acyclic (has no directed cycle).

In our notation, the vertex set X is a partition of some underlying set $\mathcal{X}$; that is, a vertex $x \in X$ is identified with a subset of $\mathcal{X}$. This notation allows us to describe our main algorithm in which sets of vertices are successively contracted. We will refer to a partition X' as *coarser* than a partition X (of the underlying set $\mathcal{X}$) if every element of X' is a union of elements of X. Similarly, we refer to X as a *finer* partition than X'.

For a non-empty set $A \subseteq X$, the *subgraph of G induced by A* is the graph with vertex set A and with all arcs of U with both endpoints in A. For the graph (X, U), the *strong component of $x \in X$* is the union of all $y \in X$ for which there is a walk (possibly empty) from x to y and y to x. Note, the strong components of G determine a partition of $\mathcal{X}$, which we denote by X^*. We define the *condensed graph of G*, Condense(G), to be the acyclic graph formed by contracting the strong components of G and deleting all loops. (See Section (4.1) for the precise definition.) A graph G is *strongly-connected* if there is a walk between every pair of vertices, namely, if G has exactly one strong component.

Our definition of balanced depends critically on the notion of a cutset of a graph.

Definition 1 *A subset $C \subset \mathcal{X}$ is* compatible with the partition X *if for every $x \in X$ either*

1. *$C \cap x = x$, or*
2. *$C \cap x = \emptyset$.*

It is easy to see that C is compatible with X if and only if C is a union of elements of X.

Definition 2 *For C that is compatible with X, the* cutset of G determined by C, *$\omega\{C; G\}$, is defined as*

$$\omega\{C; G\} = \{\omega^+(C; G), \omega^-(C; G)\},$$

where

$$\begin{aligned}\omega^+(C;G) &= \{u=(x,y)\in U \mid x\subseteq C,\ \textit{and}\ y\subseteq \mathcal{X}-C\}\,,\\ \omega^-(C;G) &= \{u=(x,y)\in U \mid x\subseteq \mathcal{X}-C,\ \textit{and}\ y\subseteq C\}\,.\end{aligned}$$

If there is no possibility of confusion, we will delete the dependence on G,

Definition 3 *A weighted graph $G=(X,U,g)$ is* balanced at (a compatible set) C *iff*

$$\max_{u\in\omega^+(C)} g_u = \max_{u\in\omega^-(C)} g_u.$$

Definition 4 *A graph $G=(X,U,g)$ is* balanced *iff it is balanced at every compatible set C.*

3 Problem Statement

We first show that if a graph G is balanced, then every arc lies in a strong component of G and, therefore, G is the union of the subgraphs induced by the strong-components. Equivalently, every connected component of G is strongly connected.

Lemma 1 *A weighted directed graph $G=(X,U,g)$ is balanced if and only if*

(i) For $u=(x,y)\in U$, x and y are contained in the same strong component of G,

(ii) Every subgraph induced by a strong component is balanced.

Proof: ($\Rightarrow$) Let $G=(X,U,g)$ be balanced, and let $u=(x,y)\in U$ be an arc for which x and y are in different strong components. Define $C\subset\mathcal{X}$ to be the the union of all vertices z for which there exists a walk from z to x, and consider $\omega(C)$. It follows that $\omega^+(C)\neq\emptyset$, but that $\omega^-(C)=\emptyset$. The balance condition at C requires that

$$\max_{u=(x,y)\in\omega^+(C)} g_u = \max_{u=(x,y)\in\omega^-(C)} g_u,$$

which cannot be satisfied since the maximum over an empty set is defined to be $-\infty$.

Part (ii) follows directly from Part (i), since if C is contained in a strong component of G then $\omega\{C;G\}$ and $\omega\{C;H\}$ coincide, where H is the subgraph induced by the strong component.

($\Leftarrow$) The converse is obvious.

∎

Lemma (1) implies that it is not possible to balance arbitrary graphs. Therefore, we describe an efficient algorithm for the following problem:

Problem 2 *Given a weighted graph (X, U, g), find vertex weights $\pi_x, x \in X$, such that the subgraph induced by every strong component of (X, U, f) is balanced, where*

$$f_u = \pi_x + g_u - \pi_y, \quad \textit{for } u = (x, y) \in U.$$

We call vertex weights $\pi_x, x \in X$ that solve Problem (2) *balancing weights* for (X, U, g) and refer to the resulting arc-weight function $f_u, u \in U$ as *balanced* on strong components.

By taking logarithms, it is easy to see that the following problem is equivalent to Problem (2):

Problem 3 *Given a weighted graph (X, U, g) with $g_u > 0$, find vertex weights $\pi_x > 0, x \in X$, for which every strong component of (X, U, f) is balanced, where*

$$f_u = \pi_x g_u \pi_y^{-1}, \quad \text{for every } u = (x, y) \in U.$$

The additive form of Problem (2) is more natural for describing the algorithm.

4 Technical Operations

Our balancing algorithm is composed of a sequence alternating two basic operations—contraction and reweighting. We describe each of these.

4.1 The Operation of Contraction (and Marking)

The operation of contraction involves identifying vertices of G and deleting loops of the resulting graph contained in a marked set of the vertices.

Definition 5 *A weighted graph $G = (X, U, g)$ together with a set $M \subseteq \mathcal{X}$ that is compatible with X is called a* weighted marked graph *and denoted by $(X, U, g; M)$.*

Let $G = (X, U, g; M)$ be a weighted marked graph, and let X' be a coarser partition of $\mathcal{X}$ than X. For $x \in X$, let A_x be the element of X' containing x. Let M' be compatible with X' and contain M. In the operation of contraction we will refer to the set of *deleted arcs* defined as

$$V = \{u = (x, y) \in U \mid A_x = A_y \subseteq M'\}$$

Definition 6 *The* contraction of G with respect to the pair (X', M'), *written $G/(X'; M')$, is the weighted marked graph $(X', U', g'; M')$ satisfying the following conditions:*

(i) there is a 1-1 and onto mapping $\phi: U' \to U - V$ such that $\phi(A_x, A_y) = (x, y)$, for $u = (x, y) \in U - V$, and

(ii) $g'_u = g_{\phi(u)}$.

Intuitively, in the contracted graph $G/(X'; M')$ all vertices of X contained in a vertex of X' are identified; all loops at marked vertices of the resulting graph are then deleted. Normally, we will identify an arc in $G/(X'; M')$ with its image under $\phi(\cdot)$ in G, so that the set of arcs in the contracted graph can be viewed as a subset of the arcs of the original graph. Thus we will use the notation: Let $u = (A_x, A_y)$ be an arc of $G/(X'; M')$ corresponding to arc $u = (x, y)$ of G. Alternatively, to avoid double subscripts we may also write: Let $u = (A, B)$ be an arc of $G/(X'; M')$ corresponding to arc $u = (x, y)$ of G (where A and B are the vertices of $G/(X'; M')$ containing x and y, respectively).

Note that the *condensed graph of G* is defined as

$$\text{Condense}(G) = G/(X^*; \mathcal{X})$$

where X^* is the partition of $\mathcal{X}$ determined by the strong components of G.

In our algorithm, we shall consider the important case in which the partition is induced by a cycle μ of G, that is, when one element of X' is the union of the vertices of μ and the others are the remaining elements of X. The set of marked vertices, M', is the union of M and the vertices of μ. In this case we denote the contracted graph by G/μ. Thus, for a sequence of graphs

$$G = H^0 \to H^1 \to H^2 \to \cdots \to H^k,$$

where H^{i+1} is constructed from H^i by contraction, the vertex set of each graph is a partition of the underlying set X and is a *coarser* than the preceeding term.

4.2 The operation of reweighting

Each time a maximum-mean cycle is computed, a corresponding set of vertex weights is computed and used to *reweight* the graph. This operation is called *reweighting* G.

Definition 7 *Let* $G = (X, U, g)$ *be a weighted graph, and let* $\pi_x, x \in X$ *be a set of vertex weights for* G. *The graph* $G_\pi = (X, U, f)$ *where*

$$f_u = \pi_x + g_u - \pi_y, \quad u = (x, y) \in U,$$

is the reweighted graph of G *(with respect to* π*).*

5 Uniqueness

Before describing the algorithm for showing the existence of balancing weights, we show that the weights are determined uniquely up to an additive constant on every strong component.

