Balanced outcomes in wage bargaining

Pingzhong Tang and **Dingli Yu**
Institute for Interdisciplinary Information Sciences,
Tsinghua University, Beijing, China
{kenshinping, leo.dingliyu}@gmail.com

Abstract

Balanced outcomes are a subset of core outcomes that take into consideration fairness and agents' power in bargaining networks. In this paper, following the seminal works by (Cook and Yamagishi 1992) and (Kleinberg and Tardos 2008) on modeling and computing balanced outcomes in unit-capacity trading networks, we explore this concept further by considering its generalization in the so-called wage bargaining network where agents on one side (the employers side) may have multiple capacity. It turns out that previous definitions do not trivially extend to this setting. Our first contribution is to incorporate insights from the bargaining theory and define a generalized notion of balanced outcomes in wage bargaining networks.

We then consider computational aspects of this newly proposed solutions. We show that there are polynomial-time combinatorial algorithms to compute such solutions in both unweighted and weighted graphs. Our algorithms and proofs are enabled by novel generalizations of techniques proposed by Kleinberg and Tardos and an original technique proposed in this paper called "loose chain".

1 Introduction

Wage has been one of the primary incentive instruments in social employment relationships. Employers concern their costs and employees pursue higher wages. In this paper, we aim toward a theory to formally investigate the following problem,

what are the stable and fair wage outcomes in a society?

from a perspective that incorporates both game and graph theories. We formally model and justify the concept of minimum wage that each employee should be paid given its ability and position in the social graph and develop efficient algorithms to compute it.

The social employment relationship considered in this paper can be formally modeled as a weighted bipartite graph, with employers on the one side and employees on the other side. If an employee is eligible to work for an employer, there is an edge between them. In addition, the corresponding weight of an edge denotes the value produced by the employee under this matching. In this paper, we assume the values produced by employees are additive, in the sense that the overall value produced by a set of employees is the sum of their individual values. We also assume each employer cannot hire more than a certain amount of employees, which is defined in the graph as its vertex capacity. On the other hand, each employee can only work for at most one employer. For example, in Figure 1, x_1, x_2, x_3, x_4 are four employers with capacity 2, 2, 1, 1 respectively, and $y_1, y_2, y_3, y_4, y_5, y_6$ are six employees with only one capacity each. In addition, the weights of all edges between X and Y are 1.

Given such a social graph instance, we model the wage bargaining interactions between employers and employees as a cooperative game. In the standard cooperative game theory, the most important solution concept is the *core*, which subscribes outcomes in which no subset of agents want to deviate from the current outcome. Even though core outcomes are stable (employees may not switch to other employers), they are still insufficient for the wage bargaining game because they are not necessarily fair in the sense that one side of the graph may get more than they deserve (employees may request for more wages). Over the years, there have been other alternative solution concepts for many different purposes. As we will review later, few of them are satisfactory to model the situation of wage bargaining.

In this paper, we generalize the notion of *balanced outcome*, originally proposed by Cook and Yamagishi (1992) and reinvestigated by Kleinberg and Tardos (2008) in the context of social exchange networks, to the context of wage bargaining. In the original model of social exchange networks, each agent has capacity of at most one and the social relation is modeled as a general graph; while in our model, we have a bipartite graph and each employer has multiple capacity.

Although the two models appear to be similar, there is a fundamental difference between them. The size of a

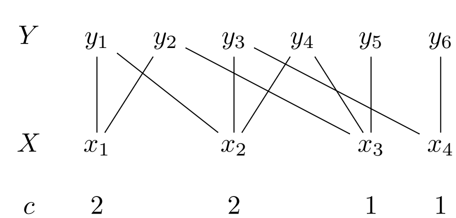

Figure 1: An example of unweighted wage bargaining model

coalition in previous models is always restricted to be either one or two, while the size of a coalition in our model (which contains one employee and several employers) can be any positive integers. For the previous case of bargaining between two agents, one can directly apply the standard Nash bargaining solution. In particular, two agents will agree on a division that is the middle point between the extremes of their alternate options. However, in the multi-agent case, alternate options of an agent may intersect with the others'. It is therefore not clear how to divide payoffs regarding to the extremes of alternate options for multiple agents. A careful definition of a multi-agent bargaining solution in this context is in need.

To address this difficulty, our first effort is to define a bargaining solution for general cooperative games. Our definition essentially differs from the existing concepts such as Shapley value (Shapley 1953), bargaining set (Aumann and Maschler 1964) and nucleolus (Schmeidler 1969) proposed in the literature.

When defining the new concept, we consider the following desiderata:

1. It must be a non-trivial subset of the core;
2. It must reduce to the definition of balanced outcomes in social exchange network;
3. It is efficiently computable in wage bargaining network.

It can be verified that none of the concepts above satisfy all the desiderata.

1.1 Our contribution

In this paper, we propose a new bargaining solution for multiple agents in Section 3. In particular, for a payoff vector γ of a cooperative game (N, v), a multi-agent coalition S is balanced if S can be partitioned into sub-coalitions $C_1, \ldots, C_k$ such that for all C_i,

$$\sigma(C_i, S \setminus C_i) = \min_{U \subsetneq S} \sigma(U, S \setminus U),$$

where

$$\sigma(S, T) = \min\{\gamma(U) - v(U) : U \subseteq N \setminus T, S \subseteq U\}.$$

Intuitively, each sub-coalition is given an identical "incentive" value v_e (which can be any positive number) if they break the current coalition. Think of it as a cooperative game on S, where for each sub-coalition $T \subsetneq S$, the payoff of T is the "incentive" value minus the cost for T deviating from S, that is, $v_e - \sigma(T, S \setminus T)$. If it turns out the core payoff vector is all 0 under a certain positive "incentive" value, we say the coalition S is balanced. This definition satisfies the first two desiderata: it is based on the core, and consistent with balanced outcomes in social exchange networks. Built on this new solution, we naturally extend the concept of balanced outcome in wage bargaining.

