On the Approximability of the Maximum Weighted Balance Problem

Katia S. Guimarães*  
{katia@di.ufpe.br}  

Liliane R. B. Salgado†  
{lrb@di.ufpe.br}  

Departamento de Informática  
Univ. Federal de Pernambuco  
50732-970 Recife. Pe, Brazil  

Abstract

The Maximum Weighted Balance problem is a basic problem in network design. In this article we exhibit a new definition of this problem and we define a polynomially bounded version of it using scaling technique. Based on that we specify an AP-reduction between them. We also present an approximate solution preserving approximation within 2 for MAX Polynomially Bounded Weighted Balance. Theses results we introduce are used to prove the pertinence of MAX Weighted Balance to APX, moreover we show a 3-approximate polynomial time algorithm for this problem.

Key words: MAX Weighted Balance. AP-reducibility, scaling technique. approximation algorithm, APX.

1 Introduction

Approximation algorithms are an usual strategy to solve NP-hard optimization problems. However it is known that even to calculate approximate solutions for these problems is computationally hard. Moreover NP-optimization problems exhibit different approximation properties which oscilate between having a polynomial-time approximation scheme and being non-approximable within any constant. As a consequence, the issue of determining under what conditions and by means of what methods we can design $r$-approximate polynomial-time algorithms is widely recognized as being relevant from practical and theoretical point of views.

In this paper we focus on the weighted version of the Maximum Balance problem which maximizes the number of paths that connect pairs of vertices and pass through a common edge $e$ (flow through edge $e$).

In Sec. 2 we present some basic definitions. In Sec. 3 we introduce a brief survey of the Maximum Balance and define MAX Weighted Balance in a different approach from that used in [Sal96]. We also define a polynomially bounded version of MAX Weighted Balance and specify an AP-reduction from MAX Weighted Balance to MAX Polynomially Bounded Weighted Balance in Sec. 4. In the following section we prove that both problems belongs to APX. We do that by presenting a 2-approximate solution for the polynomially bounded version and we determine the existence of a 3-approximate polynomial-time algorithm for the arbitrarily weighted version through of an AP-reduction. We consider only positive weights and we answer an open question for the MAX Weighted Balance. Sec. 6 presents conclusions and future work.

*Research partially supported by CNPq (Brazil), Grant no. 521700/96-5.  
†Research partially supported by FACEPE (PE, Brazil), Grant no. BPD-0709-1.05/96.
2 Preliminaries

We now introduce some basic definitions useful through of this paper.

Definition 1 ([ACP95]) A NP Optimization (NPO) problem $A$ is a fourtuple $(I_A, \text{sol}_A, m_A, \text{Goal})$ such that:

- $I_A$ is the set of the instances of $A$ and it is recognizable in polynomial time.

- Given a instance $x$ of $I_A$, $\text{sol}_A(x)$ denotes the set of feasible solutions of $x$. An polynomial $p$ exists such that, for any $x$ and for any $y \in \text{sol}_A(x)$, $|y| \leq p(|x|)$. Moreover, for any $x$ and for any $y$ such that $|y| \leq p(|x|)$, it is decidable in polynomial time whether $y \in \text{sol}_A(x)$.

- Given an instance $x$ and a feasible solution $y$ of $x$, $m_A(x, y)$ denotes the positive integer measure of $y$. The function $m$ is computable in polynomial time and is also called the objective function.

- $\text{Goal} \in \{\text{max}, \text{min}\}$.

The class NPO is the set of all NPO problems.

Definition 2 An NPO problem $A$ is said to be polynomially bounded if there is a polynomial $p$ such that $\text{opt}_A(x) \leq p(|x|)$ for all $x \in I_A$.

Definition 3 Let $A$ be an NPO problem. Given an instance $x$ and a feasible solution $y$ of $x$, the ratio bound of $y$ (with respect to $x$) is defined as

$$R_A(x, y) = \max \left( \frac{m_A(x, y)}{\text{opt}_A(x)}, \frac{\text{opt}_A(x)}{m(x, y)} \right).$$

The ratio bound is always a number greater than or equal to 1 and is as close to 1 as the solution is close to an optimum solution.

Definition 4 Let $r : N \to [1, \infty)$. We say that an algorithm $T$ for an optimization problem $A$ is $r(n)$-approximate if, for any instance $x$ of size $n$, the ratio bound of the feasible solution $T(x)$ with respect to $x$ is at most $r(n)$. If a problem $A$ admits an $r$-approximate polynomial-time algorithm for some constant $r > 1$, then we say that $A$ belongs to the class APX.