Theorem 2 *Let* $G = (X, U, g)$ *be a strongly-connected weighted graph. If* π *and* σ *are balancing weights for* G, *then* $\pi_x - \sigma_x = c$ *for some constant* c, *and, therefore, the arc-weight function* f *defined by*

$$f_u = \pi_x + g_u - \pi_y, \quad \text{for } u = (x, y) \in U$$

is the unique arc-weight function for which (X, U, f) *is balanced.*

Proof: Let π_x and σ_x, $x \in X$, be vertex weights for which G_π and G_σ are balanced. For $u = (x, y) \in U$, define

$$\begin{aligned} f_u &= \pi_x + g_u - \pi_y, \quad \text{and} \\ h_u &= \sigma_x + g_u - \sigma_y, \end{aligned}$$

Note that

$$h_u = \tau_x + f_u - \tau_y, \quad u = (x, y) \in U, \tag{1}$$

where

$$\tau_x = \sigma_x - \pi_x, \quad x \in X.$$

Define $Z \subseteq X$ by

$$Z = \left\{ z \in X \mid \tau_z = \max_{x \in X} \tau_x \right\}$$

and define $C \subseteq \mathcal{X}$ by

$$C = \bigcup_{z \in Z} z$$

Suppose $C \neq \mathcal{X}$ and consider the cutset determined by C. The balance conditions with respect to h and equation (1) imply

$$\max_{u=(x,y)\in\omega^+(C)} \{\tau_x + f_u - \tau_y\} = \max_{u=(x,y)\in\omega^-(C)} \{\tau_x + f_u - \tau_y\},$$

where $\omega^+(C), \omega^-(C) \neq \emptyset$ since G is strongly-connected. But since $\tau_x - \tau_y > 0$, for $u = (x, y) \in \omega^+(C)$, and $\tau_x - \tau_y < 0$, for $(x, y) \in \omega^-(C)$, this contradicts the balance condition for f. This contradiction implies that $C = \mathcal{X}$, namely that τ_x is constant on X, and, therefore, $f_u = h_u$, $u \in U$. ∎

It follows from Theorem (2) that for an arbitrary graph G, if $\pi_x, x \in X$ are vertex weights which balance every strong component of G, then $\sigma_x, x \in X$, also balance every strong component if and only if $\pi - \sigma$ is constant on strong components.

6 Computing Maximum-Mean cycles

For $G = (X, U, g)$, let

$$\lambda = \max_{\mu\in\Phi} \left\{ \frac{1}{|\mu|} \sum_{u\in\mu} g_u \right\}$$

be the maximum cycle-mean of (X, U, g). We describe a variant of Karp's algorithm [7] for computing the maximum-mean cycle of G.

Although our treatment of the subproblem for computing a maximum-mean cycle is closely related to [7], there are two differences which are important for this paper. First, our treatment shows that Karp's algorithm can be extended to arbitrary graphs, whereas the original version required that G be strongly-connected. Second, we compute vertex weights $\sigma_x, x \in X$, using the output from the maximum-cycle mean algorithm which are used for solving Problem (2). There is also a trivial difference that we compute maximum-mean cycles rather that minimum-mean cycles. (Karp's algorithm also finds maximum-mean cycles, by multiplying the weights by -1.) Since we use the constructions introduced in the proof in later results, we have included a modification of Karp's proof to show that our algorithm is correct. Related modifications can be found in [5].

For each $x \in X$, let $F_k(x), k = 0, 1, 2, \ldots, n$ be the maximum weight over the set of walks of length k that terminate at x. The $F_k(v)$'s can be computed in time $O(nm)$ using the recurrence

$$\begin{aligned} F_0(x) &= 0, \quad x \in X \\ F_{k+1}(x) &= \max_{u=(y,x)\in\omega^-(x)} \{F_k(y) + g_u\}, \quad k = 0, 1, 2, \ldots, n-1. \end{aligned}$$

Define

$$\pi_x^0 = \max_{0 \le k \le n-1} \{F_k(x)\}, \quad \text{for } x \in X. \tag{2}$$

Note that if G has no positive cycle, then π_x^0 is the maximum of the weights of all walks ending at x. In this case, we call a walk ν that ends at x and that has weight π_x^0 a *maximal weight walk ending at* x.

The first two parts of Lemma (3) are contained in [4]. Note that we do not assume that G has a cycle in this Lemma.

Lemma 3 *Let $G = (X, U, g)$ be a weighted graph and suppose that all cycle weights of G are nonpositive. Let π_x^0 be given by (2) for $x \in X$, and let*

$$f_u = \pi_x^0 + g_u - \pi_y^0, \quad \text{for } u = (x, y) \in U.$$

Then the following are true:

(i) $f_u \le 0$, *for all* $u \in U$;

(ii) If G has a cycle μ of weight 0 *and u is any arc of μ, then* $f_u = 0$;

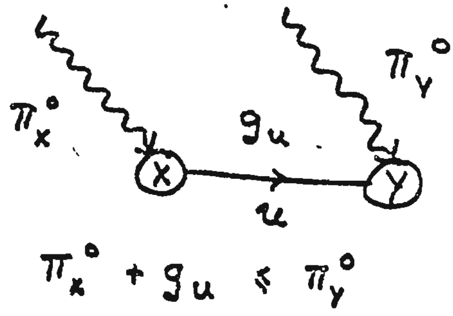


Figure 1: $f_u = \pi_x^0 + g_u - \pi_y^0 \leq 0$

(iii) *Let μ be a cycle of weight 0 and let x and y be vertices of μ. Let η be a maximal weight walk ending at x and let ν be an extension of η along μ ending at y. Then ν is a maximal weight walk ending at y (see Fig. (2).*

Proof: (i) Since π_x^0 and π_y^0 are the weights of maximal weight walks ending at x and y respectively, and since $\pi_x^0 + g_u, u = (x,y)$ is the weight of a walk ending at y, we immediately deduce (i) (see, Fig. (1).

(ii) Let μ be a cycle of weight 0. Since the weight of any cycle is unaffected by replacing g_u by f_u, we have

$$\sum_{u \in \mu} f_u = 0.$$

It follows that $f_u = 0$ for u lying on μ, since, by part (i), $f_u \leq 0$.

(iii) Let ν be composed of η and the walk ρ from x to y along the cycle μ. Let $u = (x', y')$ be an arc of μ. Then from (ii) we have

$$g_u = \pi_{y'}^0 - \pi_{x'}^0,$$

and, therefore,

$$g_\rho = \pi_y^0 - \pi_x^0.$$

Therefore,

$$g_\nu = g_\eta + g_\rho = \pi_x^0 + \pi_y^0 - \pi_x^0 = \pi_y^0,$$

and (iii) is proved (see, Fig. (2)) ▮

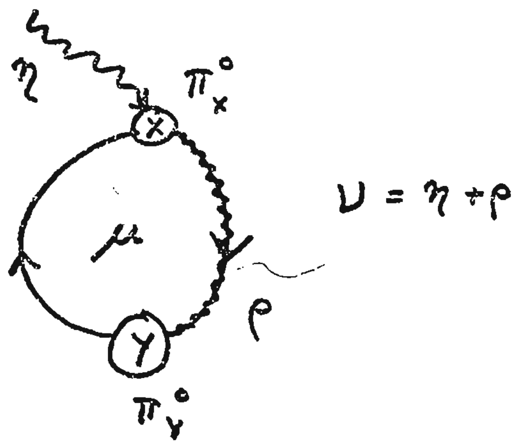


Figure 2: Extension of maximal walk in cycle of weight 0.

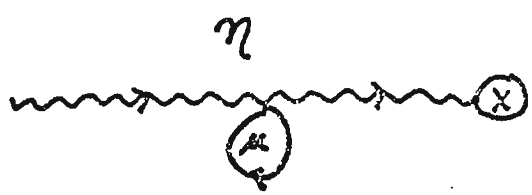


Figure 3: $F_n(x) = g_\eta + g_\mu$.

Theorem 4 *For every weighted graph $G = (X, U, g)$ the maximum cycle-mean λ is given by:*

$$\lambda = \max_{x \in X} \left\{ \min_{0 \leq k \leq n-1} \left\{ \frac{F_n(x) - F_k(x)}{n - k} \right\} \right\}. \tag{3}$$

Proof: First, note that G contains no cycle if and only if $F_n(x) = -\infty$ for every $x \in X$. Therefore, since $F_0(x) = 0$, for every $x \in X$, the theorem is true if G is acyclic. (Recall, $-\infty - (-\infty)$ if defined to be ∞.)