Our goal is then to verify the last desideratum, and further, to solve the structure of all balanced outcomes and efficiently find the optimal balanced outcome for either the employees or the employers. In Section 4, starting from the special wage bargaining problem characterized by unweighted graphs with unique perfect matching (Theorem 1), we then analyze the weighted graphs (Theorem 2). In both cases, we put forward efficient algorithms (Algorithm 1, 2) to find the employer (employee) optimal balanced outcomes. Our algorithms are enabled by novel generalizations of the techniques proposed by Kleinberg and Tardos (2008), which may be of independent interests.

1.2 Additional related works

The literature that investigates the relative power of bilateral relationship between pairs of agents dates back to the network exchange theory initiated by (Emerson 1962). The basic idea is to view this bilateral relation as a way to produce a joint value for both agents. The two agents then divide up this value by a bargaining procedure known as *social exchange*, which takes into consideration the relative *power* of each agent in the network. Social scientists have designed various experiments to validate the theory (Cook and Emerson 1978).

Another related literature is assignment games, initiated by Shapley and Shubik (1971) on two-sided markets with unit-capacity. They derive their landmark result that any maximum matching of the graph corresponds to a core assignment in the induced game.

Kleinberg and Tardos (2008) proved that balanced outcomes exist if and only if stable outcomes exist, and provided an algorithm that efficiently computes the set of balanced outcomes. Whether this result can extend to models where agents have multiple capacity is the main theme of this paper. The model and idea above are further extended to three-sided markets, with an additional layer called traders between buyers and sellers (Blume et al. 2007) and the core of the resulting cooperative games is characterized.

There are also various papers concerning the balancedness and complexity of solutions in cooperative games which are close to our modeling, such as (Chalkiadakis et al. 2010; Greco et al. 2009; Dang et al. 2006).

2 Preliminaries

A cooperative game models competitions and cooperations between groups of players ("coalitions").

Definition 1. *A cooperative game is defined as a pair (N, v), where*

- *N is the set of agents;*
- *$v : 2^N \to \mathbb{R}$ is the coalition function characterizing the maximum payoff that each subset of players can gain without other players.*

Here we assume the utility can be transferred between individuals, that is, it is a transferable utility game. Also, we assume the coalition function is superadditive:

$$v(S) + v(T) \leq v(S \cup T)$$

for any pair of disjoint subsets S and T. Therefore, $v(N)$ can be seen as the social welfare of the whole society.

Definition 2. *In a cooperative game (N, v), an outcome is defined as a pair (B, γ), where B is a set of coalitions (a* *partition of N), $\gamma : N \to \mathbb{R}$ is the payoff vector (function). A feasible outcome (B, γ) satisfies that for all $S \in B$,*

$$\gamma(S) = v(S),$$

where $\gamma(S) \triangleq \sum_{x \in S} \gamma_x$.

By definition, every feasible outcome (B, γ) satisfies

$$\gamma(N) = \sum_{S \in B} \gamma(S) = \sum_{S \in B} v(S) \leq v(N),$$

which means the total payoff of players cannot exceed the maximum social welfare. Throughout the paper, whenever an outcome is mentioned, it is feasible. Moreover, we assume $\forall S \in B$, S is the minimal set satisfying $\gamma(S) = v(S)$, that is, S cannot be divided into smaller coalitions.

Definition 3. *In a cooperative game (N, v), an outcome (B, γ) is stable if and only if:*

- *$\forall x \in N, \gamma_x \geq v(\{x\})$;*
- *$\forall x, y \in N, \gamma_x + \gamma_y \geq v(\{x, y\})$.*

Definition 4. *In a cooperative game (N, v), an outcome (B, γ) is in core if and only if:*

$$\forall S \subseteq N, \gamma(S) \geq v(S).$$

(B, γ) is also called a core outcome of (N, v).

As in a core outcome, $\gamma(N) \geq v(N)$, and also $\gamma(N) \leq v(N)$, so $\gamma(N) = v(N)$. This means a core outcome must reach the maximum social welfare.

Definition 5. *A social exchange network is a cooperative game (N, v), characterized by a graph $G = (V, E, \omega)$, where*

- *$\omega : E \to \mathbb{R}^+$ is the edge weight function;*
- *$N = V$;*
- *$\forall S \subseteq N$, $v(S)$ is the value of the maximum weighted matching of the induced graph $G[S]$[1].*

In a social exchange network (N, v) characterized by a graph $G = (V, E, \omega)$, consider any subset of players S. Let M be one of the maximum weighted matchings of $G[S]$. Then a stable outcome (B, γ) satisfies

$$\gamma(S) \geq \sum_{\{x,y\} \in M} \gamma_x + \gamma_y \geq \sum_{\{x,y\} \in M} v(\{x, y\}) = v(S).$$

So in a social exchange network, a stable outcome is also a core outcome.

Now suppose the social exchange network has two different maximum matchings M and M', and (M, γ) is a stable (or core) outcome, then

$$v(N) = \gamma(N) \geq \sum_{S \in M'} \gamma(S) \geq \sum_{S \in M'} v(S) = v(N).$$

So $\gamma(S) = v(S)$ for all $S \in M'$, which means (M', γ) is also a stable outcome. As a result, if one wants to find a stable outcome, it is sufficient to find an arbitrary maximum matching, and then solve for the payoff vectors. As all restrictions of γ are linear, therefore linear programming can be used to find a stable outcome. Despite of linear programming, combinatorial algorithms are also found to solve the payoff vectors (Aspvall and Shiloach 1979). Furthermore, combinatorial algorithms reveal clearer intuitions and structures than those found by linear programming.

