Cooperative Task-Processing Networks: Parallel Computation of Non-trivial Volunteering Equilibria*

OSU CSE Tech. Report OSU-CISRC-3/11-TR05

Theodore P. Pavlic†
Computer Science & Engineering
The Ohio State University
Columbus, OH 43210 USA
pavlic.3@osu.edu

Kevin M. Passino
Electrical & Computer Engineering
The Ohio State University
Columbus, OH 43210 USA
passino@ece.ohio-state.edu

March 9, 2011

Abstract

This work gives the complete details of a novel framework for the analysis and design of distributed agents that must complete externally generated tasks but also can volunteer to process tasks encountered by other agents. A distributed asynchronous volunteering policy is presented that dynamically adjusts task flow around the network of agents. It is shown that even though agents independently adjust their tendency to volunteer to process tasks from other agents, the set of all volunteering tendencies converges to the unique Nash equilibrium of a cooperation game. An artificial cooperation trading economy ensures that at the equilibrium, non-zero cooperation tendencies are possible and vary across agents. In particular, an agent with relatively high task-encounter rate not only provides more incentive for connected neighbors to cooperate with it but also has less incentive to volunteer to cooperate with other agents. The framework is shown via simulation to be applicable to autonomous air vehicles, and the mathematical results of the paper are also shown to be consistent with classic studies of cooperation from science.

Keywords: distributed multi-agent control, game theory, cooperation, Nash equilibria, mobile agents, parallel asynchronous computation

*This work was partially supported by the National Science Foundation under Grant No. EECS-0931669.

†Corresponding author.

Contents

1 Introduction

In this work, we consider a network of autonomous agents for which some agents are responsible for processing tasks from one or more external sources. When a task arrives at one of these agents, the agent may advertise the task to other agents connected to it. If none of the connected agents volunteer to process the task, it must be processed by the advertising agent; otherwise, the task is processed by one of the volunteering agents. Agents that volunteer for tasks may also be connected to incoming task flows for which they can advertise task encounters. In general, an agent in the network may advertise task encounters to others, volunteer to process advertised tasks from others, or do both. Our challenge is to define a distributed asynchronous algorithm for automatically tuning how often each agent volunteers to process advertised tasks so that the set of volunteering tendencies across the network converges to a Nash equilibrium with cooperative features. Later in this section, the motivations for such a task-processing network of selfish cooperative agents are discussed. In particular, Section 1.1 describes existing results in the design of cooperative agents and highlights the contribution made by this work, and Section 1.2 provides background for understanding how this work relates to classic studies of the emergence of cooperation in natural systems.

The rest of this document is organized as follows. In Section 2, the task-processing network framework is defined and example task-processing networks are described. The optimization game is presented in Section 3, and an asynchronous distributed computation method that ensures convergence to the game's Nash equilibrium is given in Section 4. In Section 5, results from a simulated task-processing network of autonomous air vehicles are presented, and conclusions and areas of future research are discussed in Section 6. Mathematical conventions are listed in Appendix A, and technical details omitted from the main document are included in the other appendices.

1.1 Cooperative agent design

Grid computing [11] is an existing approach for achieving cooperative task processing across a group of networked agents. System designers work under the assumption of heterogeneous agents with conflicting priorities. They borrow from the economic theories of mechanism design [36, Ch. 23] and implementation theory [43, Ch. 10] to design mechanisms (e.g., brokering agents) and protocols that either encourage resource sharing [3, 4, 9, 12, 13, 23, 30, 46, 48, 50] or discourage exploitation [42, 45] among groups of agents. The common element of these different methods of *distributed algorithmic mechanism design* (DAMD) [18] is that the designer has no direct control over individual agents; instead, they control the structure of the interactions between agents on a given network. Hence, DAMD is not appropriate for the design of the task-processing networks themselves.

Methods exist for the design of networks of interconnected task-processing agents that have desirable task flow characteristics. For example, a *flexible manufacturing system* (FMS) includes several machines that switch their current processing to one of several input task flows and then produce output task flows for other machines in the system. Perkins and Kumar [44] show that distributed scheduling policies exist that guarantee such systems will have finite upper bounds on all buffers of tasks. Similarly, Cruz [15] shows how special network elements can be combined to form queueing systems with output traffic flows that are guaranteed to have finite burstiness constraints so long as the input flows also satisfy similar constraints. These methods are not intended to describe how agents can dynamically adjust task flow to exploit unused processing ability on idle connected agents.

Because an optimal task flow configuration may be unknown, inaccessible, or changing over time, task-processing agents may need to use feedback to acquire and stabilize the optimal taskhandling behavior. For example, a set of autonomous air vehicles (AAV) deployed for distributed search, surveillance, or task processing can coordinate their actions in order to converge on a holistically optimal behavior [19, 20, 22]. However, the coordination required between agents can be prohibitive. Additionally, the single optimality criteria being maximized ignores fatigue on individual agents. For example, in a smart power grid [26], it may be desirable for distributed power stations to share load; however, a single overloaded station should not result in a cascade of self-sacrificing failures. Here, non-cooperative game theory is used to develop totally asynchronous and distributed algorithms for task-processing agents that both respect local processing priorities while also sharing the processing burden of highly loaded neighbors.

Non-cooperative game theory has been traditionally used to design optimal control strategies [7, 34]; however, it can also be used to design simple selfish strategies that nonetheless assist neighbors. Several such techniques already exist for designing policies on nodes in *ad hoc* multi-hop communication networks [1, 2, 10]. In these cases, nodes can forward packets from other nodes in order to reduce network congestion or improve communication diversity, but nodes resist using all local resources for assisting other nodes. A salient feature of these forwarding networks is that packets can be duplicated or dropped at any time. Hence, these networks are ill-equipped to model task-processing scenarios where tasks that enter the network must be assigned and processed by exactly one agent. Instead, our approach passes volunteering requests around a network and uses an economics-inspired task-processing network game to determine how best to respond to these requests. The resulting volunteering policy is sensitive to both local processing requests and the presence of other agents on the network that can volunteer as well.

1.2 Connections to classical cooperation work

Within a biological organism, specialized organs cooperate with each other because they are mutually dependent. Each organ performs certain vital functions for the others because they perform vital functions for it. Likewise, related individuals in a family perform costly acts for each other in order to ensure the longevity of the family. However, cooperation among distantly related individuals may not be likely, and cooperation among unrelated individuals is apparently irrational. As shown by Hamilton [25], a cooperative act between two related individuals should be taken if the cost-to-benefit ratio of the act is less than their relatedness. However, this simple rule does not explain altruistic acts between two unrelated individuals.

Trivers [49] suggests that benefits via reciprocity can be a surrogate for benefits via relatedness. Hence, cooperation among unrelated individuals who are certain to interact in the future (e.g., altruism between two friends) may be explained by a pattern of reciprocity; selfless acts in the past may actually be an investment in reciprocal acts in the future. Motivated by this idea, Axelrod [5] develops precise behavioral protocols and shows in computer simulation that stable patterns of cooperative reciprocity are possible. Additionally, several studies [e.g., 14, 16, 17, 37] have documented the existence of these reciprocity protocols in nature. Modern theoretical work studies how unrelated individuals can be similarly coupled if they are forced to interact along vertices of a graph [33, 40, 41]. Nowak [39] summarizes these results and shows that sufficient conditions for cooperation in every case are described by a generalization of Hamilton's rule. In particular, an altruistic act is favorable when the cost-to-benefit ratio of the act is less than a measure of the likelihood that the two individuals will interact again. In this work, it is shown that this general rule extends to engineering examples as well and can be used in the design of distributed task-processing agents.

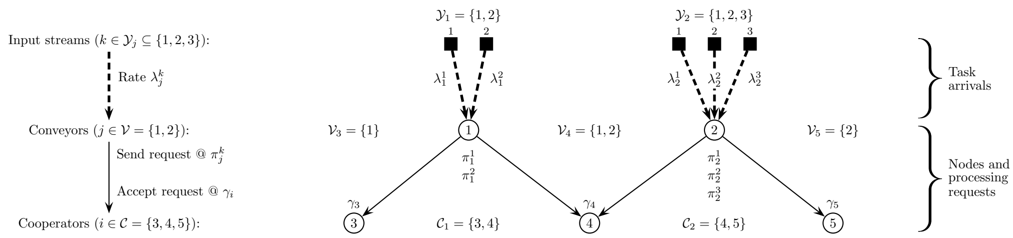


Fig. 2.1: Simple flexible manufacturing system example.

2 Task-processing networks

A *task-processing network* (TPN) models how connected agents can share the burden of processing tasks. Tasks arrive at individual agents that can process them at some cost to themselves (e.g., due to limited resources or material fatigue). In order to reduce the local task-processing cost, an agent can send requests to other nearby agents to process each task. At each request, those nearby agents can choose ignore the request or volunteer to process the corresponding task. Definition 2.1 below describes a generic TPN precisely, and two example TPNs follow it. In Section 3, the optimal ignore–accept mixed equilibrium is characterized, and in Section 4, a distributed and totally asynchronous algorithm is provided that is guaranteed to converge to this equilibrium.

Definition 2.1. (Task-processing network) Take a finite set $\mathcal{A} \subset \mathbb{N}$ of *task-processing agents* and a set $\mathcal{P} \subseteq \{(i,j) \in \mathcal{A}^2 : i \neq j\}$ of directed arcs connecting distinct agents. For each agent $i \in \mathcal{A}$,

$$\mathcal{V}_i \triangleq \{j \in \mathcal{A} : (j,i) \in \mathcal{P}\} \qquad \text{and} \qquad \mathcal{C}_i \triangleq \{j \in \mathcal{A} : (i,j) \in \mathcal{P}\}$$

are respectively the sets of *conveyors* and *cooperators* connected to agent i. Hence, $\mathcal{V} \triangleq \{j \in \mathcal{A} : \mathcal{C}_j \neq \emptyset\} = \bigcup_{i \in \mathcal{A}} \mathcal{V}_i$ and $\mathcal{C} \triangleq \{i \in \mathcal{A} : \mathcal{V}_i \neq \emptyset\} = \bigcup_{j \in \mathcal{A}} \mathcal{C}_j$ are respectively the sets of all conveyors and cooperators in the network. Assume that:

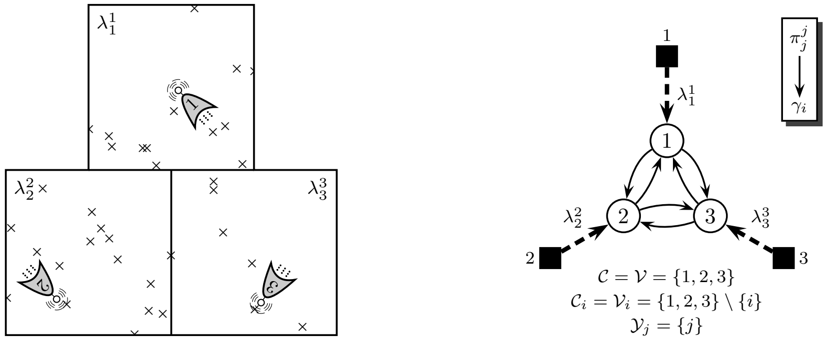
(i) For all $i \in \mathcal{A}$, there exists a finite and possibly empty set $\mathcal{Y}_i \subset \mathbb{N}$ of *task types* such that for all $k \in \mathcal{Y}_i$, tasks of type k arrive at agent i from an external source at average rate $\lambda_i^k \in \mathbb{R}_{>0}$. Each external source of tasks is assumed to be independent of all other sources.

(ii) If $j \in \mathcal{V}$, then there exist $k \in \mathcal{Y}_j$ with $\pi_j^k \neq 0$ where $\pi_j^k \in [0,1]$ represents the probability that conveyor j advertises an incoming k-type task to its connected cooperators $\mathcal{C}_j$. If $j \in \mathcal{V}$ does not advertise a task to its connected cooperators, the task will be processed by agent j.

(iii) If $i \in \mathcal{C}$, then there is some $\gamma_i \in [0,1]$ that represents the probability that agent i will volunteer for an advertised task from one of its connected conveyors $\mathcal{V}_i$. Any task arriving at conveyor $j \in \mathcal{V}$ that is advertised to cooperators $\mathcal{C}_j$ will be processed with uniform probability by exactly one of the cooperators that volunteer for it; if no cooperators volunteer for the task, then it is processed by conveyor j.

The graph $\mathcal{G} \triangleq (\mathcal{A}, \mathcal{P})$, rates, and probabilities defined above characterize a *task-processing network.*

The simple TPN shown in Fig. 2.1 represents a flexible manufacturing system (FMS) similar to the systems described by Perkins and Kumar [44]. Tasks of types 1, 2, and 3 arrive according to independent Poisson processes. Type-1 and type-2 tasks arrive at agent 1, and all three types of tasks arrive at agent 2. For tasks of type $k \in \mathcal{Y}_1 = \{1,2\}$, agent 1 advertises task arrivals



(a) AAV-patrolled territories. (b) AAV task-processing network.

Fig. 2.2: A task-processing network formed by three autonomous air vehicles (AAV).

to agents 3 and 4 with probability π_1^k. Likewise, agent 2 advertises arrivals of tasks of type $k \in \mathcal{Y}_2 = \{1, 2, 3\}$ to agents 4 and 5 with probability π_2^k. The system designer can choose different probabilities for each task type based on the specialized abilities of each agent. Each agent $i \in \{3, 4, 5\}$ volunteers for an advertised task with probability γ_i independent of task type. Hence, in this TPN, agents 1 and 2 are conveyors and agents 3, 4, and 5 are cooperators.

In the FMS example, the set of conveyors and the set of cooperators are disjoint. In a general TPN, an agent can be both a cooperator and a conveyor. For example, the fully-connected TPN shown in Fig. 2.2(b) models an autonomous air vehicle (AAV) patrol scenario shown in Fig. 2.2(a) that is similar to others in resource allocation literature [e.g., 19, 20, 22]. Each AAV $i \in \{1, 2, 3\}$ continuously searches its territory for tasks (e.g., targets) to process, and these tasks are generated (i.e., found) at rate $\lambda_i > 0$. When a task is found, the AAV advertises the task to both of its neighbors. If neither neighbor volunteers for processing, the AAV processes the task itself. In this fully-connected topology, all agents are both cooperators and conveyors. Although this network has several cycles, tasks do not move around the network — if a volunteering cooperator is given a task for processing, it cannot generate a new task-processing request for that task; it must process it itself.

Task-processing networks describe a broad range of applications. The AAV example above can also serve as a model of a mobile software agent [31, 32, 35, 46, 51] that patrols for tasks to process or any general group of networked processors [e.g., 21]. Additionally, by converting encounter rates to energetic rates (i.e., power demand), TPNs can model the behavior of smart power grids [26] made up of stations that request assistance from neighbors. That is, cooperator stations adjust additional supply provided in response to demand requests from remote conveyor stations.

