Distributed linear programming and bargaining in exchange networks

Dean Richert Jorge Cortés

Abstract— **Many engineering, economic, and social scenarios are modeled as neighboring agents in a network interacting with each other. In the setup we consider, neighboring agents (i) bargain over the possibility of matching with at most one other agent and (ii) agree on how to allocate a common good between them. In particular, we examine stable and fair outcomes called Nash bargaining solutions. Our main contribution is the design of continuous-time distributed dynamics that converge to these Nash solutions. The technical approach leads us to develop distributed dynamics for linear programming, the results of which are of independent interest. We invoke Lyapunov techniques to prove convergence and draw results from nonsmooth and set-valued analysis of dynamical systems. In the literature pertinent to bargaining problems of the form we consider, this control perspective is unique.**

I. Introduction

In an exchange network, neighboring agents have a common good they can split between each other. We consider a type of coalitional game on an exchange network where agents are interested in stabilizing coalitions of size two (called a match). A stable outcome arises when none of the agents benefit by unilaterally deviating from their match. We also impose a notion of fairness (called balance) in the final allocation to each player: paired agents should benefit equally from their match. Such stable and balanced outcomes are called Nash bargaining solutions. Our motivation for this paper comes from many engineering applications that can be modeled by games of this form. Examples include task assignment [1], resource allocation [2], pairs of UAVs performing point-to-point reconfiguration [3], large-scale data processing [4], and wireless networks with multi-hop relaying [5]. Several sociology [6] and economic applications, such as matching in labor markets [7], also exist.

A challenging question we ask is whether players can autonomously find such outcomes using only local interactions. We interpret these interactions as bargaining among agents. When posed formally, this problem amounts to distributedly (i) solving a linear program and (ii) satisfying some nonlinear constraints within the set of solutions to this linear program. Accordingly, we devote part of this paper to studying distributed linear programming which is of independent interest to the field of distributed optimization in multi-agent settings.

Literature review. Bargaining problems of the type we consider are posed on dyadic-exchange networks, so called because agents can match with at most one other agent [8]. Bipartite matching and many assignment problems [9] are special cases of the dyadic-exchange network. Centralized methods for finding Nash bargaining solutions were developed in [10], [11]. In terms of discrete-time distributed implementations, [12] provides dynamics that converge to (balanced) allocations whereas [13] provides dynamics that converge to stable and balanced (i.e., Nash) outcomes. In contrast, our work develops continuous-time dynamics from a distributed optimization viewpoint. Our dynamics are completely novel and do not correspond to a continuous-time version of the dynamics in [12]. One of the main advantages of considering continuous-time dynamics is that they allow us to use powerful Lyapunov techniques to establish asymptotic convergence and, at the same time, they open the way to the study of their robustness properties.

Another body of literature relevant to this work concerns linear programming (as a general reference, see [14]). Distributed implementation has become a subject of interest. This is motivated by the need to solve large scale problems [15] and systems where the data of the problem is naturally dispersed over a graph [16], [17]. Some discrete-time solutions exist such as projection methods [18], [19] and the distributed simplex algorithm [20]. The method we propose is implemented in continuous-time and applicable to a broad class of multi-agent scenarios. In fact, a form of the algorithm we present appears in [21] with the major difference that our dynamics are smooth in some of the variables.

Statement of contributions. The contributions of this paper are three-fold. First, with regards to distributed linear programming, we present a saddle-point method for solving constrained linear programs. To our knowledge, we are the first to (i) prove convergence of these dynamics to the set of solutions of a completely general linear program, and (ii) study these dynamics in the context of distributed implementation; in particular, the multi-agent setting.

Second, we setup the dyadic-exchange bargaining problem and introduce three notions of a bargaining outcome (stable, balanced, and Nash). For each case we develop continuous-time distributed dynamics that converge to them. Our analysis combines nonsmooth and set-valued dynamical systems theory, and results from linear and convex optimization. Simulations validate our results.

Third, our presentation of the material is done from a control and dynamical systems perspective. Thus, we provide new insights into distributed linear programming and bargaining in exchange networks. Moreover, we contribute a framework that we believe is amenable for extensions to this work.

For reasons of space, all proofs will appear elsewhere.

