

An FPTAS for Bargaining Networks with Unequal Bargaining Powers

Yashodhan Kanoria*
Department of Electrical Engineering
Stanford University
Email: ykanoria@stanford.edu

Abstract

Bargaining networks model social or economic situations in which agents seek to form the most lucrative partnership with another agent from among several alternatives. There has been a flurry of recent research studying Nash bargaining solutions (also called ‘balanced outcomes’) in bargaining networks, so that we now know when such solutions exist, and also that they can be computed efficiently, even by market agents behaving in a natural manner.

In this work we study a generalization of Nash bargaining, that models the possibility of unequal ‘bargaining powers’. This generalization was introduced in [KB+10], where it was shown that the corresponding ‘unequal division’ (UD) solutions exist if and only if Nash bargaining solutions exist, and also that a certain local dynamics converges to UD solutions when they exist. However, the bound on convergence time obtained for that dynamics was exponential in network size for the unequal division case. This bound is tight, in the sense that there exists instances on which the dynamics of [KB+10] converges only after exponential time. Other approaches, such as the one of Kleinberg and Tardos, do not generalize to the unsymmetrical case. Thus, the question of computational tractability of UD solutions has remained open.

In this paper, we provide an FPTAS for the computation of UD solutions, when such solutions exist. On a graph $G = (V, E)$ with weights (i.e. pairwise profit opportunities) uniformly bounded above by 1, our FPTAS finds an ϵ -UD solution in time $\text{poly}(|V|, 1/\epsilon)$. We also provide a fast *local* algorithm for finding ϵ -UD solution, providing further justification that a market can find such a solution.

*Part of this work was done while the author was visiting Microsoft Research New England. The author is supported by a 3Com Corporation Stanford Graduate Fellowship.

1 Introduction

Bargaining networks serve as a model for various social or economic interactions where agents seek to form pairs for mutual benefit (e.g. [CY92, Wil99, LY+01]). Situations which can be modeled as such include a housing market with buyers and sellers, a job market with job seekers and employers, or individuals seeking to form relationships and pair up. Bargaining networks are also referred to in the literature as ‘assignment markets’ [Ro84] or ‘exchange networks’ [SW93, KT08].

A bargaining network is an undirected graph, with weights on the edges representing potential profits if the corresponding pair of agents ‘trade’ with each other (see Section 1.1 for formal definitions). Profit from a trade is split between the participating agents as per a mutual agreement. Agents are constrained on the number of trades they can participate in. A natural postulate in this setting is that an outcome should be *stable*, i.e. no pair of agents should be able to do better by each abandoning a current partner and trading with each other instead. The solution concept of ‘balanced outcomes’ [Ro84, CY92, KT08] postulates further that each pair of agents that trade must play the pairwise Nash bargaining solution [Nas50], given the behavior of the rest of the network. Thus, the ‘edge surplus’ (cf. Eq. (1)), or the excess over the sum of ‘best alternatives’ for each of the two parties, is postulated to be split equally. This is called the *balance* condition.

However, it is natural to expect that such symmetry is rare in practice, and that some players tend to have greater ‘bargaining power’ than others. Such bargaining power can arise due to a variety of reasons. For example, a more patient player has more bargaining power, all else being equal. This phenomenon is well known in the Rubinstein game [R82] where nodes alternately make offers to each other until an offer is accepted – the node with less time discounting earns more in the subgame perfect Nash equilibrium.

Empirical findings confirm this. A recent experimental study of such networks [CJ+10] found that individual differences played a part in determining outcomes, including the observation that patience correlated positively with earnings. A previous study even estimates and ‘corrects’ for the effects of particular subject pairs to better uncover network structure effects [SW93]. This leads us to ask if the concept of ‘balanced outcomes’ can be suitably generalized to account for such asymmetry. It turns out that there is, in fact a simple generalization to the unsymmetrical case. Our previous work [KB+10] introduced the generalized concept of unsymmetrical ‘unequal division’ (UD) solutions, and also characterized the existence of such solutions.

Somewhat surprisingly, the various algorithms devised to compute solutions in the symmetric setting fail to generalize to the unequal division setting (see also Section 1.2.1). For example, the algorithm of Kleinberg and Tardos [KT08] proceeds via a sequence of linear programs that maximize the minimum ‘slack’. This does not seem to have a simple generalization to the asymmetric case. Thus, the question of computational tractability of solutions for the unsymmetrical case in bargaining networks has been open.

Besides computational tractability, another important question is “Can a market find the solution concept on its own?” In this line of work, one looks for simple, local mechanisms that converge to a solution concept. Azar et al [AB+09] proposed such a local mechanism for the bargaining networks problem. The convergence result in that work showed an exponential bound on convergence time for the symmetric case. Also, it does not generalize to the unsymmetrical case. Our recent work on local dynamics in bargaining networks [KB+10] introduced a new analysis technique that provides a proof of convergence even for the unsymmetrical case, and a polynomial bound on convergence to an approximate solution for the symmetrical case. However, a crucial issue (see Section 4 of this paper) led to a worst case exponential time to convergence in the unsymmetrical case. In this paper we resolve this issue, providing a new efficient local algorithm for the unsymmetrical case.

Contributions. This work makes the following contributions in the context of bargaining networks:

- We establish computational tractability for bargaining networks with unequal bargaining powers by providing the first FPTAS for the corresponding ‘unequal division’ solutions.
- We provide a simple local algorithm and show that it converges fast to approximate unequal division solutions. Specifically, it is a two phase algorithm: (i) The first phase consists of finding the maximum weight matching and a stable allocation using belief propagation [BSS05]. (ii) The second phase consists of unsymmetrical edge balancing of the allocation, converging to an approximate solution in polynomial time.

We note that the local algorithm we provide is similar to the one given by Azar et al [AB+09] for the symmetrical case. In that work also there is a phase of matching using belief propagation, followed by a phase of edge balancing. However, several critical differences in both the design and the analysis of the algorithm enable us to overcome limitations of their approach.

1.1 Model

A bargaining network consists of an undirected graph $G = (V, E)$ with positive weights on the edges, denoted by $(w_e, e \in E) \in (0, W]^{|E|}$ (where $W > 0$ denotes an arbitrary bound on weights). Edges represent potential ‘trades’, and weights are the corresponding ‘profits’. Players are constrained on the number of trades they are allowed to participate in. For simplicity, we will work with the *one exchange rule*, i.e. each player is allowed to participate in at most one trade. All our results easily generalize to the case of arbitrary integral constraints on number of trades for each player.

If a pair of players trade with each other, the profit must be divided between them. Thus, a *trade outcome* or just an *outcome* consists of a matching M between players, and an *allocation* $\underline{\gamma} \in \mathbb{R}_+^{|V|}$ such that $\gamma_i + \gamma_j = w_{ij}$ for each pair $(i, j) \in M$, and for each node $k \in V$ that is unmatched under M , $\gamma_k = 0$.

Given a trade outcome $(\underline{\gamma}, M)$, we define implicit *offers* on all edges not in M . Let $(x)_+ \equiv \max(x, 0)$. For any $(i, j) \in E \setminus M$, node i offers node j an amount $(w_{ij} - \gamma_i)_+$, the idea being that i should be willing to switch partners if she can earn even slightly more. Thus, each node has a set of well defined ‘alternatives’ to its current partner in M . A natural postulate is that an outcome should be *stable*, i.e. for each node i , γ_i should be no smaller than the best alternative of node i (if i is unmatched under M , she should receive no non-zero offers). The stability condition can be concisely written as $\gamma_i + \gamma_j \geq w_{ij}$ for all $(i, j) \in E \setminus M$.

Let ∂i denote the set of neighbors of node i in G . For each edge $(ij) \in M$, we define the ‘edge surplus’ as the excess of w_{ij} over the sum of best alternatives, i.e.,

$$\text{Surp}_{ij}(\underline{\gamma}) = w_{ij} - \max_{k \in \partial i \setminus j} (w_{ik} - \gamma_k)_+ - \max_{l \in \partial j \setminus i} (w_{jl} - \gamma_l)_+. \quad (1)$$

We can think of each node in the network as having an inherent ‘bargaining power’, such that Surp_{ij} should be split in a manner determined by the bargaining powers of i and j . We adopt a general model where the surplus is postulated to be split as per a fraction $r_{ij} \in (0, 1)$ for each matched edge $(ij) \in M$. We call this *correct division*. Each r_{ij} can be an arbitrary number in the interval $(0, 1)$, independently for all edges.