3 Cooperation game among selfish agents

In a task-processing network, the probability (i.e., *cooperation propensity*) $\gamma_i \in [0, 1]$ that cooperator $i \in \mathcal{C}$ will volunteer for an advertised task from its connected conveyors must be chosen. It is assumed that this choice must be done in a distributed fashion and it is impractical for agents to coordinate in order to maximize some global utility. So each agent independently chooses a cooperation policy that maximizes its individual utility (i.e., agents are selfish). Hence, optimality is given in terms of the Nash equilibrium from Definition D.16 in Appendix D.

To inform each cooperator how to choose this policy, the network's designer assigns cost and rewards to agent operations in a common currency (e.g., proportional to dollars of net profit) that is called *points* here. In particular,

- Agent $i \in \mathcal{A}$ receives $(b_i^k - c_i^k)$ net points for processing a locally generated task of type $k \in \mathcal{Y}_i$.
- Conveyor $i \in \mathcal{V}$ receives r_i^k when a task of type $k \in \mathcal{Y}_i$ from i is processed by a $\mathcal{C}_i$ cooperator.
- If cooperator $j \in \mathcal{C}_i$ volunteers and is selected to process a task of type $k \in \mathcal{Y}_i$ from conveyor $i \in \mathcal{V}$, then cooperator j pays cost c_{ij}^k to process that task.

However, these costs and benefits alone do not provide cooperators with any incentive to volunteer to process conveyor tasks, and so an adaptive payment mechanism is required. Consider conveyor $j \in \mathcal{V}$ and task type $k \in \mathcal{Y}_j$. If one or more cooperators in $\mathcal{C}_j$ volunteer frequently to process requests from agent j, the other cooperators in the set should conserve resources by volunteering infrequently. To ensure this qualitative behavior, each cooperator $i \in \mathcal{C}_j$ receives volunteering payment $q_{ij}^k p_j^k(Q_j)$ from conveyor $j \in \mathcal{V}_i$ where:

- $Q_j \triangleq \sum_{k \in \mathcal{C}_j} \gamma_k$ is the *total quantity of cooperation propensity* available to conveyor j.
- $p_j^k(Q_j)$ is a decreasing *payment function* that represents the price that conveyor j pays to its connected cooperators each time they volunteer for a task of type $k \in \mathcal{Y}_j$.
- $q_{ij}^k \in \mathbb{R}_{>0}$ is a value factor that scales payment $p_j^k(Q_j)$ from conveyor j into the currency of cooperator $i \in \mathcal{C}_j$ (i.e., i perceives $q_{ij}^k p_j^k(Q_j)$ value from the contribution $p_j^k(Q_j)$ from j).

So if any cooperator $i \in \mathcal{C}_j$ increases its cooperation propensity γ_i, it increases how often it receives payment $p_j^k(Q_j)$ while also decreasing the payment itself. For each cooperator $i \in \mathcal{C}_j$, these two pressures encourage cooperation propensity (i.e., $\gamma_i > 0$) and resource conservation (i.e., $\gamma_i < 1$).

To maximize net points earned over a long run time, each agent chooses a policy that maximizes its own expected rate of point accumulation. So for a given vector $\underline{\gamma} = [\gamma_{c_1}, \gamma_{c_2}, \dots, \gamma_{c_{|\mathcal{C}|}}]^\top \in [0,1]^{|\mathcal{C}|}$ of cooperation policies (where unique $c_k \in \mathcal{C}$ for all $k \in \{1, 2, \dots, |\mathcal{C}|\}$), the utility (i.e., long-term rate of point gain) returned to cooperator $i \in \mathcal{C}$ is

$$U_i(\underline{\gamma}) \triangleq \underbrace{b_i + \overbrace{\left(1 - \prod_{j \in \mathcal{C}_i} (1 - \gamma_j)\right)}^{\Pr(\text{Volunteer from } \mathcal{C}_i | \text{Advertisement from } i)} r_i - Q_i p_i(Q_i)}_{\text{Conveyor part — constant with respect to } \gamma_i} + \underbrace{\gamma_i \sum_{j \in \mathcal{V}_i} \left(p_{ij}(Q_j) - \overbrace{\mathrm{SOBP}_1(\mathcal{C}_j \setminus \{i\})}^{\Pr(i \text{ awarded task from } j | i \text{ volunteers})} c_{ij}\right)}_{\text{Cooperator part — } \gamma_i \text{ and } Q_j \text{ vary with } \gamma_i} \tag{3.1}$$

where

$$\underbrace{\begin{aligned} b_i &\triangleq \sum_{k \in \mathcal{Y}_i} \lambda_i^k \left(b_i^k - c_i^k\right), \\ r_i &\triangleq \sum_{k \in \mathcal{Y}_i} \lambda_i^k \pi_i^k \left(r_i^k - \left(b_i^k - c_i^k\right)\right), \\ p_i(Q_i) &\triangleq \sum_{k \in \mathcal{Y}_i} \lambda_i^k \pi_i^k p_i^k(Q_i), \end{aligned}}_{\text{Costs and benefits of local processing on } i \in \mathcal{V}} \quad \text{and} \quad \underbrace{\begin{aligned} c_{ij} &\triangleq \sum_{k \in \mathcal{Y}_j} \lambda_j^k \pi_j^k c_{ij}^k, \\ p_{ij}(Q_j) &\triangleq \sum_{k \in \mathcal{Y}_j} \lambda_j^k \pi_j^k q_{ij}^k p_j^k(Q_j). \end{aligned}}_{\substack{\text{Costs and benefits to } i \in \mathcal{C} \\ \text{for volunteering for tasks from } j \in \mathcal{V}_i}} \tag{3.2}$$

The SOBP (i.e., the sum of binomial products) in Eq. (3.1) is defined in Eq. (C.3) from Appendix C. In particular, $\mathrm{SOBP}_1(\mathcal{C}_j \setminus \{i\})$ is the probability that cooperator i will be chosen to process an advertised task from conveyor j given that it volunteers for it. Hence, for $j \in \mathcal{V}_i$, the impact of cost rate c_{ij} decreases as other cooperators from $\mathcal{C}_j$ increase their own cooperation propensity because the probability that agent i will be selected decreases. So for a conveyor $j \in \mathcal{V}$, its connected cooperators $\mathcal{C}_j$ form a Cournot oligopoly [38] (i.e., a set of independent agents that provide a service for a demand-driven price) with a positive externality [6] (i.e., the cost of processing decreases as more cooperators enter the market). The underbraced *cooperator part* of the utility function shows that cooperator i must set its cooperation propensity γ_i (i.e., its quantity of supplied cooperation) based on the summed returns from several such markets.

4 Distributed computation of the Nash equilibrium

Let $n \triangleq |\mathcal{C}|$. Because there is no coordination between players, the n-dimensional play space is the Cartesian product $\prod_{i \in \mathcal{C}} [0,1] = [0,1]^n$, and the collection of cooperation policies across all cooperators is the vector $\underline{\gamma} \triangleq [\gamma_{c_1}, \gamma_{c_2}, \ldots, \gamma_{c_n}]^\top \in [0,1]^n$ (where unique $c_k \in \mathcal{C}$ for all $k \in \{1, 2, \ldots, n\}$). For each $i \in \mathcal{C}$, it is assumed that the utility function $U_i : [0,1]^n \mapsto \mathbb{R}$ is twice-continuously differentiable, and so, by Weirstrass' theorem, U_i is bounded above and below and achieves its extrema. Following Propositions D.9 and D.10 in Appendix D, the Nash equilibria of the cooperation game can be found by solving n separate one-dimensional variational inequality problems. In particular, $\underline{\gamma}^* \in [0,1]^n$ is a Nash equilibria of the cooperation game if and only if, for all $i \in \mathcal{C}$,

$$(\gamma_i - \gamma_i^*) \nabla_i U_i(\underline{\gamma}^*) \leq 0 \qquad \text{for all } \gamma_i \in [0,1] \tag{4.1}$$

where

$$\nabla_i U_i(\underline{\gamma}) \triangleq \frac{\partial U_i(\underline{\gamma})}{\partial \gamma_i} = \sum_{j \in \mathcal{V}_i} \Big(\overbrace{p_{ij}(Q_j) + \gamma_i p'_{ij}(Q_j)}^{\frac{\partial}{\partial \gamma_i}(\gamma_i p_{ij}(Q_j))} - \mathrm{SOBP}_1(\mathcal{C}_j \setminus \{i\}) c_{ij} \Big)$$

is the block gradient from Definition D.7. So in a local neighborhood of the Nash equilibrium $\gamma^* \in [0,1]^n$, any unilateral perturbation of a coordinate of γ^* will result in equal or reduced utility.

A closed-form solution to the constrained variational inequality problem in Eq. (4.1) is difficult to find in general. In particular, because the play space is a Cartesian product of 1-dimensional $[0,1]$ factor spaces, Eq. (4.1) is equivalent to the condition that for all $i \in \mathcal{C}$,

$$\begin{cases} \overbrace{\sum_{j \in \mathcal{V}_i} p_{ij}(Q_j^*)}^{\text{Marginal benefit of cooperation}} \leq \overbrace{\sum_{j \in \mathcal{V}_i} \mathrm{SOBP}_1^*(\mathcal{C}_j \setminus \{i\}) c_{ij}}^{\text{Marginal cost of cooperation}} & \overbrace{\text{if } \gamma_i^* = 0,}^{\text{Nash cooperation level}} \\ \sum_{j \in \mathcal{V}_i} \big(p_{ij}(Q_j^*) + \gamma_i^* p'_{ij}(Q_j^*) \big) = \sum_{j \in \mathcal{V}_i} \mathrm{SOBP}_1^*(\mathcal{C}_j \setminus \{i\}) c_{ij} & \text{if } \gamma_i^* \in (0,1), \\ \underbrace{\sum_{j \in \mathcal{V}_i} \big(p_{ij}(Q_j^*) + p'_{ij}(Q_j^*) \big)}_{\frac{\partial}{\partial \gamma_i} \left(\sum_{j \in \mathcal{V}_i} \gamma_i p_{ij}(Q_j) \right) \Big|_{\underline{\gamma} = \underline{\gamma}^*}} \geq \sum_{j \in \mathcal{V}_i} \underbrace{\mathrm{SOBP}_1^*(\mathcal{C}_j \setminus \{i\})}_{\Pr(i \text{ receives } j\text{'s task} | i \text{ volunteers})|_{\underline{\gamma} = \underline{\gamma}^*}} c_{ij} & \text{if } \gamma_i^* = 1 \end{cases} \tag{4.2}$$

where, for all $j \in \mathcal{V}$, Q_j^* and SOBP^* are respectively equivalent to Q_j and SOBP when $\underline{\gamma} = \underline{\gamma}^*$. The existence alone of a solution to the n simultaneous nonlinear equations of the form of Eq. (4.2) is not guaranteed in general. However, as discussed in Appendix D, variational inequalities over product spaces are well suited for parallel and asynchronous computation. Under special conditions on each utility function, a unique Nash equilibrium is guaranteed to exist, and each of its coordinates in Eq. (4.1) can be computed independently in the distributed and asynchronous fashion described by Assumption 4.1.

Assumption 4.1. (Totally asynchronous distributed iteration) *Take* $(c_1, c_2, \ldots, c_n) \triangleq \mathcal{C}$ *to represent the* n *distinct cooperators of* $\mathcal{C}$*. Let* $\mathcal{T} \triangleq \mathbb{W}$ *to be the indices of a sequence of physical times, and let* $\{\underline{\gamma}(t)\}_{t\in\mathcal{T}} \triangleq \{(\gamma_{c_1}(t), \gamma_{c_2}(t), \ldots, \gamma_{c_n}(t))\}$ *be a sequence of iterated calculations in the* $[0,1]^n$ *play space. For each* $i \in \mathcal{C}$*, subset* $\mathcal{T}^i \subseteq \mathcal{T}$ *corresponds to the times when coordinate* $\gamma_i(t)$ *is computed. Additionally, for each* $i, j \in \mathcal{C}$ *and each* $t \in \mathcal{T}$*, there is an index* $\tau_j^i(t) \in \mathcal{T}$ *of the least-outdated version of coordinate* γ_j *available for the computation of coordinate* γ_i *with transition mapping* $T_i : [0,1]^n \mapsto [0,1]$ *at time* t *such that* $0 \leq \tau_j^i(t) \leq t$*. That is, an outdated state estimate*

$$\underline{\gamma}^i(t) \triangleq (\gamma_{c_1}^i(t), \gamma_{c_2}^i(t), \ldots, \gamma_{c_n}^i(t)) \triangleq (\gamma_{c_1}(\tau_{c_1}^i(t)), \gamma_{c_2}(\tau_{c_2}^i(t)), \ldots, \gamma_{c_n}(\tau_{c_n}^i(t)))$$

is available for the computation $\gamma_i(t+1) = T_i(\underline{\gamma}^i(t))$ *for each* $t \in \mathcal{T}$ *and* $i \in \mathcal{C}$*. It is assumed that*

(i) *Set* $\mathcal{T}^i$ *is countably infinite (i.e.,* $|\mathcal{T}^i| = |\mathcal{T}| = |\mathbb{N}|$*) for all* $i \in \mathcal{C}$*.*

(ii) *If subsequence* $\{t_k\}$ *of* $\mathcal{T}^i$ *is such that* $\lim_{k\to\infty} t_k = \infty$*, then* $\lim_{k\to\infty} \tau_j^i(k) = \infty$ *for all* $i, j \in \{1, 2, \ldots, n\}$*. That is,* $\liminf_{t\to\infty} \tau_j^i(t) = \infty$ *for all* $i, j \in \{1, 2, \ldots, m\}$*.*

For all $t \in \mathcal{T}$*, sequence* $\{\underline{\gamma}(t)\}$ *is generated by the* totally asynchronous distributed iteration (TADI)

$$\gamma_i(t+1) \triangleq \begin{cases} T_i(\underline{\gamma}^i(t)) & \text{if } t \in \mathcal{T}^i, \\ \gamma_i(t) & \text{if } t \notin \mathcal{T}^i \end{cases} \tag{4.3}$$

where $\underline{\gamma}(t) \triangleq (\gamma_{c_1}(t), \gamma_{c_2}(t), \ldots, \gamma_{c_n}(t))$*.*

For each $i \in \mathcal{C}$, the transition mapping $T_i : [0,1]^n \mapsto [0,1]$ in Eq. (4.3) is defined by

$$T_i(\underline{\gamma}) \triangleq \min\{1, \max\{0, \gamma_i + \sigma_i \nabla_i U_i(\underline{\gamma})\}\}$$

for all $\underline{\gamma} \in [0,1]^n$ where $\sigma_i \in \mathbb{R}_{>0}$ is a *step size* that scales movement along the utility gradient $\nabla_i U_i$. The corresponding TADI-generated $\{\underline{\gamma}(t)\}$ sequence represents the collective motion of n self-interested agents that each climb their respective gradient in order to maximize their expected rate of point return. That is, Eq. (4.3) may be viewed as a dynamical system model of coupled agents that each take independent actions. For example, in the synchronous case,

$$\gamma_i(t+1) = \min\{1, \max\{0, \gamma_i(t) - \sigma_i \sum_{j\in\mathcal{V}_i} \text{SOBP}_1(\mathcal{C}_j \setminus \{i\}) c_{ij} + \underbrace{\sigma_i \sum_{j\in\mathcal{V}_i} \overbrace{\left(p_{ij}(Q_j) + \gamma_i(t) p'_{ij}(Q_j)\right)}^{u_{ij}(\underline{\gamma}) \triangleq \frac{\partial}{\partial \gamma_i} \gamma_i p_{ij}(Q_j)}}_{\triangleq u_i(\underline{\gamma}) = \sum_{j\in\mathcal{V}_i} u_{ij}(\underline{\gamma})}\}\}$$

for all $i \in \mathcal{C}$. Here, the underbraced payment expression $u_i(\underline{\gamma})$ may be viewed as a feedback control on the behavior of cooperator agent i. By Proposition C.6, there exists a constant $\underline{\text{SOBP}} > 0$ such that $\text{SOBP}_1(\Gamma) \geq \underline{\text{SOBP}}$ for all $\Gamma \subseteq \mathcal{C}$. So, assuming that $c_{ij} > 0$ for all $i, j \in \mathcal{A}$, the undriven response of the system (i.e., the response when $u_i \equiv 0$ for all $i \in \mathcal{C}$) reaches $\underline{\gamma}(T) = \underline{0}$ in some finite time $T \in \mathbb{W}$. That is, the intrinsic agent behavior is not to cooperate. For each $i \in \mathcal{C}$, it is desirable to find a control law $u_i : [0,1]^n \mapsto [0,1]$ that feeds forward the payment $\sum_{j\in\mathcal{V}_i} p_{ij}(Q_j(\underline{\gamma}))$ to destabilize the no-cooperation equilibrium and provides feedback $\gamma_i \sum_{j\in\mathcal{V}_i} p'_{ij}(Q_j(\underline{\gamma}))$ to stabilize the Nash equilibrium. Hence, payment is a control mechanism that both establishes and stabilizes cooperation.