Definition 5 An NPO problem $A$ belongs to the class PTAS if it admits a polynomial-time approximation scheme, that is, an algorithm $T$ such that, for any instance $x$ of $A$ and for any rational $r > 1$, $T(x, r)$ returns a feasible solution whose performance ratio is at most $r$ in time bounded by $q_r(|x|)$ where $q_r$ is a polynomial.

Definition 6 ([CKST95, Tre96]) Let $A$ and $B$ be two NPO problems. $A$ is said to be AP-reducible to $B$, in symbols $A \leq_{AP} B$, if two functions $f$ and $g$, and a positive constant $\alpha$ exist such that:

1. For any $x \in I_A$ and for any $r > 1$, $f(x, r) \in I_B$.

2. For any $x \in I_A$, for any $r > 1$, and for any $y \in \text{sol}_B(f(x, r))$, $g(x, y, r) \in \text{sol}_A(x)$.
3. $f$ and $g$ are computable by two algorithms $T_f$ and $T_g$, respectively, whose running time is polynomial for any fixed $r$.

4. For any $x \in I_A$, for any $r > 1$, and for any $y \in \text{sol}_B(f(x, r))$.
   \[
   R_B(f(x, r), y) \leq r \text{ implies } R_A(x, g(x, y, r)) \leq 1 + \alpha(r - 1).
   \]

Sometimes $(f, g, \alpha)$ is called an $\alpha$-AP-reduction from $A$ to $B$, and we write $A \leq^\alpha_{\text{AP}} B$.

According to the above definition, functions like $2^{1/(r-1)n}$ or $n^{1/(r-1)}$ are admissible bounds on the computation time of $f$ and $g$, while this is not true for functions like $n^r$ or $2^n$. Therefore the computation time does not increase when the ratio bound decreases. As a result the AP-reducibility preserves membership in PTAS and is efficient even when poor ratio bounds are required (to preserve membership in logAPX and polyAPX). As far as it is known the AP-reducibility is the strictest one appearing in the literature that allows to obtain natural APX-completeness results (for instance, the APX-completeness of Max Sat).

**Definition 7** ([CGM83]) A 1-constrained spanning tree problem is that associated to the restriction $(C, \Delta)$ and denoted by $(C, \Delta)$, where $\Delta \in \{\leq, \geq\}$ a relational symbol and $C$ is an integer valued function defined over the set of all pairs $(T, \rho)$ such that $T$ is a tree and $\rho$ is a vertex of $T$ called root (it is optional in the notation).

In its decision version the question is: Is there a spanning tree $T$ of $G$ such that $C(T, \rho) \Delta W$?

**Definition 8** ([CGM86]) A weighted 1-constrained spanning tree problem is denoted by $(R, C, \Delta)$ with $R \subseteq \mathbb{Z}$. It is associated to a restriction $(R, C, \Delta)$ and an integer valued function $w : E \to R$, where $C$ and $\Delta$ are as defined in Def 7.

**Definition 9** A weighted 1-constrained spanning tree problem is uniform when $R = \{1\}$.

3 The Maximum Weighted Balance Problem

Maximum Balance problem is a 1-constrained spanning tree problem associated to network design. As a direct application we can mention the partitioning of a network into two connected balanced components. In the study of its computational complexity are important the analyses of function max Flow$(T)$ showed in [CGM80, CGM83, CGM86]. This function is defined as follows: $\text{max}_e \text{flow}(T) = \max_{e \in T} [w(e) \cdot f(e, T)]$, where $f(e, T)$ denotes the number of paths which connect pairs of vertices and pass through of a common edge $e$ (flow through edge $e$).

3.1 A Brief Report

The Balance problem is the uniform case of max Flow$(T)$. As a consequence, the NP-completeness proof showed in [CGM80] to $(\{1\}, \text{max}_e \text{flow}(T), \leq)$ is also sufficient to classify $(\text{Balance}(T), \leq)$ as an NP-complete problem. In addition to that, the intractability of $(\text{Balance}(T), \geq)$ was proved in [CGM83].