Definition 1.1. A problem instance I consists of an undirected graph $G = (V, E)$, with positive weights $(w_e)_{e \in E}$ and split fractions $(r_{ij})_{(ij) \in E} \in (0, 1)^{|E|}$. An arbitrary direction is chosen on each edge for purposes of specifying the split fraction. If r_{ij} is specified, then it is implicit that $r_{ji} = 1 - r_{ij}$.

Definition 1.2 (Correct division). *An outcome $(\underline{\gamma}, M)$ is said to satisfy correct division if, for all $(ij) \in M$,*

$$\gamma_i = \max_{k \in \partial i \setminus j} (w_{ik} - \gamma_k)_+ + r_{ij} \text{Surp}_{ij} \quad (2)$$

where $r_{ji} = 1 - r_{ij}$ and Surp_{ij} is defined by Eq. (1).

Note that it follows from Eq. (2) and Eq. (1) that $\gamma_j = w_{ij} - \gamma_i = \max_{l \in \partial j \setminus i} (w_{jl} - \gamma_l)_+ + r_{ji} \text{Surp}_{ij}$.

Definition 1.3. *An outcome $(\underline{\gamma}, M)$ is said to be an unequal division (UD) solution if it is stable and satisfies correct division (cf. Definition 1.2).*

1.2 Related work

We present here a short review of relevant related work.

Recall the linear programming relaxation of the maximum weight matching problem

$$\begin{aligned} & \text{maximize} && \sum_{(i,j) \in E} w_{ij} x_{ij}, \\ & \text{subject to} && \sum_{j \in \partial i} x_{ij} \leq 1 \quad \forall i \in V, \quad x_{ij} \geq 0 \quad \forall (i,j) \in E. \end{aligned} \quad (3)$$

The dual problem to (3) is:

$$\begin{aligned} & \text{minimize} && \sum_{i \in V} y_i, \\ & \text{subject to} && y_i + y_j \geq w_{ij} \quad \forall (i,j) \in E, \quad y_i \geq 0 \quad \forall i \in V \end{aligned} \quad (4)$$

Sotomayor [Sot05] characterized the existence of stable outcomes in exchange networks.

Lemma 1.4 ([Sot05, KT08]). *Stable outcomes exist if and only if the LP (3) has an integral optimum. Further, if $(\underline{\gamma}, M)$ is a stable outcome, then $\underline{\gamma}$ is an optimum solution of the dual LP (4) and M is a maximum weight matching. Conversely, if the LP (3) has an integral optimum, then for any maximum weight matching M^* and any optimum \underline{y}^* of the dual LP (4), (\underline{y}^*, M^*) is a stable outcome.*

The above lemma follows from the stability condition $\gamma_i + \gamma_j \geq w_{ij}$ for all $(ij) \in M$. It implies, in particular, that all instances on bipartite graphs possess stable outcomes.

There have been several recent works on the symmetrical ‘balanced outcome’ solution concept (corresponding to $r_{ij} = 1/2$ for all $(ij) \in E$), following a paper by Kleinberg and Tardos [KT08, AB+09, BH+10, KB+10].

Though our previous work [KB+10] focuses on the symmetrical case, it also introduces unequal division solutions. Further, it shows that unequal division solutions exist if and only if Nash bargaining solutions exist.

Theorem 1.5 ([KB+10]). *A problem instance admits a UD solution if and only if it admits a stable outcome (which occurs iff the LP (3) has an integral optimum).*

This generalizes a result of Kleinberg and Tardos for existence of balanced outcomes [KT08].

[KB+10] also shows that a certain local dynamics converges to UD solutions, when such solutions exist. However, the bound on time to convergence is exponential in the network size (in contrast to the symmetrical case), and this bound turns out to be tight in worst case (see Section 4). Here, we resolve this issue, providing a new FPTAS for computing approximate UD solutions.

1.2.1 Relationship to Cooperative games

Recent work by Bateni et al [BH+10] shows that the bargaining network setting can be viewed as a cooperative game, making this problem susceptible to a large body of literature (see also [Ro84]). This literature defines various solution concepts such as nucleolus, kernel and prekernel, and also investigates means to compute these solutions for various classes of games.¹ It is noteworthy that all the solution concepts studied are symmetric in the players. Whereas such concepts may form a reasonable predictive framework in the absence of player specific information, we also want to ask “Can the players reach an appropriate ‘solution’ when there is asymmetry?” To this end, we would like to establish computational tractability in the asymmetric case.

However, a little investigation reveals that the approaches devised to compute various (symmetric) solution concepts rely heavily on the symmetry in their respective definitions. For instance, the polynomial time algorithm in Faigle et al [FKK01] for finding a point in the least core intersection prekernel uses two components –a transfer scheme and a linear programming based update– neither of which work in the unsymmetrical case.

The situation is similar with regard to simple ‘transfer schemes’ that converge to a solution concept. For the general cooperative game problem, Maschler proposed a simple transfer scheme to approximate points in the prekernel. A version of this scheme was shown to converge by Stearns, and a simpler proof of convergence was provided by Faigle et al [FKK01], in the general cooperative game setting. However, both proofs suffer from two drawbacks: (a) they depend on the symmetry of the solution concepts, (b) the bound on convergence time (if any) is exponential in network size. Essentially the same transfer scheme was used in Azar et al [AB+09] for bargaining networks (see [BH+10] for the connection), and the proof of convergence suffered from the same drawbacks.

The current work addresses computational tractability for the asymmetric case in the bargaining network setting, where an appropriate asymmetric solution concept can be readily defined.

1.3 Outline of the paper

We present our FPTAS in Section 2, along with a proof that it returns an ϵ -UD solution in polynomial time. We present a fast local algorithm for this problem in subsection 2.1. Each of the algorithms involve an iterative ‘rebalancing’ phase. Section 3 contains proofs of some key Lemmas used. In Section 4, we demonstrate the importance of ensuring that we stay within the subset of *stable* allocations in our iterative updates. This insight is critically used in our construction of an FPTAS. Appendix B shows a polynomial time local ‘reduction’ from the problem of finding an ϵ -UD solution to the problem of finding a maximum weight matching.

2 Main results

First we define an approximate version of correct division, asking that Eq. (2) be satisfied to within an additive ϵ , for all matched edges.

Definition 2.1 (ϵ -Correct division). *An outcome $(\underline{\gamma}, M)$ is said to satisfy ϵ -correct division if, for all $(ij) \in M$,*

$$|\gamma_i - \max_{k \in \partial i \setminus j} (w_{ik} - \gamma_k)_+ - r_{ij} \text{Surp}_{ij}(\underline{\gamma})| \leq \epsilon \quad (5)$$

where $\text{Surp}_{ij}(\cdot)$ is defined by Eq. (1).

¹[BH+10] shows that stable, balanced outcomes in bargaining networks correspond to the core intersection prekernel.

We define approximate UD solutions as follows:

Definition 2.2. *An outcome $(\underline{\gamma}, M)$ is an ϵ -UD solution for $\epsilon \geq 0$ if it is stable and it satisfies ϵ -correct division (cf. Definition 2.1).*

This is analogous to the definition of ϵ -Nash equilibrium (see, e.g. [DP09]).

It follows from Lemma 1.4 that ϵ -UD solutions exist iff the LP (3) admits an integral optimum. This is the same as the requirement for existence of UD solutions (see Theorem 1.5). Our main result is the following:

Theorem 2.3. *There is a $\text{Poly}(|V|, 1/\epsilon)$ algorithm such that for any problem instance with weights uniformly bounded by 1, i.e. $(w_e, e \in E) \in (0, 1]^{|E|}$:*

- *If the instance admits a UD solution, the algorithm finds an ϵ -UD solution.*
- *If the instance does not admit a UD solution the algorithm returns a message UNSTABLE.*

Our approach to finding an ϵ -UD solution consists of two main steps:

1. Find a maximum weight matching M^* and a dual optimum $\underline{\gamma}$ (solution to the dual LP (4)). Thus, form a stable outcome $(\underline{\gamma}, M^*)$. Else certify that the instance has no UD solution.
2. Iteratively update the allocation $\underline{\gamma}$ without changing the matching. Updates are local, and are designed to converge fast to an allocation satisfying the ϵ -correct division solution *while maintaining stability*. Thus, we arrive at an ϵ -UD solution.

As mentioned earlier, this is similar to the approach of [AB+09]. The crucial differences (enabling our results) are: (i) we stay within the space of stable outcomes, and (ii) our analysis of convergence.