4.1 Stabilizing payment functions

Under the control interpretation where, for all $i \in \mathcal{C}$, $u_i \triangleq \sum_{j\in\mathcal{V}_i} p_{ij}(Q_j) + \gamma_i \sum_{j\in\mathcal{V}_i} p'_{ij}(Q_j)$ is the sum of a feed-forward and a feedback control law, intuition suggests that a nontrivial Nash equilibrium can be stabilized by the control if, for each $j \in \mathcal{V}_i$, the nonlinear feedback gain $p'_{ij}(Q_j)$ is strictly negative everywhere with greater action at low cooperation levels. These conditions are made more precise by Definition 4.1 of a stabilizing payment function p_{ij} with $j \in \mathcal{V}_i$.

Definition 4.1. (Stabilizing payment function) For $k \in \mathbb{N}$, a *stabilizing payment function (SPF)* $p : [0, k] \mapsto \mathbb{R}$ is a twice-continuously-differentiable function such that:

(i) It is strictly decreasing. In particular, $p'(Q) \triangleq \mathrm{d}p(Q)/\mathrm{d}Q < 0$ for all $Q \in [0, k]$.

(ii) It is convex. In particular, $p''(Q) \triangleq \mathrm{d}^2 p(Q)/\mathrm{d}^2 Q \geq 0$ for all $Q \in [0, k]$.

(iii) Its convexity is eventually dominated by its slope. In particular,

$$\gamma p''(Q) \leq -p'(Q) \qquad \text{for all } Q \in [\gamma, k-(1-\gamma)] \text{ with } \gamma \in [0,1]. \tag{4.4}$$

Consider cooperator $i \in \mathcal{C}$ and a connected conveyor $j \in \mathcal{V}_i$. Under the control law interpretation, condition (i) guarantees that the nonlinear feedback gain $p'_{ij}(Q_j)$ is always negative. As shown in the proof of Proposition 4.1, the SPF conditions guarantee that the payment slope is bounded away from zero, which is equivalent to requiring that the negative feedback control law never vanishes. Likewise, condition (ii) of Definition 4.1 states that the feedback gain should relax as the total quantity $Q_j \triangleq \sum_{k\in\mathcal{C}_j} \gamma_k$ of cooperation increases, and condition (iii) ensures that the relaxation of the feedback p'_{ij} is sufficiently moderate. That is, condition (iii) of Definition 4.1 states that

$$\frac{\mathrm{d}}{\mathrm{d}\gamma_i}\Big(\overbrace{\gamma_i p'_{ij}(\gamma_i + \underbrace{(Q_j-\gamma_i)}_{\triangleq \kappa})}^{\text{Stabilizing feedback}}\Big) \leq 0 \qquad \text{for all } \gamma_i \in [0,1] \text{ and } \kappa \in [0, \underbrace{|\mathcal{C}_j|}_{\triangleq k} - 1].$$

Because p_{ij} is convex, the function $f_1(\kappa) \triangleq \gamma p'_{ij}(\gamma+\kappa)$ is increasing for any $\gamma > 0$. However, for any $\kappa \in [0, k-1]$, the continuous function $f_2(\gamma) \triangleq \gamma p'(\gamma+\kappa)$ is initially decreasing because $f_2(0) = 0$ and $f_2(\gamma) < 0$ for all $\gamma \in (0,1]$. The requirement in item (iii) is that f_2 be decreasing for all $\gamma \in [0,1]$ and all $\kappa \in [0, k-1]$. That is, the magnitude of the feedback control action should decelerate, but it should not decrease.

Proposition 4.1. (Non-vanishing negative feedback) *For $k \in \mathbb{N}$ and any stabilizing payment function $p : [0,k] \mapsto \mathbb{R}$, $p(0) > p(Q) > p(k)$ and $p'(0) \leq p'(Q) \leq p'(k) < 0$ for all $Q \in (0,k)$.*

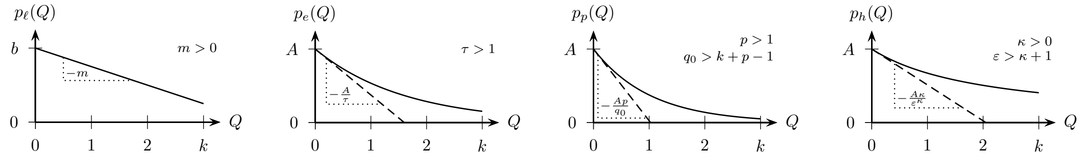
Proof of Proposition 4.1 given in Appendix B. □

As shown in Proposition B.1, the set of SPFs is closed under conical combinations (i.e., it is a filled cone). So for $i \in \mathcal{C}$, if p_{ij} is an SPF for all $j \in \mathcal{V}_i$, then the sum $\sum_{j\in\mathcal{V}_i} p_{ij}(Q_j)$ is itself an SPF. Additionally, by the definition of $p_{ij}(Q_j)$ in Eq. (3.2), if $p^k_j(Q_j)$ is an SPF for all $j \in \mathcal{V}$ and $k \in \mathcal{Y}_j$, then $p_{ij}(Q_j)$ will also be an SPF for all $i \in \mathcal{C}$.

Four example SPFs are shown in Fig. 4.1. Each payment function meets the simpler condition in Proposition 4.2; however, using the weaker condition (iii) of Definition 4.1, it is only necessary for $\varepsilon \geq \kappa$ in (d). Additionally, the polynomial function in (c) is an extension of the linear function in (a).

Proposition 4.2. (Sufficient conditions for payment stabilization) *Take $k \in \mathbb{N}$ and function $p : [0,k] \mapsto \mathbb{R}$. If $0 \leq p''(Q) < -p'(Q)$ for all $Q \in [0,k]$, then p is a stabilizing payment function.*

Proof of Proposition 4.2 given in Appendix B. □



(a) $b - mQ$ (b) $A\exp(-Q/\tau)$ (c) $A(1 - Q/q_0)^p$ (d) $A\varepsilon^\kappa/(\varepsilon + Q)^\kappa$

Fig. 4.1: Sample stabilizing payment (i.e., inverse-demand) functions.

4.2 Topological constraints

Ensuring that p_{ij} is an SPF for all $j \in \mathcal{V}_i$ does not guarantee that the n independent agents will achieve of a stable Nash equilibrium. As shown in Eq. (4.2), if the marginal cost of cooperation (MCC) varies greatly along TADI trajectories, then convergence to a unique Nash equilibrium may be impossible. However, if the Hessian of each agent's utility function meets certain diagonal dominance conditions, then the agent's progress in moving toward the Nash equilibrium will be dominated by its own actions, and the agent will consistently move in a productive direction.

The MCC associated with a given cooperator $i \in \mathcal{C}$ depends primarily upon the number of other cooperators connected to each conveyor $j \in \mathcal{V}_i$. If the topology of the task-processing network meets special conditions involving the set of cooperators connected to each conveyor, then there exists a tractable bound on the variation in the MCC. These conditions will be precisely specified in Theorem 4.1 using Definition 4.2.

Definition 4.2. (k-conveyor) Conveyor $i \in \mathcal{V}$ is a *k-conveyor* if it has exactly $k \in \mathbb{N}$ outgoing connections to cooperators (i.e., if $k = |\mathcal{C}_i|$).

4.3 Convergence result

Theorem 4.1 gives sufficient conditions for convergence to the Nash equilibrium.

Theorem 4.1. (Convergence to the Nash equilibrium of the cooperation game) *Assume that*

(i) *For all $i \in \mathcal{C}$ and $j \in \mathcal{V}_i$, p_{ij} is a stabilizing payment function.*

(ii) *For all $j \in \mathcal{V}$, $|\mathcal{C}_j| \leq 3$ (i.e., no conveyor can have more than 3 outgoing links to cooperators).*

(iii) *For $i \in \mathcal{C}$ and $j \in \mathcal{V}_i$, if j is a 3-conveyor, then there must be some $k \in \mathcal{V}_i$ that is a 2-conveyor.*

Define $T : [0,1]^n \mapsto [0,1]^n$ by $T(\underline{\gamma}) \triangleq (T_1(\underline{\gamma}), T_2(\underline{\gamma}), \ldots, T_n(\underline{\gamma}))$ where, for each $i \in \mathcal{C}$,

$$T_i(\underline{\gamma}) \triangleq \min\{1, \max\{0, \gamma_i + \sigma_i \nabla_i U_i(\underline{\gamma})\}\} \qquad \text{where } \frac{1}{\sigma_i} \geq 2|\mathcal{V}_i| \max_{k \in \mathcal{V}_i} |p'_{ik}(0)| \tag{4.5}$$

for all $\underline{\gamma} \in [0,1]^n$. If

$$\min_{j \in \mathcal{V}_i} |p'_{ij}(|\mathcal{C}_j|)| > \left(|\mathcal{V}_i| - \frac{1}{2}\right) \max_{j \in \mathcal{V}_i} |c_{ij}| \qquad \text{for all } i \in \mathcal{C}, \tag{4.6}$$

then the TADI sequence $\{\underline{\gamma}(t)\}$ generated with mapping T and the outdated estimate sequence $\{\underline{\gamma}^i(t)\}$ for all $i \in \mathcal{C}$ each converge to the unique Nash equilibrium of the cooperation game.

Proof of Theorem 4.1 given in Appendix B. □

The restriction in Eq. (4.6) is similar to the network generalization of Hamilton's rule [25] (i.e., benefit/cost > 1/relatedness where relatedness = 1/(average number of connections)) discussed by Ohtsuki et al. [41] and Nowak [39]. In particular, as the number of connected conveyors increases, a cooperator's *relatedness* to each of them decreases, and stable cooperation may require increased benefits (i.e., steeper payment slopes to dominate uncertain costs). Additionally, if σ_i is picked to satisfy Eq. (4.5) for each $i \in \mathcal{C}$, then Proposition B.2 gives a sufficient condition that simplifies Eq. (4.6). The complete proof of Theorem 4.1 is given in Appendix B. Most of the proof is a specialized combination of Propositions D.3 and D.4 that have proofs given by Bertsekas and Tsitsiklis [8]. However, the novel result is the relationship between the assumptions of Theorem 4.1 and the assumptions on which Propositions D.3 and D.4 are predicated, and so that relationship is discussed in detail here.

By assumption (i) (i.e., all payment functions are stabilizing), for any $\underline{\gamma} \in [0,1]^n$ and $i \in \mathcal{C}$,

$$\nabla^2_{ii} U_i(\underline{\gamma}) \triangleq \frac{\partial^2 U_i(\underline{\gamma})}{\partial {\gamma_i}^2} = \sum_{j \in \mathcal{V}_i} \big(2p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big) = \sum_{j \in \mathcal{V}_i} \overbrace{p'_{ij}(Q_j)}^{<0} + \sum_{j \in \mathcal{V}_i} \overbrace{\big(p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big)}^{\leq 0} < 0,$$

and

$$\begin{aligned}\nabla^2_{ii} U_i(\underline{\gamma}) &= \sum_{j \in \mathcal{V}_i} \big(2p'_{ij}(Q_j) + \overbrace{\gamma_i p''_{ij}(Q_j)}^{\geq 0}\big) \geq \sum_{j \in \mathcal{V}_i} 2p'_{ij}(Q_j) = -2 \sum_{j \in \mathcal{V}_i} |p'_{ij}(Q_j)| \\ &\geq -2 \sum_{j \in \mathcal{V}_i} \max_{k \in \mathcal{V}_i} |p'_{ik}(0)| = -2|\mathcal{V}_i| \max_{k \in \mathcal{V}_i} |p'_{ik}(0)| \geq -2|\mathcal{V}_i| \max_{k \in \mathcal{V}_i} |p'_{ik}(0)|. \end{aligned} \tag{4.7}$$

So, by the assumed limits on step size σ_i given in Eq. (4.5),

$$0 > \nabla^2_{ii} U_i(\underline{\gamma}) \geq -\frac{1}{\sigma_i} \qquad \text{or, equivalently,} \qquad 0 < |\nabla^2_{ii} U_i(\underline{\gamma})| \leq 2|\mathcal{V}_i| \max_{k \in \mathcal{V}_i} |p'_{ik}(0)| \leq \frac{1}{\sigma_i}$$

for all $i \in \mathcal{C}$. So the TADI step size and each agent's utility function's concavity are inversely related. For example, if a cooperator services a large number of incoming conveyors or if a cooperator is connected to a conveyor with a very steep payment function, then small perturbations in its level of cooperation will bring large changes in the amount of payment received. In this case, the cooperator must sample its utility gradient very finely by making only small changes in its cooperation level at each TADI step. The "2" is present in the bound in Eq. (4.7) because each payment function controls the utility gradient through sum of both feed-forward p_{ik} and feedback $\gamma_i p'_{ik}$ payment, and hence the curvature is twice affected by the payment slope.