The authors are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA 92093, USA, {drichert,cortes}@ucsd.edu

II. Preliminaries

With regards to notation, the set of real numbers is denoted $\mathbb{R}$. The nonnegative part of a set $X \subset \mathbb{R}^n$ is given by X_+. If $x \in \mathbb{R}^n$ then we use $x \geq 0$ (resp. $x > 0$) to mean that all components of x are nonnegative (resp. positive). Given two sets A and B we use $A \setminus B$ to mean those elements that are in A but not B. The set A is convex if it fully contains the segment connecting any two points in A. We use $||\cdot||$ to denote any norm. The set $\mathbb{B}(x,\delta) \subset \mathbb{R}^n$ is the open ball centered at $x \in \mathbb{R}^n$ with radius $\delta > 0$. Given a matrix A, the ℓ^{th} row of A is denoted $\mathrm{row}_\ell(A)$. A function $f : X \to \mathbb{R}$ defined on the convex set $X \subset \mathbb{R}^n$ is convex if $f(kx + (1-k)y) \leq kf(x) + (1-k)f(y)$ for all $x, y \in X$ and $k \in [0,1]$. f is concave if $-f$ is convex. The function $L : X \times Y \to \mathbb{R}$ defined on the convex set $X \times Y \subset \mathbb{R}^n \times \mathbb{R}^m$ is convex-concave if it is convex on X and concave on Y. A point $(\bar{x}, \bar{y}) \in X \times Y$ is a saddle point of L if $L(\bar{x}, y) \geq L(\bar{x}, \bar{y}) \geq L(x, \bar{y})$ for all $(x, y) \in X \times Y$. Finally, a set-valued map $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ maps elements in $\mathbb{R}^n$ to subsets of $\mathbb{R}^n$.

A. Nonsmooth analysis

The following exposition on nonsmooth analysis is taken from [22]. A function $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz at $x \in \mathbb{R}^n$ if there exists an $\delta_x > 0$ and $L_x \geq 0$ such that $|f(x) - f(y)| \leq L_x ||x - y||$ for all $y \in \mathbb{B}(x, \delta_x)$. When f is locally Lipschitz at all $x \in \mathbb{R}^n$ we simply call f locally Lipschitz. If f is convex, then it is locally Lipschitz. Let $\Omega_f, S \subset \mathbb{R}^n$ be the set of points where f is not differentiable and a set of measure zero, respectively. Then, if f is locally Lipschitz, the generalized gradient of f at $x \in \mathbb{R}^n$ is

$$\partial f(x) = \mathrm{co}\left\{ \lim_{i\to\infty} \nabla f(x_i) : x_i \to x, x_i \notin S \cup \Omega_f \right\},$$

where $\mathrm{co}\{\cdot\}$ denotes the convex hull of a set of points. A set-valued map $F : X \subset \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is upper semi-continuous if for all $x \in X$ and $\epsilon \in (0, \infty)$ there exists a $\delta_x \in (0, \infty)$ such that $F(y) \subseteq F(x) + \mathbb{B}(0, \epsilon)$ for all $y \in \mathbb{B}(x, \delta_x)$.

Lemma II.1 (Properties of the generalized gradient). *If $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz at $x \in \mathbb{R}^n$, then $\partial f(x)$ is nonempty, convex, and compact. Moreover, $x \mapsto \partial f(x)$ is locally bounded and upper semi-continuous.*

If $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ and $y \in \mathbb{R}^n$, then $\partial_x f(x, y)$ is used to denote the generalized gradient of $x \mapsto f(x, y)$.

B. Set-valued dynamical systems

This section follows the exposition in [23] for a tutorial on this subject. A time-invariant set-valued dynamical system is given by the differential inclusion

$$\dot{x}(t) \in F(x(t)), \tag{1}$$

where $t \in \mathbb{R}_{\geq 0}$ and $F : X \subset \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is a set valued map. If F is locally bounded, upper semi-continuous and takes nonempty, convex, and compact values then there exists an absolutely continuous curve $x : \mathbb{R}_{\geq 0} \to X$ (called a trajectory or solution) satisfying (1) almost everywhere from any initial condition $x(0) \in X$.

C. Graph theory

A weighted undirected graph is a triplet $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ where $\mathcal{V} = \{1, \ldots, n\}$ are vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are edges, and $W \in \mathbb{R}^{|\mathcal{E}|}_{>0}$ is a vector of edge weights indexed by edges in $\mathcal{G}$. The neighbors of a vertex i are denoted $\mathcal{N}_i = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$. A matrix $A \in \mathbb{R}^{m\times n}$ is called compatible with $\mathcal{G}$ if $a_{\ell,i}, a_{\ell,j} \neq 0 \Rightarrow (i,j) \in \mathcal{E}$ for every $\ell = 1, \ldots, m$ and $i, j \in \mathcal{V}$ (the reverse implication need not be true).

III. Distributed linear programming

In this section we introduce continuous-time dynamics for linear programming and present them in the context of a multi-agent system. We use these dynamics in the network bargaining scenario introduced in Section IV. Nevertheless, the results presented here are of independent interest.

A. Saddle-point dynamics for linear programming

Consider the linear program in standard form,

$$\min \quad c^T x \tag{2a}$$
$$\text{s.t.} \quad Ax = b, \quad x \geq 0, \tag{2b}$$

where $x, c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, and $b \in \mathbb{R}^m$. In this paper, we only consider feasible linear programs. Accordingly, a solution of (2) is denoted x_*. At times we make reference to the dual formulation of (2)

$$\max \quad -b^T z \tag{3a}$$
$$\text{s.t.} \quad A^T z - \lambda + c = 0, \quad \lambda \geq 0, \tag{3b}$$

where z_* denotes a solution of (3). The following result relates the solutions of (2) to the saddle points of a modified Lagrangian function.