First let us focus on obtaining an FPTAS using the steps above. Later we describe how to make the algorithm local.

Step 1 can be carried out by finding a maximum weight matching M^* (e.g. [GT91]) and also solving the the dual linear program (4). For the dual LP, let \mathcal{V} be the optimum value and let $\underline{\gamma}$ be an optimum solution. We now use Lemma 1.4. If the weight of M^* is smaller than \mathcal{V} , we return UNSTABLE, since we know that no stable outcome exists, hence no UD solution (or ϵ -UD solution) exists. Else, $(\underline{\gamma}, M^*)$ is a stable outcome. This completes step 1! The computational effort involved is $\text{poly}(|V|)$.

In step 2, we fix the matching M^* , and rebalance the edges iteratively. It turns out to be crucial that our iterative updates preserve stability. Section 4 demonstrates that the rebalancing procedure can take an exponentially large time to reach an approximate UD solution if stability is not preserved.

We motivate the rebalancing procedure briefly, before we give a detailed description and state results. Imagine an edge $(i, j) \in M^*$. Since we start with a stable outcome, the edge weight w_{ij} is at least the sum of the best alternatives, i.e. $\text{Surp}_{ij} \geq 0$. Suppose we change the division of w_{ij} into γ'_i, γ'_j so that the Surp_{ij} is divided as per the prescribed split fraction r_{ij} . Earnings of all other nodes are left unchanged. Since $r_{ij} \in (0, 1)$, γ'_i is at least as large as the best alternative of i , as was the case for γ_i . This leads to $\gamma'_i + \gamma_k \geq w_{ik}$ for all $k \in \partial i \setminus j$. A similar argument holds for node j . In short, *stability is preserved!*

It turns out that the analysis of convergence is simpler if we analyze synchronous updates, as opposed to asynchronous updates as described above. Moreover, we find that simply choosing an appropriate ‘damping factor’ allows us to ensure that stability is preserved even with synchronous updates. We use a powerful technique introduced in our recent work [KT08] to prove convergence.

Table 1: Local algorithm that converts stable outcome to ϵ -UD solution

EDGE REBALANCING(Instance I , Stable outcome $(\underline{\gamma}, M)$, Damping factor κ , Error target ϵ)	
1:	Check $\kappa \in (0, 1/2]$, $\epsilon > 0$, $(\underline{\gamma}, M)$ is stable outcome
2:	If (Check fails) Return ERROR
3:	$\underline{\gamma}^0 \leftarrow \underline{\gamma}$
4:	$t \leftarrow 0$
5:	Do
6:	ForEach $(i, j) \in M$
7:	$\gamma_i^{\text{reb}} \leftarrow \max_{k \in \partial i \setminus j} (w_{ik} - \gamma_k^t)_+ + r_{ij} \text{Surp}_{ij}(\underline{\gamma}^t)$
8:	$\gamma_j^{\text{reb}} \leftarrow \max_{l \in \partial j \setminus i} (w_{jl} - \gamma_l^t)_+ + r_{ji} \text{Surp}_{ij}(\underline{\gamma}^t)$
9:	End ForEach
10:	ForEach $i \in V$ that is unmatched under M
11:	$\gamma_i^{\text{reb}} \leftarrow 0$
12:	End ForEach
13:	If $(\ \underline{\gamma}^{\text{reb}} - \underline{\gamma}^t\ _\infty \leq \epsilon)$ Break Do
14:	$\underline{\gamma}^{t+1} = \kappa \underline{\gamma}^{\text{reb}} + (1 - \kappa) \underline{\gamma}^t$
15:	$t \leftarrow t + 1$
16:	End Do
17:	Return $(\underline{\gamma}^t, M)$

Table 1 shows the algorithm EDGE REBALANCING we use to complete step 2. Note that each iteration of the loop can requires $O(|E|)$ simple operations.

Correctness of EDGE REBALANCING:

A priori, it is not clear that $\underline{\gamma}^t$ computed by EDGE REBALANCING is a stable allocation (or even an allocation) corresponding to M , for $t > 0$. The following lemma eliminates this concern.

Lemma 2.4. *If EDGE REBALANCING is given a valid input satisfying the ‘Check’ on line 1, then $(\underline{\gamma}^t, M)$ is a stable outcome for all $t \geq 0$.*

This guarantees that EDGE REBALANCING returns an ϵ -UD solution if it terminates (unless it returns ERROR). The lemma is a straightforward consequence of the constraint $\kappa \leq 1/2$ (proof in Section 3).

Convergence of EDGE REBALANCING:

Next we need to show that the rebalancing algorithm terminates fast at an ϵ -UD solution. Note that the termination condition $\|\underline{\gamma}^{\text{reb}} - \underline{\gamma}^t\|_\infty \leq \epsilon$ on Line 13 is equivalent to ϵ -correct division.

Lemma 2.5. *For any instance with weights bounded by 1, i.e. $(w_e, e \in E) \in (0, 1]^{|E|}$, if EDGE REBALANCING is given a valid input, it terminates in T iterations, where*

$$T \leq \left\lceil \frac{1}{\pi \kappa (1 - \kappa) \epsilon^2} \right\rceil, \quad (6)$$

and returns an outcome satisfying ϵ -correct division (cf. Definition 2.1). Here $\pi = 3.14159 \dots$

The proof is in Section 3.

Using Lemmas 2.4 and Lemmas 2.5, we immediately obtain our main result, Theorem 2.3.

Proof of Theorem 2.3. We showed that step 1 can be completed in time $\text{poly}(|V|)$. If the instance has no UD solutions then the algorithm returns UNSTABLE. Else we obtain a stable outcome and proceed to step 2.

Step 2 is performed using EDGE REBALANCING. The input is the instance, the stable outcome obtained from step 1, $\kappa = 1/2$ (for example) and the target error value $\epsilon > 0$. Lemmas 2.4 and 2.5 show that EDGE REBALANCING terminates after at most $\lceil 1/(\pi\kappa(1-\kappa)\epsilon^2) \rceil$ iterations, returning a outcome that is stable and satisfies ϵ -correct division, i.e. an ϵ -UD solution. Moreover, each iteration requires $O(|E|)$ simple operations. Hence, step 2 is completed in $O(|E|/\epsilon^2)$ simple operations.

The total number of operations required by the entire algorithm is thus $\text{poly}(|V|, 1/\epsilon)$. \square

2.1 A Fast Local Algorithm

Our algorithm EDGE REBALANCING for step 2 is local/distributed, with each matched edge in the graph being updated according to the same, time invariant rule. This rule is a simple function of the edge parameters (weight, split fraction), and the current earnings of nodes in the 1-hop neighborhood. Only the termination condition is centrally computed, but even that *can be replaced with fixed time* $T_* = \lceil 1/(\pi\kappa(1-\kappa)\epsilon^2) \rceil$ at which to terminate (cf. Section 3, Remark 3.1). Note that T_* is *independent of network size*. It is also worth mention that since stability is preserved, no player ever has incentive to change her partner. Thus, EDGE REBALANCING constitutes a plausible model for behavior of market participants, after they have attained a stable outcome. Damping can be interpreted as inertia to change in the status quo.

We now show that step 1 can also be accomplished by a fast local algorithm, when the LP (3) has a unique optimum (this condition is generic, see Remark A.1, Appendix A).

The local algorithm we use is belief propagation for maximum weight matching [BSS05, BB+07, SMW07, HuJ07]. This is a message passing algorithm with iterative updates of a ‘message’ vector $\underline{m} \in [0, W]^{2|E|}$. There are two messages on each edge (i, j) , denoted by $m_{i \rightarrow j}$ and $m_{j \rightarrow i}$, one in each direction. The algorithm performs iterative updates according to

$$m_{i \rightarrow j}^{t+1} = \left(w_{ij} - \max_{k \in \partial i \setminus j} m_{k \rightarrow i}^t \right)_+ \quad (7)$$

The initialization is the all-zero message vector $\underline{m}^0 = \underline{0}$. We denote this algorithm by BP-MWM.