Definition 6. *In a social exchange network (N, v) with an outcome (B, γ), define the best alternate option of a player x as*

$$\alpha_x \triangleq \max_{y, \{x,y\} \notin B} v(\{x, y\}) - \gamma_y.$$

Definition 7. *In a social exchange network (N, v), a core outcome (B, γ) is balanced if and only if for all $\{x, y\} \in B$,*

$$\gamma_x - \alpha_x = \gamma_y - \alpha_y.$$

It has been proved in (Kleinberg and Tardos 2008) that balanced outcomes in a social exchange network exist if and only if stable outcomes exist.

Definition 8. *An assignment game is a social exchange network characterized by a bipartite graph.*

Since assignment games always have stable outcomes (Shapley and Shubik 1971), balanced outcomes always exist as well.

3 A multi-agent bargaining solution

We now define a bargaining problem for multi-agent coalitions and balanced outcomes in cooperative games.

Definition 9. *In a cooperative game (N, v), define the minimum slack of set S over set T with respect to payoff vector γ to be*

$$\sigma(S, T) = \min\{\gamma(U) - v(U) : U \subseteq N \setminus T, S \subseteq U\}.$$

In a multi-agent coalition C, for any sub-coalition S, assume every agent agrees that the cost of S deviating from C is $\sigma(S, C \setminus S)$. Then it can be modeled as a cooperative game on C, with coalition function (v_e will be explained later)

$$v'(S) = \begin{cases} v_e - \sigma(S, C \setminus S), & \text{if } S \neq C; \\ 0, & \text{if } S = C. \end{cases}$$

However, if $v_e = 0$, there is only one core outcome of this cooperative game, $(\{C\}, \vec{0})$, because $\forall S \neq C, v'(S) < 0$.

In order to make the cooperative game non-trivial, each coalition S is given an "incentive" value v_e if S breaks the current coalition C. Define v_e to be a *good incentive value* if the payoff of each agent remains unchanged regardless of whether they break the coalition C or not. Namely, the cooperative game has at least the following two core outcomes:

$$(\{C\}, \vec{0}), (B', \vec{0}),$$

where B' is a partition of C.

Now define the coalition to be balanced if there exists a good incentive value, and define an outcome is balanced if all coalitions of the outcome are balanced. The following simplified definition is an equivalent to what we have defined so far.

[1] The induced subgraph $G[S]$ is a subgraph formed from S and any edges whose endpoints are both in S.

Definition 10. *In a cooperative game (N, v), an outcome (B, γ) is balanced if and only if*

- *(B, γ) is in the core.*
- *$\forall S \in B$, there exists a non-trivial partition of S, $C = \{C_1, \dots, C_k\}$ such that*

$$\forall T \in C, \sigma(T, S \setminus T) = \min_{U \subsetneq S} \sigma(U, S \setminus U).$$

Then one can check the desiderata given in Section 1: It is defined on the core; If this definition is applied on social exchange networks, it is consistent with the original definition. Also, we will show this definition is computable in wage bargaining problem in the next section.

We note without proof that, another interesting property, from the perspective of non-transferable utility game, is that the balance of a coalition C implies the existence of the core in the hedonic game (Dreze and Greenberg 1980) $(C, (\succeq_i)_{i \in C})$, where $\forall S, T \subsetneq C, \forall i \in S \cap T$,

$$S \succeq_i T \Leftrightarrow \sigma(S, C \setminus S) \leq \sigma(T, C \setminus T),$$

and

$$\forall S \subsetneq C, \forall i \in S, S \succeq_i C.$$

4 Balanced outcomes in wage bargaining

First, we define the wage bargaining as a cooperative game.

Definition 11. *A wage bargaining problem is a cooperative game (N, v), characterized by a bipartite graph $G = (V, E, \omega, c)$, where*

- *$N = V = X \cup Y$ (X is the set of employers and Y is the set of employees);*
- *every edge in E has a endpoint in X and another in Y;*
- *$\omega : E \to \mathbb{R}^+$ is the edge weight (denoting the value produced by the employer and employee under the matching);*
- *$c : X \to \mathbb{N}$ is the maximum capacity of nodes in X (an employer x can hire at most c_x employees);*
- *$v(S)$ is the maximum weighted matching of the induced graph $G[S]$ under the capacity restriction c.*

In this section, we will simplify the concept of balanced outcome in wage bargaining problem and introduce our algorithms which efficiently computes balanced outcomes in wage bargaining. Using organization similar to Kleinberg and Tardos' (Kleinberg and Tardos 2008), we will first consider the easy case where the graph is unweighted and has a unique perfect matching, which requires simpler notations and offers better insights. We will then extend to the general case with weights. Compared to Kleinberg and Tardos' setting, our setting is more complicated. In particular, for each employer x, the employee y with the minimum $\omega_{xy} - \gamma_y$ need to be identified. To handle the difficulty, a kind of new structure called "loose chain" is invented by us to cooperate with the original structures: chains("tight chains" in our notation) and free cycles (free cycles is not required in the unique matching case).

4.1 Simplifications

Before we dig into the balanced outcomes, we prove the existence of core outcomes. In a wage bargaining problem (N, v), if each $x \in X$ is replaced by c_x nodes of capacity 1 and edges of x are also copied c_x times, it becomes an assignment game (N', v'). Let f be the function mapping nodes in the new game back to nodes in the original game. From a core outcome (B', γ') of the new game, one can naturally obtain an outcome of the original game (B, γ) such that $\gamma(S) = \gamma'(f^{-1}(S))$, then

$$\forall S \in N, \gamma(S) = \gamma'(f^{-1}(S)) \geq v'(f^{-1}(S)) = v(S),$$

which indicates (B, γ) is in the core. Since assignment games always have core outcomes, wage bargaining problems also have core outcomes by the analysis above.

We are now ready to apply balanced outcomes in the wage bargaining problem. Let $G = (V, E, \omega, c)$ be a wage bargaining problem, with two color sets X and Y. Suppose G has a balanced outcome (M, γ). Since balanced outcomes are always core outcomes, M must be a maximum matching of G in order to maximize social welfare. For example,

$$M = \{\{x_1, y_1, y_2\}, \{x_2, y_3, y_4\}, \{x_3, y_5\}, \{x_4, y_6\}\}$$

is the only possible coalition set in the wage bargaining problem described in Figure 1. Note here M is a set of coalitions, not a set of edges. We slightly abuse the notation and use $(x, y) \in M$ to denote that (x, y) is a matching edge in M.