Proposition III.1 (Solutions of linear program as saddle points). *Define $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ as*

$$L(x,y) = c^T x + \frac{1}{2}(Ax - b)^T(Ax - b) + y^T(Ax - b) + \gamma \sum_{i=1}^n \max\{0, -x_i\},$$

where $\gamma > 0$. Then L is convex in x and concave (in fact, linear) in y. Moreover, for γ sufficiently large,

(i) *if $x_* \in \mathbb{R}^n$ is a solution of (2) then there exists $y \in \mathbb{R}^m$ such that (x_*, y) is a saddle point of L,*
(ii) *if (x_*, y) is a saddle point of L, then $x_* \in \mathbb{R}^n$ is a solution of (2).*

A lower bound can be characterized for γ. However, for the main results of this paper, it will be seen that only the existence of γ is essential.

Remark III.2 (Modified Lagrangian). Note that L is simply the standard Lagrangian function augmented with two terms. The nonsmooth max term is an exact penalty function enforcing $x \geq 0$. The term $(Ax - b)^T(Ax - b)$ appears for technical reasons and is used to prove convergence results for the dynamics we propose next. •

The power of the above result is two-fold. First, we are justified in studying the saddle points of L rather than trying to directly solve the constrained optimization problem. Second, since L is convex-concave, we are inspired to consider saddle-point dynamics. Typically, differentiability of the Lagrangian function is a desirable property when designing saddle-point dynamics. For example, in [24], the inequality constraints are enforced by means of an inexact log-barrier function. The tools of nonsmooth analysis in systems theory allow us to deal with the non-differentiable nature of L. As a result, we are able to design dynamics that converge exactly to desired solutions.

Let us consider the saddle-point dynamics based on the generalized gradients of L defined by:

$$\dot{x} + c + A^T(y + Ax - b) \in \gamma\partial \max\{0, -x\}, \tag{4a}$$

$$\dot{y} = Ax - b, \tag{4b}$$

where $\max\{0, -x\} = (\max\{0, -x_i\}, \dots, \max\{0, -x_n\}) \in \mathbb{R}^n_{\geq 0}$. Since the dynamics (4) are derived from the gradients of a locally Lipschitz function, it satisfies the sufficient conditions for the existence of solutions (cf. Lemma II.1 and Section II-B). The following result states the asymptotic convergence of the above dynamics to solutions of (2).

Proposition III.3 (Asymptotic convergence of saddle-point dynamics). *For $\gamma > 0$ sufficiently large, the projection onto the first (resp. second) component of any trajectory of* (4) *asymptotically converges to a solution of* (2) *(resp.* (3)*).*

In order to implement (4), it seems that γ must be known. However, (as we mentioned earlier) it turns out that we can implement (4) based only on the knowledge of γ's existence. To achieve this, consider the following *nominal flow function*

$$F^{\text{nom}}(x, y) = -c - A^T(y + Ax - b).$$

Note in particular that $F^{\text{nom}}(x, y) \equiv -\partial_x L(x, y)$ on the domain $\mathbb{R}^n_{>0} \times \mathbb{R}^m$. Then, we propose the alternative discontinuous dynamics for convergence to solutions of (2),

$$\dot{x}_i = \begin{cases} F_i^{\text{nom}}(x, y), & x_i > 0, \\ \max\{0, F_i^{\text{nom}}\}, & x_i = 0, \end{cases} \quad \forall i \in \{1, \dots, n\}, \tag{5a}$$

$$\dot{y} = Ax - b. \tag{5b}$$

The following states the convergence properties of (5).

Proposition III.4 (Asymptotic convergence of (5)**).** *For any initial condition $(x_0, y_0) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^m$, the projection onto the first (resp. second) component of any trajectory of* (5) *asymptotically converges to a solution of* (2) *(resp.* (3)*).*

B. Multi-agent setting

Let us now study the distributed implementation of (5) in a multi-agent setting. For clarity, it is helpful to express the nominal flow function component-wise,

$$\begin{aligned} F_i^{\text{nom}}(x, y) &= -c_i - \sum_{\ell=1}^{m} a_{\ell,i}\Big[y_\ell + \sum_{k=1}^{n} a_{\ell,k}x_k - b_\ell\Big], \quad (6) \\ &= -c_i - \sum_{\{\ell\,:\,a_{\ell,i}\neq 0\}} a_{\ell,i}\Big[y_\ell + \sum_{\{k\,:\,a_{\ell,k}\neq 0\}} a_{\ell,k}x_k - b_\ell\Big]. \end{aligned}$$