Lemma 2.6 ([BSS05, BB+07, SMW07]). *Suppose LP (3) has a unique optimum. BP-MWM converges iff the optimum is integral. Further, if LP (3) has an integral optimum corresponding to matching M^* , then the messages converge in $\lceil \frac{2|V|W}{g} \rceil$ iterations to a fixed point \underline{m}^* satisfying the following. For any $i \in V$, if i is matched under M^* to j , then $\arg\max_{k \in \partial i} m_{k \rightarrow i}^* = \{j\}$ and $m_{j \rightarrow i}^* > 0$. If i is unmatched under M^* then $m_{k \rightarrow i}^* = 0$ for every $k \in \partial i$.*

Note how the condition for convergence of BP-MWM is the same as that for the existence of UD solutions! Here g is difference in the weights of the heaviest and next heaviest corner of the matching polytope (given by the constraints of LP (3)). We call it the *LP gap*.

In the case that LP (3) has a unique optimum, BP-MWM thus answers “Does the LP (3) have an integral optimum?” If yes, it also finds the maximum weight matching.

In fact, one also can find an optimum solution to the dual LP (4) from \underline{m}^* , when BP-MWM converges (see also [KB+10, Appendix F.1]). Consider any $i \in V$. Sort the messages $(m_{k \rightarrow i}^*, k \in \partial i)$

is non-increasing order. Denote the value of the first item in the sorted list by $\mu_i(1)$ and the next value by $\mu_i(2)$. Define $y_i^* \equiv (\mu_i(1) + \mu_i(2))/2$. The following is proved in Appendix A.

Proposition 2.7. *The construction above produces \underline{y}^* , an optimal solution to the dual LP (4).*

Thus, we obtain a stable outcome (\underline{y}^*, M^*) from the BP fixed point \underline{m}^* (see Lemma 1.4).

We mention here that BP-MWM can be interpreted as a bargaining process [AB+09, Appx A].

Remark 2.8. *The performance of BP-MWM seems to be lacking in two respects. First, it fails when the LP (3) has an integral optimum that is not unique. Second, the bound on convergence time depends inversely on LP gap g which may be arbitrarily small (in fact the bound is tight in worst case). We make three comments on this issue:*

- (i) *These ‘flaws’ appear to be inevitable. We are not aware of any local algorithm for maximum weight matching that overcomes them.*
- (ii) *For any instance on a bipartite graph, the LP gap g is larger than inverse polynomial in $|V|$ with probability close to 1 under small random perturbations [KB+10, Lemma 1]. Thus, BP-MWM is likely to converge in time $\text{poly}(|V|)$ on bipartite graphs as per this ‘smoothed analysis’.*
- (iii) *Appendix B shows that if we are given a maximum weight matching M^* for an instance possessing a UD solution, then we can locally construct a stable outcome in $\text{poly}(|V|)$ operations. Using this, we obtain a local polynomial time reduction from the problem of finding an ϵ -UD solution to the sub-problem of finding a maximum weight matching.*

3 Proofs of Lemmas 2.4 and 2.5

Proof of Lemma 2.4. We prove this lemma by induction on time t . Clearly (γ^0, M) is a stable outcome, since the input is valid. Suppose (γ^t, M) is a stable outcome.

Consider any $(i, j) \in M$. It is easy to verify that $\gamma_i^{\text{reb}} + \gamma_j^{\text{reb}} = w_{ij}$, for $\underline{\gamma}^{\text{reb}}$ computed from $\underline{\gamma}^t$ in Lines 8-11 of EDGE REBALANCING. Also, we know that $\gamma_i^t + \gamma_j^t = w_{ij}$. It follows that $\gamma_i^{t+1} + \gamma_j^{t+1} = w_{ij}$ as needed. For $i \in V$ unmatched under M , $\gamma_i^t = 0$ by hypothesis and $\gamma_i^{\text{reb}} = 0 \Rightarrow \gamma_i^{t+1} = 0$ as needed.

Consider any $(i, k) \in E \setminus M$. We know that $\gamma_i^t + \gamma_k^t \geq w_{ik}$. We want to show the corresponding inequality at time $t+1$. Define $\sigma_{ik}^t \equiv \gamma_i^t + \gamma_k^t - w_{ik} \geq 0$.

Claim: $\gamma_i^{\text{reb}} \geq \gamma_i^t - \sigma_{ik}^t$

If we prove the claim, it follows that a similar inequality holds for γ_k^{reb} , and hence $\gamma_i^{\text{reb}} + \gamma_k^{\text{reb}} \geq \gamma_i^t + \gamma_k^t - 2\sigma_{ik}^t = w_{ik} - \sigma_{ik}^t$. It then follows from the definition in Line 14 that $\gamma_i^{t+1} + \gamma_k^{t+1} \geq w_{ik}$, for any $\kappa \in (0, 1/2]$. This will complete our proof that $(\underline{\gamma}^{t+1}, M)$ is a stable outcome.

Let us now prove the claim. Suppose i is matched under M . Using the definition in Line 7 (Line 8 contains a symmetrical definition), $\gamma_i^{\text{reb}} \geq \max_{k' \in \partial i \setminus j} (w_{ik'} - \gamma_{k'}^t)_+$ since $\text{Surp}_{ik}(\underline{\gamma}^t) \geq 0$. Hence,

$$\gamma_i^{\text{reb}} \geq (w_{ik} - \gamma_k^t)_+ \geq (w_{ik} - \gamma_k^t) = \gamma_i^t - \sigma_{ik}^t$$

as needed. If i is not matched under M , then $\gamma_i^t = \gamma_i^{\text{reb}} = 0$, so the claim follows from $\sigma_{ik}^t \geq 0$. \square

Proof of Lemma 2.5. This result is proved using the powerful technique introduced in our recent work [KB+10]. We show that the iterative updates of EDGE REBALANCING can be written as

$$\underline{\gamma}^{t+1} = \kappa \mathsf{T} \underline{\gamma}^t + (1 - \kappa) \underline{\gamma}^t \quad (8)$$

where T is a non-expansive self mapping of a bounded convex subset of a normed linear space.

The linear space we consider is simply $\mathbb{R}_+^{|V|}$. Let $\mathcal{A}_M \subseteq [0, W]^{|V|}$ be the set of allocations corresponding to matching M . It is easy to see that \mathcal{A}_M is a bounded convex set. We define $\mathsf{T} : \mathcal{A}_M \rightarrow \mathcal{A}_M$ as the product of two operators, ‘rebalancing’ operator $\mathsf{T}^{\text{reb}} : \mathcal{A}_M \rightarrow \mathcal{A}_M^{\text{ext}}$ and a ‘thresholding’ operator $\mathsf{T}^{\text{thr}} : \mathcal{A}_M^{\text{ext}} \rightarrow \mathcal{A}_M$. Here $\mathcal{A}_M^{\text{ext}} \supseteq \mathcal{A}_M$ is set of allocations corresponding to matching M , with the non-negativity constraint relaxed. We define T^{reb} as follows. For each $i \in V$ that is unmatched under M , $(\mathsf{T}^{\text{reb}}\underline{\gamma})_i \equiv 0$. For each $(i, j) \in M$,

$$(\mathsf{T}^{\text{reb}}\underline{\gamma})_i \equiv \max_{k \in \partial i \setminus j} (w_{ik} - \gamma_k)_+ + r_{ij} \text{Surp}_{ij}(\underline{\gamma}) \quad (9)$$

$$(\mathsf{T}^{\text{reb}}\underline{\gamma})_j \equiv w_{ij} - (\mathsf{T}^{\text{reb}}\underline{\gamma})_i = \max_{l \in \partial j \setminus i} (w_{jl} - \gamma_l)_+ + r_{ji} \text{Surp}_{ij}(\underline{\gamma}) \quad (10)$$

Note that $\underline{\gamma}^{\text{reb}}$ as defined in Lines 6-12 of EDGE REBALANCING is exactly $\mathsf{T}^{\text{reb}}\underline{\gamma}^t$. Also note that $\mathsf{T}^{\text{reb}}\underline{\gamma} \in \mathcal{A}_M^{\text{ext}}$ as required.

We define T^{thr} as follows. For each $i \in V$ that is unmatched under M , $(\mathsf{T}^{\text{thr}}\underline{\gamma})_i \equiv 0$. For each $(i, j) \in M$, there are three cases.

$$\gamma_i < 0: (\mathsf{T}^{\text{thr}}\underline{\gamma})_i \equiv 0, (\mathsf{T}^{\text{thr}}\underline{\gamma})_j \equiv w_{ij}$$

$$\gamma_j < 0: (\mathsf{T}^{\text{thr}}\underline{\gamma})_i \equiv w_{ij}, (\mathsf{T}^{\text{thr}}\underline{\gamma})_j \equiv 0$$

$$\gamma_i \geq 0, \gamma_j \geq 0: (\mathsf{T}^{\text{thr}}\underline{\gamma})_i \equiv \gamma_i, (\mathsf{T}^{\text{thr}}\underline{\gamma})_j \equiv \gamma_j$$

Note that $\gamma_i < 0$ and $\gamma_j < 0$ cannot occur simultaneously since $\gamma_i + \gamma_j = w_{ij}$. Also note that the result of operating with T^{thr} is in \mathcal{A}_M .