Now suppose that the maximum cycle-mean λ is 0. Let $x \in X$, and let ν be a walk of length n and weight $F_n(x)$ ending at x. Then ν must contain some cycle μ; let η be the walk ending at x formed by deleting μ from ν (see, Fig. (3)). Since the length of η is less than n, there is some j, $0 \leq j \leq n-1$, such that

$$F_n(x) = g_\eta + g_\mu \leq g_\eta \leq F_j(x).$$

It follows that

$$\min_{0\le k\le n-1}\{F_n(x)-F_k(x)\}\le 0,\quad \text{for every } x\in X. \tag{4}$$

Now let μ be a cycle of weight 0 and let x be any vertex of μ. Let j, $0\le j\le n-1$, be the length of the maximal weight walk η ending at x. Let ν be the extension of this walk along μ such that ν has length n, and suppose that ν ends at y (see, Fig. (2)). Then, from Part (iii) of Lemma (3), it follows that $F_n(y)=\pi_y^0$, and, therefore, by Part (ii)

$$\min_{0\le k\le n-1}\{F_n(y)-F_k(y)\}\ge 0. \tag{5}$$

The result follows from (4) and (5).

We now turn to the case of general finite λ. Consider the graph with arc weights $g_u-\lambda$. Then for all k, $0\le k\le n-1$, and for all $x\in X$

$$\left(\frac{F_n(x)-F_k(x)}{n-k}\right)$$

is decreased by λ. But the maximum cycle-mean is now 0, and, therefore, the result follows from the second part of the proof. ∎

As a corollary to Lemma (3), we have the following important result (see Theorem 7.5 of [4]).

Corollary 5 *Let $G=(X,U,g)$ be a weighted graph with maximum cycle-mean λ (possibly, $-\infty$).*

(i) Then

$$\lambda=\inf_{\sigma\in\Re^n}\left\{\max_{u=(x,y)\in U}\{\sigma_x+g_u-\sigma_y\}\right\},$$

(ii) In the case $\lambda>-\infty$, define

$$\pi_x=\max_{0\le k\le n-1}\{F_k(x)-k\lambda\}. \tag{6}$$

Then

$$\lambda=\max_{u=(x,y)\in U}\{\pi_x+g_u-\pi_y\}. \tag{7}$$

If μ is any maximum-mean cycle for G, then

$$\pi_x+g_u-\pi_y=\lambda,\quad \textit{for } u=(x,y)\in\mu. \tag{8}$$

Proof: (i) First, suppose that G is acyclic (i.e., $\lambda = -\infty$). For $x \in X$, let α_x be the length (number of arcs) of the longest walk in G ending at x. Let M be any positive number and set $\sigma_x = M\alpha_x$. Since $\alpha_y \geq \alpha_x + 1$, for $u = (x, y) \in U$, we have

$$\sigma_x + g_u - \sigma_y \leq -M + g_u.$$

Since M is arbitrary, (i) follows in the acyclic case.

(ii) Now suppose that G has a cycle. Note that replacing the arc weights by $\sigma_x + g_u - \sigma_y$, $u = (x, y) \in U$, does not change the weight of any cycle. Therefore, one of the altered weights must always be as large as λ. It follows that

$$\lambda \leq \inf_{\sigma \in \Re^n} \left\{ \max_{u=(x,y)\in U} \{\sigma_x + g_u - \sigma_y\} \right\}.$$

Let π_x be defined by (6). Since the π's are the weights of maximum-weight walks with respect to the arc weights $g_u - \lambda$, it follows that

$$\pi_x + (g_u - \lambda) \leq \pi_y, \quad \text{for any } u = (x, y) \in U,$$

or, equivalently,

$$\pi_x + g_u - \pi_y \leq \lambda, \quad \text{for any } u = (x, y) \in U.$$

If u is any arc on a maximum-mean cycle μ, then by Part (ii) of Lemma (3)

$$\pi_x + g_u - \pi_y = \lambda, \quad \text{for any } u = (x, y) \in \mu,$$

and the result follows. ∎

The weights $\pi_x, x \in X$ defined by (6) are called *optimal weights* for G and are essential for computing balancing weights for G.

Once the $F_k(x)$'s are computed, λ and the vertex weights π_x can be computed in time $O(n^2)$. Further, if x is any vertex at which the maximum in (3) is attained, then any cycle contained in a maximal weight walk of length n ending at x is a maximum-mean cycle. Let $\nu = (u_0, u_1, \ldots, u_{n-1})$ where $u_i = (x_i, x_{i+1})$ be a walk of length n ending at x. Then ν has maximal weight if and only if

$$F_{i+1}(x_{i+1}) = F_i(x_i) + g_{u_i}, \quad \text{for } i = 0, 1, 2, \ldots, n-1. \tag{9}$$

Starting with vertex $x_n = x$, we can scan $\omega^-(x)$ and find a $u_n = (x_{n-1}, x_n)$ satisfying (9) with $i = n-1$. We then scan $\omega^-(x_{n-1})$ to find arc u_{n-1} satisfying (9) with $i = n-2$. We continue in this fashion until we encounter a repeated vertex, at which point we have found a maximum-mean cycle. Since no vertex is scanned twice, each arc of G is examined at most once. Therefore, the number of operations needed to find a maximum-mean cycle given the weights $F_k(x)$ is bounded by $O(m)$.

7 Some Technical Lemmas

The next three lemmas are needed to prove that the balancing algorithm described in Section (8) is correct.

Lemma 6 *Let $G = (X, U, g; M)$ be a weighted marked graph. Let X' be a partition of $\mathcal{X}$ that is coarser than X and finer than X^* (the partition determined by the strong components of G), and let $M \subseteq M' \subseteq \mathcal{X}$. Then the strong components of G and $G/(X'; M')$ coincide.*

Proof: (1) Let $(u_1, u_2, \ldots, u_k)$ be a walk between two vertices x and y contained in the same component of G. If we delete from this walk every arc whose endpoints are contained in the same vertex of X', then we are left with a walk from A_x to A_y in $G/(X'; M')$. Therefore, the components of G are contained in the components of $G/(X'; M')$.

(2) Conversely, let A and B be vertices of $G/(X'; M')$ belonging to the same component. Then there is a walk $(v_1, v_2, \ldots, v_k)$ in $G/(X'; M')$ from A to B. Let $v_i = (x_i, y_i)$ be the corresponding arcs in G (under the mapping ϕ). Let x and y be any vertices of G contained in A and B, respectively. Define $y_0 = x$ and $x_{k+1} = y$. Then y_i and x_{i+1}, $i = 0, 1, \ldots k$, are contained in the same vertex of $G/(X'; M')$ and, therefore, are contained in the same component of G (since X' is finer than X^*). Therefore there exists a walk in G from y_i to x_{i+1} (which could be empty) and also a walk from x to y (see Fig. (4)). It follows that the components of $G/(X'; M')$ are contained in the components of G. ∎

Lemma 7 *Let $G = (X, U, g; M)$ be a weighted marked graph and let $G' = (X', U', g'; M') = G/(X'; M')$ be the contraction of G with respect to (X', M').*

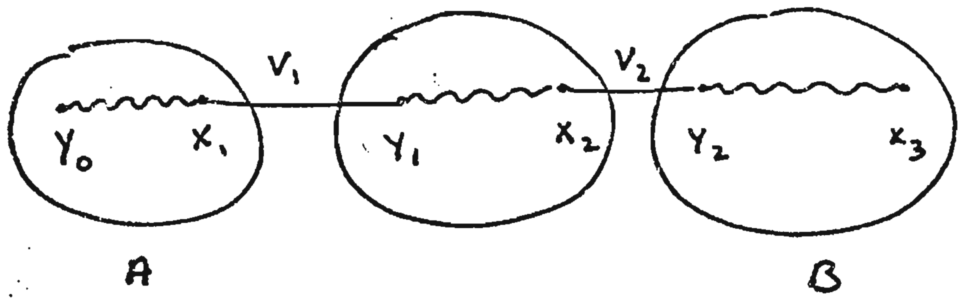


Figure 4: Strong components of G and $G/(X';M')$ coincide.