As discussed, to ensure a kind of diagonal dominance of each agent's utility Hessian (i.e., the Jacobian of each utility gradient), the topology of the task-processing network must be limited. So take $\underline{\gamma} \in [0,1]^n$ and cooperator $i \in \mathcal{C}$. For another cooperator $\ell \in \mathcal{C} \setminus \{i\}$, if $\ell \notin \mathcal{C}_j$ (i.e., ℓ is not an outgoing cooperator for j), then $\partial Q_j / \partial \gamma_\ell = 0$ and $\partial \, \mathrm{SOBP}_1(\mathcal{C}_j - \{i\}) / \partial \gamma_\ell = 0$ where $Q_j \triangleq \sum_{k \in \mathcal{C}_j} \gamma_k$ and SOBP is from Definition C.1. So by introducing SOMS from Proposition C.11,

$$\begin{aligned} 0 \leq \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} |\nabla^2_{i\ell} U_i(\underline{\gamma})| &\triangleq \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left| \frac{\partial^2 U_i(\underline{\gamma})}{\partial \gamma_i \partial \gamma_\ell} \right| \\ &= \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left| \sum_{j \in \mathcal{V}_i} [\ell \in \mathcal{C}_j] \left(p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j) + \overbrace{\mathrm{SOMS}_2(\mathcal{C}_j \setminus \{i, \ell\})}^{\partial/\partial \gamma_\ell \, \mathrm{SOBP}_1(\mathcal{C}_j - \{i\})} c_{ij} \right) \right| \end{aligned}$$

where $[\cdot]$ is the Iverson bracket [27]. That is, for a propositional statement S, $[S] = 1$ if S is true, and $[S] = 0$ otherwise. As described by Knuth [29], introducing the Iverson bracket here will allow the restriction on index set to be manipulated algebraically. Hence,

$$\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} |\nabla^2_{i\ell} U_i(\underline{\gamma})| \leq \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \sum_{j \in \mathcal{V}_i} [\ell \in \mathcal{C}_j] \left(\underbrace{|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)|}_{\leq 0} + |\mathrm{SOMS}_2(\mathcal{C}_j \setminus \{i, \ell\})| \, |c_{ij}| \right).$$

By Propositions C.14 and C.15, $0 < \mathrm{SOMS}_2(\Gamma) \leq 1/2$ for all $\Gamma \subseteq \mathcal{C}$, and so

$$\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left|\nabla^2_{i\ell} U_i(\underline{\gamma})\right| \leq \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \sum_{j \in \mathcal{V}_i} [\ell \in \mathcal{C}_j] \left(\left|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right| + \frac{1}{2}|c_{ij}| \right).$$

Furthermore, because these two finite sums can be transposed,

$$\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left|\nabla^2_{i\ell} U_i(\underline{\gamma})\right| \leq \sum_{j \in \mathcal{V}_i} \left(\left|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right| + \frac{1}{2}|c_{ij}| \right) \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} [\ell \in \mathcal{C}_j].$$

Hence, the second sum is a count of all elements in $(\mathcal{C} \setminus \{i\}) \cap \mathcal{C}_j$. That is,

$$\begin{aligned}
\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left|\nabla^2_{i\ell} U_i(\underline{\gamma})\right| &\leq \sum_{j \in \mathcal{V}_i} \left(\left|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right| + \frac{1}{2}|c_{ij}| \right) \underbrace{|\{\ell \in \mathcal{C} : \ell \in \mathcal{C}_j \setminus \{i\}\}|}_{\substack{\text{Number of non-}i\text{ cooperators} \\ \text{connected to } j}} \\
&= \sum_{j \in \mathcal{V}_i} \left(\left|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right| + \frac{1}{2}|c_{ij}| \right) |\mathcal{C}_j \setminus \{i\}| \,,
\end{aligned}$$

and, because $j \in \mathcal{V}_i$ if and only if $i \in \mathcal{C}_j$,

$$\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left|\nabla^2_{i\ell} U_i(\underline{\gamma})\right| \leq \sum_{j \in \mathcal{V}_i} \left(\left|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right| + \frac{1}{2}|c_{ij}| \right) (|\mathcal{C}_j| - 1) \,.$$

However, by assumption (ii), each conveyor $j \in \mathcal{V}$ has no more than three outgoing connections to cooperators (i.e., $|\mathcal{C}_j| \leq 3$). Additionally, by assumption (iii), if $j \in \mathcal{V}_i$ is a 3-conveyor (i.e., it has 3 outgoing cooperator connections), then there must be some other conveyor $m \in \mathcal{V}_i \setminus \{j\}$ that is a 2-conveyor. So letting $m \in \mathcal{V}_i$ be the 2-conveyor that is guaranteed to exist,

$$\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left|\nabla^2_{i\ell} U_i(\underline{\gamma})\right| \leq \overbrace{2 \sum_{j \in \mathcal{V}_i \setminus \{m\}} \left(\underbrace{\left|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right|}_{\leq 0} + \frac{1}{2}|c_{ij}| \right)}^{\substack{\text{Doubled contribution to sum from} \\ \text{other cooperators connected to} \\ \text{assumed 3-conveyors in } \mathcal{V}_i \setminus \{m\}}} + \overbrace{\underbrace{\left|p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)\right|}_{\leq 0} + \frac{1}{2}|c_{im}|}^{\substack{\text{Contribution to sum from} \\ \text{other cooperator of 2-conveyor } m \in \mathcal{V}_i}}.$$

Because of item (iii) in Definition 4.1 of an SPF,

$$\begin{aligned}\sum_{\substack{\ell\in\mathcal{C}\\ \ell\neq i}}\left|\nabla^2_{i\ell}U_i(\underline{\gamma})\right| &\le \sum_{j\in\mathcal{V}_i\setminus\{m\}}\Big(-\left(2p'_{ij}(Q_j)+2\gamma_i p''_{ij}(Q_j)\right)+|c_{ij}|\Big)-\left(p'_{im}(Q_m)+\gamma_i p''_{im}(Q_m)\right)+\frac{1}{2}|c_{im}|\\ &=\sum_{j\in\mathcal{V}_i\setminus\{m\}}\Big(-\left(2p'_{ij}(Q_j)+\gamma_i p''_{ij}(Q_j)\right)+|c_{ij}|\Big)-\overbrace{\sum_{j\in\mathcal{V}_i\setminus\{m\}}\gamma_i p''_{ij}(Q_j)}^{\ge 0}\\ &\qquad-\left(p'_{im}(Q_m)+\gamma_i p''_{im}(Q_m)\right)+\frac{|c_{im}|}{2},\end{aligned}$$

and so, due to the convexity of stabilizing payment functions,

$$\sum_{\substack{\ell\in\mathcal{C}\\ \ell\neq i}}\left|\nabla^2_{i\ell}U_i(\underline{\gamma})\right| \le -\sum_{j\in\mathcal{V}_i\setminus\{m\}}\left(2p'_{ij}(Q_j)+\gamma_i p''_{ij}(Q_j)\right)-\left(p'_{im}(Q_m)+\gamma_i p''_{im}(Q_m)\right)+\sum_{j\in\mathcal{V}_i\setminus\{m\}}|c_{ij}|+\frac{|c_{im}|}{2}.$$

Because $\mathcal{A}$ is finite, $\mathcal{V}_i\subseteq\mathcal{A}$ is finite, and so

$$\begin{aligned}\sum_{\substack{\ell\in\mathcal{C}\\ \ell\neq i}}\left|\nabla^2_{i\ell}U_i(\underline{\gamma})\right| &\le -\sum_{j\in\mathcal{V}_i\setminus\{m\}}\left(2p'_{ij}(Q_j)+\gamma_i p''_{ij}(Q_j)\right)-\left(p'_{im}(Q_m)+\gamma_i p''_{im}(Q_m)\right)\\ &\qquad+\left(|\overbrace{\mathcal{V}_i\setminus\{m\}}^{m\in\mathcal{V}_i}|+\frac{1}{2}\right)\max_{j\in\mathcal{V}_i}|c_{ij}|\\ &=-\sum_{j\in\mathcal{V}_i\setminus\{m\}}\left(2p'_{ij}(Q_j)+\gamma_i p''_{ij}(Q_j)\right)-\left(p'_{im}(Q_m)+\gamma_i p''_{im}(Q_m)\right)+\left(|\mathcal{V}_i|-\frac{1}{2}\right)\max_{j\in\mathcal{V}_i}|c_{ij}|,\end{aligned}$$

and

$$\sum_{\substack{\ell\in\mathcal{C}\\ \ell\neq i}}\left|\nabla^2_{i\ell}U_i(\underline{\gamma})\right| \le -\sum_{j\in\mathcal{V}_i\setminus\{m\}}\left(2p'_{ij}(Q_j)+\gamma_i p''_{ij}(Q_j)\right)-\left(p'_{im}(Q_m)+\gamma_i p''_{im}(Q_m)\right)+\left(|\mathcal{V}_i|-\frac{1}{2}\right)\max_{j\in\mathcal{V}_i}|c_{ij}|.$$

By expanding the summation's index set to include $m\in\mathcal{V}_i$ and subtracting the new contribution,

$$\begin{aligned}\sum_{\substack{\ell\in\mathcal{C}\\ \ell\neq i}}\left|\nabla^2_{i\ell}U_i(\underline{\gamma})\right| \le -\sum_{j\in\mathcal{V}_i}&\left(2p'_{ij}(Q_j)+\gamma_i p''_{ij}(Q_j)\right)+\left(2p'_{im}(Q_m)+\gamma_i p''_{im}(Q_m)\right)\\ &-\left(p'_{im}(Q_m)+\gamma_i p''_{im}(Q_m)\right)+\left(|\mathcal{V}_i|-\frac{1}{2}\right)\max_{j\in\mathcal{V}_i}|c_{ij}|,\end{aligned}$$

and by canceling some of the resulting terms,

$$\sum_{\substack{\ell\in\mathcal{C}\\ \ell\neq i}}\left|\nabla^2_{i\ell}U_i(\underline{\gamma})\right| \le -\overbrace{\sum_{j\in\mathcal{V}_i}\left(2p'_{ij}(Q_j)+\gamma_i p''_{ij}(Q_j)\right)}^{\nabla^2_{ii}U_i(\underline{\gamma})}+\overbrace{p'_{im}(Q_m)}^{<0}+\left(|\mathcal{V}_i|-\frac{1}{2}\right)\max_{j\in\mathcal{V}_i}|c_{ij}|$$

where the Hessian diagonal term $\nabla^2_{ii} U_i(\underline{\gamma})$ is evident. So

$$\begin{aligned}
\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left| \nabla^2_{i\ell} U_i(\underline{\gamma}) \right| &\leq -\nabla^2_{ii} U_i(\underline{\gamma}) - |p'_{im}(Q_m)| + \left(|\mathcal{V}_i| - \frac{1}{2} \right) \max_{j \in \mathcal{V}_i} |c_{ij}| \\
&\leq -\nabla^2_{ii} U_i(\underline{\gamma}) - \min_{j \in \mathcal{V}_i} |p'_{ij}(Q_j)| + \left(|\mathcal{V}_i| - \frac{1}{2} \right) \max_{j \in \mathcal{V}_i} |c_{ij}| \\
&= -\nabla^2_{ii} U_i(\underline{\gamma}) - \underbrace{\left(\min_{j \in \mathcal{V}_i} |p'_{ij}(Q_j)| - \left(|\mathcal{V}_i| - \frac{1}{2} \right) \max_{j \in \mathcal{V}_i} |c_{ij}| \right)}_{> 0 \text{ by Eq. (4.6)}},
\end{aligned}$$

and, by the assumption in Eq. (4.6), the underbraced expression is strictly positive. Hence,

$$\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left| \nabla^2_{i\ell} U_i(\underline{\gamma}) \right| < -\nabla^2_{ii} U_i(\underline{\gamma}). \tag{4.8}$$

Because $\nabla^2_{ii} U_i(\underline{\gamma}) < 0$, Eq. (4.8) states that $|\nabla^2_{ii} U_i(\underline{\gamma})| > \sum_{\ell \in \mathcal{C}, \ell \neq i} \left| \nabla^2_{i\ell} U_i(\underline{\gamma}) \right|$. So assumptions (ii) and (iii) and Eq. (4.6) ensure strict diagonal dominance in the i^{th} row of each utility Hessian. For each agent, its corresponding utility function is not only concave along its play dimension, but its concavity dominates the curvature along any other direction.

The strict diagonal dominance of the i^{th} row of the utility Hessian also implies properties of the TADI dynamical system. Precisely, the Jacobian of vector-valued function $[\nabla_{c_1} U_{c_1}, \nabla_{c_2} U_{c_2}, \ldots, \nabla_{c_n} U_{c_n}]^\top$ (where unique $c_k \in \mathcal{C}$ for all $k \in \{1, 2, \ldots, n\}$) is strictly row diagonally dominant. Moreover, local stability of the synchronous approximation of the TADI follows from the stability of this Jacobian linearization. Using the payment-as-control interpretation, these conditions ensure controllability. In particular, Eq. (4.6) is equivalent to

$$\overbrace{\min_{j \in \mathcal{V}_i} 2|p'_{ij}(|\mathcal{C}_j|)|}^{\substack{\text{Maximum variation} \\ \text{of payment control} \\ \text{due to self movement}}} > \overbrace{\Bigg(\underbrace{2|\mathcal{V}_i| - 1}_{\substack{\text{Twice the number of 3-conveyors} \\ \text{plus a 2-conveyor}}} \Bigg) \max_{j \in \mathcal{V}_i} |c_{ij}|}^{\substack{\text{Maximum variation of intrinsic cost} \\ \text{due to movement of others}}} \qquad \text{for all } i \in \mathcal{C}. \tag{4.9}$$

The "2" in the expression on the left-hand side of Eq. (4.9) reflects the dual payment action from feed-forward and feedback controls, and the overbraced expression on the right-hand side reflects the bound on the 3-conveyor-caused dual change and the 2-conveyor-caused single change in SOBP that is guaranteed by assumptions (ii) and (iii). So the number of incoming conveyors $\max_{i \in \mathcal{C}} |\mathcal{V}_i|$ or the magnitude of cooperation cost $\max_{i \in \mathcal{C}, j \in \mathcal{V}_i} |c_{ij}|$ can only be increased if the minimum payment slope $\min_{j \in \mathcal{V}_i} |p'_{ij}(\mathcal{C}_j)|$ can also be increased. If sufficiently large feedback gain (i.e., $\sum_{j \in \mathcal{C}_i} p'_{ij}$) is unavailable, the evolution of the TADI trajectory may be too sensitive to cost variations (i.e., the payment signal may fail to dominate the intrinsic cost signal).