Consider the composite operator $\mathsf{T} \equiv \mathsf{T}^{\text{thr}}\mathsf{T}^{\text{reb}}$. If T operates on a stable outcome, the output of operator T^{reb} is a non-negative allocation (since $\text{Surp}_{ij} \geq 0$ for every $(i, j) \in M$) with earnings of unmatched nodes being 0, and T^{thr} acts simply as an identity operator. It follows (using Lemma 2.4) that Lines 6-12 define $\underline{\gamma}^{\text{reb}} = \mathsf{T}\underline{\gamma}^t$. Thus, we have verified that the iterative updates of EDGE REBALANCING (Line 14) correspond to Eq. (8).

Next, we show that T is non expansive in sup norm, i.e. for any $\underline{\gamma}^a, \underline{\gamma}^b \in \mathcal{A}_M$,

$$\|\mathsf{T}\underline{\gamma}^a - \mathsf{T}\underline{\gamma}^b\|_\infty \leq \|\underline{\gamma}^a - \underline{\gamma}^b\|_\infty \quad (11)$$

We prove this by showing that each of T^{reb} and T^{thr} is non-expansive in sup norm.

Consider T^{reb} . Take any $(i, j) \in M$. Rewriting Eq. (9) using Eq. (1), we have

$$(\mathsf{T}^{\text{reb}}\underline{\gamma})_i = r_{ij}w_{ij} + (1 - r_{ij}) \max_{k \in \partial i \setminus j} (w_{ik} - \gamma_k)_+ - r_{ij} \max_{l \in \partial j \setminus i} (w_{jl} - \gamma_l)_+$$

Now $x \mapsto (w - x)_+$ is non-expansive, and the ‘max’ operator is non-expansive. Hence, using the triangle inequality we obtain

$$|(\mathsf{T}^{\text{reb}}\underline{\gamma}^a)_i - (\mathsf{T}^{\text{reb}}\underline{\gamma}^b)_i| \leq \|\underline{\gamma}^a - \underline{\gamma}^b\|_\infty$$

and similarly for j . For each k that is unmatched under M , $(\mathsf{T}^{\text{reb}}\underline{\gamma})_k = 0$. It follows that T^{reb} is non-expansive in sup norm.

Next consider T^{thr} . For each k that is unmatched under M , $(\mathsf{T}^{\text{thr}}\underline{\gamma})_k = 0$. For any $(i, j) \in M$, we can write $(\mathsf{T}^{\text{thr}}\underline{\gamma})_i = \max(\min(\gamma_i, w_{ij}), 0)$. Since the ‘max’ and ‘min’ operators are non-expansive, it follows that T^{thr} is non-expansive in sup norm.

Thus, we have shown that T is a non-expansive self mapping of a bounded convex set of diameter 1 (since $W = 1$). Also, $\underline{\gamma}^t$ is obtained via iterative updates as per Eq. (8). The main theorem in [BB96] tells us that

$$\|\mathsf{T}\underline{\gamma}^t - \underline{\gamma}^t\|_\infty \leq \frac{1}{\sqrt{\pi\kappa(1-\kappa)}t} \quad (12)$$

Eq. (6) follows. Also, $\|\mathsf{T}^{\text{reb}}\underline{\gamma}^T - \underline{\gamma}^T\|_\infty = \|\underline{\gamma}^{\text{reb}} - \underline{\gamma}^T\|_\infty \leq \epsilon$ implies ϵ -correct division for $(\underline{\gamma}^T, M)$. \square

Remark 3.1. In light of Eq. (12), we could have simply used a fixed termination time of $T_* = \lceil 1/(\pi\kappa(1-\kappa)\epsilon^2) \rceil$, instead of the termination condition in Line 13. Eq. (12) guarantees that $\underline{\gamma}^{T_*}$ satisfies the ϵ -correct division condition.

Remark 3.2. If we remove the termination condition on Line 13 of EDGE REBALANCING (and iterate forever), [I76, Corollary 1] tells us that we converge to some $\underline{\gamma}^*$ such that $\mathsf{T}\underline{\gamma}^* = \underline{\gamma}^*$, i.e. we reach an exact UD solution. (Note that Lemma 2.4 gives stability of the iterates, and stability of the limit point $\underline{\gamma}^*$ follows.) As a corollary, we recover Theorem 1.5 on existence of UD solutions.

4 Stability is Critical

This section demonstrates that our approach of starting with a stable allocation, and ensuring that stability is preserved, plays a critical role in our construction of an FPTAS using iterative edge rebalancing.

Let $n \equiv |V|$. Appendix C shows the following. There is a sequence of instances $(I_n, n \geq 8)$, such that for each instance in the sequence the following holds. (a) The instance admits a UD solution. (b) There is an outcome $(\underline{\gamma}, M^*)$ on a maximum weight matching M^* such that:

1. The outcome satisfies ϵ -correct division for $\epsilon = 2^{-cn}$.
2. (Stability violation) There is a ‘bad’ edge $(i, j) \notin M^*$ such that $\gamma_i + \gamma_j \leq w_{ij} - 1$

where $c > 0$ is a constant. Split fractions are bounded within $[r, 1-r]$ for arbitrary desired $r \in (0, 1/2)$ (c depends on r). Also, the weights are uniformly bounded by a constant $W(r)$.

We now describe the implications of such a construction. Suppose we perform edge balancing on the example outcome (as per Eq. (8), using operator T defined there), i.e. $\underline{\gamma}^0 \equiv \underline{\gamma}$. We know that $\|\mathsf{T}\underline{\gamma}^0 - \underline{\gamma}^0\|_\infty \leq \epsilon$, since $\underline{\gamma}^0$ satisfies ϵ -correct division. Define $\mathsf{T}_\kappa \equiv \kappa\mathsf{T} + (1-\kappa)\mathsf{I}$, where I is the identity operator. Eq. (8) simply corresponds to iterating with T_κ , i.e. $\underline{\gamma}^t = \mathsf{T}_\kappa^t \underline{\gamma}^0$. Clearly, $\|\mathsf{T}_\kappa \underline{\gamma}^0 - \underline{\gamma}^0\|_\infty \leq \epsilon$. Also, it follows from non-expansivity of T (as per Eq. (11)) that T_κ is non-expansive in sup norm. As a consequence $\|\mathsf{T}_\kappa \underline{\gamma}^t - \underline{\gamma}^t\|_\infty \leq \epsilon$ for all $t \geq 0$. Thus, successive iterates differ by at most ϵ in sup norm, meaning that no coordinate changes by more than ϵ per iteration. Suppose we want to reach a configuration that satisfies both (1/2)-stability ($\gamma_k + \gamma_l \geq w_{kl} - 1/2$ for each $(k, l) \in E$) and the (1/2)-correct division condition. One of γ_i and γ_j must change by at least $1/4$ for the ‘bad’ edge (i, j) to satisfy (1/2)-stability, i.e. $\gamma_i + \gamma_j \geq w_{ij} - 1/2$. But this will take at least $1/(4\epsilon) = 2^{\Omega(n)}$ iterations!

Thus, *it can take exponential time to reach an approximate UD solution if we do not stay within the space of stable outcomes while rebalancing.*

Remark 4.1. Essentially the same construction and reasoning shows that the dynamics of [KB+10] can take exponential time to reach an ϵ -UD solution.

Further directions. It remains open whether there is a polynomial algorithm that finds an exact UD solution.

Second, it would be interesting to identify other classes of games where solution concepts that are not symmetrical in the players can be naturally defined and motivated. Various classes of cooperative games seem like particularly suitable candidates.

Third, though we have found a fast local algorithm for finding ϵ -UD solutions, it does not constitute a natural description of market behavior of the type proposed in [KB+10]. However, as discussed in Section 4, there are instances where that dynamics does *not* quickly reach a solution in the unsymmetrical case. So it is unclear how to resolve this question.

Acknowledgements. The author would like to thank Andrea Montanari, Mohsen Bayati, R. Ravi and Mohammad Hossein Bateni for helpful discussions.