As the capacity for each employee is restricted to 1, every coalition in the wage bargaining problem contains one employer and multiple employees.

Now, consider a coalition $\{x, y_1, \dots, y_k\}$ that contains one employer and k employees.

Definition 12. *Let the slack of the best alternate option for x[2] be*

$$\sigma_x = \min_{(x,y) \in M} (\omega_{xy} - \gamma_y) - \max\{0, \max_{(x,y') \in E \setminus M} \omega_{xy'} - \gamma_{y'}\},$$

and the slack of the best alternate option for y_i without x be

$$\sigma_{y_i} = \min\{\gamma_{y_i}, \min_{(x',y_i) \in E \setminus M} \min_{(x',y') \in M} \omega_{x'y'} - \gamma_{y'} + \gamma_{y_i} - \omega_{x'y_i}\}.$$

Then the coalition $\{x, y_1, \dots, y_k\}$ is balanced if and only if

- *For all $i \in [k]$[3], $\sigma_{y_i} \geq \sigma_x$;*
- *There exists $i \in [k]$, such that $\sigma_{y_i} = \sigma_x$ and*

$$\omega_{xy_i} - \gamma_{y_i} = \min_{(x,y) \in M} \omega_{xy} - \gamma_y.$$

For example, if we set

$$\gamma_{y_1} = 0.75, \gamma_{y_2} \in [0.5, 0.75],$$

[2] An alternate option for x is a set of agents including x, but not including all the other agents that are in the current coalition of x. The best alternate option for x is the alternate option S that minimize the slack $\gamma(S) - v(S)$ among all alternate options for x. The slack of the best alternate option for each agent should be equal to the incentive value, if the coalition is balanced.

[3] $[k] = \{1, \dots, k\}$.

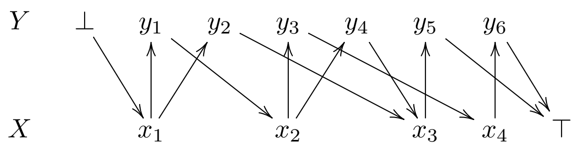

Figure 2: Visualize G as a directed graph

$$\gamma_{y_3} = \gamma_{y_4} = 0.5, \gamma_{y_5} = \gamma_{y_6} = 0.25$$

in Figure 1, then from the definition, $\sigma_v = 0.25$ for all $v \in V$ except

$$\sigma_{y_2} = \gamma_{y_2} - 0.25,$$

which satisfies the balanced condition above for every coalition. Therefore, this outcome is balanced and it is in fact the only balanced outcome. One can verify if Definition 10 is applied in wage bargaining problems, it is consistent with this definition.

In order to further simplify the problem, the graph is modified accordingly and some concepts are defined as follows.

One can visualize G as a directed graph with edges in M oriented from X to Y while all other edges oriented from Y to X. Also, two additional nodes are added to the graph: a node $\perp$ linking to all nodes in X with weight 1, and an node $\top$ linked from all nodes in Y with weight 1. These two nodes are defined to ensure there is always a non-matching edge for each node, so $\gamma_\top = \gamma_\perp = 1$. Write

$$V^+ = V \cup \{\perp, \top\}.$$

To illustrate, we now convert Figure 1 to Figure 2. Some edges from $\perp$ and some edges to $\top$ are omitted.

Now for a payoff vector γ, define its correspond labeling η on V^+, such that $\eta_\perp = 1, \eta_\top = 1$, and for each $y \in Y$, $\eta_y = \gamma_y$, for each $x \in X$,

$$\eta_x = \min_{x \to y} \omega_{xy} - \gamma_y.$$

For any edge (y, x) from Y to X define the edge slack

$$\sigma_{yx} = \eta_y + \eta_x - \omega_{xy},$$

then we can simplify σ_x and σ_y as

$$\sigma_x = \min_{y \to x} \sigma_{yx}, \sigma_y = \min_{x \leftarrow y} \sigma_{yx}.$$

It can be seen that an outcome (M, γ) is in core if and only if γ's corresponding labeling η satisfies

$$\forall v \in V, \sigma_v \geq 0.$$

Definition 13. *Define an outcome (M, γ) is balanced if and only if γ's corresponding labeling η is balanced. In particular, η satisfies*

- $\forall v \in V, \sigma_v \geq 0.$
- $\sigma_y \geq \sigma_x$ *if* $(x, y) \in M$;
- $\forall x, \exists (x, y) \in M$, *such that* $\eta_x = \omega_{xy} - \eta_y$ *and* $\sigma_y = \sigma_x$.

This definition is equivalent to Definition 12, and is easier to analyze. Therefore, in what follows, we will compute balanced labelings instead.

4.2 Wage bargaining in unweighted graphs with a unique perfect matching

As the first step toward a general characterization of balanced outcomes (in fact, balanced labelings), we consider the unweighted graphs instances where there is a unique perfect matching. Let $G = (V, E, \omega, c)$ characterize a wage bargaining problem, where $\omega \equiv 1$ and G has a unique perfect matching M.

Now we can write $\zeta_x = \eta_x, \zeta_y = 1 - \eta_y$ for convenience. Then we have

- $\zeta_\perp = 0, \zeta_\top = 1$;
- $\zeta_x = \min_{x \to y} \zeta_y$;
- $\sigma_{yx} = \zeta_y - \zeta_x$;
- The corresponding payoff vector of η is in core if for each edge from y to x, $\zeta_y \leq \zeta_x$. In other words, $\zeta_{v_1} \leq \zeta_{v_2}$ if v_1 can reach v_2 in the directed graph.