4.4 Finding necessary conditions on network topology for stabilization

As shown in the derivation of Eq. (4.9), every 3-conveyor contributes two payment slope p'_{ij} terms to $\nabla^2_{i\ell} U_i$ that are cancelled by the two slope terms in $\nabla^2_{ii} U_i$. Hence, when 3-conveyors are connected to a cooperator, the cooperator loses control of its utility gradient along its cooperation coordinate unless there exists a 2-conveyor that it can dominate. So 2-conveyors are themselves stabilizers that allow a cooperator $i \in \mathcal{C}$ to focus its decision making on the conveyors in $\mathcal{V}_i$ for

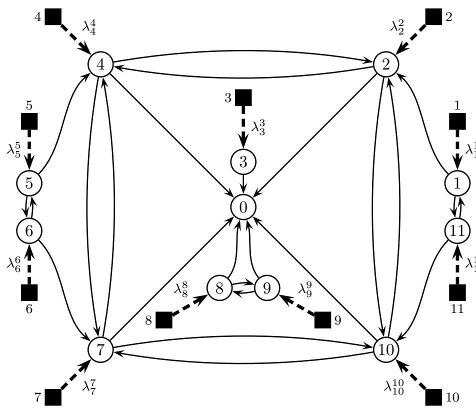


Fig. 4.2: Many-agent task-processing network with stable topology.

which there is only one other cooperator competing for payment. For example, in the complex TPN in Fig. 4.2, the 3-conveyors in the network (e.g., 2, 4, 7, and 10) could destabilize the gradient ascent if the 2-conveyors (e.g., 1, 5, 6, 8, 9, and 11) were not also present. It may be possible to weaken Theorem 4.1 to allow for conveyors with $n > 3$ outgoing connections to cooperators so long as the slopes of the n-conveyor payment functions can be dominated by those of other 1-conveyors.

5 Example: Simulation of cooperative AAV scenario

Consider an AAV scenario like the one shown in Fig. 2.2. Assume that $\pi_i^k = 1$, $c_{ij}^\ell = 0.1$, and $q_{ij}^\ell = 1$ for all $i \in \mathcal{A}$, $j \in \mathcal{A} \setminus \{i\}$, $k \in \mathcal{Y}_i$, and $\ell \in \mathcal{Y}_j$. Also assume that $\lambda_1^1 = 0.6$, $\lambda_3^3 = 1.7$, $0 < \lambda_2 \leq 5$, and the linear payment function $p_i^i(Q_i) \triangleq 1 - Q_i/\lambda_i^i$ for all $i \in \mathcal{A}$. Hence, the three otherwise equivalent agents face different task encounter rates, and their payment functions have slopes that are inversely proportional to each encounter rate. So agents associated with higher encounter rates have a higher demand for cooperation and thus have inelastic payment functions (i.e., cooperation retains its high value even when a high quantity is available).

A conservative choice of step size $\sigma_\ell \triangleq 1/(4 \max_{i \in \mathcal{A}, j \in \mathcal{V}_i} p_{ij}'([0,0,0]^\top)$ for all $\ell \in \mathcal{A}$ yields a convergent TADI for this scenario. MATLAB simulation results summarized in Fig. 5.1 show how the resulting Nash equilibrium $\underline{\gamma}^*$ depends upon the AAV encounter rates. In particular, the Nash equilibrium has the desirable feature that $\lambda_i > \lambda_j$ implies that $\gamma_i^* < \gamma_j^*$ for all $i, j \in \mathcal{A}$. So agents that are locally busy are less willing to cooperate, and agents that are relatively idle are more willing to cooperate. In Fig. 5.1, as λ_2 increases, payment function p_2 to agents 1 and 3 becomes shallower and causes the optimal γ_1^* and γ_3^* to increase. However, as γ_1^* (or γ_3^*) increases, payment $p_3(Q_3)$ (or $p_1(Q_1)$) to agent 2 is depressed and γ_2^* decreases. Moreover, at point b when the ascent of γ_1^* truncates, the rate that γ_2^* decreases shallows. At point c when γ_3^* also truncates, the γ_2^* graph flattens entirely. Hence, to reduce the load on the saturated cooperators, agent 2 reciprocates for their cooperation by not reducing its own cooperation level to zero. So even though each agent's own encounter rate has no direct relationship to its TADI-directed movement, a desirable coupling between encounter rates and optimal cooperation levels emerges.

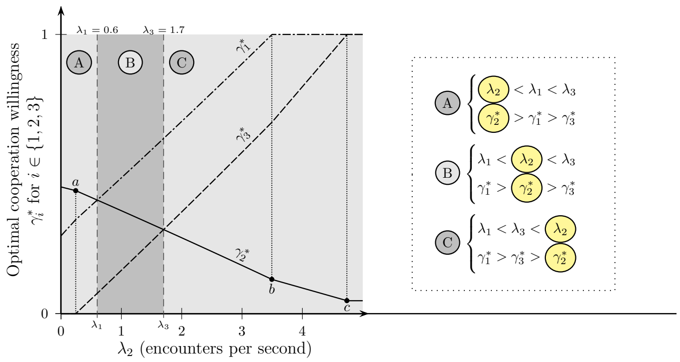


Fig. 5.1: AAV optimal cooperation propensity as encounter rates vary.

6 Conclusion

A framework for cooperative task processing on a network has been presented. Using this framework, a particular totally asynchronous cooperative control policy was shown to stabilize the Nash equilibrium of a cooperation game. By introducing a cooperation-trading economy into the formulation, the agents individually climb their own local utility functions yet still achieve an equilibrium where task processing is shared among different agents. The present work adjusts each agent's overall cooperation propensity in order to maximize economic returns over a lifetime of task encounters and processing requests. Future work should address the case where each agent associates a different cooperation propensity with each of its connected conveyors. Likewise, forwarding probabilities could be considered to be decision variables that should be adjusted across each agent's connected cooperators. The present work associates only one decision variable with each distributed agent, and so it makes the simplifying assumption that those variables come from a Cartesian product space. However, if future frameworks place multiple decision variables on a single distributed agent, that assumption may be relaxed.

A weakness of the present work is that it implicitly assumes that agents either have infinite processing capacity or that all tasks have negligible processing time. Processing and switching durations are central motivations for the work of Perkins and Kumar [44] just as finite capacity motivates the work of Cruz [15]. These effects can incorporated by explicitly modeling the average processing time of each task. In particular, the present work optimizes the long-term rate of gain of each agent based on rewards issued at the instant each task arrives, and this rate will be depressed by the processing time of each task. Moreover, the time spent processing a task represents a opportunity cost due to the lost time available for encountering other tasks that return higher profit. Because each arrival is independent, the average reward results of Johns and Miller [28] for Markov renewal–reward processes can be used to model the long-term rate of gain in this case. Hence, the utility functions discussed in this work can be easily modified to include these effects. If analytically tractable, optimal results can be found that account for appreciable processing times.

A Mathematical symbols and notation

$\equiv$	identically equal
$\approx$	approximately equal
$\triangleq$	defined as
$\subset$ ($\supset$)	strict (i.e., not equal) subset (superset) of
$\subseteq$ ($\supseteq$)	subset (superset) of
$\{\ldots : P\}$	a subset of set $\{\ldots\}$ where predicate P holds for every element of the set
$\emptyset$	empty set (i.e., the null set $\{\}$)
$\mathcal{X}$ ($\mathscr{X}$)	set $\mathcal{X}$ (set of sets $\mathscr{X}$)
$\wp(\mathcal{X})$	power set of $\mathcal{X}$ (i.e., $\{\mathcal{Y} : \mathcal{Y} \subseteq \mathcal{X}\}$)
$\mathcal{X} \setminus \mathcal{Y}$	set difference (i.e., all elements of $\mathcal{Y}$ removed from $\mathcal{X}$)
$\lvert\mathcal{X}\rvert$	cardinality (i.e., generalization of size) of set $\mathcal{X}$
$[S]$	Iverson bracket [27, 29] for statement S (i.e., $[S] = 1$ or 0 if S is true or false)
inf (sup)	infimum (supremum) — least upper (greatest lower) bound
$\limsup \mathcal{X}$	superior/outer limit of set $\mathcal{X}$ (e.g., $\inf_{n\to\infty} \sup_{m\geq n} S_m$ for sequence (S_m))
$\liminf \mathcal{X}$	inferior/inner limit of set $\mathcal{X}$ (e.g., $\sup_{n\to\infty} \inf_{m\geq n} S_m$ for sequence (S_m))
$\lim \mathcal{X}$	limit of set $\mathcal{X}$ (i.e., value upon which $\liminf \mathcal{X}$ and $\limsup \mathcal{X}$ agree)
$\overline{\mathcal{X}}$	closure of set $\mathcal{X}$ (i.e., limit points of the set)
$\mathbb{N}$	natural numbers (i.e., $\{1, 2, \ldots\}$)
$\mathbb{W}$	whole numbers (i.e., $\{0, 1, 2, \ldots\}$)
$\mathbb{Z}$	integers (i.e., $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$)
$\mathbb{Q}$	rational numbers (i.e., $\{p/q : p \in \mathbb{Z}, q \in \mathbb{N}\}$)
$\mathbb{R}$	real numbers (i.e., $\overline{\mathbb{Q}}$)
$\mathbb{R}_{\leq 0}$ ($\mathbb{R}_{\geq 0}$)	non-positive (non-negative) real numbers
$\mathbb{R}_{<0}$ ($\mathbb{R}_{>0}$)	strictly negative (positive) real numbers
$\overline{\mathbb{R}}$	extended real numbers (i.e., compactification $\mathbb{R} \cup \{-\infty, \infty\}$)
$\to$	tends to (i.e., denotes a limit)
$\underline{x}^\top$ ($A^\top$)	transpose of vector $\underline{x}$ (matrix A)
$\lVert\underline{x}\rVert_p$ ($\lVert A\rVert_p$)	(induced) p-norm of vector $\underline{x}$ (matrix A)
$\mathcal{B}_\varepsilon(\underline{x}^*)$	in normed vector space $\mathcal{X}$, open ball $\{\underline{x} \in \mathcal{X} : \lVert\underline{x} - \underline{x}^*\rVert < \varepsilon\}$
$\mathcal{B}^p_\varepsilon(\underline{x}^*)$	p-ball (i.e., open ball $\mathcal{B}_\varepsilon(\underline{x}^*)$ under p-norm $\lVert\cdot\rVert_p$)
$\mathcal{B}^\infty_\varepsilon(\underline{x}^*)$	∞-ball (i.e., open ball $\mathcal{B}_\varepsilon(\underline{x}^*)$ under maximum norm $\lVert\cdot\rVert_\infty$)
$f : \mathcal{X} \mapsto \mathcal{Y}$	function from $\mathcal{X}$ into $\mathcal{Y}$
f' ($f'(x_0)$)	first derivative $\mathrm{d}f/\mathrm{d}x$ (at point $x_0 \in \overline{\mathcal{X}}$) of function $f : \mathcal{X} \mapsto \mathcal{Y}$ where $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}$
f'' ($f''(x_0)$)	second derivative $\mathrm{d}^2 f/\mathrm{d}x^2$ (at point $x_0 \in \overline{\mathcal{X}}$) of function $f : \mathcal{X} \mapsto \mathcal{Y}$ where $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}$
∇f	gradient (transposed Jacobian) of scalar-valued (vector-valued) function f
$\nabla^2 f$	Hessian of scalar-valued function f (i.e., $\nabla\nabla f$)
$\nabla_i f$	For $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n$, i^{th} block gradient of scalar-valued function f
$\nabla^2_{ij} f$	$\nabla_i \nabla_j f$ (when $\mathcal{X}_i \subseteq \mathbb{R}$ for all i, $\nabla^2_{ij} f$ is i^{th} row and j^{th} column of Hessian $\nabla^2 f$)

B Proofs of central results

B.1 Properties of stabilizing payment functions

Proof of Proposition 4.1. Stabilizing payment function p is differentiable, defined over a compact set, and strictly decreasing, and so the bounds on p are clear by Weierstrass' theorem. Payment slope p' is differentiable, defined over a compact set, and nondecreasing, and so p' is bounded by $p'(0)$ and $p'(k)$. However, because p is strictly decreasing, $p'(k) < 0$. □

Proof of Proposition 4.2. Function p is strictly decreasing and convex. Additionally, because $\gamma \in [0,1]$, then $\gamma p''(Q) \leq p'(Q)$ for all $Q \in [0,k]$. So all conditions of Definition 4.1 are met. □

Proposition B.1. (Conical combinations of stabilizing payment functions) *Take $k \in \mathbb{N}$ and a set of stabilizing payment functions $\{p_1, p_2, \dots, p_m\}$ where $p_j : [0,k] \mapsto \mathbb{R}$ for all $j \in \{1,2,\dots,m\}$. Assume that there are constants $\{\alpha_1, \alpha_2, \dots, \alpha_m\}$ where*

(i) *$\alpha_i \in \mathbb{R}_{\geq 0}$ for all $i \in \{1,2,\dots,m\}$.*

(ii) *There is some $j \in \{1,2,\dots,m\}$ such that $\alpha_j > 0$.*

Then the nontrivial conical combination $p : [0,k] \mapsto \mathbb{R}$ defined by

$$p(Q) \triangleq \alpha_1 p_1(Q) + \alpha_2 p_2(Q) + \cdots + \alpha_m p_m(Q)$$

for all $Q \in [0,k]$ is also a stabilizing payment function.

Proof of Proposition B.1. The function p is clearly twice-continuously differentiable as it is a linear combination of twice-continuously-differentiable functions. To be a stabilizing payment function, p must satisfy requirements (i), (ii), and (iii) of Definition 4.1.

(i) Assume that $i \in \{1,2,\dots,m\}$ is such that $\alpha_i > 0$. Then, for each $Q \in [0,k]$,

$$p'(Q) = \alpha_1 p_1'(Q) + \alpha_2 p_2'(Q) + \cdots + \alpha_i p_i'(Q) + \cdots + \alpha_n p_n'(Q) \leq \alpha_i p_i'(Q) < 0,$$

and so p is strictly decreasing.

(ii) For each $Q \in [0,k]$,

$$p''(Q) = \alpha_1 p_1''(Q) + \alpha_2 p_2''(Q) + \cdots + \alpha_n p_n''(Q) \geq 0,$$

and so p is a convex function.

(iii) For each $\gamma \in [0,1]$ and each $Q \in [\gamma, k-(1-\gamma)]$,

$$\begin{aligned} p'(Q) + \gamma p''(Q) &= \alpha_1 p_1'(Q) + \cdots + \alpha_n p_n'(Q) + \gamma\left(\alpha_1 p_1''(Q) + \cdots + \alpha_n p_n''(Q)\right) \\ &= \alpha_1\left(p_1'(Q) + \gamma p_1''(Q)\right) + \cdots + \alpha_n\left(p_n'(Q) + \gamma p_n''(Q)\right) \leq 0, \end{aligned}$$

and so $\gamma p''(Q) \leq -p'(Q)$. □

B.2 Nash convergence in the cooperation game

What follows is a complete proof of Theorem 4.1, which is the principal result of this work. As discussed in Section 4, the proof uses numerical constraints on the distributed algorithm and topological constraints on the task-processing network to restrict aspects of the curvature of each agent's utility function. Those bounds can be combined with results from Bertsekas and Tsitsiklis [8] to show asymptotic convergence to the Nash equilibrium of the cooperation game. In particular, provided that the utility function is so constrained, the desired result follows almost entirely from Propositions D.3 and D.4 that are included in Appendix D along with other supporting material from Bertsekas and Tsitsiklis [8]. To aid in the interpretation of the predicated constraints, specialized versions of those proofs are applied in line with this proof.