If C is compatible with X', then $\omega\{C;G\} = \omega\{C;G'\}$. That is,

$$u = (x,y) \in \omega^+(C;G) \iff u = (A_x, A_y) \in \omega^+(C;G'), \quad \textit{and}$$
$$u = (x,y) \in \omega^-(C;G) \iff u = (A_x, A_y) \in \omega^-(C;G').$$

Proof: (1) Let $u = (x,y)$ be an arc of $\omega^+(C;G)$. Then $x \subseteq C$ and $y \subseteq \mathcal{X} - C$. Since C is compatible with X', either $A_x \cap C = A_x$ or $A_x \cap C = \emptyset$. But $x \subseteq A_x$ and $x \cap C = x$ imply that $A_x \cap C \neq \emptyset$. Therefore, $A_x \cap C = A_x$. Similarly, $y \subseteq \mathcal{X} - C$ implies $A_y \cap C \neq A_y$, and, therefore, $A_y \cap C = \emptyset$. It follows that $(A_x, A_y) \in U'$ and $(A_x, A_y) \in \omega^+(C;G')$

(2) Conversely, let $u = (A_x, A_y) \in U'$ be an arc of $\omega^+(C;G')$. Then the corresponding arc $u = (x,y)$ of U clearly satisfies $x \subseteq C$ and $y \subseteq \mathcal{X} - C$ and, therefore, is in $\omega^+(C;G)$.

Similarly, we can show that $u = (x,y) \in \omega^-(C;G)$ if and only if $u = (A_x, A_y) \in \omega^-(C;G')$. ▮

The following lemma is crucial for the inductive proof of the correctness of the algorithm. It states that if the operations of reweighting and contracting are applied twice in succession, then the resulting weighted marked graph can be generated directly by reweighting and contracting the original graph.

Lemma 8 *Let $G = (X, U, g; M)$ be a weighted marked graph; let $G' = G_\pi/(X';M')$, and let $G'' = G'_{\pi'}/(X'';M'')$. Then $G'' = G_{\pi''}/(X'';M'')$, where*

$$\pi''_x = \pi_x + \pi'_A, \quad \textit{for } x \in X,$$

and A is the element of X' containing x.

Proof: Let $G' = (X', U', g'; M')$ and $G'' = (X'', U'', G''; M'')$. First, we observe that X'' is coarser than X' and that X' is coarser than X. Therefore, X'' is coarser than X. Also, M'' is compatible with X'' and contains M. Therefore, contraction of G with respect to (X'', M'') is well-defined.

Let V and V' be the sets of deleted arcs when G is contracted to G' and G' is contracted to G'', respectively. Also, let $\phi: U' \to U - V$ and $\phi': U'' \to U' - V'$ be the 1–1 and onto mappings satisfying the conditions in the definition of contraction. Let $\psi = \phi \circ \phi'$. We claim that ψ is the mapping required to show that $G'' = G_{\pi''}/(X''; M'')$. Clearly, ψ is a 1-1 mapping from U'' to U. Further, $\phi'(U'') = U' - V'$ and $\phi(U') = U - V$. Therefore,

$$\begin{aligned}
\psi(U'') &= \phi(U' - V'),\\
&= \phi(U') - \phi(V'),\\
&= U - V - \phi(V'),\\
&= U - (V \cup V'),
\end{aligned}$$

since $\phi(V')$ and V' are identified. Thus, to show that ψ maps onto $U - V''$ it suffices to show that $V'' = V \cup V'$, where V'' is the set of deleted arcs in the contraction $G_{\pi''}/(X''; M'')$. For $x \in X$, let A_x and $\bar{A}_x$ be the elements of X' and X'', respectively, containing x.

(1) $V'' \subseteq V \cup V'$.

Suppose $u = (x, y) \in V''$. Then $\bar{A}_x = \bar{A}_y \subseteq M''$. If $A_x \neq A_y$, then $u \in V'$. If $A_x = A_y$, then $u \in V$ if $A_x \subseteq M'$; otherwise $u \in V'$.

(2) $V \cup V' \subseteq V''$.

Let $u = (x, y) \in V$; then $A_x = A_y$ and, therefore, $\bar{A}_x = \bar{A}_y$. Also, $A_x \subseteq M'$, which implies that $\bar{A}_x \cap M'' \neq \emptyset$, since $M \subseteq M' \subseteq M''$. Therefore, the compatibility of M'' with X'' implies that $\bar{A}_x \subseteq M''$. It follows that $u \in V''$.

Let $u = (A_x, A_y)$, and let $\bar{A} \subseteq M''$ be the element of X'' containing A_x and A_y. Then $\bar{A}_x = \bar{A}_y = \bar{A}$, and, therefore, $u \in V''$.

Finally, to see that the weights are correct, let $u = (\bar{A}, \bar{B})$ be an arc of G'', and assume that

$$\begin{aligned}
\phi'(\bar{A}, \bar{B}) &= (A, B), \quad \text{and}\\
\phi(A, B) &= (x, y).
\end{aligned}$$

Then

$$
\begin{aligned}
g''_u &= \pi'_A + g'_u - \pi'_B, \\
&= \pi'_A + \pi_x + g_u - \pi_y - \pi'_B, \\
&= \pi''_x + g_u - \pi''_y.
\end{aligned}
$$

This completes the proof of the lemma. ∎

8 The Balancing Algorithm

We are now ready to state our balancing algorithm.

The Balancing Algorithm

Input: A weighted marked graph $G = (X, U, g; M)$, with $M = \emptyset$.

Output: (i) Vertex weights $\pi_x, x \in X$, such that every subgraph induced by a strong component of the reweighted graph G_π is balanced,

(ii) The acyclic graph $H = \text{Condense}(G_\pi)$.

(Recall, the weight of $u = (x, y)$ in G_π is $\pi_x + g_u - \pi_y$.)

0: (Initialization) Let $G^i = (X, U, g^i; M^i)$, and $H^i = (X^i, U^i, h^i; M^i)$. Set $i = 0$, $\pi_x = 0$, $G^0 = H^0 = G$. (Note, $M^0 = \emptyset$.)

1: (Termination) If H^i is acyclic, set $\pi = \pi^i$, $H = H^i$ and STOP.

2: (Compute Optimal Cycle) Find a maximum-mean cycle μ^i and optimal weights σ^i for H^i. Set X^{i+1} equal to the partition induced by μ^i, and set M^{i+1} equal to the union of the vertices of μ^i and M^i (see Section (4.1)).

3: (Reweight and Contract H^i) Form $\bar{H} = H^i_{\sigma^i}$ (the reweighting of H^i with respect to the vertex weights σ^i) and contract to form $H^{i+1} = \bar{H}/\mu^i$. That is, for arc $u = (\bar{A}, \bar{B}) \in U^{i+1}$ corresponding to arc $u = (A, B) \in U^i$ (see Fig. (5)),

$$h^{i+1}_u = \sigma^i_A + h^i_u - \sigma^i_B$$

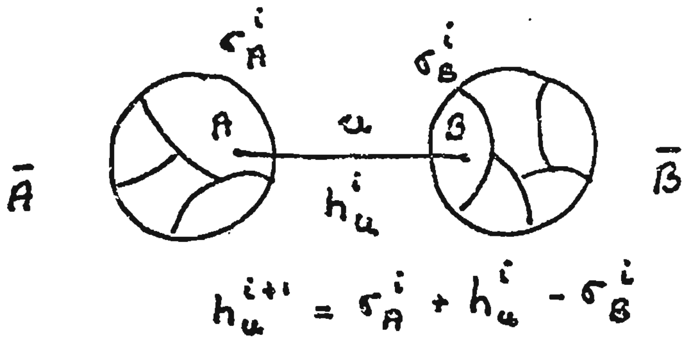


Figure 5: Arc weight in the Contracted Graph

4: (Reweight G) Let $G^{i+1} = (X, U, g^{i+1}; M^{i+1})$ be the weighted, marked graph defined by

$$\begin{aligned} \pi_x^{i+1} &= \pi_x^i + \sigma_A^i, \quad x \in X, x \subseteq A \in X^i \\ g_u^{i+1} &= \pi_x^{i+1} + g_u - \pi_y^{i+1}, \quad u = (x, y) \in U. \end{aligned}$$

5: (Increment) Set $i = i + 1$ and return to Step (1).