Since M is the only perfect matching in G, the directed graph induced by M must be acyclic. Figure 2 is an example.

Theorem 1. *There exists a balanced labeling for any wage bargaining problem in unweighted graphs with a unique perfect matching. All balanced labelings can also be efficiently generated. Moreover, the optimal labeling for employers or employees can be computed in polynomial time.*

Proof sketch. To prove the theorem, it suffices to give an algorithm that constructs the labeling η, and then verify its properties. The basic idea of the algorithm is to label nodes with small σ first so that the constraints of balanced outcomes are met during the algorithm. In each stage of the algorithm, a set of nodes (called an anchored chain, which will be introduced later) with the same σ labeled. An anchored chain will then be classified into either a loose chain or a tight chain in order to handle the cases when $\sigma_x \leq \sigma_y$ or when $\sigma_x = \sigma_y$ for $(x, y) \in M$. Then by using Lemma 1, we prove the returned labeling by Algorithm 1 is always balanced. On the other hand, in Lemma 2, we further show that all balanced labelings can be generated. Moreover, by adding rules in Algorithm 1, we prove Lemma 3 — the balanced labelings that maximize X or Y's payoff can be generated respectively.

Proof. Consider the following algorithm that generates a labeling:

Algorithm 1. *The algorithm proceeds in stages. At the start of each stage s, let B_s be the subset that has already been labeled. Say a chain*

$$C = b_1 \to b_2 \to \cdots \to b_r$$

is anchored if $r > 2$, and the endpoints b_1 and b_r are in B_s. At the end of the stage, choose an anchored chain C, and label $b_2, \ldots, b_{r-1}$ such that

- *For any edge (x, y) in C, set $\zeta_x \leq \zeta_y$; and if $x \neq b_1$, set $\zeta_x = \zeta_y$.*
- *For any edge (y, x) in C, set $\sigma_y = \sigma_{yx}$; and if $x \neq b_r$, set $\sigma_x = \sigma_{yx}$.*

- *set* $\sigma_{b_2} = \cdots = \sigma_{b_{r-1}}$.

Define $\sigma(C)$ *to be the range of* σ_{b_2} *if* C *is labeled in this stage. Let*

$$\overline{\sigma}(C) = \sup \sigma(C),$$

σ_ℓ *be the maximum* σ *in* B_s*, and* $\sigma_r = \min_C \overline{\sigma}(C)$. Choose *a pair* (σ, C) *such that*

- $\sigma \in [\sigma_\ell, \sigma_r] \cap \sigma(C)$*;*
- *If* $b_2 \in Y$*, for any anchored chain*

$$b_1 \to b_2 \to c_3 \to \cdots \to c_m,$$

$\zeta_{c_m} - \zeta_{b_r} \geq \frac{m-r}{2}\sigma$.

Finally, label C *with* $\sigma_{b_2} = \sigma$.

For Figure 2, the labeling by Algorithm 1 would be: in the first stage, set $\zeta_{x_1} = \zeta_{y_1} = 0.25, \zeta_{x_2} = \zeta_{y_3} = 0.5, \zeta_{x_4} = \zeta_{y_6} = 0.75$; in the second stage, set $\zeta_{y_4} = 0.5, \zeta_{x_3} = \zeta_{y_5} = 0.75$; in the last stage, set $\zeta_{y_2} \in [0.25, 0.5]$.

The first observation about the algorithm is that if a node y is labeled, its matching point x must be labeled as well. So for each anchored chain

$$C = b_1 \to b_2 \to \cdots \to b_r,$$

b_r is always in X. For convenience, say C is a loose chain if $b_1 \in X$, otherwise C is a tight chain. Then, it is not hard to show that, if C is a loose chain,

$$\sigma(C) = \left[\sigma_{b_1}, \frac{\zeta_{b_r} - \zeta_{b_1}}{(r-1)/2}\right];$$

otherwise,

$$\sigma(C) = \left\{\frac{\zeta_{b_r} - \zeta_{b_1}}{r/2}\right\}.$$

Note it is yet unclear whether there is always a pair (σ, C) that satisfies the algorithm's conditions. We now formally prove the following lemma.

Lemma 1. *In stage* s*, suppose the algorithm chooses a pair* (σ, C) *to label. Then in stage* $s+1$*, any anchored chain* C' *satisfies* $\overline{\sigma}(C') \geq \sigma$*.*

Proof. Let $C = b_1 \to b_2 \to \cdots \to b_r$. If $b_1 \in Y$, the lemma holds trivially. Suppose by contradiction, there exists an anchored chain C' such that $\overline{\sigma}(C') < \sigma$ in stage $s+1$, one of endpoints of C' must be in C; otherwise, C' is also available in stage s, which contradicts with $\sigma_r \geq \sigma$. Therefore, we can connect part of C with C' and obtain another chain C'' such that $\overline{\sigma}(C'') < \sigma$ in stage s, which also contradicts with $\sigma_r \geq \sigma$

If $b_1 \in X$, three cases need to be considered: both endpoints of C' are in C; only the first node of C' is in C; only the last point of C' is in C. Notice the last case is the same as above. For the first case, suppose C' is an anchored chain of length m from b_i to b_j. Consider the chain

$$C'' = b_1 \to \cdots \to b_{i-1} \to C' \to b_{j+1} \to \cdots \to b_r,$$

its length $m + r - j + i - 1 \leq r$ according to the second condition of choosing the pair (σ, C), so $j - i + 1 \geq m$. Then

$$\overline{\sigma}(C') = (\zeta_{b_j} - \zeta_{b_i})/\lfloor m/2 \rfloor \geq \sigma.$$

For the second case, suppose the first node of C' is b_i. As

$$\zeta_{b_r} = \zeta_{b_i} + \lfloor (r-i+1)/2 \rfloor \sigma,$$

substitute ζ_{b_r} in the second condition of the algorithm, $\overline{\sigma}(C') \geq \sigma$ is then obvious. □

This lemma shows that condition $\sigma_\ell \leq \sigma_r$ holds throughout the proceeding. Moreover, if the algorithm chooses $\overline{\sigma}(C) = \sigma_r$, the second condition for choosing pair (σ, C) always holds, so the algorithm can always choose a valid pair (σ, C). As σ is always increasing, which means if nodes are balanced after labeling, they will continue to be balanced. So the algorithm will always output a balanced labeling. In fact, if the algorithm always chooses $\overline{\sigma}(C) = \sigma_r$, it is exactly the algorithm that proves Theorem 2.2 in the Kleinberg and Tardos' paper (Kleinberg and Tardos 2008), and also the algorithm that maximizes Y's payoff.