In [CGM86] it was observed that if $(R, \text{Balance}(T), \leq)$ and $(R, \text{Balance}(T), \geq)$ are NP-complete for $R = 1$ then they are strongly NP-complete when $R = N$ or $R = Z$. Besides, it is possible to extend these considerations to graphs with weighted vertices [Sal96]. The NP-completeness proof is the same. All we need is to consider all vertices with weight 1 and to conclude that an extension to $N$ or $Z$ results in a strongly NP-complete problem in those cases.
The optimization version of \( \langle \text{Balance}(T), \geq \rangle \) is the Maximum Balance or \textbf{MAX Balance}. It searches for a spanning tree \( T^* \) which maximizes the function \( \text{Balance}(T) \) over all spanning trees \( T \) of \( G \). It means,

\[
\text{Balance}(T^*) = \max_{e \in T^*} f(e, T^*) = \max_T \max_{e \in T} f(e, T) = b^*.
\]

If we let \( e = (x, y) \) be an edge of a spanning tree \( T \) of \( G \), \( N_x \) and \( N_y \) be the number of vertices of two subtrees of \( T \) obtained by removing edge \( e \), then we can define \( f(e, T) = N_x \cdot N_y \), where we consider the first tree \( T_x \) containing \( x \) and the other \( T_y \) vertex \( y \).

A detailed survey of the Balance problem can be found in [Sal96].

### 3.2 A New Definition of the Problem

In order to consider graphs with weighted vertices we can generalize \( N_x \) and \( N_y \) to denote the sum of weights of the vertices in the subtrees \( T_x \) and \( T_y \). It means define \( f(e, T) = S_x \cdot S_y \), where \( S_x \) and \( S_y \) are the mentioned sums.

\[ \text{Observation 1} \quad \text{Balance}(T) \text{ is a function of type } f(t) = t(s - t), \text{ which strictly increases in the interval } (-\infty, [s/2]). \text{ As a consequence, the maximum value is reached at } t = [s/2]. \]

We can now observe that for each edge \( e \) of \( T \), we have \( S_x \leq [s/2] \) and \( S_y \geq [s/2] \), or vice versa. Without loss of generality, we assume that \( S_x \leq [s/2] \).

We also realize that \( S_y \) is uniquely determined by \( S_x \). As a result, we can specify the maximum number of paths which connect pairs of vertices and pass through a common edge \( e \) only maximizing \( \text{Balance}(T) \) defined as follows:

\[
\text{Balance}(T) = \max_{e \in T} S_x = \max_{e \in T} \sum_{u \in T_x} w(u),
\]

where \( w(u) \) indicates the weight of vertex \( u \).

\[ \text{Definition 10} \quad \textbf{MAX Weighted Balance} \text{ is an NPO problem with:} \]

- **Instance:** an undirected connected graph \( G = (V, E) \) with edge set \( E \) and vertex set \( V = \{v_1, \ldots, v_n\} \) labeled with integers \( w(v_1), \ldots, w(v_n) \) smaller or equal to \([M/2]\) and such that \( \sum_{i=1}^{n} w(v_i) = M \).

- **Feasible Solution:** a spanning tree \( T \) of \( G \).

- **Objective Function:** \( \text{Balance}(T) = \max_{e \in T} S_x = \max_{e \in T} \sum_{v_i \in T_x} w(v_i) \).

- **Goal:** maximization

The optimization problem defined this way is equivalent to that using objective function \( \text{Balance}(T) = \max_{e \in T} S_x \cdot S_y \), for which it has been shown [Sal96] that there is a 9/8-approximate polynomial-time algorithm for instances with polynomially bounded positive weights.

An open question is if \textbf{MAX Weighted Balance} with arbitrary weights belongs to \textbf{APX}. In this paper, we answer this question for the case when only positive weights are allowed.
Figure 1: Interval $I_0$

\[
\begin{array}{c}
M \frac{1}{2} \quad M \frac{1}{2} \quad \frac{M}{2} \quad \frac{2}{2} \\
\frac{n}{2} \quad \frac{n}{2} + 1 \quad \frac{n}{2} \\
\end{array}
\]

Figure 2: Interval $I_1$

\[
\begin{array}{c}
M \frac{1}{2} \quad M \frac{1}{2} \quad \frac{M}{2} \quad \frac{2}{2} \\
\frac{n}{2} \quad \frac{n}{2} + 1 \quad \frac{n}{2} \\
\end{array}
\]

Figure 3: Interval $I_n$

\[
\begin{array}{c}
M \frac{1}{2} \quad M \frac{1}{2} \quad \frac{M}{2} \quad \frac{2}{2} \\
\frac{n}{2^k} \quad \frac{n}{2^k} + 1 \quad \frac{n}{2^k} \\
\end{array}
\]
4 Our reduction

In this section we introduce a polynomially bounded weighted version of the Maximum Weighted Balance problem and we reduce MAX Weighted Balance to it.