Notice that we do not make any assumptions about the connectivity of the input graph. Vertex weights π for which every strong component of G_π is balanced could also be computed by first finding the strongly connected components of G and then balancing each component separately. We prefer, however, to present the algorithm in the more general context of arbitrary graph. See, for example, [1] for a discussion of algorithms for computing the strong components of a graph.

The next three lemmas establish important properties of the balancing algorithm which are need to verify that the algorithm is correct. First, we define a *descent* function for the algorithm. For an arbitrary graph $G = (X, U)$, define $\Theta(G)$ by

$$\Theta(G) = |X| + \text{the number of vertices of } X \text{ containing a loop.}$$

Lemma 9 *The following are true:*

(i) At each iteration of the balancing algorithm,

$$\Theta\left(H^{i+1}\right) < \Theta\left(H^i\right).$$

(ii) The algorithm terminates with $H = Condense(G_\pi)$ after, at most, $2n$ contraction-reweighting operations.

Proof: (i) The new vertex of X^{i+1} formed by μ^i is always loopless, and new loops are never created at the remaining vertices of X^{i+1}. Therefore, a contraction-reweighting in the balancing algorithm cannot increase the number of vertices with a loop. If $|X^{i+1}| = |X^i|$, then μ^i must be a loop and the corresponding vertex of X^{i+1} must be loopless. Therefore, a contraction-reweighting operation either reduces the number of vertices containing a loop or contracts two vertices.

(ii) Termination of the balancing algorithm in at most $2n$ iterations follows immediately from Part (i), and, therefore, the algorithm must terminate with an acyclic graph H. It follows by induction using Lemma (6) that the components of H coincide with the components of G. But since H is acyclic these must be the vertices of H. ∎

Lemma 10 *Suppose the balancing algorithm terminates after k contraction-reweighting operations; let λ^i be the mean weight of the maximum-mean cycle μ^i computed at the ith iteration. Then*

1. $h_u^{i+1} \leq \lambda^i$, *for* $u \in U^{i+1}$, *and*
2. $\lambda^0 \geq \lambda^1 \geq \lambda^2 \geq \cdots \geq \lambda^k$.

Proof: Claim (1) follows directly from Corollary (5); claim (2) is an immediate consequence of claim (1) and the definition of λ^i. ∎

In the next lemma, we describe the relationship between the graphs G^i and H^i, which are defined in the balancing algorithm. We will use these properties to prove that the graph G_π is balanced at strong components.

Lemma 11 *The following are true:*

(i) At iteration i of the balancing algorithm,

$$H^i = (G_{\pi^i}) \,/\, \left(X^i; M^i\right).$$

(ii) Let $u = (A, B) \in U^i$ be an arc of H^i corresponding to arc $u = (x, y) \in U$ of G. For g^{i+1} defined in Step (4) of the balancing algorithm

$$g_u^{i+1} \quad = \quad \sigma_A^i + h_u^i - \sigma_B^i, \quad u \in U^i. \tag{10}$$

(iii) Let $u \in U - U^i$, the set of arcs deleted up to and including the ith contraction-reweighting operation. Then

$$g_u^k = g_u^i, \quad \textit{for } k \geq i \quad u \in U - U^i.$$

Proof: (i) Using the definitions of X^{i+1}, M^{i+1} and σ^i in Step (2) of the balancing algorithm, it follows that

$$H^{i+1} = (H_{\sigma^i}) / \left(X^{i+1}; M^{i+1}\right).$$

Since $H^0 = G^0 = G$, the result follows by induction from Lemma (8).

(ii) The result is clearly true for $i = 0$, since $\pi_x^1 = \sigma_x^0$. Using the definitions of g^{i+1} and π^{i+1}in Step (4), induction, and the definition of h^i, it follows that for $i > 0$

$$\begin{aligned}
g_u^{i+1} &= \pi_x^{i+1} + g_u - \pi_y^{i+1}, \\
&= \pi_x^i + \sigma_A^i + g_u - \pi_y^i - \sigma_B^i, \\
&= \sigma_A^i + g_u^i - \sigma_B^i, \\
&= \sigma_A^i + \sigma_A^{i-1} + h_u^{i-1} - \sigma_B^{i-1} - \sigma_B^i, \\
&= \sigma_A^i + h_u^i - \sigma_B^i.
\end{aligned}$$

(iii) If arc $u = (x, y)$ is in $U - U^i$, then x and y are in the same element of the partition X^i; that is, $x, y \subseteq A \in X^{i+1}$ for some vertex A of H^i. Using the definition of π^{i+1} in Step (4), if $k = i + 1$ then

$$\begin{aligned}
g_u^k &= \pi_x^k + g_u - \pi_y^k, \\
&= \pi_x^i + \sigma_A^i + g_u - \pi_y^i - \sigma_A^i, \\
&= g_u^i.
\end{aligned}$$

The result now follows by induction. ∎

Note that in Lemma (11) Part (ii) we are claiming more than $g^{i+1} = h^{i+1}$, since u is only required to be in U^i, whereas h^{i+1} is defined only for $u \in U^{i+1}$. We are claiming that g^{i+1} agrees with the arc weights in the reweighting of H^i by σ^i prior to the contraction step in which the arcs in $U^i - U^{i+1}$ are deleted.

We are now ready to prove the main theorem of the paper.

Theorem 12 *Let G be a arbitrary weighted graph and let π be the weights from the balancing algorithm. Then every strong component of G_π is balanced.*

Proof: Let $\bar{H}$ be the subgraph of G induced by a strong component X_i^*. Let C be a subset of X_i^* that is compatible with X, and let $\omega\{C;\bar{H}\}$ be the corresponding cutset of $\bar{H}$ determined by C. We must show that

$$\max_{u=(x,y)\in\omega^+(C;\bar{H})} \{\pi_x + g_u - \pi_y\} = \max_{u=(x,y)\in\omega^-(C;\bar{H})} \{\pi_x + g_u - \pi_y\} \tag{11}$$

We can assume that $\emptyset \subset C \subset X_i^*$, since otherwise this equality is vacuously satisfied.

Consider the sequence of partitions,

$$X = X^0 \to X^1 \to \cdots \to X^k = X^*,$$

generated by the balancing algorithm. There is a j, $0 < j < k$, for which C is not compatible with X^j, since C is compatible with X but is not compatible with X^*. Let j be the smallest such index.

Because X^j is formed from X^{j-1} by identifying the vertices of μ^{j-1}, the cycle μ^{j-1} must intersect both $\omega^+(C;H^{j-1})$ and $\omega^-(C;H^{j-1})$. Also Corollary (5) implies that in the reweighting of H^{j-1} by σ^{j-1}, the maximum weight occurs at each arc of μ^{j-1} and equals λ^{j-1}. That is,

$$\sigma_A^{j-1} + h_u^{j-1} - \sigma_B^{j-1} = \lambda^{j-1}, \quad \text{for } u = (A,B) \in \mu^{j-1}.$$

Therefore, the reweighted graph of H^{j-1} must be balanced at C. That is,

$$\max_{u=(A,B)\in\omega^+(C;H^{j-1})} \left\{\sigma_A^{j-1} + h_u^{j-1} - \sigma_B^{j-1}\right\} = \max_{u=(A,B)\in\omega^-(C;H^{j-1})} \left\{\sigma_A^{j-1} + h_u^{j-1} - \sigma_B^{j-1}\right\},$$

and both maxima must be attained at some arc of μ^{j-1}.

It follows from Lemma (7) that $\omega\{C;G\}$ and $\omega\{C;H^{j-1}\}$ coincide. Combined with Lemma (11) part (ii), it follows that for $i = j$

$$\max_{u\in\omega^+(C;G)} g_u^i = \max_{u\in\omega^-(C;G)} g_u^i. \tag{12}$$

The arcs of μ^{j-1} are contained in the set of arcs deleted by the $(j-1)$st contraction operation. Therefore, Lemma (11) Part (iii) implies that

$$g_u^k = g_u^j, \quad \text{for } u \in \mu^{j-1}.$$

Further, since $\omega\{C;G\}$ and $\omega\{C;H^{j-1}\}$ coincide, Lemma (10) implies that

$$g_u^k \le \lambda^{j-1}$$

for any u in the cutset $\omega\{C;G\}$. Therefore (12) holds for $i = k$.