Suppose the interconnections between agents are modeled by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$. Then we say that the dynamics (5) are *distributed over* $\mathcal{G}$ if

(i) for each $i \in \mathcal{V} = \{1, \dots, n\}$, agent i knows
   a) $c_i \in \mathbb{R}$,
   b) every $b_\ell \in \mathbb{R}$ for which $a_{\ell,i} \neq 0$,
   c) the non-zero elements of every $\text{row}_\ell(A) \in \mathbb{R}^n$ for which $a_{\ell,i} \neq 0$,

(ii) agent i has control over the variable $x_i \in \mathbb{R}$, and

(iii) agents can observe or measure the variables controlled by neighboring agents (i.e., A is compatible with $\mathcal{G}$).

Remark III.5 (Implementation of (5b)**).** Component-wise, the dynamics (5b) are

$$\dot{y}_\ell = \sum_{\{i\,:\,a_{\ell,i}\neq 0\}} a_{\ell,i}x_i - b_\ell. \tag{7}$$

Since A is compatible with $\mathcal{G}$ and agent i knows the non-zero elements of every $\text{row}_\ell(A)$ for which $a_{\ell,i} \neq 0$, the dynamics (7) are implementable by an agent i who requires y_ℓ in its computation of (6). These dynamics are shared among $i \in \mathcal{V}$ for which $a_{\ell,i} \neq 0$ (i.e., neighbors of i). •

In solving (2), agents are only interested in computing their local component of the solution, not the entire solution itself. Our dynamics are consistent with this fact. This is in contrast to the dynamics proposed in [18], [19], which would not be truly distributed for the class of problems we consider. Problems for which (5) is distributed over a network of agents include (among others) bipartite matching, task assignment, resource allocation, and bargaining on exchange networks.

IV. Bargaining in exchange networks

In this section, we introduce the bargaining problem on an exchange network. The problem setup is as follows. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ be an undirected weighted graph. In a bargaining network, vertices correspond to agents (or players) and edges connect agents who have the ability to negotiate with each other. Should they come to an agreement, the edge weight $w_{i,j}$ is divided between i and j accordingly. On one hand, agents are selfish and seek to maximize the amount they receive. However if two agents cannot come to an agreement they forfeit the entire amount $w_{i,j}$. In this paper, we consider bargaining outcomes of the following form.

Definition IV.1 (Outcomes). An *outcome* is a pair (M, α), where $M \subset \mathcal{E}$ is a *matching* (i.e., a set of edges without common vertices) and $\alpha \in \mathbb{R}^n$ is an *allocation* to each agent. An outcome has the additional property that $\alpha_i + \alpha_j = w_{i,j}$ if $(i, j) \in M$ and $\alpha_k = 0$ if agent k is not part of any edge in M. We consider three classes of outcomes.

Stable: An outcome (M, α^s) such that $\alpha^s \geq 0$ and

$$\alpha_i^s + \alpha_j^s \geq w_{i,j}, \quad \forall (i,j) \in \mathcal{E}.$$

Balanced: An outcome (M, α^b) where for all $(i,j) \in M$,

$$\alpha_i^b - \max_{k \in \mathcal{N}_i \setminus j} \{w_{i,k} - \alpha_k^b\}_+ = \alpha_j^b - \max_{\ell \in \mathcal{N}_j \setminus i} \{w_{j,\ell} - \alpha_\ell^b\}_+.$$

Nash: An outcome (M, α^N) that is stable and balanced. •

Outcomes of these type are appropriate in the context of *dyadic-exchange* networks, so called because each player is allowed to pair with at most one other player. A stable outcome ensures that no agent can strictly increase its allocation by unilaterally trading with an alternative player. On the other hand, in a balanced outcome, matched agents benefit equally from the matching (with respect to their next-best-alternative). The above definition of a Nash outcome is an extension of the classical two player Nash bargaining solution [25] to multi-player bargaining networks. See Figure 1 for specific examples that motivate these outcomes.

The problem we solve is to develop distributed dynamics that converge to stable, balanced, and Nash outcomes.

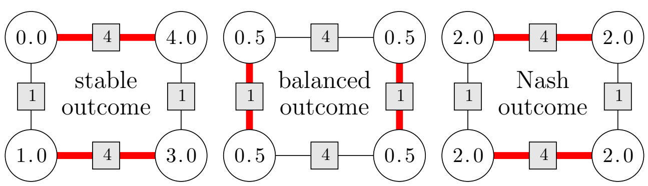


Fig. 1. For each outcome, edge weights are in grey boxes, matches are thicker edges, and allocations are in the vertex circle. In the stable outcome, the 0.0 allocation hardly seems fair to that node. In the balanced outcome, agents can receive a higher allocation by deviating from their matches. Nash outcomes do not exhibit either of these shortcomings.