At first, note that $1 \leq \max_{e \in T} \sum_{u \in T_e} w(u) \leq |M/2|$. By Obs. 2, we can assume that $1 \leq S_x^{n-1} \leq |M/2|$.

**Observation 2** In Sec. 3.2 we define $S_x$ for each edge $e$ in $T$. In other words, $S_x^i$ for $i = 1, ..., n-1$. Thus, without loss of generality, we now specify that $S_x^{n-1} = \max\{S_x^i\}$.

Let us now consider a polynomially bounded version of the MAX Weighted Balance (MWB), called MAX Polynomially Bounded Weighted Balance (MPBWB). In order to get it we use the scaling technique in a way which generalizes that applied by Crescenzi and Trevisan [CT94] to define MAX Polynomially Bounded Weighted SAT.

We have to look for the optimum in the interval $1, ..., |M/2|$ and the reduction to MPBWB maps this interval into $1, ..., |n/2|$. Before showing it we prove some claims.

**Claim 1** $\forall x \in R$, $\lfloor \frac{x}{2} \rfloor = \lfloor \frac{x}{2} \rfloor$.

**Proof of Claim 1.** Assume that $x = n + \omega$ with $n = \lfloor x \rfloor$ and $0 \leq \omega < 1$. Thus $\frac{n}{2} \leq \lfloor \frac{n}{2} \rfloor + \frac{1}{2}$.

Because if $n$ is even then $\frac{n}{2} = \lfloor \frac{n}{2} \rfloor$, otherwise $\frac{n}{2} = \lfloor \frac{n}{2} \rfloor + \frac{1}{2}$.

As a result $\frac{n}{2} \leq \lfloor \frac{n}{2} \rfloor + \frac{1}{2} + \frac{\omega}{2} \Rightarrow \lfloor \frac{n}{2} \rfloor \leq \frac{n}{2} < \lfloor \frac{n}{2} \rfloor + 1 \Rightarrow \lfloor \frac{n}{2} \rfloor = \lfloor \frac{n}{2} \rfloor = \lfloor \frac{x}{2} \rfloor$. \hfill $\Box$

**Claim 2** $\forall x \in R$, $2\lfloor \frac{x}{2} \rfloor \leq |x| \leq 2\lfloor \frac{x}{2} \rfloor + 1$.

**Proof of Claim 2.** Observe that $\forall n \in N$, $y \in R$, $n|y| \leq |ny| \leq n|y| + (n-1)$.

In fact, assume that $y = m + \omega$ with $m = \lfloor y \rfloor$ and $0 \leq \omega < 1$. Thus $ny = nm + n\omega$ where $0 \leq n\omega < n$. Then we have $n|y| \leq |ny| < n|y| + n \Rightarrow n|y| \leq |ny| \leq n|y| + (n-1)$.

Now we consider $n = 2$ and $y = \frac{x}{2}$ and conclude our proof. \hfill $\Box$

Intuitively the MPBWB is obtained by splitting the interval $[1, 2|M/2| + \lfloor M/2 \rfloor/n/2 \rfloor$ into $j + 1$ intervals $I_s = \left[ \frac{\lfloor M/2 \rfloor}{2^s}, 2 \cdot \frac{\lfloor M/2 \rfloor}{2^s} + \frac{\lfloor M/2 \rfloor}{2^{s+1}} \right]$ for $s = 0, ..., j$ and $t = 0, ..., k$. Where $j = \min\{h \mid \lfloor \frac{M/2}{2^h} \rfloor \text{ is equal to } 1\}$ and $k = \min\{h \mid \lfloor \frac{n/2}{2^h} \rfloor \text{ is equal to } 1\}$. 