Lemma 2. *Any balanced labeling* η^* *can be generated by the algorithm.*

Proof. Let σ^* be the corresponding slack of η^*, following the steps of the algorithm, it suffices to prove the algorithm can always label a node with the smallest σ^* among all unlabeled nodes. Consider a node v with the smallest σ^*, if $v \in X$, there exists a node $y \in Y$ such that $\sigma^*_{yv} = \sigma^*_v$. Then if y is not labeled, $\sigma^*_y = \sigma^*_{yv} = \sigma^*_v$. Moreover, if y's matching node x is also not labeled, $\sigma^*_x = \sigma^*_y$, and so on. Trace upwards until there is a labeled node. Similarly, trace downwards until there is a labeled node. Finally, an anchored chain with σ^*_v is found, and it obviously satisfies the conditions in the algorithm.

Now it remains to prove $\sigma^*_v \leq \overline{\sigma}(C)$ for any anchored chain

$$C = b_1 \to b_2 \to \ldots \to b_r.$$

Notice that for any edge (x, y) in C, $\zeta^*_x \leq \zeta^*_y$, and for any edge (y, x) in C, $\zeta^*_y + \sigma^*_v \leq \zeta^*_x$. So

$$\zeta_{b_1} + \lfloor r/2 \rfloor \sigma^*_v \leq \zeta_{b_r},$$

which indicates $\overline{\sigma}(C) \geq \sigma^*_v$. □

Lemma 3. *There are algorithms generating the balanced labelings that maximize* X *or* Y*'s payoff.*

Proof. In order to maximize Y's payoff, only a slight adjustment is needed in Algorithm 1: at the end of each stage, choose a pair (σ, C) such that the original conditions are satisfied, and $\overline{\sigma}(C) = \sigma_r$. The correctness will be proved by induction on stages. First assume the following: if the first s stages of the algorithm is fixed, only considering stages after stage s, the labeling on $V \setminus B_s$ generated by the revised algorithm is the best among all labelings generated by the original algorithm. Then suppose in stage s, the original algorithm chooses a pair (σ, C) that $\overline{\sigma}(C) > \sigma_r$. It suffices to prove σ and C can be transformed without decreasing Y's payoff until $\overline{\sigma}(C) = \sigma_r$ is satisfied.

First, try to add a very small $\Delta\sigma$ to σ so that $(\sigma + \Delta\sigma, C)$ still satisfies the original conditions, and then run the revised algorithm. As a result, all ζ_v for $v \notin B_s$ decrease a non-negative times of $\Delta\sigma$. If it is impossible to add a small

$\Delta\sigma$, which indicates the inequality in the second conditions of (σ, C) is tight, another chain C' can be easily found to swap with C. Again, add small $\Delta\sigma$ to σ and swap chains if necessary, until $\sigma = \sigma_r$. Now one can assume $\sigma = \sigma_r$, and observe that if $\sigma_r \neq \overline{\sigma}(C)$, the next stage will still have the same σ_r until $\sigma_r = \overline{\sigma}(C)$. Then by swapping the order of these stages, $\sigma_r = \overline{\sigma}(C)$ can be achieved in stage s.

Maximizing X's payoff is very similar to maximizing Y's payoff: one only needs to choose the pair (σ, C) that minimizes σ at the end of each stage. We omit the proof here because it is the same as the proof of maximizing Y's. □

The algorithm is clearly polynomial-time. In fact, for any $b_1, b_r \in B_s$, only the longest chain from b_1 to b_r needs to be considered. □

4.3 Wage bargaining in weighted graphs

In this section, built on previous model and analysis on the unweighted graphs, we discuss the general case where the graphs are weighted. In the previous subsection, there is a nice property that the graph is acyclic, so we do not need to consider cycles. Moreover, it is yet unknown whether the previous structure generalizes in weighted graphs.

Theorem 2. *There exists a balanced labeling for any wage bargaining problem. The set of balanced labelings can also be efficiently generated. Moreover, the optimal labeling for employers or employees can be computed in polynomial time.*

In order to tackle the difficulties and prove Theorem 2, we propose an new algorithm and again analyze its properties. The idea of algorithm is similar to Algorithm 1, so is the structure of the proof. Particularly, the algorithm will extend Algorithm 1 by adding the free cycle structure and specifying weights. Free cycles will be used to handle the case that unlabeled nodes with the smallest σ form cycles, which does not happen in the last subsection.

Proof. We propose a new algorithm as follows:

Algorithm 2. *Label 0 on the nodes in Y that are not part of any maximum matching.*

The algorithm proceeds in stages.