Proof of Theorem 4.1. Define a mapping $R : [0,1]^n \mapsto \mathbb{R}^n$ by $R(\underline{\gamma}) \triangleq (R_1(\underline{\gamma}), R_2(\underline{\gamma}), \ldots, R_n(\underline{\gamma}))$ where, for each $i \in \{1, 2, \ldots, n\}$,

$$R_i(\underline{\gamma}) \triangleq \gamma_i + \sigma_i \nabla_i U_i(\underline{\gamma})$$

for each $\gamma \in [0,1]^n$. By Propositions D.7 and D.8, the mapping T is the orthogonal projection of R onto the Cartesian product space $[0,1]^n$ of real intervals. In particular, $T(\underline{\gamma}) = [R(\underline{\gamma})]^+$ and $T_i(\underline{\gamma}) = [R_i(\underline{\gamma})]^+$ for each $i \in \{1, 2, \ldots, n\}$.

Let $\underline{x}, \underline{y} \in [0,1]^n$. By the projection theorem in Proposition D.6, the orthogonal projection $[\cdot]^+$ is non-expansive with respect to the ℓ_2-norm. Hence,

$$\begin{aligned}
\|T(\underline{x}) - T(\underline{y})\|_\infty &= \max_{i \in \mathcal{C}} |T_i(\underline{x}) - T_i(\underline{y})| = \max_{i \in \mathcal{C}} \|T_i(\underline{x}) - T_i(\underline{y})\|_2 \\
&\leq \max_{i \in \mathcal{C}} \|R_i(\underline{x}) - R_i(\underline{y})\|_2 = \max_{i \in \mathcal{C}} |R_i(\underline{x}) - R_i(\underline{y})| \\
&= \|R(\underline{x}) - R(\underline{y})\|_\infty.
\end{aligned} \tag{B.1}$$

Because $[0,1]^n$ is a convex set, $t\underline{x} + (1-t)\underline{y} \in [0,1]^n$ for all $t \in [0,1]$. So, for each $i \in \mathcal{C}$, define function $g_i : [0,1] \mapsto \mathbb{R}$ by the convex combination

$$g_i(t) \triangleq tR_i(\underline{x}) + (1-t)R_i(\overbrace{t\underline{x} + (1-t)\underline{y}}^{\in [0,1]^n}) = tx_i + (1-t)y_i + \sigma_i \nabla_i U_i(t\underline{x} + (1-t)\underline{y}),$$

which shares being continuously differentiable with $\nabla_i U_i$. Then, for each $i \in \mathcal{C}$, $R_i(\underline{x}) = g_i(1)$ and $R_i(\underline{y}) = g_i(0)$. Furthermore, by the fundamental theorem of calculus,

$$\begin{aligned}
\|T(\underline{x}) - T(\underline{y})\|_\infty \leq \|R(\underline{x}) - R(\underline{y})\|_\infty &= \max_{i \in \mathcal{C}} |R_i(\underline{x}) - R_i(\underline{y})| \\
&= \max_{i \in \mathcal{C}} |g_i(1) - g_i(0)| = \max_{i \in \mathcal{C}} \left| \int_0^1 g_i'(t)\, \mathrm{d}t \right| \\
&\leq \max_{i \in \mathcal{C}} \int_0^1 |g_i'(t)|\, \mathrm{d}t \leq \max_{i \in \mathcal{C}} \max_{t \in [0,1]} |g_i'(t)| \int_0^1 \mathrm{d}t = \max_{i \in \mathcal{C}} \max_{t \in [0,1]} |g_i'(t)|.
\end{aligned}$$

Then, by applying the chain rule to g_i' for each $i \in \mathcal{C}$,

$$\|R(\underline{x}) - R(\underline{y})\|_\infty \leq \max_{i \in \mathcal{C}} \max_{t \in [0,1]} |x_i - y_i + \sigma_i (\nabla \nabla_i U_i(\underbrace{t\underline{x} + (1-t)\underline{y}}_{\in [0,1]^n}))^\top (\underline{x} - \underline{y})|,$$

but $t\underline{x} + (1-t)\underline{y} \in [0,1]^n$ for all $t \in [0,1]$; so

$$\|R(\underline{x}) - R(\underline{y})\|_\infty \leq \max_{i \in \mathcal{C}} \max_{\underline{z} \in [0,1]^n} |x_i - y_i + \sigma_i (\nabla \nabla_i U_i(\underline{z}))^\top (\underline{x} - \underline{y})|.$$

By Eq. (D.2) that follows from Definition D.7 of the block gradient in a product space, row vector $\nabla \nabla_i U_i^\top = \begin{bmatrix} \nabla_{i1}^2 U_i & \nabla_{i2}^2 U_i & \cdots & \nabla_{in}^2 U_n \end{bmatrix}$. So, by expanding the dot product,

$$\begin{aligned}
\|R(\underline{x}) - R(\underline{y})\|_\infty &\leq \max_{i \in \mathcal{C}} \max_{\underline{z} \in [0,1]^n} \left| \left(1 + \sigma_i \nabla_{ii}^2 U_i(\underline{z})\right)(x_i - y_i) + \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \sigma_i \nabla_{i\ell}^2 U_i(\underline{z})(x_\ell - y_\ell) \right| \\
&\leq \max_{i \in \mathcal{C}} \max_{\underline{z} \in [0,1]^n} \left(\left|1 + \sigma_i \nabla_{ii}^2 U_i(\underline{z})\right| |x_i - y_i| + \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \sigma_i |\nabla_{i\ell}^2 U_i(\underline{z})| |x_\ell - y_\ell| \right) \\
&\leq \max_{i \in \mathcal{C}} \max_{\underline{z} \in [0,1]^n} \left(\left|1 + \sigma_i \nabla_{ii}^2 U_i(\underline{z})\right| + \sum_{\substack{j \in \mathcal{C} \\ j \neq i}} \sigma_i |\nabla_{i\ell}^2 U_i(\underline{z})| \right) \|\underline{x} - \underline{y}\|_\infty.
\end{aligned} \tag{B.2}$$

However, by assumption (i), for any $\underline{\gamma} \in [0,1]^n$ and $i \in \mathcal{C}$,

$$\nabla^2_{ii} U_i(\underline{\gamma}) \triangleq \frac{\partial^2 U_i(\underline{\gamma})}{\partial {\gamma_i}^2} = \sum_{j \in \mathcal{V}_i} \big(2p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big) = \sum_{j \in \mathcal{V}_i} \overbrace{p'_{ij}(Q_j)}^{<0} + \sum_{j \in \mathcal{V}_i} \overbrace{\big(p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big)}^{\leq 0} < 0, \tag{B.3}$$

and

$$\begin{aligned}\nabla^2_{ii} U_i(\underline{\gamma}) &= \sum_{j \in \mathcal{V}_i} \big(2p'_{ij}(Q_j) + \overbrace{\gamma_i p''_{ij}(Q_j)}^{\geq 0}\big) \geq \sum_{j \in \mathcal{V}_i} 2p'_{ij}(Q_j) = -2 \sum_{j \in \mathcal{V}_i} |p'_{ij}(Q_j)| \\ &\geq -2 \sum_{j \in \mathcal{V}_i} \max_{k \in \mathcal{V}_i} |p'_{ik}(0)| = -2|\mathcal{V}_i| \max_{k \in \mathcal{V}_i} |p'_{ik}(0)| \geq -2|\mathcal{V}_i| \max_{k \in \mathcal{V}_i} |p'_{ik}(0)|.\end{aligned}$$

So, by the assumed limits on step size σ_i given in Eq. (4.5),

$$-\frac{1}{\sigma_i} \leq \nabla^2_{ii} U_i(\underline{\gamma}) < 0 \tag{B.4}$$

for all $i \in \mathcal{C}$. Hence, following Eq. (B.2),

$$\begin{aligned}\|T(\underline{x}) - T(\underline{y})\|_\infty &\leq \max_{i \in \mathcal{C}} \max_{\underline{z} \in [0,1]^n} \Bigg(\underbrace{\big|1 + \overbrace{\sigma_i \nabla^2_{ii} U_i(\underline{z})}^{\geq -1}\big|}_{\geq 0} + \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \sigma_i |\nabla^2_{i\ell} U_i(\underline{z})| \Bigg) \|\underline{x} - \underline{y}\|_\infty \\ &= \max_{i \in \mathcal{C}} \max_{\underline{z} \in [0,1]^n} \Bigg(1 + \sigma_i \nabla^2_{ii} U_i(\underline{z}) + \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \sigma_i |\nabla^2_{i\ell} U_i(\underline{z})| \Bigg) \|\underline{x} - \underline{y}\|_\infty \\ &= \max_{i \in \mathcal{C}} \max_{\underline{z} \in [0,1]^n} \Bigg(1 + \sigma_i \Bigg(\nabla^2_{ii} U_i(\underline{z}) + \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} |\nabla^2_{i\ell} U_i(\underline{z})| \Bigg) \Bigg) \|\underline{x} - \underline{y}\|_\infty. \end{aligned} \tag{B.5}$$

So take $\underline{\gamma} \in [0,1]^n$ and $i \in \mathcal{C}$. For another cooperator $\ell \in \mathcal{C} \setminus \{i\}$, if $\ell \notin \mathcal{C}_j$ (i.e., ℓ is not an outgoing cooperator for j), then

$$\frac{\partial Q_j}{\partial \gamma_\ell} = 0 \qquad \text{and} \qquad \frac{\partial \, \mathrm{SOBP}_1(\mathcal{C}_j - \{i\})}{\partial \gamma_\ell} = 0,$$

where $Q_j \triangleq \sum_{k \in \mathcal{C}_j} \gamma_k$ and SOBP is from Definition C.1. So, by introducing SOMS from Proposition C.11,

$$\begin{aligned} 0 \leq \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} |\nabla^2_{i\ell} U_i(\underline{\gamma})| &\triangleq \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left| \frac{\partial^2 U_i(\underline{\gamma})}{\partial \gamma_i \partial \gamma_\ell} \right| \\ &= \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left| \sum_{j \in \mathcal{V}_i} [\ell \in \mathcal{C}_j] \left(p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j) + \overbrace{\mathrm{SOMS}_2(\mathcal{C}_j \setminus \{i, \ell\})}^{\partial/\partial\gamma_\ell \, \mathrm{SOBP}_1(\mathcal{C}_j - \{i\})} c_{ij} \right) \right| \end{aligned}$$

where $[\cdot]$ is the Iverson bracket [27, 29] (i.e., $[S] = 1$ when statement S is true and $[S] = 0$ otherwise). Hence, by the triangle inequality,

$$\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} |\nabla^2_{i\ell} U_i(\underline{\gamma})| \leq \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \sum_{j \in \mathcal{V}_i} [\ell \in \mathcal{C}_j] \Big(\underbrace{\big|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big|}_{\leq 0} + |\mathrm{SOMS}_2(\mathcal{C}_j \setminus \{i, \ell\})| \, |c_{ij}| \Big).$$

By Propositions C.14 and C.15, $0 < \mathrm{SOMS}_2(\Gamma) \leq 1/2$ for all $\Gamma \subseteq \mathcal{C}$, and so

$$\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left|\nabla^2_{i\ell} U_i(\underline{\gamma})\right| \leq \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \sum_{j \in \mathcal{V}_i} [\ell \in \mathcal{C}_j] \left(\left|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right| + \frac{1}{2}|c_{ij}| \right).$$

Furthermore, because these two finite sums can be transposed,

$$\begin{aligned}
\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left|\nabla^2_{i\ell} U_i(\underline{\gamma})\right| &\leq \sum_{j \in \mathcal{V}_i} \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} [\ell \in \mathcal{C}_j] \left(\left|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right| + \frac{1}{2}|c_{ij}| \right) \\
&= \sum_{j \in \mathcal{V}_i} \left(\left|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right| + \frac{1}{2}|c_{ij}| \right) \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} [\ell \in \mathcal{C}_j].
\end{aligned}$$

Hence, the second sum is a count of all elements in $(\mathcal{C} \setminus \{i\}) \cap \mathcal{C}_j$. That is,

$$\begin{aligned}
\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left|\nabla^2_{i\ell} U_i(\underline{\gamma})\right| &\leq \sum_{j \in \mathcal{V}_i} \left(\left|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right| + \frac{1}{2}|c_{ij}| \right) \underbrace{\left|\{\ell \in \mathcal{C} : \ell \in \mathcal{C}_j \setminus \{i\}\}\right|}_{\substack{\text{Number of non-}i\text{ cooperators} \\ \text{connected to } j}} \\
&= \sum_{j \in \mathcal{V}_i} \left(\left|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right| + \frac{1}{2}|c_{ij}| \right) \left|\mathcal{C}_j \setminus \{i\}\right|,
\end{aligned}$$

and, because $j \in \mathcal{V}_i$ if and only if $i \in \mathcal{C}_j$,

$$\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left|\nabla^2_{i\ell} U_i(\underline{\gamma})\right| \leq \sum_{j \in \mathcal{V}_i} \left(\left|p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right| + \frac{1}{2}|c_{ij}| \right) \left(|\mathcal{C}_j| - 1\right).$$