216
After that, each $I_s$ is subdivided into $\left\lfloor \frac{\lfloor M/2 \rfloor}{2^s} \right\rfloor$ intervals

$$
\left[ \left\lfloor \frac{\lfloor M/2 \rfloor}{2^s} \right\rfloor + i_s \cdot \frac{\lfloor M/2 \rfloor}{2^s}, \left\lfloor \frac{\lfloor M/2 \rfloor}{2^s} \right\rfloor + (i_s + 1) \cdot \frac{\lfloor M/2 \rfloor}{2^s} \right]
$$

for $i_s = 0, \ldots, \left\lfloor \frac{\lfloor n/2 \rfloor}{2^s} \right\rfloor$ such that any solution in an interval $i_s$ is assigned a new measure equal to $\left\lfloor \frac{\lfloor n/2 \rfloor}{2^s} \right\rfloor + i_s$ as showed in Figs. 1, 2, 3 and 4.

Observe that each $I_s$ strictly contains possible values to $m_{MBWB}$ that we are interested. In other words, using $I_0$ we map the value $\left\lfloor M/2 \right\rfloor$ and with $I_s$ for $s = 1, \ldots, j$ we map the values ranging from $\left\lfloor \frac{\lfloor M/2 \rfloor}{2^s} \right\rfloor$ to $2 \cdot \left\lfloor \frac{\lfloor M/2 \rfloor}{2^s} \right\rfloor$.

Note that only if $M = n$ we have $j = u$, otherwise when $M > n$ we need to determine how to continue our partition until to reach the value $j$. This situation is explained in Fig. 4, when we indicate how to map $I_s$ for $s = u, \ldots, j$ into the same interval $\left[ \left\lfloor \frac{\lfloor n/2 \rfloor}{2^s} \right\rfloor, 2 \cdot \left\lfloor \frac{\lfloor n/2 \rfloor}{2^s} \right\rfloor \right]$.

Formally, $MPWB$ and $MBWB$ are equal except for the measure function which is defined as follows.

$$
m_{MPWB}(a, T) = \max_{e \in T} \left\{ \frac{\lfloor n/2 \rfloor}{2^t} + \frac{\lfloor M/2 \rfloor}{2^t} (S_e - \left\lfloor \frac{\lfloor M/2 \rfloor}{2^t} \right\rfloor) \right\}
$$

$$
= \left\lfloor \frac{\lfloor n/2 \rfloor}{2^t} \right\rfloor + \frac{\lfloor M/2 \rfloor}{2^t} (S_e - \left\lfloor \frac{\lfloor M/2 \rfloor}{2^t} \right\rfloor)
$$

for all pairs $(s, t)$.

We denote $m_{MPWB}(a, T)$ as the measure function of $MPWB$. According to the above definition, for any instance $a$ of $MPWB$ and for any spanning tree $T$, $m_{MPWB}(a, T) \leq \lfloor n/2 \rfloor$ and this problem is indeed polynomially bounded.

**Theorem 1** $MBWB \leq AP MPBWP$

**Proof.** Let $a = (V, E, w_1, \ldots, w_n, M)$ be a instance of $MBWB$ and $T$ a spanning tree problem such that $R_{MPWB}(a, T) \leq r$. Moreover, let

$$
i_T = \left\lfloor \frac{\lfloor n/2 \rfloor}{2^t} \right\rfloor \cdot \left( S_e - \left\lfloor \frac{\lfloor M/2 \rfloor}{2^t} \right\rfloor \right).
$$

Thus $R_{MPWB}(a, T) = \frac{\lfloor n/2 \rfloor}{2^t} + i_T$ and

$$
R_{MBWB}(a, T) = \frac{\max_{e \in T} \left\{ \frac{\lfloor n/2 \rfloor}{2^t} \right\}}{\frac{\max_{e \in T} \left\{ \frac{\lfloor n/2 \rfloor}{2^t} \right\}}{\max_{e \in T} \left\{ \frac{\lfloor M/2 \rfloor}{2^t} \right\}}
$$

$$
\leq \frac{\left\lfloor \frac{\lfloor M/2 \rfloor}{2^t} \right\rfloor + \left( \frac{\lfloor n/2 \rfloor}{2^t} \right)}{\left\lfloor \frac{\lfloor M/2 \rfloor}{2^t} \right\rfloor + \left( \frac{\lfloor n/2 \rfloor}{2^t} \right)}
$$

$$
= \frac{\left\lfloor \frac{\lfloor M/2 \rfloor}{2^t} \right\rfloor \cdot (i_T + \frac{\lfloor n/2 \rfloor}{2^t})}{\left\lfloor \frac{\lfloor M/2 \rfloor}{2^t} \right\rfloor \cdot (i_T + \frac{\lfloor n/2 \rfloor}{2^t})}
$$

$$
= R_{MPWB}(a, T) + \frac{1}{i_T + \frac{\lfloor n/2 \rfloor}{2^t}}
$$

217
\[ \leq R_{MPBWB}(a, T) + \frac{1}{\lfloor \frac{n/2}{2^t} \rfloor} \cdot \frac{1}{r(t-1)}. \]