References

- [AS] B. Aspvall and Y. Shiloach, “A polynomial time algorithm for solving systems of linear inequalities with two variables pre inequality,” 20th IEEE Symposium Foundations of Computer Science, Puerto Rico, October 1979.
- [AB+09] Y. Azar, B. Birnbaum, L. Elisa Celis, N. R. Devanur and Y. Peres, “Convergence of Local Dynamics to Balanced Outcomes in Exchange Networks,” 50th IEEE Symposium Foundations of Computer Science, Atlanta, November 2009.
- [BB96] J. Baillon and R. E. Bruck, “The rate of asymptotic regularity is $O(1/\sqrt{n})$,” in: A.G. Kartsatos (ed.), Theory and applications of nonlinear operators of accretive and monotone type, Lecture Notes in Pure and Appl. Math. 178, Marcel Dekker, Inc., New York, 1996, 51-81.
- [BH+10] M. Bateni, M. Hajiaghayi, N. Immorlica and H. Mahini, “The cooperative game theory foundations of network bargaining games,” Intl. Colloquium on Automata, Languages and Programming, 2010.
- [BSS05] M. Bayati, D. Shah and M. Sharma, “Max-Product for Maximum Weight Matching: Convergence, Correctness, and LP Duality,” IEEE Trans. Inform. Theory, **54** (2008) 1241-1251
- [BB+07] M. Bayati, C. Borgs, J. Chayes, R. Zecchina, “On the exactness of the cavity method for Weighted b-Matchings on Arbitrary Graphs and its Relation to Linear Programs,” arXiv:0807.3159, (2007)
- [B88] D. P. Bertsekas, “The Auction Algorithm: A Distributed Relaxation Method for the Assignment Problem,” Annals of Operations Research, Vol. 14, pp. 105-123, 1988.
- [CJ+10] T. Chakraborty, S. Judd, M. Kearns, J. Tan, “A Behavioral Study of Bargaining in Social Networks,” to appear in Proc. EC, 2010.
- [CKK09] T. Chakraborty, M. Kearns and S. Khanna, “Network Bargaining: Algorithms and Structural Results” 10th ACM Conference on Electronic Commerce, Stanford, July 2009.
- [CY92] K. S. Cook and T. Yamagishi, “Power exchange in networks: A power-dependence formulation,” Social Networks, **14** (1992) 245-265
- [DP09] C. Daskalakis, C. Papadimitriou, “On oblivious PTAS’s for nash equilibrium,” Proc. STOC 2009.
- [EOB78] M. Edelstein and R.C. O’Brien, “Nonexpansive mappings, asymptotic regularity, and successive approximations,” J. London Math. Soc. 1 (1978), 547-554
- [FKK01] U. Faigle, W. Kern, and J. Kuipers, “On the computation of the nucleolus of a cooperative game,” International Journal of Game Theory, 30 (2001), pp. 79-98.
- [GS62] D. Gale and L. S. Shapley, “College admissions and the stability of marriage,” Amer. Math. Monthly 69 (1962), 9-15.

- [GT91] H. N. Gabow and R. E. Tarjan, “Faster scaling algorithms for general graph-matching problems,” *J. ACM*, 38(4):815-853, 1991.
- [I76] S. Ishikawa, “Fixed points and iteration of a nonexpansive mapping in a Banach space,” *Proc. American Mathematical Society*, Vol. 59, No. 1, 1976.
- [HuJ07] B. Huang, T. Jebara, “Loopy belief propagation for bipartite maximum weight b-matching,” *Artificial Intelligence and Statistics (AISTATS)*, March, 2007.
- [KB+10] Y. Kanoria, M. Bayati, C. Borgs, J. Chayes, and A. Montanari, “Fast Convergence of Natural Bargaining Dynamics in Exchange Networks,” *arXiv:1004.2079* (April 2010).
- [KT08] J. Kleinberg and E. Tardos, “Balanced outcomes in social exchange networks,” in *STOC*, 2008.
- [K01] U. Kohlenback, “A Quantitative Version Of A Theorem Due To Borwein-Reich-Shafrir,” *Numerical Functional Analysis and Optimization*, Vol. 22, Issue 5-6, August 2001.
- [LY+01] J.W. Lucas, C.W. Younts, M.J. Lovaglia, and B. Markovsky, “Lines of power in exchange networks,” *Social Forces*, 80 (2001) 185-214
- [M53] W. R. Mann, “Mean value methods in iteration,” *Proc. Amer. Math Soc.*, 4 (1953), 506–510
- [Nas50] J. Nash, “The bargaining problem,” *Econometrica*, **18** (1950) 155-162
- [Ro84] S.C. Rochford, “Symmetric pairwise-bargained allocations in an assignment market,” in *J. Economic Theory*, **34** (1984) 262-281.
- [R82] A. Rubinstein: Perfect equilibrium in a bargaining model. *Econometrica* 50, 97-109 (1982).
- [SMW07] S. Sanghavi, D. Malioutov, A. Willsky, “Linear Programming Analysis of Loopy Belief Propagation for Weighted Matching,” *Neural Information Processing Systems (NIPS)*, 2007
- [Sot05] M. Sotomayor, “On The Core Of The One-Sided Assignment Game,” 2005, http://www.usp.br/feaecon/media/fck/File/one_sided_assignment_game.pdf.
- [SW93] J. Skvoretz and D. Willer, “Exclusion and power: A test of four theories of power in exchange networks,” *American Sociological Review*, 58 (1993) 801-818
- [Wil99] D. Willer (ed.) *Network Exchange Theory*, Praeger, 1999

A Appendix to Section 2.1

Remark A.1. Fix a graph $G = (V, E)$ and maximum weight $W > 0$. We argue that the condition “LP (3) has a unique optimum” is generic in each of two different cases:

- (i) **All instances:** Let $\mathbf{G} = (0, W]^{|E|}$ be the set of all instances. Then the subset of instances with unique optimum is both open and dense in \mathbf{G} .
- (ii) **Instances with integral optimum:** Let $\mathbf{G}_I \subset (0, W]^{|E|}$ be the set of instances having an integral optimum. Let $\mathbf{G}_{UI} \subset \mathbf{G}_I$ be the set of instances having a unique integral optimum. It turns out that \mathbf{G}_I has dimension $|E|$ (i.e. the class of instances having an integral optimum is large) and that \mathbf{G}_{UI} is both open and dense in \mathbf{G}_I .

Proof of Proposition 2.7. Take any edge $(i, j) \in M^*$. From Lemma 2.6, we know that $m_{j \rightarrow i}^* > 0$. It follows that $m_{j \rightarrow i}^* = w_{ij} - \max_{l \in \partial j \setminus i} m_{l \rightarrow i}^*$. But $m_{j \rightarrow i}^* = \mu_i(1)$ and $\max_{l \in \partial j \setminus i} m_{l \rightarrow i}^* = \mu_j(2)$ by Lemma 2.6. Thus we obtain

$$\mu_i(1) = w_{ij} - \mu_j(2) \quad (13)$$

Similarly, we have

$$\mu_j(1) = w_{ij} - \mu_i(2) \quad (14)$$

Combining Eq. (13), (14), we obtain $y_i^* + y_j^* = w_{ij}$ as required.

Take any edge $(i, j) \notin M^*$. From Lemma 2.6, we know that $m_{i \rightarrow j} \leq \mu_j(2)$. Also, $\max_{k \in \partial i \setminus j} m_{k \rightarrow i} = \mu_i(1)$. It follows that $(w_{ij} - \mu_i(1))_+ \leq \mu_j(2) \Rightarrow w_{ij} \leq \mu_i(1) + \mu_j(2)$. Similarly, we obtain $w_{ij} \leq \mu_j(1) + \mu_i(2)$. Combining, it follows that $y_i^* + y_j^* \geq w_{ij}$ as required.

Note also that for any $i \in V$ not matched under M^* , it follows from Lemma 2.6 that $y_i^* = 0$. Thus, we have shown that \underline{y}^* is a feasible point for the dual LP (4), which also satisfies $\sum_{i \in V} y_i^* = \text{weight of } M^*$, i.e. it achieves the value of the primal LP (3). Hence, \underline{y}^* is an optimum solution to the dual LP (4). \square

B Local polynomial time ‘reduction’ to maximum weight matching

In this section we prove the following:

Claim B.1. Given a maximum weight matching M^* for an instance possessing a UD solution, an ϵ -UD solution can be constructed by a local algorithm with computational effort $\text{poly}(|V|, 1/\epsilon)$.