A. Distributed convergence to stable outcomes

It has been established [10] that when a stable outcome (M, α^s) exists, the matching M is a maximum weight matching on $\mathcal{G}$. A *maximum weight* matching is one in which the sum of the edge weights in the matching is maximal. Mathematically, they are constructed from solutions to

$$\begin{aligned} \max \quad & \textstyle\sum_{(i,j)\in\mathcal{E}} w_{i,j} m_{i,j} \\ \text{s.t.} \quad & \textstyle\sum_{j\in\mathcal{N}_i} m_{i,j} \leq 1, \quad \forall i \in \mathcal{V}, \\ & m_{i,j} \in \{0,1\}, \quad \forall (i,j) \in \mathcal{E}, \end{aligned}$$

where the matching induced by $(i,j) \in M \Leftrightarrow m_{i,j} = 1$ is well-defined. The linear programming relaxation is

$$\begin{aligned} \max \quad & \textstyle\sum_{(i,j)\in\mathcal{E}} w_{i,j} m_{i,j} \\ \text{s.t.} \quad & \textstyle\sum_{j\in\mathcal{N}_i} m_{i,j} \leq 1, \quad \forall i \in \mathcal{V}, \\ & m_{i,j} \geq 0, \quad \forall (i,j) \in \mathcal{E}, \end{aligned} \tag{8}$$

with the associated dual

$$\begin{aligned} \min \quad & \textstyle\sum_{i\in\mathcal{V}} \alpha_i^s \\ \text{s.t.} \quad & \alpha_i^s + \alpha_j^s \geq w_{i,j}, \quad \forall (i,j) \in \mathcal{E}, \\ & \alpha_i^s \geq 0, \quad \forall i \in \mathcal{V}. \end{aligned} \tag{9}$$

An interesting fact relating stable outcomes and solutions to (8) appears in [10], [12], [13].

Lemma IV.2 (Existence of stable outcomes). *A stable outcome exists if and only if* (8) *yields an integral solution.*

A systematic method for determining whether (8) yields an integral solution remains an open problem. From here on, we assume that a stable outcome exists and the maximum weight matching is unique (a standard assumption in exchange network bargaining, see e.g. [13]). Besides technical implications, requiring uniqueness has a practical motivation. For example, if an agent will receive the same allocation regardless of the matching, it is unclear with whom it will choose to match with. Consequently, when a stable outcome exists, finding one is a matter of solving the relaxed maximum weight matching problem, where the matching is induced from the solution of (8) and the allocation is any solution to (9). Moreover, the results of Section III allow agents to distributedly compute this outcome. To put (9) in standard form, introduce slack variables $s_{i,j}$ for each $(i,j) \in \mathcal{E}$,

$$\begin{aligned} \min \quad & \textstyle\sum_{i\in\mathcal{V}} \alpha_i^s \\ \text{s.t.} \quad & \alpha_i^s + \alpha_j^s - s_{i,j} = w_{i,j}, \quad \forall (i,j) \in \mathcal{E}, \\ & \alpha_i^s \geq 0 \quad \forall i \in \mathcal{V}, \quad \text{and} \quad s_{i,j} \geq 0 \quad \forall (i,j) \in \mathcal{E}. \end{aligned}$$

Directly substituting into (6), we obtain the following distributed nominal flow functions for each agent $i \in \mathcal{V}$,

$$\mathcal{S}_i^{\text{nom}}(\alpha^s, s, \mu) = -1 - \sum_{j\in\mathcal{N}_i} \Big[\mu_{i,j} + \alpha_i^s + \alpha_j^s - s_{i,j} - w_{i,j}\Big],$$

and for each $(i,j) \in \mathcal{E}$

$$\mathcal{S}_{i,j}^{\text{nom}}(\alpha^s, s, \mu) = -\mu_{i,j} - \alpha_i^s - \alpha_j^s + s_{i,j} + w_{i,j}.$$

Then, the dynamics (5) for each agent $i \in \mathcal{V}$ are

$$\dot{\alpha}_i^s = \begin{cases} \mathcal{S}_i^{\text{nom}}(\alpha^s, s, \mu), & \alpha_i^s > 0, \\ \max\{0, \mathcal{S}_i^{\text{nom}}(\alpha^s, s, \mu)\}, & \alpha_i^s = 0, \end{cases} \tag{10a}$$

and for each edge $(i,j) \in \mathcal{E}$ agents i and j share

$$\dot{s}_{i,j} = \begin{cases} \mathcal{S}_{i,j}^{\text{nom}}(\alpha^s, s, \mu), & s_{i,j} > 0, \\ \max\{0, \mathcal{S}_{i,j}^{\text{nom}}(\alpha^s, s, \mu)\}, & s_{i,j} = 0, \end{cases} \tag{10b}$$

$$\dot{\mu}_{i,j} = \alpha_i^s + \alpha_j^s - s_{i,j} - w_{i,j}. \tag{10c}$$

We use the notation s to represent the vector of slacks indexed by edges in $\mathcal{G}$ (likewise for μ).