However, by assumption (ii), each conveyor $j \in \mathcal{V}$ has no more than three outgoing connections to cooperators (i.e., $|\mathcal{C}_j| \leq 3$). Additionally, by assumption (iii), if $j \in \mathcal{V}_i$ is a 3-conveyor (i.e., it has 3 outgoing cooperator connections), then there must be some other conveyor $m \in \mathcal{V}_i \setminus \{j\}$ that is a 2-conveyor. So, letting $m \in \mathcal{V}_i$ be the 2-conveyor that is guaranteed to exist,

$$\begin{aligned}
\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left|\nabla^2_{i\ell} U_i(\underline{\gamma})\right| &\leq \overbrace{2\sum_{j \in \mathcal{V}_i \setminus \{m\}} \left(\Big|\underbrace{p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)}_{\leq 0}\Big| + \frac{1}{2}|c_{ij}| \right)}^{\substack{\text{Doubled contribution to sum from} \\ \text{other cooperators connected to} \\ \text{assumed 3-conveyors in } \mathcal{V}_i \setminus \{m\}}} + \overbrace{\Big|\underbrace{p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)}_{\leq 0}\Big| + \frac{1}{2}|c_{im}|}^{\substack{\text{Contribution to sum from} \\ \text{other cooperator of 2-conveyor } m \in \mathcal{V}_i}} \\
&= 2\sum_{j \in \mathcal{V}_i \setminus \{m\}} \left(\underbrace{-\left(p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\right)}_{\geq 0} + \frac{1}{2}|c_{ij}| \right) \underbrace{-\left(p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)\right)}_{\geq 0} + \frac{1}{2}|c_{im}| \\
&= \sum_{j \in \mathcal{V}_i \setminus \{m\}} \left(-\left(2p'_{ij}(Q_j) + 2\gamma_i p''_{ij}(Q_j)\right) + |c_{ij}| \right) - \left(p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)\right) + \frac{1}{2}|c_{im}| \\
&= \sum_{j \in \mathcal{V}_i \setminus \{m\}} \left(-\Big(2p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\Big) + |c_{ij}| \right) \\
&\quad - \left(p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)\right) + \frac{|c_{im}|}{2}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j \in \mathcal{V}_i \setminus \{m\}} \Big(-\big(2p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big) + |c_{ij}| \Big) - \overbrace{\sum_{j \in \mathcal{V}_i \setminus \{m\}} \gamma_i p''_{ij}(Q_j)}^{\geq 0} \\
&\quad - (p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)) + \frac{|c_{im}|}{2},
\end{aligned}
$$

and so, due to the convexity of stabilizing payment functions,

$$
\begin{aligned}
\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left| \nabla^2_{i\ell} U_i(\underline{\gamma}) \right| &\leq -\sum_{j \in \mathcal{V}_i \setminus \{m\}} \big(2p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big) + \sum_{j \in \mathcal{V}_i \setminus \{m\}} |c_{ij}| - (p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)) + \frac{|c_{im}|}{2} \\
&= -\sum_{j \in \mathcal{V}_i \setminus \{m\}} \big(2p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big) - (p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)) + \sum_{j \in \mathcal{V}_i \setminus \{m\}} |c_{ij}| + \frac{|c_{im}|}{2}.
\end{aligned}
$$

Because $\mathcal{A}$ is finite, $\mathcal{V}_i \subseteq \mathcal{A}$ is finite, and so

$$
\begin{aligned}
\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left| \nabla^2_{i\ell} U_i(\underline{\gamma}) \right| &\leq -\sum_{j \in \mathcal{V}_i \setminus \{m\}} \big(2p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big) - (p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)) \\
&\quad + \left(|\overbrace{\mathcal{V}_i \setminus \{m\}}^{m \in \mathcal{V}_i}| + \frac{1}{2} \right) \max_{j \in \mathcal{V}_i} |c_{ij}| \\
&= -\sum_{j \in \mathcal{V}_i \setminus \{m\}} \big(2p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big) - (p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)) + \left(|\mathcal{V}_i| - \frac{1}{2} \right) \max_{j \in \mathcal{V}_i} |c_{ij}|,
\end{aligned}
$$

and

$$
\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left| \nabla^2_{i\ell} U_i(\underline{\gamma}) \right| \leq -\sum_{j \in \mathcal{V}_i \setminus \{m\}} \big(2p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big) - (p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)) + \left(|\mathcal{V}_i| - \frac{1}{2} \right) \max_{j \in \mathcal{V}_i} |c_{ij}|.
$$

So, by expanding the index set of the summation to include $m \in \mathcal{V}_i$ and subtracting the new term outside the summation,

$$
\begin{aligned}
\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left| \nabla^2_{i\ell} U_i(\underline{\gamma}) \right| &\leq -\sum_{j \in \mathcal{V}_i} \big(2p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big) + (2p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)) \\
&\quad - (p'_{im}(Q_m) + \gamma_i p''_{im}(Q_m)) + \left(|\mathcal{V}_i| - \frac{1}{2} \right) \max_{j \in \mathcal{V}_i} |c_{ij}| \\
&= -\overbrace{\sum_{j \in \mathcal{V}_i} \big(2p'_{ij}(Q_j) + \gamma_i p''_{ij}(Q_j)\big)}^{\nabla^2_{ii} U_i(\underline{\gamma})} + \overbrace{p'_{im}(Q_m)}^{<0} + \left(|\mathcal{V}_i| - \frac{1}{2} \right) \max_{j \in \mathcal{V}_i} |c_{ij}| \\
&= -\nabla^2_{ii} U_i(\underline{\gamma}) - |p'_{im}(Q_m)| + \left(|\mathcal{V}_i| - \frac{1}{2} \right) \max_{j \in \mathcal{V}_i} |c_{ij}| \\
&\leq -\nabla^2_{ii} U_i(\underline{\gamma}) - \min_{j \in \mathcal{V}_i} |p'_{ij}(Q_j)| + \left(|\mathcal{V}_i| - \frac{1}{2} \right) \max_{j \in \mathcal{V}_i} |c_{ij}| \\
&= -\nabla^2_{ii} U_i(\underline{\gamma}) - \underbrace{\left(\min_{j \in \mathcal{V}_i} |p'_{ij}(Q_j)| - \left(|\mathcal{V}_i| - \frac{1}{2} \right) \max_{j \in \mathcal{V}_i} |c_{ij}| \right)}_{> 0 \text{ by Eq. (4.6)}}.
\end{aligned}
$$

So, by the assumption in Eq. (4.6), the underbraced expression is strictly greater than zero. Hence,

$$\sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} \left| \nabla^2_{i\ell} U_i(\underline{\gamma}) \right| < -\nabla^2_{ii} U_i(\underline{\gamma}). \tag{B.6}$$

Furthermore, by the bounds on $\nabla^2_{ii} U_i(\underline{\gamma})$ in Eq. (B.4),

$$-\frac{1}{\sigma_i} \leq \nabla^2_{ii} U_i(\underline{\gamma}) + \left| \nabla^2_{i\ell} U_i(\underline{\gamma}) \right| < 0.$$

Hence, following Eq. (B.5),

$$\|T(\underline{x}) - T(\underline{y})\|_\infty \leq \underbrace{\max_{i \in \mathcal{C}} \max_{\underline{z} \in [0,1]^n} \underbrace{\left(1 + \sigma_i \underbrace{\left(\overbrace{\nabla^2_{ii} U_i(\underline{z}) + \sum_{\substack{\ell \in \mathcal{C} \\ \ell \neq i}} |\nabla^2_{i\ell} U_i(\underline{z})|}^{\in \left[-\frac{1}{\sigma_i}, 0\right)} \right)}_{\in [-1,0)} \right)}_{\in [0,1)}}_{\triangleq \alpha \in [0,1)} \|\underline{x} - \underline{y}\|_\infty,$$

So there is an $\alpha \in [0, 1)$ such that

$$\|T(\underline{x}) - T(\underline{y})\|_\infty \leq \alpha \|\underline{x} - \underline{y}\|_\infty.$$

Thus, the projection mapping T is a contraction mapping with modulus α. By Proposition D.2(a), there exists a unique $\underline{\gamma}^* \in [0, 1]^n$ such that $T(\underline{\gamma}^*) = \underline{\gamma}^*$ (i.e., $\underline{\gamma}^*$ is a fixed point of the contraction T). Further, convergence of the sequence $\{\underline{\gamma}(t)\}_{t \in \mathcal{T}}$ generated by the TADI iteration mapping T in Eq. (4.5) to $\underline{\gamma}^*$ is guaranteed by Proposition D.14. In particular, the sequence of sets $\{\mathcal{X}(k)\}_{k \in \mathbb{W}}$ defined, for all $k \in \mathbb{W}$, by

$$\begin{aligned}
\mathcal{X}(k) &\triangleq \left\{ \underline{\gamma} \in [0,1]^n : \|\underline{\gamma} - \underline{\gamma}^*\|_\infty \leq \alpha^k \|\underline{\gamma}(0) - \underline{\gamma}^*\|_\infty \right\} \\
&= \left\{ (\gamma_{c_1}, \gamma_{c_2}, \ldots, \gamma_{c_n}) \in [0,1]^n : |\gamma_i - \gamma_i^*| \leq \alpha^k \max_{j \in \mathcal{C}} |\gamma_j(0) - \gamma_j^*| \text{ for all } i \in \mathcal{C} \right\} \\
&= \prod_{i \in \mathcal{C}} \underbrace{\{ \gamma_i \in [0,1] : |\gamma_i - \gamma_i^*| \leq \alpha^k \max_{j \in \mathcal{C}} |\gamma_j(0) - \gamma_j^*| \}}_{\triangleq \mathcal{X}_i(k)}
\end{aligned}$$

meets the general (i.e., asynchronous) convergence conditions given in Proposition D.12 that:

(i) For each $i \in \mathcal{C}$ and $k \in \mathbb{W}$, $\cdots \subset \mathcal{X}_i(k+1) \subset \mathcal{X}_i(k) \subset \cdots \subset \mathcal{X}_i(0) \subseteq [0, 1]$. Additionally, $\underline{\gamma}(0) \in \mathcal{X}(0)$.

(ii) For all $k \in \mathbb{W}$ and all $\underline{\gamma} \in \mathcal{X}(k)$, $T(\underline{\gamma}) \in \mathcal{X}(k+1)$. Additionally, if $\{\underline{y}^k\}$ is a sequence such that $\underline{y}^k \in \mathcal{X}(k)$ for every $k \in \mathbb{W}$, then $\lim_{k \to \infty} \underline{y}^k = \underline{\gamma}^*$, which is the fixed point of the TADI mapping T.

Additionally, for any ε, there exists a $k \in \mathbb{W}$ such that $\mathcal{X}(k) \subseteq \mathcal{B}^\infty_\varepsilon(\underline{\gamma}^*)$ where open ball $\mathcal{B}^\infty_\varepsilon(\underline{\gamma}^*) \triangleq \{\underline{\gamma} \in [0,1]^n : \|\underline{\gamma} - \underline{\gamma}^*\|_\infty < \varepsilon\}$. So, by Proposition D.13, the TADI-generated sequence $\{\underline{\gamma}(t)\}$ and the outdated estimate sequences $\{\underline{\gamma}^i(t)\}$ for all $i \in \mathcal{C}$ each converge to fixed point $\underline{\gamma}^*$. The sets $\mathcal{X}(k)$ for all $k \in \mathbb{W}$ are analogous to level sets of a Lyapunov function; they guarantee the continual reduction of the distance between the asynchronous algorithm's trajectory and the fixed point $\underline{\gamma}^*$. By Proposition D.11, the fixed point $\underline{\gamma}^*$ that $\{\underline{\gamma}(t)\}$ converges to is the unique solution to the separable variational inequality problem in Eq. (4.1). By Proposition D.10, the variational inequality solution $\underline{\gamma}^*$ is the Nash equilibrium of the cooperation game. □

Proposition B.2. (Sufficient condition for diagonal dominance) *For all $i \in \mathcal{C}$, if step size σ_i is such that*

$$\overbrace{\frac{1}{\sigma_i} \geq 2|\mathcal{V}_i| \max_{k\in\mathcal{V}_i} |p'_{ik}(0)|}^{\textit{From Eq. (4.5)}} \tag{B.7a}$$

and

$$2 \min_{j\in\mathcal{V}_i} |p'_{ij}(|\mathcal{C}_j|)| > \left(\frac{1}{\sigma_i \max_{k\in\mathcal{V}_i} |p'_{ik}(0)|} - 1 \right) \max_{j\in\mathcal{V}_i} |c_{ij}|, \tag{B.7b}$$

then

$$\underbrace{\min_{j\in\mathcal{V}_i} |p'_{ij}(|\mathcal{C}_j|)| > \left(|\mathcal{V}_i| - \frac{1}{2} \right) \max_{j\in\mathcal{V}_i} |c_{ij}|}_{\textit{Eq. (4.6)}} \qquad \textit{for all } i \in \mathcal{C}.$$

Proof of Proposition B.2. By Eq. (B.7a), for all $i \in \mathcal{C}$,

$$\frac{1}{\sigma_i \max_{k\in\mathcal{V}_i} |p'_{ik}(0)|} \geq 2|\mathcal{V}_i|,$$

and so Eq. (B.7b) implies

$$2 \min_{j\in\mathcal{V}_i} |p'_{ij}(|\mathcal{C}_j|)| > (2|\mathcal{V}_i| - 1) \max_{j\in\mathcal{V}_i} |c_{ij}| \qquad \text{for all } i \in \mathcal{C}.$$

Hence,

$$\min_{j\in\mathcal{V}_i} |p'_{ij}(|\mathcal{C}_j|)| > \left(|\mathcal{V}_i| - \frac{1}{2} \right) \max_{j\in\mathcal{V}_i} |c_{ij}| \text{for all } i \in \mathcal{C}. \qquad \square$$

C Combinatorics applied to volunteering

The principal results used in this work are Propositions C.11, C.14, and C.15, which follow primarily from Theorems C.1, C.2, and C.3. These results serve to assist in the analysis of volunteering problems like the one given in Example C.1.

Example C.1. (Volunteering) You are going to volunteer for a job. Two other individuals will volunteer (independently) with probabilities γ_1 and γ_2, respectively. If $n \in \{1, 2, 3\}$ individuals volunteer, any one of them will be asked to complete the job with uniform probability $1/n$. Given that you volunteer, the probability that you will be asked to do the job is

$$(1-\gamma_1)(1-\gamma_2) + \frac{1}{2}\gamma_1(1-\gamma_2) + \frac{1}{2}(1-\gamma_1)\gamma_2 + \frac{1}{3}\gamma_1\gamma_2. \tag{C.1}$$

That is, given that you volunteer, there is a $1/(k+1)$ probability that you will be asked to complete the job when $k \in \{0, 1, 2\}$ other individuals also volunteer.

The probability in Eq. (C.1) from Example C.1 matches the SOBP expression in Eq. (C.3) from Definition C.1 below with $g = 1$, $\Omega = \{\gamma_i\}_{i\in\{1,2\}}$, and $\Gamma = \{1, 2\}$.

C.1 Definitions: SOBP and SOMS

In the following, let $\mathcal{I} \subset \mathbb{W}$ be a finite index set, and let

$$\Omega \triangleq \{\gamma_i\}_{i\in\mathcal{I}} \tag{C.2}$$

be an indexed family with where $\gamma_i \in \mathcal{X} \subseteq \mathbb{R}$ for each $i \in \mathcal{I}$.

Definition C.1. (Sum of binomial products) For $\Gamma \subseteq \mathcal{I}$, $g \in \mathbb{N}$, and $m \triangleq |\Gamma|$, the *sum of binomial products*

$$\begin{aligned}
\mathrm{SOBP}_g(\Gamma) \triangleq \quad & \frac{1}{g} \prod_{i\in\Gamma}(1-\gamma_i) \\
&+ \frac{1}{g+1} \sum_{i\in\Gamma}\left(\gamma_i \prod_{j\in\Gamma\setminus\{i\}}(1-\gamma_j)\right) \\
&+ \frac{1}{g+2} \sum_{\{i,j\}\subseteq\Gamma}\left(\gamma_i\gamma_j \prod_{k\in\Gamma\setminus\{i,j\}}(1-\gamma_k)\right) \\
&+ \cdots \\
&+ \frac{1}{g+\ell} \sum_{\substack{\mathcal{C}\subseteq\Gamma\\ |\mathcal{C}|=\ell}}\left(\left(\prod_{i\in\mathcal{C}}\gamma_i\right)\left(\prod_{k\in\Gamma\setminus\mathcal{C}}(1-\gamma_k)\right)\right) \\
&+ \cdots \\
&+ \frac{1}{g+m} \prod_{i\in\Gamma}\gamma_i,
\end{aligned} \tag{C.3}$$

and g is called the *seed*.