Our definition of ϵ -UD solutions retains a strict version of stability while relaxing the balance requirement to ϵ balance (cf. Definition 1.3). We use max-product belief propagation to find a stable allocation, given a maximum weight matching M^* . This is achieved locally and in polynomial time.

Consider the standard *undamped synchronous BP updates* given by:

$$\begin{aligned} m_{i \rightarrow j}^t &= (w_{ij} - \alpha_{i \setminus j}^t)_+ \\ \alpha_{i \setminus j}^{t+1} &= \max_{k \in \partial i \setminus j} m_{k \rightarrow i}^t \end{aligned} \quad (15)$$

This is equivalent to the update rule Eq. (7).

We use a carefully chosen initialization (different from the usual all-zero) to achieve our objective:

$$m_{i \rightarrow j}^0 = \begin{cases} w_{ij} & \text{if } (ij) \in M^* \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

Let the version of max-product BP message passing defined by (15) and (16) be denoted by \mathcal{A} . Our key result on \mathcal{A} is the following:

Claim B.2. *Algorithm \mathcal{A} converges to an exact fixed point in $2|E|$ iterations.*

The fixed points of update rule (15) can be characterized similarly to the fixed points of the ‘natural dynamics’ in our previous work [KB+10, Section 3].

Lemma B.3. *Consider an instance having an integral optimum to LP (3), corresponding to matching M^* . The update rule (15) has at least one fixed point. Let $(\underline{\alpha}^*, \underline{m}^*)$ be a fixed point. Then*

$$\begin{aligned} w_{ij} - \alpha_{i \setminus j}^* - \alpha_{j \setminus i}^* &\geq 0 \quad \forall (ij) \in M^* \\ w_{ij} - \alpha_{i \setminus j}^* - \alpha_{j \setminus i}^* &\leq 0 \quad \forall (ij) \notin M^* \end{aligned}$$

Also, for every $(ij) \notin M^*$, we have $\alpha_{i \setminus j}^* = m_{k \rightarrow i}^*$, where $(i, k) \in M^*$.

The above lemma follows directly from the arguments in ([KB+10], Appendix F.1).

We now show how Claim B.2 implies Claim B.1.

Proof of Claim B.1. Using Lemma B.3, we can show that for any fixed point \underline{m}^* of Eq. (15) for an instance such that LP (3) has an integral optimum M^* , the following holds: For any $i \in V$, if i is matched under M^* to j then $j \in \arg\max_{k \in \partial i} m_{k \rightarrow i}^*$ and $m_{j \rightarrow i}^* = w_{ij} - \alpha_{j \setminus i}^*$, whereas if i is unmatched under M^* then $m_{k \rightarrow i}^* = 0$ for every $k \in \partial i$. We use the construction for \underline{y}^* described in Section 2.1, and essentially the same proof of Proposition 2.7 goes through. A stable allocation (\underline{y}^*, M^*) thus follows from any fixed point \underline{m}^* of \mathcal{A} , where M^* is the given maximum weight matching.

Starting from a stable allocation, an ϵ -UD solution can be constructed with effort $O(|E|/\epsilon^2)$ using EDGE BALANCING as described in Section 2. The claim follows. \square

We devote the rest of this subsection to the proof of Claim B.2.

Next, we define a useful partial ordering on message vectors \underline{m} .

Definition B.4. *We say $\underline{m} \preceq \hat{\underline{m}}$ if the following hold:*

$$\begin{aligned} m_{i \rightarrow j} &\geq \hat{m}_{i \rightarrow j} \quad \forall (ij) \in M^* \\ m_{i \rightarrow j} &\leq \hat{m}_{i \rightarrow j} \quad \forall (ij) \notin M^* \end{aligned}$$

Lemma B.5. *Let \underline{m}^* be a fixed point of update rule (15). Algorithm \mathcal{A} satisfies*

$$\underline{m}^t \preceq \underline{m}^{t+1} \preceq \underline{m}^* \quad \forall t \geq 0. \quad (17)$$

Also, for all $(ij) \notin M^*$, we have

$$\alpha_{i \setminus j}^{t+1} = m_{k \rightarrow i}^t \quad \forall t \geq 0,$$

where $(i, k) \in M^*$.

Proof. It is trivial to check validity for $t = 0$. Suppose the result is true up to $t - 1$. Take any fixed point \underline{m}^* . Then

$$\underline{m}^t \preceq \underline{m}^* \quad (18)$$

Now consider any unmatched edge $(ij) \notin M^*$. If i is matched, say $(i, k) \in M^*$, it follows from Eq. (18) and Lemma B.3 that $\alpha_{i \setminus j}^{t+1} = m_{k \rightarrow i}^t$ as needed. Further, $m_{k \rightarrow i}^t \leq m_{k \rightarrow i}^{t-1} = \alpha_{i \setminus j}^t$, leading to

$$\begin{aligned} \alpha_{i \setminus j}^{t+1} &\leq \alpha_{i \setminus j}^t \\ \Rightarrow m_{i \rightarrow j}^{t+1} &\geq m_{i \rightarrow j}^t. \end{aligned}$$

as needed. Else if i is unmatched under M^* , $m_{k \rightarrow i}^t \leq m_{k \rightarrow i}^* = 0$ for all $k \in \partial i$. Hence, $\alpha_{i \setminus j}^{t+1} = \alpha_{i \setminus j}^t = 0$, leading to $m_{i \rightarrow j}^{t+1} = m_{i \rightarrow j}^t = w_{ij}$. This suffices.

On the other hand, for every matched edge (i, k) ,

$$\begin{aligned} \alpha_{i \setminus k}^{t+1} &= \max_{j \in \partial i \setminus k} m_{j \rightarrow i}^t \geq \max_{j \in \partial i \setminus k} m_{j \rightarrow i}^{t-1} = \alpha_{i \setminus k}^t \\ \Rightarrow m_{i \rightarrow k}^{t+1} &\leq m_{i \rightarrow k}^t. \end{aligned}$$

as needed.

The second inequality $\underline{m}^{t+1} \preceq \underline{m}^*$ can be established similarly.

Induction completes the proof. \square

Next, we present a key construction leading to a proof of Claim B.2 for the case that LP (3) has a unique optimum:

Choose a fixed point \underline{m}^* . A *critical path* P leading to a message $m_{i_1 \rightarrow i_0}^*$ is constructed as follows:

CRITICAL PATH(Instance G , BP-fixed point \underline{m}^*)	
1:	$k \leftarrow 1$
2:	While $m_{i_k \rightarrow i_{k-1}}^* > 0$
3:	Find $i_{k+1} \in \operatorname{argmax}_{j \in \partial i_k \setminus i_{k-1}} m_{j \rightarrow i_1}^*$.
4:	$k \leftarrow k + 1$
5:	If $(i_k, i_{k-1}) = (i_l, i_{l-1})$ for some $l < k$ then
6:	Break While;
7:	End If
8:	End While
9:	Return $(i_k, i_{k-1}, \dots, i_0)$

It is easy to see that a critical path can have at most $2|E| + 1$ directed edges since the path is terminated if a directed edge repeats, and there are $2|E|$ distinct directed edges in the graph.

Proof of Claim B.2: Unique LP optimum case. Take a fixed point \underline{m}^* . Consider any edge (i_1, i_0) . Let its critical path be $(i_k, i_{k-1}, \dots, i_0)$. There are two cases:

Case (i): $m_{i_k \rightarrow i_{k-1}}^* = 0$

This is the case where no directed edge repeats. In this case, we claim that $m_{i_1 \rightarrow i_0}^{k-1} = m_{i_1 \rightarrow i_0}^*$. We simply start with the evident $m_{i_k \rightarrow i_{k-1}}^0 = m_{i_k \rightarrow i_{k-1}}^*$ and move sequentially along the critical path. Lemmas B.3 and B.5 ensure that $\alpha_{i_{k-1} \setminus i_{k-2}}^1 = m_{i_k \rightarrow i_{k-1}}^0$, $\alpha_{i_{k-2} \setminus i_{k-3}}^2 = m_{i_{k-1} \rightarrow i_{k-2}}^1$ and so on. The monotonicity established plays a key role here. This leads to $m_{i_{k-1} \rightarrow i_{k-2}}^1 = m_{i_{k-1} \rightarrow i_{k-2}}^*$, $m_{i_{k-2} \rightarrow i_{k-3}}^2 = m_{i_{k-2} \rightarrow i_{k-3}}^*$ and so on, leading to the result.