By the construction process already illustrated, we have \( \lfloor \frac{n/2}{2^t} \rfloor \geq 1 \Rightarrow \frac{1}{\lfloor \frac{n/2}{2^t} \rfloor} \leq 1 \) for \( t = 0, \ldots, \). Thus, \( R_{MWB}(a, T) \leq R_{MPBWB}(a, T) + 1 \Rightarrow R_{MWB}(a, T) \leq r + 1 = 1 + \frac{r}{(r-1)} \cdot (r-1) \)

Now we define an AP-reduction between MWB and MPBWB.

1. For any \( a \in I_{MWB} \) and for any \( r > 1, f(a, r) = a. \)

2. For any \( x \in I_{MWB} = I_{MPBWB}, \) for any \( r > 1 \) and for any \( T \in sol_{MPBWB}(f(a, r)), g(a, T, r) = T. \)

3. \( \alpha = \frac{r}{(r-1)} \)

Assume now \( r > 1, \) let \( a \) be an instance and \( T \) a solution such that \( R_{MPBWB} \leq r. \) Then we show that

\[ R_{MWB}(a, T) \leq r + 1 = 1 + \frac{r}{(r-1)} \cdot (r-1) = 1 + \alpha(r-1). \]

Then the AP-conditions are satisfied and that concludes our proof. \( \square \)

5 MWB Belongs To APX

At first, we show that MPBWB has a 2-approximate polynomial-time algorithm. In order to get it we modify the approximate solutions introduced in [GMM95] as follows.

For each \( i = 1, \ldots, n-1 \) let \( T_i^x \) and \( T_i^y \) be two trees obtained from \( T \) by removal of edge \( i \) (any ordering of the edges from 1 to \( n - 1 \) is acceptable here); moreover set \( \alpha_i = \lfloor \frac{\lfloor n/2 \rfloor}{2^t} \rfloor + \lfloor \frac{\lfloor n/2 \rfloor}{2^t} \rfloor \cdot \lfloor \frac{\lfloor n/2 \rfloor}{2^t} \rfloor \rfloor \) and \( \beta_i = M - \alpha_i. \) Then we have \( \alpha_i \leq \lfloor n/2 \rfloor \) and \( \beta_i \geq M - \lfloor n/2 \rfloor. \)

Using that approach, we substitute in the approximate solution to 2-connected graphs the following points:

1. The first optimality test \( |\alpha_{n-1} - \beta_{n-1}| \leq 1 \) is replaced to \( |\alpha_{n-1} - \beta_{n-1}| = M - 2 \cdot \lfloor n/2 \rfloor. \)

2. The update condition \( \alpha_{n-1} + \alpha_i \leq \lfloor n/2 \rfloor \) or \( \alpha_{n-1} < \beta_i/2 \) is restricted to \( \alpha_{n-1} + \alpha_i \leq \lfloor n/2 \rfloor \)

We denote the modified algorithm by \( MaxBal2_{MPBWB} \) and now we are able to prove Teo. 2.

**Observation 3** By the construction process of MAX Polynomially Bounded Weighted Balance we have that \( m_{MPBWB} = \alpha_{n-1} \) and its optimum value is reached when \( S_{x}^{n-1} = \lfloor M/2 \rfloor \) which implies \( \alpha_{n-1} = \lfloor n/2 \rfloor. \)

**Theorem 2** Let \( k \geq 2. \) For any 2-connected graph \( G \) algorithm MaxBal2_{MPBWB} returns in polynomial time a spanning tree \( T \) of \( G \) whose measure \( b \) is at least \( 1/k \) times the measure \( b^* \) of an optimum solution tree \( T^*. \)

**Proof.** If \( |\alpha_{n-1} - \beta_{n-1}| = M - 2 \cdot \lfloor n/2 \rfloor, \) i.e., if \( \alpha_{n-1} = \lfloor n/2 \rfloor, \) then \( m_{MPBWB} \) is maximum and \( T = T^*. \)
If $\alpha_{n-1} \geq \beta_{n-1}/k$ we conclude that

$$\frac{b^*}{b} = \frac{m_{MPBW}(T^*)}{m_{MPBW}(T)} \leq \frac{\lceil n/2 \rceil}{\alpha_{n-1}} \leq \frac{k \cdot \lceil n/2 \rceil}{\beta_{n-1}} \leq k,$$

and $T$ is the required approximate solution.