Proposition IV.3 (Convergence to stable outcomes). *For any initial condition* $(\alpha_0^s, s_0, \mu_0) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \times \mathbb{R}^{|\mathcal{E}|}$,

(i) *the projection onto the first component of any trajectory of* (10) *asymptotically converges to the set of solutions of* (9) *(denote this set by* A^s*),*

(ii) *if* (8) *yields a unique integral solution then the projection onto the second component of any trajectory of* (10) *asymptotically converges to a point* s*. In this case, the matching induced by the implication*

$$(i,j) \in M \quad \Leftrightarrow \quad s_{i,j} = 0,$$

is well-defined and is a maximum weight matching. Thus, (M, α^s) is a stable outcome for any $\alpha^s \in A^s$.

Moreover, the dynamics (10) *are distributed over $\mathcal{G}$.*

B. Distributed convergence to balanced outcomes

Here, we propose distributed dynamics that converge to balanced outcomes. We assume that a matching M has been given and that only allocations are being negotiated. This assumption is dropped when we consider Nash outcomes.

The following concepts will be used in our discussion. Suppose $(i, j) \in M$. The *best allocation* that i could expect to receive by matching with a neighbor other than j is

$$\mathtt{ba}_{i\setminus j}(\alpha^b) = \max_{k\in\mathcal{N}_i\setminus j}\{w_{i,k} - \alpha^b_k\}_+.$$

Moreover, the argument of the above function is the set of *best neighbors* (possibly empty)

$$\mathtt{bn}_{i\setminus j} = \operatorname{argmax}_{k\in\mathcal{N}_i\setminus j}\{w_{i,k} - \alpha^b_k\}_+.$$

Since $\alpha^b_i + \alpha^b_j = w_{i,j}$ for $(i, j) \in M$ in an outcome, the balance condition for two matched agents can be restated as

$$\alpha^b_i = \frac{1}{2}(w_{i,j} + \mathtt{ba}_{i\setminus j}(\alpha^b) - \mathtt{ba}_{j\setminus i}(\alpha^b)),$$
$$\alpha^b_j = \frac{1}{2}(w_{i,j} - \mathtt{ba}_{i\setminus j}(\alpha^b) + \mathtt{ba}_{j\setminus i}(\alpha^b)).$$

The above formulation inspires the following distributed dynamics, whose equilibria are by construction allocations in a balanced outcome,

$$\dot{\alpha}^b_i = \begin{cases} \frac{1}{2}(w_{i,j} + \mathtt{ba}_{i\setminus j}(\alpha^b) - \mathtt{ba}_{j\setminus i}(\alpha^b)) - \alpha^b_i, & \\ & \text{if } (i,j)\in M, \\ -\alpha^b_i, & \text{otherwise.}\end{cases} \quad (11)$$

At any time t and for $j \in \mathcal{N}_i$, the quantity $w_{i,j} - \alpha^b_i(t)$ has the interpretation of "i's offer to j". Note that (11) are continuous and require 2-hop information because i updates its offer to j based on $\mathtt{ba}_{j\setminus i}$. The design of (11) is motivated (yet distinctly different) from the dynamics proposed in [12].

Proposition IV.4 (Asymptotic convergence to balanced outcomes). *Given a matching M, the dynamics* (11) *globally asymptotically converge to an allocation α^b such that (M, α^b) is a balanced outcome. Moreover, the dynamics* (11) *are distributed with respect to 2-hop neighborhoods over $\mathcal{G}$.*

C. Distributed convergence to Nash outcomes

In this section, we propose distributed dynamics that converge to Nash outcomes. The design of these dynamics is inspired from the following result, adapted slightly from [12].

Proposition IV.5 (Balanced implies stable). *Assume there exists a stable outcome and M is a maximum weight matching. Then any balanced outcome (M, α^b) is also stable and thus Nash.*

In a nutshell, the dynamics we propose next combine the fact that (i) the distributed dynamics of Section IV-A converge to a maximum weight matching and (ii) given such a maximum weight matching, the distributed dynamics of Section IV-B converge to a balanced (and therefore Nash) outcome. We further exploit the fact that agents are able to execute the balancing dynamics (11) before (10) have converged. To do so, agents predict with whom (if any) they will be matched in a final Nash outcome. An agent makes this prediction based on the current value of the slack variables $s(t) \in \mathbb{R}^{|\mathcal{E}|}$ in (10). In other words, agent i predicts its *partner* by

$$\mathcal{P}_i(s) = \{j \in \mathcal{N}_i : s_{i,j} < s_{i,k},\ \forall k \in \mathcal{N}_i \setminus j\}.$$