Case (ii): $(i_k, i_{k-1}) = (i_l, i_{l-1})$ for some $l < k$

In this case we know that all messages along the critical path are strictly positive on unmatched edges, and hence are not thresholded at 0. It follows, going around the directed alternating cycle, $\mathcal{C} = (i_k, \dots, i_l)$ that for any directed edge $(\ell, \ell - 1) \in \mathcal{C}$

$$m_{\ell \rightarrow \ell-1}^* \leq m_{\ell \rightarrow \ell-1}^* + \text{wt. of unmatched edges in } \mathcal{C} - \text{wt. of matched edges in } \mathcal{C} < m_{\ell \rightarrow \ell-1}^*$$

since we have assumed that LP (3) has a unique solution. This is a contradiction.

Thus, Case (ii) never arises. Case (i) implies that $k \leq 2|E|$. Hence all messages converge to values at \underline{m}^* in $2|E|$ iterations. \square

Note that the proof above implies that the fixed point \underline{m}^* is unique!

Sketch of proof of Claim B.2: non-unique LP optimum. Claim B.2 holds also for the case where LP (3) is tight but not pointed. We only briefly sketch the proof in this case. In the non-unique optimum case, max product may have multiple fixed points. However, there is a unique smallest fixed point $\underline{m}^{*,\text{bot}}$ with respect to the partial ordering defined in B.4, and algorithm \mathcal{A} converges to this fixed point by monotone convergence (cf. Eq. (17)).

In fact, we can show that the same bound $2|E|$ holds on the time to convergence. To prove this we compare against the special fixed point $\underline{m}^{*,\text{bot}}$, whose minimality plays a crucial role. We use a similar critical path construction as for the unique optimum case. However, we have to be more careful here: we break ties in selecting an element of $\arg\max_{j \in \partial i_k \setminus i_{k-1}} m_{j \rightarrow i_1}^*$ non-deterministically (*there exists* a sequence of tie-breaking choices such that ...). The same cases (i) and (ii) arise in the proof of message convergence using the critical path (cf. proof for unique LP optimum). Case (i) goes through as before. For Case (ii), we use the minimality of $\underline{m}^{*,\text{bot}}$ to arrive at a contradiction. \square

C An Example showing that Stability is Critical

Let $n = |V|$. In this section, we construct a sequence of instances $(I_n, n \geq 8)$, such that for each instance in the sequence the following holds. (a) The instance admits a UD solution. (b) There is an outcome $(\underline{\gamma}, M^*)$ on a maximum weight matching M^* such that:

1. The outcome satisfies ϵ -correct division for $\epsilon = 2^{-cn}$.
2. (Stability violation) There is a ‘bad’ edge $(i, j) \notin M^*$ such that $\gamma_i + \gamma_j \leq w_{ij} - 1$

where $c > 0$ is a constant. Split fractions are bounded within $[r, 1-r]$ for arbitrary desired $r \in (0, 1/2)$ (c depends on r). Also, the weights are uniformly bounded by a constant $W(r)$.

Such a construction implies that we cannot hope to converge in worst case polynomial time to an approximate UD solution, if we start the rebalancing process (cf. Table 1) with an arbitrary allocation corresponding to M^* . This is discussed in Section 4. Thus, our strategy of staying within the space of *stable* configurations plays a critical role.

We now define the instance I_n . Let us first consider $n = 8N$, where $N \in \mathbb{Z}$. Later we show how to extend the construction to arbitrary $n \geq 8$. The graph $G_n = (V_n, E_n)$ we will consider is a simple ‘ring’. More precisely, $V_n = \{1, 2, \dots, n\}$ and $E_n = \{(1, 2), (2, 3), \dots, (n-1, n), (n, 1)\}$. All edges have the same weight W . This graph has two integral maximum weight matchings (cf. Remark C.1 below), we pick $M^* = \{(1, 2), (3, 4), \dots, (n-1, n)\}$. Given any $r \in (0, 1/2)$, we define the split fractions as follows:

$$\begin{aligned} r_{1,2} &= r_{3,4} = \dots = r_{2N-1,2N} = r \\ r_{4N,4N-1} &= r_{4N-2,4N-3} = \dots = r_{2N+2,2N+1} = r \end{aligned}$$

Note that the values of the split fraction on the edges $(2, 3), (4, 5), \dots \notin M^*$ are irrelevant, given our choice of matching. As before, $r_{i,i+1} = 1 - r_{i+1,i}$ is implicit.

For $l > 4N$, we define split fractions in a symmetrical way. Define ‘reflection’ $\mathcal{R} : \{4N + 1, 4N + 2, \dots, 8N\} \rightarrow \{1, 2, \dots, 4N\}$ as

$$\mathcal{R}(l) = 8N - l + 1 \quad (19)$$

We set $r_{2i+1,2i+2} = r_{\mathcal{R}(2i+1),\mathcal{R}(2i+1)}$ for all $i \in \{2N, 2N + 1, \dots, 4N - 1\}$.

Note that the allocation in which each node earns $W/2$, together with matching M^* , constitutes a UD solution for the instance defined.

Now we show how to construct an outcome $(\underline{\gamma}, M^*)$ satisfying properties 1 and 2 above. Let $\beta \equiv (1 - r)/r > 1$. Define

$$\epsilon' = \frac{1}{\beta^{N-1}}.$$

For $0 \leq i \leq N - 1$ we choose

$$\gamma_{2(N-i)} = \gamma_{2(N+i)+1} = \frac{W}{2} + \frac{1}{2} + \frac{1 - \beta^{-i}}{\beta - 1} \quad (20)$$

In each case $\gamma_{2j} = W - \gamma_{2j-1}$, since we want a valid outcome. Thus, we have defined $\gamma_1, \gamma_2, \dots, \gamma_{4N}$. The remaining earnings are defined as,

$$\gamma_i = W - \gamma_{\mathcal{R}(i)} \quad \text{for } i = 4N + 1, 4N + 1, \dots, 8N \quad (21)$$

It is easy to see that this definition satisfies the fixed sum constraints on all edges in M^* .

Importantly, note that it suffices to have $W \geq 1 + 2/(\beta - 1)$ to ensure that this is a valid allocation with each $\gamma_i \in (0, W)$. Choose $W \equiv 1 + 2/(\beta - 1)$ (for example).

See that $\gamma_{6N} + \gamma_{6N+1} = W - 1$ thus satisfying property 2. We show next that $(\underline{\gamma}, M^*)$ satisfies the ϵ' -correct division condition.

Consider the edge $(2(N - i), 2(N - i) + 1)$ for $1 \leq i \leq N - 1$. It is easy to see that $\gamma_{2(N-i)} + \gamma_{2(N-i)+1} = W + \beta^{-i}$. Also, $\gamma_{2N} + \gamma_{2N+1} = W + 1$. It follows from a short calculation that the exact ‘correct division’ requirement Eq. (2) is satisfied by the matched edges $(3, 4), (5, 6), \dots, (2N - 1, 2N)$. For the matched edge $(1, 2)$, note that $\gamma_1 + \gamma_{8N} = W$, whereas $\gamma_2 + \gamma_3 = W + \beta^{-(N-1)} = W + \epsilon'$. It follows that edge $(1, 2)$ satisfies ϵ' -correct division. Similar arguments take care of all the other matched edges in the other three ‘quarters’ of the ring (in fact, the argument can be completed using symmetry). Thus, we have an outcome satisfying ϵ' -correct division (cf. Definition 2.1). Since $\epsilon' = \beta^{-(N-1)}$ it follows that property 1 above is satisfied provided c is chosen appropriately.

For $n \geq 8$, but not a multiple of 8, we simply use the construction above for $n' = 8\lfloor n/8 \rfloor$ and add a dummy component of size $n - n'$, disconnected from $G_{n'}$. Further, we fix a UD solution on the dummy component (any bipartite graph has a UD solution). Since $n' \geq n/2$, it follows that property 1 is satisfied if c is chosen appropriately. ($c \equiv (1/4) \log_2 \beta$ works for all $n \geq 8$.) Clearly, property 2 is also satisfied.

Note that it was only for the sake of simplicity that the example we gave had multiple maximum weight matchings.

Remark C.1. *Though the example constructed above has multiple maximum weight matchings, this is not necessary. We can in fact, construct examples (that admit UD solutions) with the same properties 1 (for appropriate $c > 0$) and 2 above, and $W = O(1)$, where the weight of the maximum weight matching is at least 1 more than the weight of the next heaviest matching.*