At the start of each stage s, let B_s be the subset that has already been labeled. Define a simple cycle $C = b_1 \to \cdots \to b_r \to b_1$ to be free cycle, if $b_i \notin B_s$. Without loss of generality, assume $b_1 \in X$ in any free cycle. At the end of the stage, either choose an anchored chain C and label C by the same way in Algorithm 1, or choose a free cycle C, and label $b_1, \ldots, b_r$ such that

- *For any edge (x, y) in C, set $\eta_x = \omega_{xy} - \eta_y$;*
- *For any edge (y, x) in C, set $\sigma_y = \sigma_{yx} = \sigma_x$;*
- *set $\sigma_{b_1} = \cdots = \sigma_{b_{r-1}}$.*

Define $\sigma(C)$ to be the range of σ_{b_2} if C is labeled in this stage. Let $\overline{\sigma}(C) = \sup \sigma(C)$. Let σ_ℓ be the maximum σ in B_s, and $\sigma_r = \min_C \overline{\sigma}(C)$. Choose *one of three structures C satisfying the following conditions:*

- *If C is a tight chain, there is only one way to label C such that $\overline{\sigma}(C) = \sigma_r$;*
- *If C is a loose chain, there is one degree of freedom to choose $\sigma \in [\sigma_\ell, \sigma_r]$ and determine η_{b_2}, such that for any anchored chain*

$$C' = b_2 \to c_3 \to \cdots \to c_m,$$

  $\overline{\sigma}(C') \geq \sigma$ (assume b_2 is labeled by η_{b_2}).
- *If C is a free cycle, $\sigma(C)$ contains only one value, then $\overline{\sigma}(C) = \sigma_r$ is required, but there is still one degree of freedom to choose η_{b_1}, such that for any anchored chain $C' = b_1 \to c_2 \to \cdots \to c_m$ or $C' = c_2 \to \cdots \to c_m \to b_1$, $\overline{\sigma}(C') \geq \sigma_r$ (assume b_1 is labeled by η_{b_1}).*

Finally, label C according to C's type.

First, it is still true that if a node y is labeled, its matching point x must be labeled as well. Then for each anchored chain $C = b_1 \to b_2 \to \cdots \to b_r$, if C is a loose chain,

$$\sigma(C) = \left[\sigma_{b_1}, \frac{\eta_{b_r} - \eta_{b_1} + \sum_{i=2}^{r}(-1)^i \omega_{b_{i-1}b_i}}{(r-1)/2}\right],$$

otherwise,

$$\sigma(C) = \left\{\frac{\eta_{b_1} + \eta_{b_r} - \sum_{i=2}^{r}(-1)^i \omega_{b_{i-1}b_i}}{r/2}\right\}.$$

And for any free cycle $C = b_1 \to \cdots \to b_r \to b_{r+1} = b_1$,

$$\sigma(C) = \left\{\frac{\sum_{i=2}^{r+1}(-1)^i \omega_{b_{i-1}b_i}}{r/2}\right\}.$$

Lemma 4. *In stage s, suppose Algorithm 2 chooses a structure C, and labels it with slack σ. Then in stage $s+1$, any anchored chain C' satisfies $\overline{\sigma}(C') \geq \sigma$.*

Proof. If C' is a free cycle, $\sigma(C')$ remains unchanged after stage s, so $\overline{\sigma}(C') \geq \sigma$. If C' and C are both anchored chains, we just use the same method in Lemma 1 to verify.

If C is a free cycle and C' is the anchored chain, by contradiction, we suppose $\overline{\sigma}(C') < \sigma$. If both C''s first node and last node are in C, we can find another free cycle C'' such that $\overline{\sigma}(C'') < \sigma$, so C cannot be chosen in stage s. Otherwise, we can again connect part of C with C' and get an anchored chain C'' such that $\overline{(C'')} < \sigma$ in stage s, and either the first node or the last node of C'' is b_1, which violates the condition for free cycles in the algorithm. □

As Lemma 4 holds, the argument that the algorithm can always generate one balanced labeling is also the same as the previous case.

To prove Lemma 2 under this algorithm, we only need to consider the case when a node is visited twice during the tracing, otherwise, it is the same as before. But if a node is visited twice, we naturally obtain a free cycle which obviously satisfies the condition in the algorithm. Additionally, we can also prove $\sigma_v^* \leq \overline{\sigma}(C)$ for any free cycle C using the similar argument for anchored chains.

To prove Lemma 3, we add another principle to generate the optimal labeling for X or Y. Again, choose C such that

σ is maximized/minimized in each stage. If C happens to be a free cycle, choose the smallest η_{b_1} to generate the optimal labeling for Y. The proof is still similar to the original one. On the other hand, in order to generate the optimal labeling for X, choose the largest η_{b_1} instead.

Algorithm 2 is more complex than Algorithm 1, so we need to be careful about the complexity. Although it is now hard to find the best chain from b_1 to b_r, or the free cycle with the smallest $\overline{\sigma}$, one can easily check whether there exists a structure C with $\overline{\sigma}(C) \leq \sigma$. In particular, it only involves linear inequalities, which can be solved by linear programming. So binary search can be used to obtain σ_r, and find the structure C with $\overline{\sigma}(C) = \sigma_r$. For loose chains, in order to satisfy the condition, all loose chains starting from the same b_1, b_2 split the range $[\sigma_\ell, \sigma_r]$. Each loose chain holds a subinterval, and it is easy to find the subinterval containing a certain σ. As for the range of η_{b_1} of a free cycle, it is also an interval. Anchored chains $C' = b_1 \to c_2 \to \cdots \to c_m$ restrict its upper bound, while anchored chains $C' = c_2 \to \cdots \to c_m \to b_1$ restrict its lower bound. □

5 Discussions

5.1 Efficiency via local negotiation.

In reality, bargaining usually happens locally. Therefore, one of important metrics of a solution concept in this setting is whether the concept can be reached by a small number of local negotiations. Researchers show that local bargaining in social exchange networks converges to balanced outcomes efficiently, in time roughly independent of the network size (Kanoria et al. 2011). In order to further validate our concept, in this section, we run simulations on randomly sampled wage bargaining instances with random sampled core outcomes ($|X|$ is uniformly selected in $[2, 100]$, $|Y|$ is uniformly selected in $[\alpha|X| + 1, (\alpha + 1)|X|]$). In particular, we calculate the number of wage (payoff) changes from the core outcomes to a certain approximation of balanced outcomes via local negotiation. For example, Figure 3(a) is a diagram of 100 instances with $\alpha = 4$. From the diagram, one can observe the average number of wage changes for employees is less than 10 for most of the cases, that is, the average time that employees spend on wage negotiation is independent of the size of the network. Similarly, when $\alpha =$ 8(see Figure 3(b)), the average number of wage changes is less than 7, and also independent of $|Y|$. This is even smaller than case when $\alpha = 4$.