Clearly, $\mathcal{P}_i(s)$ is at most a singleton and can be computed by i using local information. Likewise, if $\emptyset \neq \mathcal{P}_i(s) = j$, then $\mathcal{P}_j(s)$ can be computed by i with 2-hop information. Using this notion of partner, Proposition IV.3(ii) and the asymptotic convergence properties of (10) reveal the following fact: If (8) yields a unique integral solution then $\exists T > 0$ such that the matching induced by the implication

$$\begin{aligned}(i,j)\in M \quad &\Leftrightarrow \quad s_{i,j}(t) = 0,\\ &\Leftrightarrow \quad [\emptyset \neq \mathcal{P}_i(s(t)) = j] \wedge [\mathcal{P}_j(s(t)) = i],\end{aligned}$$

is well-defined and is a maximum weight matching for all $t \geq T$. Further, after time T, the following dynamics implemented in conjunction with (10) converge to allocations of a Nash outcome,

$$\dot{\alpha}^b_i = \begin{cases} \frac{1}{2}(w_{i,j} + \mathtt{ba}_{i\setminus j}(\alpha^b) - \mathtt{ba}_{j\setminus i}(\alpha^b)) - \alpha^b_i, & \\ \quad \text{if } [\emptyset \neq \mathcal{P}_i(s) = j] \wedge [\mathcal{P}_j(s) = i], & \\ -\alpha^b_i, \quad \text{otherwise.}\end{cases} \quad (13)$$

In (13), agents begin balancing their allocations if they identify each other as partners. Since the s are fed to i by (10), these dynamics are a cascade system. We retain asymptotic convergence with this cascade structure as stated next.

Theorem IV.6 (Asymptotic convergence to Nash outcome). *Suppose that* (8) *yields a unique integral solution and all agents implement* (10) *and* (13)*, which are distributed with respect to 2-hop neighborhoods over $\mathcal{G}$. Then, for any initial conditions $(\alpha^s_0, s_0, \mu_0) \in \mathbb{R}^n_{\geq 0} \times \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \times \mathbb{R}^{|\mathcal{E}|}$ and $\alpha^b_0 \in \mathbb{R}^n$,*

(i) *the projection onto the second component of any trajectory of* (10) *converges asymptotically to a point s and*

(ii) *any trajectory of* (13) *converges to a point α^N.*

Moreover, the matching induced by the implication

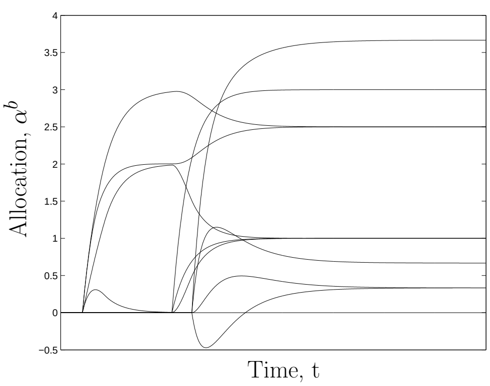
$$(i,j) \in M \quad \Leftrightarrow \quad s_{i,j} = 0.$$

is well-defined and is a maximum weight matching. Thus, (M, α^N) is a Nash outcome.

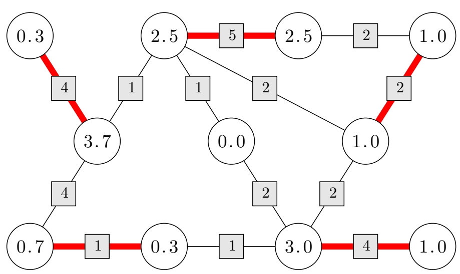
We have validated the above result in simulation. Figure 2 displays some results for an 11 node random graph.

V. Conclusions and future work

We have considered bargaining among agents in an exchange network. In particular, we studied dyadic-exchanges where agents pair with at most one other agent. Players had to decide with whom (if any) to match and agree on an allocation of a common good. We designed continuous-time distributed dynamics to converge to Nash bargaining solutions. In the final outcome, individual agents had no incentive to unilaterally deviate from their match. Of independent interest, we presented continuous-time distributed dynamics for linear programming and stated convergence results. Several simulations validated our results. Future work will include exploring finite-time and event-triggered versions of our dynamics, considering other solution concepts on dyadic-exchange networks, as well as applying our techniques to multi-exchange networks (i.e., coalitions of more than two). In addition, we would like to study the rate of convergence and robustness properties of the proposed dynamics; in particular, the effects of time delays and adversarial agents. Finally, we wish to study our dynamics in the context of specific coordination tasks such as point-to-point reconfiguration of UAV formation pairs [3].



(a) Evolution of agents' allocations (i.e., trajectories of (13)) resulting from distributed dynamics (10) and (13).



(b) Graph topology and final outcome of the simulation. Edge weights are in grey boxes, matches are thicker edges, allocations are in the vertex circles. For reasons of presentation, some values were rounded.