Since the cycle μ^{j-1} must lie entirely in the subgraph $\bar{H}$, both maxima in (12) must occur at an arc of $\omega^+(C;\bar{H})$ and $\omega^-(C;\bar{H})$, respectively, when $i = k$. Now we can apply the definition of $g_u^k = \pi_x^k + g_u - \pi_y^k, u = (x,y) \in U$ and $\pi_x = \pi_x^k$ to obtain (11). This proves the theorem. ∎

The proof of Theorem (12) actually shows a slightly stronger result. Consider the output of the balancing algorithm—Condense(G_π), the acyclic graph formed by contracting the strong components of G_π. By Lemma (10) every arc of Condense(G_π) has weight no greater than the minimum of the λ^i's computed by the algorithm. Therefore, G_π is balanced at every compatible set C which in not a union of strong components. Equivalently, if some strong component is separated by the cutset determined by C, then the balance conditions for the components imply that G_π is also balanced at C.

The acyclic graph G_π can, in the following sense, also be balanced. For an arbitrary graph $G = (X,U)$, a vertex x is an *boundary vertex of* G if either $\omega^+(x)$ or $\omega^-(x)$ is empty (i.e., if either there are no arcs directed into x or there are no arcs directed out of x). Otherwise a vertex is an ***interior*** vertex. Let ∂G denote the union of the boundary vertices of G. It is easy to see that the graph formed by contracting the boundary is strongly-connected.

Consider $H =$ Condense(G_π) the acyclic graph generated by the balancing algorithm. Let X' be the partition of X where one element is ∂H and the others are the remaining elements of X^*. Then the balancing algorithm can be applied to the strongly-connected graph $H/(X';\partial H)$. Let the resulting weights (defined on the vertices of X') be π' and define

$$\sigma_x = \pi_x + \pi'_A, \quad \text{where } x \subseteq A \in X', \quad \text{and } x \in X.$$

Then the original graph G reweighted by σ is balanced at strong components and further H is also balanced.

To conclude, we want to combine our results from Theorems (2) and (12) for the case of strongly-connected graphs.

Corollary 13 *Let $G = (X, U, g)$ be a strongly-connected, weighted graph. Then there exists vertex-weights $\pi_x, x \in X$, unique up to an additive constant, such that the reweighted graph G_π is balanced. Thus there exists a unique balanced graph obtainable from G by reweighting.*

Finally, we mention the multiplicative interpretation of Corollary (13) (see Problem (3) in Section (3)). Let A be an $n \times n$ matrix. We can associate a graph (X, U, g) to A by:

$$\begin{aligned} X &= \{1,2,3,\ldots,n\}, \\ U &= \{(i,j) \mid a_{ij} \neq 0\}, \quad \text{and} \\ g_{ij} &= a_{ij} \end{aligned}$$

Let I be any subset of $\{1,2,3,\ldots,n\}$ and let I' be the complement of I.

A matrix A is *irreducible* if and only if its associated graph is strongly-connected. Thus, by Corollary (13), if $A \geq 0$ is irreducible, there exists a diagonal matrix D with positive diagonal elements, unique up to a multiplicative constant, such that $B = DAD^{-1}$ satisfies

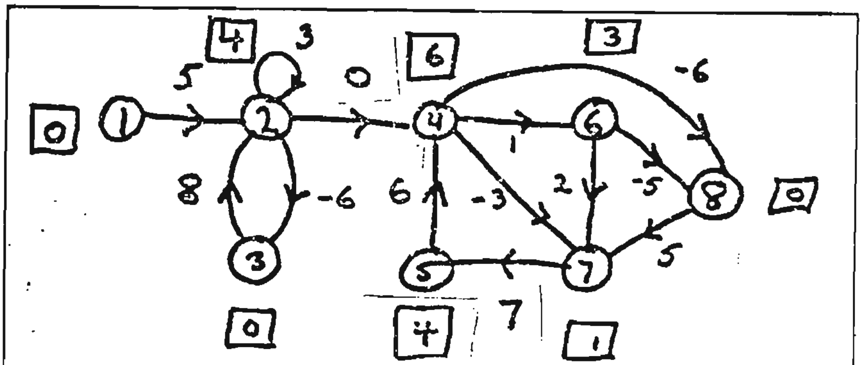
$$\max\{b_{ij} \mid i \in I, j \in I'\} = \max\{b_{ij} \mid i \in I', j \in I\}$$

for every $I \subset \{1,2,3,\ldots,n\}$.

This result is analogous in the l_∞-norm to Theorem (2) of [2], and thus provides another canonical form for diagonal similarity in the case of irreducible nonnegative matrices. For some definitions and a different canonical form for diagonal similarity without the restrictions of nonnegativity and irreducibility see [3].

9 Numerical Example

Consider the weighted directed graph G given by:



The Graph $H^0 = G$

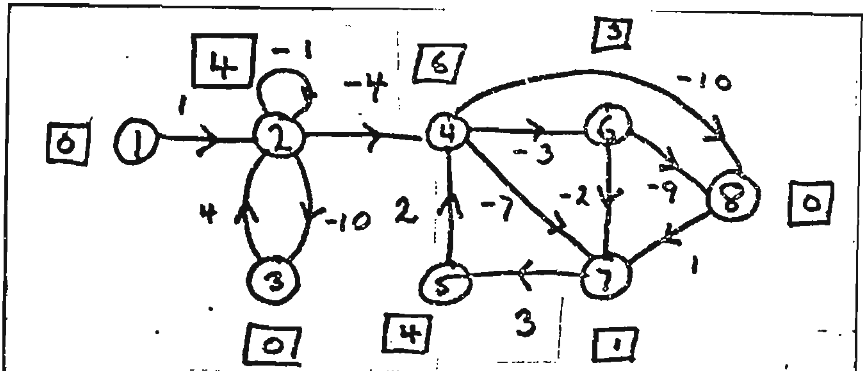
At iteration i, a maximum-mean cycle μ^i with mean λ^i is computed for H^i using the weights $F_i(x)$ and the formula described in Theorem (4). The corresponding optimal weights are computed using line (6) in Corollary (5). In the example, we will not present the calculations needed to find μ^i and λ^i. Rather, we observe that it follows from line (6) that the optimal weights corresponding to a maximum-mean cycle can be computed by first shifting every arc weight of H^i by λ^i and then computing the weights of maximal weight walks ending at each vertex. We call the graph formed from H^i by shifting arc weights by λ^i the *auxiliary graph for H^i*.

First Iteration

A maximum-mean cycle for H^0 with corresponding cycle mean λ^0 is given by:

- μ^0: $4 \to 6 \to 7 \to 5 \to 4$,
- $\lambda^0 = 4$.

Shifting the arc weights by $\lambda^0 = 4$ produces:



The Auxiliary Graph for H^0

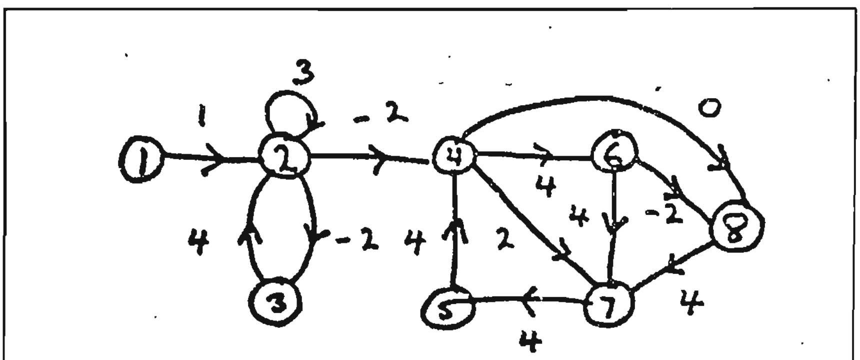
The weights of maximal weight walks with a specified terminal vertex are shown in the boxes adjacent to the vertices. Thus the vector of optimal weights σ^0 is:

- $\sigma^0 = (0, 4, 0, 6, 4, 3, 1, 0)$,

and the set of marked vertices is

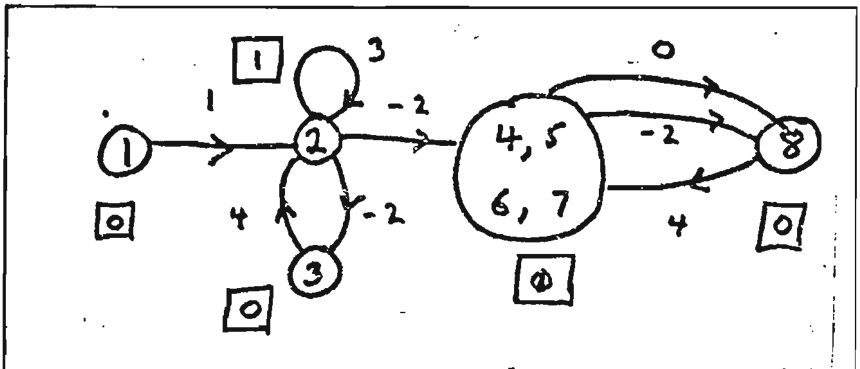
- $M^1 = \{4, 5, 6, 7\}$.