Moreover, we run the same simulation of the same instances for another existing concept — kernel (Davis and Maschler 1965). The results are shown by Figure 3(c) and 3(d). Although kernel satisfies all our desiderata (Bateni et al. 2010) mentioned in Section 1, its local negotiation efficiency is much lower than balanced outcome. Furthermore, when α changes from 4 to 8, the average number of wage changes increases in the case of kernel.

5.2 Extensions

Since we have developed a multi-agent bargaining solution and defined balanced outcomes in general cooperative games, it is possible that the concept of balanced outcomes is also applicable in other models, such as three-sided markets. As we mentioned in additional related works, the core of three-sided markets has been characterized (Blume et al. 2007), which provides a solid foundation for analysis of balanced outcomes. However, as the structure of coalitions becomes complicated, the balanced outcome also becomes hard to analyze. For example, if employees also have multiple capacity in wage bargaining (which is very rare in reality), examining balanced outcomes is already non-trivial, even in the case where everyone's capacity is 2. In that special case, coalitions can be cycles or chains of any length.

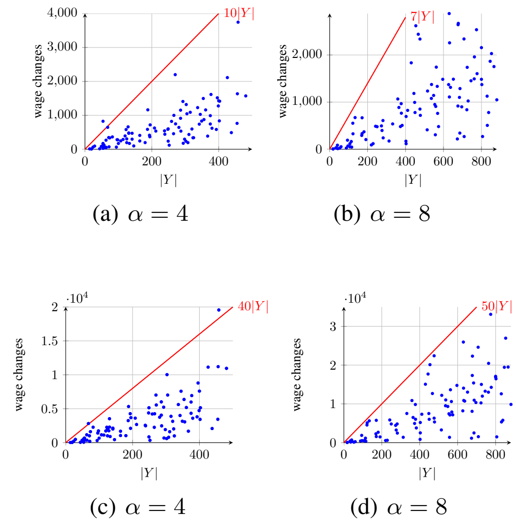

Figure 3: Each point in diagrams is a randomly generated instance, with the number of employees as the horizontal axis and the number of changes in wages during local negotiation as the vertical axis. (a) and (b) are results for balanced outcomes, while (c) and (d) are results for kernels.

6 Conclusion

We have proposed a novel bargaining solution for multiple agents and applied it to the wage bargaining setting. Furthermore, we proposed efficient algorithms to compute balanced outcomes in wage bargaining via an original technique called "loose chain".

Our bargaining solution, based on the idea of "incentive" value and essentially different to existing concepts, generalizes the concept of balance outcome in social exchange network.

Our algorithm, taking employees'(employers') benefits into consideration, can have a potentially impact in real life. Although our focus in this paper is mostly on the theoretical aspect, it is an interesting further direction to test its effectiveness in real life settings.

References

Aspvall, B., and Shiloach, Y. 1979. A polynomial time algorithm for solving systems of linear inequalities with two variables per inequality. In *Foundations of Computer Science, 1979., 20th Annual Symposium on*, 205–217. IEEE.

Aumann, R. J., and Maschler, M. 1964. The bargaining set for cooperative games. *Advances in game theory* 52:443–476.

Bateni, M.; Hajiaghayi, M.; Immorlica, N.; and Mahini, H. 2010. The cooperative game theory foundations of network bargaining games. In *International Colloquium on Automata, Languages, and Programming*, 67–78. Springer.

Blume, L.; Easley, D.; Kleinberg, J.; and Tardos, E. 2007. Trading networks with price-setting agents. In *Proceedings of the 8th ACM conference on Electronic commerce*, 143–151. ACM.

Chalkiadakis, G.; Elkind, E.; Markakis, E.; Polukarov, M.; and Jennings, N. R. 2010. Cooperative games with overlapping coalitions. *Journal of Artificial Intelligence Research* 39(1):179–216.

Cook, K. S., and Emerson, R. M. 1978. Power, equity and commitment in exchange networks. *American sociological review* 721–739.

Cook, K. S., and Yamagishi, T. 1992. Power in exchange networks: A power-dependence formulation. *Social Networks* 14(3-4):245–265.

Dang, V. D.; Dash, R. K.; Rogers, A.; and Jennings, N. R. 2006. Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *AAAI*, volume 6, 635–640.

Davis, M., and Maschler, M. 1965. The kernel of a cooperative game. *Naval Research Logistics Quarterly* 12(3):223–259.

Dreze, J. H., and Greenberg, J. 1980. Hedonic coalitions: Optimality and stability. *Econometrica: Journal of the Econometric Society* 987–1003.

Emerson, R. M. 1962. Power-dependence relations. *American sociological review* 31–41.

Greco, G.; Malizia, E.; Palopoli, L.; and Scarcello, F. 2009. On the complexity of compact coalitional games. In *IJCAI*, volume 9, 147–152.

Kanoria, Y.; Bayati, M.; Borgs, C.; Chayes, J.; and Montanari, A. 2011. Fast convergence of natural bargaining dynamics in exchange networks. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, 1518–1537. Society for Industrial and Applied Mathematics.

Kleinberg, J., and Tardos, É. 2008. Balanced outcomes in social exchange networks. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, 295–304. ACM.

Schmeidler, D. 1969. The nucleolus of a characteristic function game. *SIAM Journal on applied mathematics* 17(6):1163–1170.

Shapley, L. S., and Shubik, M. 1971. The assignment game i: The core. *International Journal of game theory* 1(1):111–130.

Shapley, L. S. 1953. A value for n-person games. *Contributions to the Theory of Games* 2(28):307–317.