Fig. 2. Simulation results for the 11 node randomly generated network shown in (b). Agents bargain with their neighbors over how to split an edge weight. An evolving allocation (see (a)) is interpreted as bargaining between neighbors. Allocations at steady-state represent an agreement between neighbors over how to split an edge weight. The final steady-state allocations are presented in (b). As predicted by Theorem IV.6, a Nash outcome is achieved.

Acknowledgments

This research was supported by Award FA9550-10-1-0499.

References

[1] B. J. Moore and K. M. Passino, "Distributed task assignment for mobile agents," *IEEE Transactions on Automatic Control*, vol. 52, pp. 749–753, April 2007.

[2] S. H. Ali, K. Lee, and V. C. M. Leung, "Dynamic resource allocation in OFDMA wireless metropolitan area networks," *IEEE Wireless Communications*, vol. 14, pp. 6–13, Feb. 2007.

[3] D. Richert and J. Cortés, "Optimal leader allocation in UAV formation pairs under costly switching," in *IEEE Conf. on Decision and Control*, (Maui, Hawaii), pp. 831–836, 2012.

[4] C. J. Romanowski, R. Nagi, and M. Sudit, "Data mining in an engineering design environment: OR applications from graph matching," *Computers and Operations Research*, vol. 33, pp. 3150–3160, Nov. 2006.

[5] Z. Zhaoyang, S. Jing, C. Hsiao-Hwa, M. Guizani, and Q. Peiliang, "A cooperation strategy based on Nash bargaining solution in cooperative relay networks," *IEEE Transactions on Vehicular Technology*, vol. 57, pp. 2570–2577, July 2008.

[6] T. Chakraborty, S. Judd, M. Kearns, and J. Tan, "A behavioral study of bargaining in social networks," in *ACM Conference on Electronic Commerce*, pp. 243–252, June 2010.

[7] A. E. Roth, "The evolution of the labor market for medical interns and residents: A case study in game theory," *Journal of Political Economy*, vol. 92, pp. 991–1016, Dec. 1984.

[8] K. S. Cook and R. M. Emerson, "Power, equity and commitment in exchange networks," *American Sociological Review*, vol. 43, pp. 721–739, Oct. 1978.

[9] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Athena Scientific, 1998.

[10] J. Kleinberg and E. Tardos, "Balanced outcomes in social exchange networks," in *Proceedings of the Annual ACM Symposium on Theory of Computing*, (Victoria, Canada), pp. 295–304, May 2008.

[11] M. Bateni, M. Hajiaghayi, N. Immorlica, and H. Mahini, "The cooperative game theory foundations of network bargaining games," in *Internation Colloquium on Automata, Languages and Programming*, (Bordeaux, France), pp. 67–78, July 2010.

[12] Y. Azar, B. Birnbaum, L. Celis, N. Devanur, and Y. Peres, "Convergence of local dynamics to balanced outcomes," 2009. Submitted. Available at http://arxiv.org/abs/0907.4356.

[13] Y. Kanoria, M. Bayati, C. Borgs, J. Chayes, and A. Montanari, "Bargaining dynamics in exchange networks," 2011. Submitted. Available at http://arxiv.org/abs/1004.2079.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.

[15] D. P. Bertsekas, *Parallel and Distributed Computations: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[16] R. Alberton, R. Carli, A. Cenedese, and L. Schenato, "Multi-agent perimeter patrolling subject to mobility constraints," in *American Control Conference*, (Montréal), pp. 4498–4503, 2012.

[17] W. Kim and A. C. Kak, "3-D object recognition using bipartite matching embedded in discrete relaxation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 224–251, March 1991.

[18] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[19] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.

[20] M. Burger, G. Notarstefano, F. Allgower, and F. Bullo, "A distributed simplex algorithm and the multi-agent assignment problem," in *American Control Conference*, (San Francisco), pp. 2639–2644, July 2011.

[21] K. Arrow, L. Hurwitz, and H. Uzawa, *A Gradient Method for Approximating Saddle Points and Constrained Maxima*. United States Army Air Forces: Rand Corporation, 1951.

[22] F. H. Clarke, *Optimization and Nonsmooth Analysis*. Canadian Mathematical Society Series of Monographs and Advanced Texts, Wiley, 1983.

[23] J. Cortés, "Discontinuous dynamical systems - a tutorial on solutions, nonsmooth analysis, and stability," *IEEE Control Systems Magazine*, vol. 28, no. 3, pp. 36–73, 2008.

[24] J. Wang and N. Elia, "A control perspective for centralized and distributed convex optimization," in *IEEE Conf. on Decision and Control*, (Orlando, Florida), pp. 3800–3805, 2011.

[25] J. Nash, "The bargaining problem," *Econometrica*, vol. 8, no. 2, pp. 155–162, 1950.