The new graph H^1 is formed by computing $H^0_{\sigma^0}$ (the reweighting of H^0 with respect to σ^0) and contracting the cycle μ^0 to a point. Since the reweighting is applied to H^0, the weights σ^0 are also shown on the graph of H^0.



The Graph $H^0_{\sigma^0}$

The new graph H^1 is:



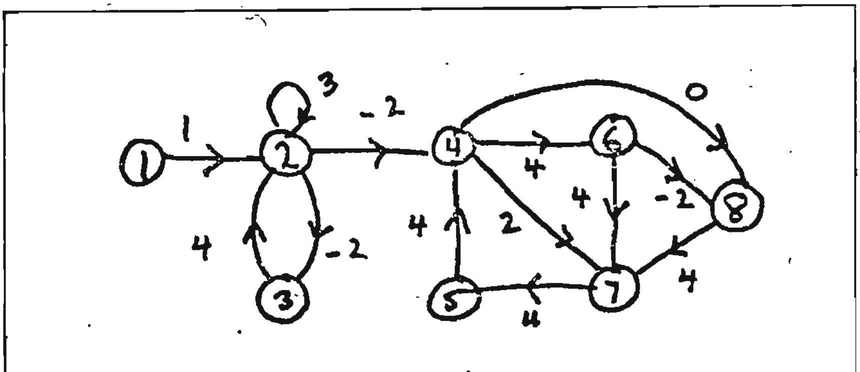
The graph H^1

Note, that only loops at marked vertices are deleted in the contraction operation.

The vector of weights π^1 is

- $\pi^1 = (0, 2, 0, 6, 4, 3, 1, 0)$,

and G^1 the reweighting of G with respect to π^1 is



The graph $G^1 = G_{\pi^1}$

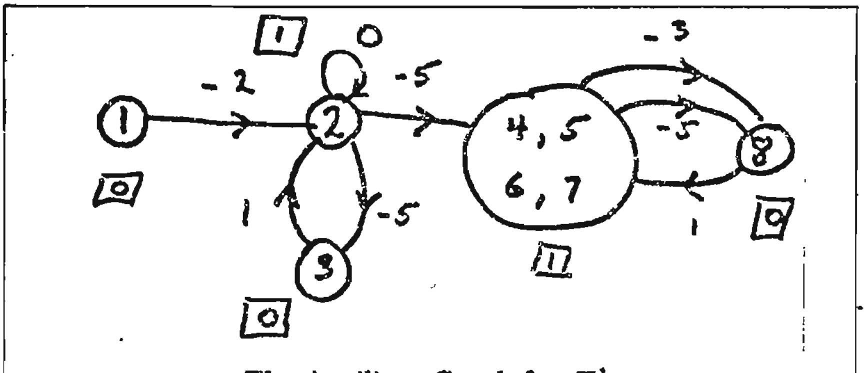
Second Iteration

A maximum-mean cycle for H^1 with corresponding cycle mean λ^1 is:

- μ^1: $2 \to 2$,

- $\lambda^1 = 3$.

The auxiliary graph with arc weights of H^1 shifted by 3 is:



The Auxiliary Graph for H^1

where the weights of maximal weight walks are shown in the boxes adjacent to the vertices.

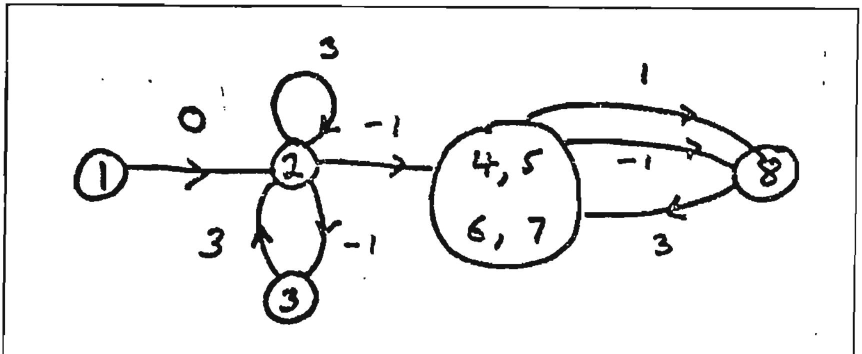
The corresponding vector of weights σ^1 is

- $\sigma^1 = (0, 1, 0, 1, 0)$,

and the set of marked vertices is

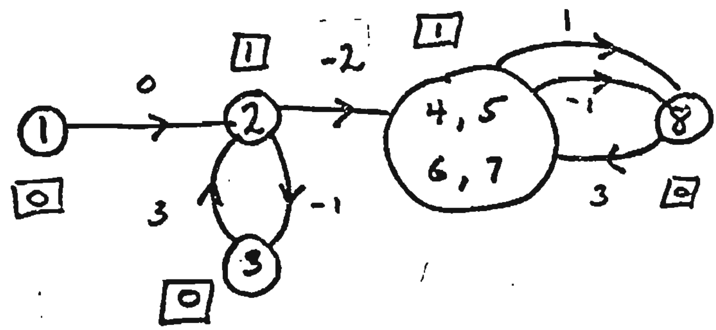
- $M^2 = \{2, 4, 5, 6, 7\}$.

The graph H^2 is computed by reweighting H^1 with respect to σ^1 and contracting μ^1 to a point.



The Graph $H^1_{\sigma^1}$

The new graph H^2 is

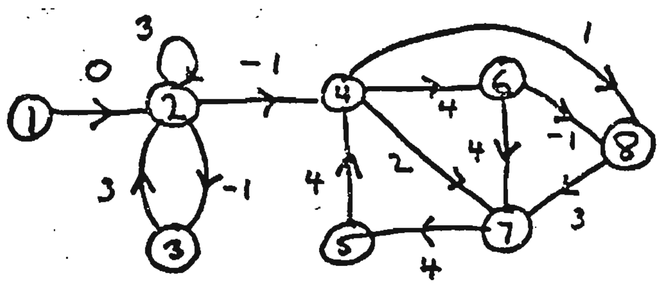


The graph H^2

The vector of weights π^2 is

- $\pi^2 = (0,4,0,6,4,3,1,0) + (0,1,0,1,1,1,1,0) = (0,5,0,7,5,4,2,0)$,

and G^2 the reweighting of G with respect to π^2 is



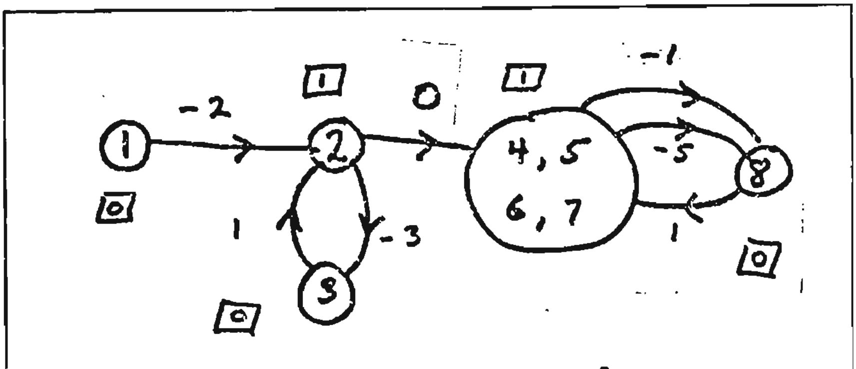
The graph $G^2 = G_{\pi^2}$.

Third Iteration

A maximum-mean cycle for H^2 with corresponding cycle mean λ^2 is

- μ^2: $\{4,5,6,7\} \to 8 \to \{4,5,6,7\}$,
- $\lambda^2 = 2$.