Otherwise suppose $\alpha_{n-1} < \beta_{n-1}/k$ and therefore $\alpha_i \leq \alpha_{n-1} < k \cdot \lceil \beta_{n-1}/k \rceil \leq \beta_i$ for each $i = 1, \ldots, n - 1$.

Observe that from $|\alpha_{n-1} - \beta_{n-1}| > m - 2 \cdot \lceil n/2 \rceil$ and $w(v_i) \leq \lfloor M/2 \rfloor$ for $i = 1, \ldots, n$ it is impossible a tree $T_{n-1}$ consisting only of vertex $y$. Therefore spite of our modification exists an edge $e$ as specified in the algorithm, since the graph is 2-connected and the removal of $y$ cannot disconnect it.

If $\alpha_{n-1} + \alpha_i \leq \lceil n/2 \rceil$ then the updating operation strictly increases the value of $m_{MPBW}$ (recall the constraining process of this objective function) from $\alpha_{n-1}$ to $(\alpha_{n-1} + \alpha_i)$.

Otherwise, if $\alpha_{n-1} + \alpha_i > \lceil n/2 \rceil$, we derive that

$$\alpha_{n-1} + \alpha_i \geq \lceil n/2 \rceil \Rightarrow \alpha_i \geq \lceil n/2 \rceil - \alpha_{n-1};$$

$$\alpha_{n-1} \geq \alpha_i \geq \lceil n/2 \rceil - \alpha_{n-1};$$

$$2\alpha_{n-1} \geq \lceil n/2 \rceil \Rightarrow \alpha_{n-1} \geq \frac{\lceil n/2 \rceil}{2}.$$

Based on that inequality we have

$$\frac{b^*}{b} \leq \frac{\lceil n/2 \rceil}{\alpha_{n-1}} \leq \frac{\lceil n/2 \rceil}{\frac{\lceil n/2 \rceil}{2}} = \frac{n/2}{\frac{n/2}{2}} = 2.$$

Now if we consider the approximate solution to any connected graph presented by Galbiati et al. [GMM95], by Obs. 3 we can maximize $m_{MPBW}$ searching for an edge $e$ whose sum of weights $S_x$ is maximum. As a consequence that algorithm can be used with a slight modification. It means that the 2-connected solution used is replaced by $Max Bal2_{MPBW}$. Because of this dependence the new algorithm becomes a solution preserving approximation within 2. Despite the new approximation constant the correctness proof of the algorithm remains equal.

Based on the above results and our AP-reduction, we can conclude the existence of a 3-approximate polynomial-time solution for MWB.

6 Conclusions and Future Work

Our main result is MAX Weighted Balance $\in$ APX in the case of positive weights. To show that, we did the following:

* We introduced a new but equivalent definition of MAX Weighted Balance;

* We applied the scaling technique used by Crescenzi and Trevisan [CT94] to define MAX Polynomially Bounded Weighted Balance;

* We defined an AP-reduction from MAX Weighted Balance to MAX Polynomially Bounded Weighted Balance, and

* We presented a 2-approximate polynomial-time algorithm for MAX Polynomially Bounded Weighted Balance, which in turn implies a 3-approximate polynomial-time algorithm for MAX Weighted Balance.

Next, we intend to extend the results introduced by Crescenzi and Trevisan [CST96] to "nice" subset problems. They studied the relative complexity of the arbitrarily weighted version, the
polynomially bounded weighted version, and the unweighted version of that class of problems. Surprisingly, they showed that for "nice" subset problems the approximation threshold was exactly the same for all three versions. We conjecture that it is also valid for a different kind of problem such as MAX Weighted Balance. The main result of this paper is a basic requirement to accomplish that.

Acknowledgements

The authors would like to thank Luca Trevisan for helpful discussions. Thanks also to Airton Castro for his remarks.

References