The auxiliary graph with arc weights of H^2 shifted by 2 together with the weights of maximal paths is



The Auxiliary Graph for H^2

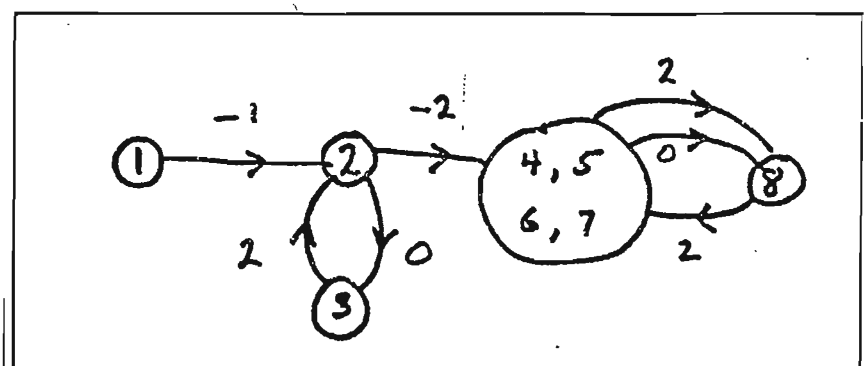
The corresponding vector of weights σ^2 is

- $\sigma^2 = (0,1,0,1,0)$,

and the set of marked vertices is

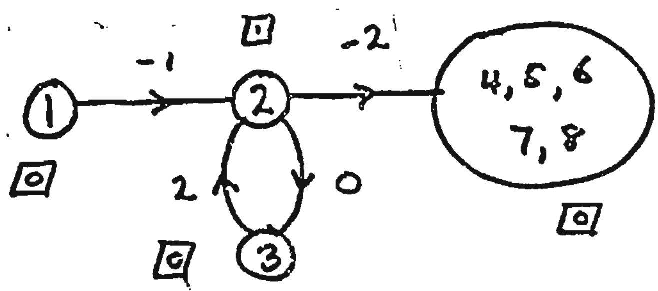
- $M^3 = \{2,4,5,6,7,8\}$.

The graph H^3 is computed by reweighting H^2 with respect to σ^2 and contracting μ^2 to a point.



The Graph $H^2_{\sigma^2}$

The new graph H^3 is

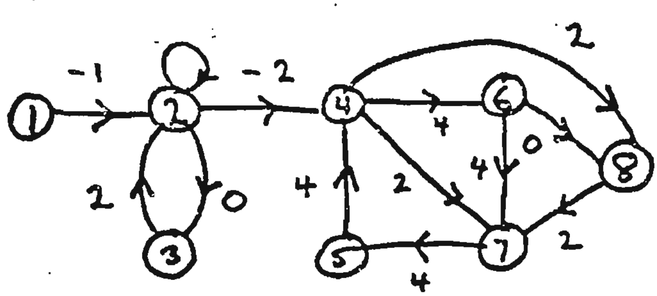


The graph H^3

The vector of weights π^3 is

- $\pi^3 = (0,5,0,7,5,4,2,0) + (0,1,0,1,1,1,1,0) = (0,6,0,8,6,5,3,0)$,

and G^3 the reweighting of G with respect to π^3 is



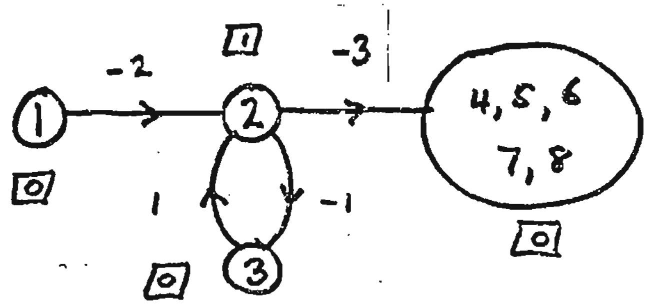
The graph $G^3 = G_{\pi^3}$.

Fourth Iteration

A maximum-mean cycle for H^3 with corresponding cycle mean λ^3 is

- μ^3: $\{4,5,6,7\} \to 8 \to \{4,5,6,7\}$,
- $\lambda^3 = 1$.

The auxiliary graph with arc weights of H^3 shifted by 1 together with the weights of maximal weight path is



The Auxiliary Graph for H^3

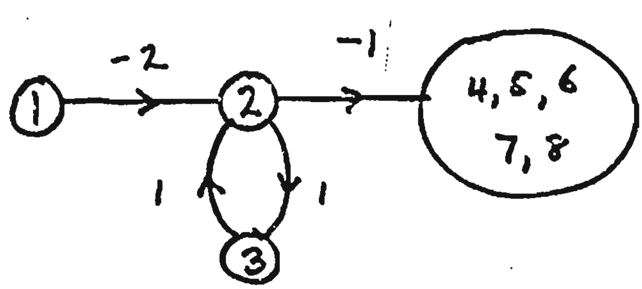
The corresponding vector of weights σ^2 is

- $\sigma^3 = (0, 1, 0, 0)$,

and the set of marked vertices is

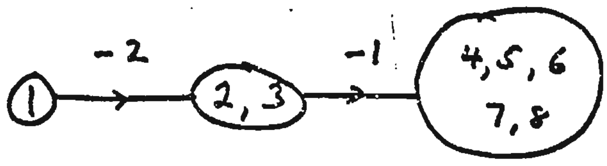
- $M^4 = \{2, 3, 4, 5, 6, 7, 8\}$.

The graph H^4 is computed by reweighting H^3 with respect to σ^3 and contracting μ^3 to a point.



The Graph $H^3_{\sigma^3}$

The new graph H^4 is

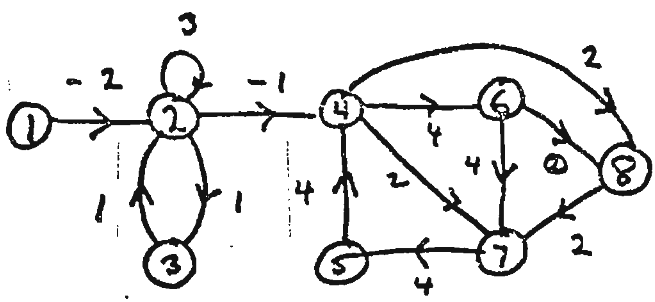


The graph H^4

The vector of weights π^4 is

- $\pi^4 = (0,6,0,8,6,5,3,0) + (0,1,0,0,0,0,0,0) = (0,7,0,8,6,5,3,0)$,

and G^4 the reweighting of G with respect to π^4 is



The graph $G^4 = G_{\pi^4}$.

Since H^4 is acyclic, the algorithm terminates, and $\pi = \pi^4$. Note that the strong components of the resulting graph $G_\pi = G^4$ are balanced.

References

[1] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Publishing Company, 1974.

[2] B. Curtis Eaves, Alan J. Hoffman, Uri G. Rothblum, and Hans Schneider. Line-sum-symmetric scalings of square nonnegative matrices. *Mathematical Programming Studies*, 25:124–141, 1985.

[3] Gernot M. Engel and Hans Schneider. Algorithms for testing the diagonal similarity of matrices and related problems. *SIAM Journal of Algebraic and Discrete Methods*, 3(4):429–438, 1982.

[4] Gernot M. Engel and Hans Schneider. Diagonal similarity and equivalence for matrices over groups with 0. *Czechoslovak Mathematical Journal*, 25:389–403, 1975.

[5] M.v. Golitschek. Optimal cycles in doubly weighted graphs and approximation of bivariate functions by univariate ones. *Numerische Mathematik*, 39:65–84, 1982.

[6] M.v. Golitschek and Hans Schneider. Applications of shortest path algorithms to matrix scalings. *Numerische Mathematik*, 44:111–126, 1984.

[7] Richard M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23:309–311, 1978.

[8] E.E. Osborne. On pre-conditoning of matrices. *Journal of the Association of Computing Machinery*, 7:338–345, 1960.

[9] Hans Schneider and Michael H. Schneider. *A simple iterative algorithm for balancing matrices.* Technical Report, Princeton University, Department of Civil Engineering, 1987.

[10] Michael H. Schneider and Stavros Zenios. *A comparative study of algorithm for matrix balancing.* Technical Report 86-10-4, University of Pennsylvania, Department of Decision Sciences, The Wharton School, January 1987.