Definition C.2. (Sum of monomial sums) For $\Gamma \subseteq \mathcal{I}$, $h \in \mathbb{N}$, and $m \triangleq |\Gamma|$, the *sum of monomial sums*

$$\begin{aligned} \mathrm{SOMS}_h(\Gamma) \triangleq \frac{1}{h} - \frac{1}{h+1}\sum_{i\in\Gamma}\gamma_i + \frac{1}{h+2}\sum_{\{i,j\}\subseteq\Gamma}\gamma_i\gamma_j \\ - \ldots +- \ldots + (-1)^\ell \frac{1}{h+\ell} \sum_{\substack{\mathcal{C}\subseteq\Gamma \\ |\mathcal{C}|=\ell}} \left(\prod_{i\in\mathcal{C}}\gamma_i\right) - \ldots +- \ldots \\ + (-1)^m \frac{1}{h+m}\prod_{i\in\Gamma}\gamma_i, \end{aligned} \tag{C.4}$$

and h is called the *seed.*

The following sections provide the analytical tools to relate SOBP to SOMS. Hence, they provide a framework in which to analyze volunteering problems like Example C.1.

C.2 Coordinate transformation

Proposition C.1. (Binomial theorem) *For $a, b \in \mathbb{R}$ and $n \in \mathbb{W}$,*

$$\begin{aligned} (a+b)^n &= \overbrace{\binom{n}{0}}^{1} a^n + \binom{n}{1} a^{n-1}b^1 + \binom{n}{2} a^{n-2}b^2 + \cdots + \binom{n}{k} a^{n-k}b^k + \cdots + \overbrace{\binom{n}{n}}^{1} b^n \\ &= \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k \end{aligned} \tag{C.5}$$

where, for all $r \in \{0, 1, \ldots, n\}$, $\binom{n}{r} \triangleq n!/(r!(n-r)!)$.

Proof of Proposition C.1 is given by Gustafson et al. [24]. □

Remark (Simple binomial theorem) Let $x \in \mathbb{R}$ and $n \in \mathbb{W}$. Then, by Proposition C.1,

$$(1+x)^n = \binom{n}{0} + \binom{n}{1}x^1 + \binom{n}{2}x^2 + \cdots + \binom{n}{n-1}x^{n-1} + \binom{n}{n}x^n = \sum_{k=0}^{n}\binom{n}{k}x^k \tag{C.6}$$

Proposition C.2. (Product of binomials) *For a set $\Gamma \subseteq \mathcal{I}$,*

$$\begin{aligned} \prod_{i\in\Gamma}(1-\gamma_i) &= \sum_{\mathcal{C}\subseteq\Gamma}(-1)^{|\mathcal{C}|}\prod_{i\in\mathcal{C}}\gamma_i \\ &= 1 - \sum_{i\in\Gamma}\gamma_i + \sum_{\{i,j\}\subseteq\Gamma}\gamma_i\gamma_j - \sum_{\{i,j,k\}\subseteq\Gamma}\gamma_i\gamma_j\gamma_k + \cdots + (-1)^\ell \sum_{\substack{\mathcal{C}\subseteq\Gamma \\ |\mathcal{C}|=\ell}}\prod_{i\in\mathcal{C}}\gamma_i + \cdots + (-1)^{|\Gamma|}\prod_{i\in\Gamma}\gamma_i. \end{aligned} \tag{C.7}$$

Proof of Proposition C.2. The claim in Eq. (C.7) is trivially true for $\Gamma = \emptyset$ (i.e., when $|\Gamma| = 0$ and $|\wp(\Gamma)| = |\{\emptyset\}| = 1$). For the purpose of induction, assume that the claim is true for $\Gamma \subseteq \mathcal{I}$

with $|\Gamma| = k \in \{0, 1, \dots, n-1\}$ and $\max \Gamma \leq \min(\mathcal{I} \setminus \Gamma)$. Let $j = \min(\mathcal{I} \setminus \Gamma)$. Then

$$\begin{aligned}
\prod_{i \in \Gamma \cup \{j\}} (1 - \gamma_i) &= (1 - \gamma_j) \prod_{i \in \Gamma} (1 - \gamma_i) \\
&= (1 - \gamma_j) \sum_{\mathcal{C} \subseteq \Gamma} (-1)^{|\mathcal{C}|} \prod_{i \in \mathcal{C}} \gamma_i \\
&= \sum_{\mathcal{C} \subseteq \Gamma} (-1)^{|\mathcal{C}|} \prod_{i \in \mathcal{C}} \gamma_i - \gamma_j \sum_{\mathcal{C} \subseteq \Gamma} (-1)^{|\mathcal{C}|} \prod_{i \in \mathcal{C}} \gamma_i \\
&= \underbrace{\sum_{\mathcal{C} \subseteq \Gamma} (-1)^{|\mathcal{C}|} \prod_{i \in \mathcal{C}} \gamma_i}_{\text{Combinations of } \Gamma} + \underbrace{\sum_{\mathcal{C} \subseteq \Gamma} (-1)^{|\mathcal{C}|+1} \prod_{i \in \mathcal{C}} \gamma_i \gamma_j}_{\text{Combinations of } \Gamma \text{ also including } j} \\
&= \sum_{\mathcal{C} \subseteq \Gamma} (-1)^{|\mathcal{C}|} \prod_{i \in \mathcal{C}} \gamma_i + \sum_{\mathcal{C} \subseteq \Gamma} (-1)^{|\mathcal{C}|+1} \prod_{i \in \mathcal{C} \cup \{j\}} \gamma_i \\
&= \underbrace{\sum_{\mathcal{C} \subseteq \Gamma} (-1)^{|\mathcal{C}|} \prod_{i \in \mathcal{C}} \gamma_i}_{\substack{\text{Combinations of} \\ \Gamma \cup \{j\} \text{ without } j}} + \underbrace{\sum_{\substack{\mathcal{C}_0 \subseteq \Gamma \\ \mathcal{C} = \mathcal{C}_0 \cup \{j\}}} (-1)^{|\mathcal{C}|} \prod_{i \in \mathcal{C}} \gamma_i}_{\text{Combinations of } \Gamma \cup \{j\} \text{ with } j} \\
&= \sum_{\mathcal{C} \subseteq (\Gamma \cup \{j\})} (-1)^{|\mathcal{C}|} \prod_{i \in \mathcal{C}} \gamma_i,
\end{aligned}$$

and so the claim is also true for the $k+1$ case. Hence, because the claim is true for the $k = 0$ case, it is true for all $k \in \{0, 1, \dots, n\}$ by induction. □

Proposition C.3. (Sum of mixed product) *Let $\Gamma \subseteq \mathcal{I}$, $m \triangleq |\Gamma|$, and $\ell \in \{0, 1, \dots, m\}$. Then*

$$\begin{aligned}
&\sum_{\substack{\mathcal{C} \subseteq \Gamma \\ |\mathcal{C}| = \ell}} \left(\left(\prod_{i \in \mathcal{C}} \gamma_i \right) \left(\prod_{k \in \Gamma \setminus \mathcal{C}} (1 - \gamma_k) \right) \right) \\
&\quad = \sum_{\substack{\mathcal{C} \subseteq \Gamma \\ |\mathcal{C}| \geq \ell}} \left((-1)^{|\mathcal{C}| - \ell} \binom{|\mathcal{C}|}{\ell} \prod_{i \in \mathcal{C}} \gamma_i \right) \\
&\quad = \binom{\ell}{\ell} \sum_{\substack{\mathcal{C} \subseteq \Gamma \\ |\mathcal{C}| = \ell}} \prod_{i \in \mathcal{C}} \gamma_i + \binom{\ell+1}{\ell} \sum_{\substack{\mathcal{C} \subseteq \Gamma \\ |\mathcal{C}| = \ell+1}} \prod_{i \in \mathcal{C}} \gamma_i + \cdots + \binom{m-1}{\ell} \sum_{\substack{\mathcal{C} \subseteq \Gamma \\ |\mathcal{C}| = m-1}} \prod_{i \in \mathcal{C}} \gamma_i + \binom{m}{\ell} \prod_{i \in \Gamma} \gamma_i.
\end{aligned} \tag{C.8}$$

Proof of Proposition C.3. By Proposition C.2,

$$\begin{aligned}
\sum_{\substack{\mathcal{C} \subseteq \Gamma \\ |\mathcal{C}| = \ell}} \left(\left(\prod_{i \in \mathcal{C}} \gamma_i \right) \left(\prod_{k \in \Gamma \setminus \mathcal{C}} (1 - \gamma_k) \right) \right) &= \sum_{\substack{\mathcal{C} \subseteq \Gamma \\ |\mathcal{C}| = \ell}} \left(\left(\prod_{i \in \mathcal{C}} \gamma_i \right) \left(\sum_{\mathcal{D} \subseteq (\Gamma \setminus \mathcal{C})} (-1)^{|\mathcal{D}|} \prod_{k \in \mathcal{D}} \gamma_k \right) \right) \\
&= \sum_{\substack{\mathcal{C} \subseteq \Gamma \\ |\mathcal{C}| = \ell}} \left(\sum_{\mathcal{D} \subseteq (\Gamma \setminus \mathcal{C})} (-1)^{|\mathcal{D}|} \left(\prod_{i \in \mathcal{C}} \gamma_i \right) \prod_{k \in \mathcal{D}} \gamma_k \right)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{\substack{\mathcal{C}\subseteq\Gamma\\|\mathcal{C}|=\ell}} \left(\sum_{\mathcal{D}\subseteq(\Gamma\setminus\mathcal{C})} (-1)^{|\mathcal{D}|} \prod_{i\in\mathcal{D}\cup\mathcal{C}} \gamma_i \right) \\
&= \sum_{\substack{\mathcal{C}\subseteq\Gamma\\|\mathcal{C}|=\ell}} \underbrace{\sum_{\substack{\mathcal{E}\subseteq\Gamma\\\mathcal{C}\subseteq\mathcal{E}}} \left((-1)^{|\mathcal{E}|-\ell} \prod_{i\in\mathcal{E}} \gamma_i \right)}_{\substack{\text{Each } \mathcal{E}\subseteq\Gamma \text{ repeats}\\ \text{for every } \mathcal{C}\subseteq\mathcal{E} \text{ set}}} \\
&= \sum_{\substack{\mathcal{E}\subseteq\Gamma\\|\mathcal{E}|\geq\ell}} (-1)^{|\mathcal{E}|-\ell} \binom{|\mathcal{E}|}{\ell} \prod_{i\in\mathcal{E}} \gamma_i \qquad \square
\end{aligned}
$$

Theorem C.1. (Binomial–monomial relationship) *Let $\Gamma \subseteq \mathcal{I}$, $m \triangleq |\Gamma|$, and $a_k \in \mathbb{R}$ for all $k \in \{0, 1, \ldots, m\}$. The expression*

$$
\begin{aligned}
& a_0 \prod_{i\in\Gamma} (1-\gamma_i) \\
&+ a_1 \sum_{i\in\Gamma} \left(\gamma_i \prod_{j\in\Gamma\setminus\{i\}} (1-\gamma_j) \right) \\
&+ a_2 \sum_{\{i,j\}\subseteq\Gamma} \left(\gamma_i\gamma_j \prod_{k\in\Gamma\setminus\{i,j\}} (1-\gamma_k) \right) \\
&+ \cdots \\
&+ a_\ell \sum_{\substack{\mathcal{C}\subseteq\Gamma\\|\mathcal{C}|=\ell}} \left(\left(\prod_{i\in\mathcal{C}} \gamma_i \right) \left(\prod_{k\in\Gamma\setminus\mathcal{C}} (1-\gamma_k) \right) \right) \\
&+ \cdots \\
&+ a_m \prod_{i\in\Gamma} \gamma_i,
\end{aligned}
\tag{C.9a}
$$

is equal to

$$
\begin{aligned}
&\overbrace{a_0}^{b_0} - \overbrace{(a_0 - a_1)}^{b_1} \sum_{i\in\Gamma} \gamma_i + \overbrace{\sum_{k=0}^{2} \binom{2}{k} (-1)^k a_k}^{b_2} \sum_{\{i,j\}\subseteq\Gamma} \gamma_i\gamma_j \\
&- \ldots +- \ldots + (-1)^\ell \underbrace{\sum_{k=0}^{\ell} \binom{\ell}{k} (-1)^k a_k}_{b_\ell} \sum_{\substack{\mathcal{C}\subseteq\Gamma\\|\mathcal{C}|=\ell}} \left(\prod_{i\in\mathcal{C}} \gamma_i \right) \\
&- \ldots +- \ldots + (-1)^m \underbrace{\sum_{k=0}^{m} \binom{m}{k} (-1)^k a_k}_{b_m} \prod_{i\in\Gamma} \gamma_i,
\end{aligned}
\tag{C.9b}
$$

where

$$
b_\ell \triangleq \sum_{k=0}^{\ell} \binom{\ell}{k} (-1)^k a_k \tag{C.9c}
$$

is the coefficient that corresponds to the sum of monomials with $\ell \in \{0, 1, \ldots, m\}$ factors.

Proof of Theorem C.1. Applying Proposition C.3 to expand each row of Eq. (C.9a) yields

$$\begin{aligned}
&a_0\Bigg(1 - \sum_{i\in\Gamma}\gamma_i + \sum_{\{i,j\}\in\Gamma}\gamma_i\gamma_j - \sum_{\{i,j,k\}\in\Gamma}\gamma_i\gamma_j\gamma_k + \cdots - + \cdots + (-1)^m\prod_{i\in\Gamma}\gamma_i\Bigg)\\
&+ a_1\Bigg(\sum_{i\in\Gamma}\gamma_i - \binom{2}{1}\sum_{\{i,j\}\in\Gamma}\gamma_i\gamma_j + \binom{3}{1}\sum_{\{i,j,k\}\in\Gamma}\gamma_i\gamma_j\gamma_k - \cdots + - \cdots - (-1)^m\binom{m}{1}\prod_{i\in\Gamma}\gamma_i\Bigg)\\
&+ a_2\Bigg(+\binom{2}{2}\sum_{\{i,j\}\in\Gamma}\gamma_i\gamma_j - \binom{3}{2}\sum_{\{i,j,k\}\in\Gamma}\gamma_i\gamma_j\gamma_k + \cdots - + \cdots + (-1)^m\binom{m}{2}\prod_{i\in\Gamma}\gamma_i\Bigg)\\
&+ \cdots\\
&+ a_m\Bigg(\binom{m}{m}\prod_{i\in\Gamma}\gamma_i\Bigg).
\end{aligned}$$

The expansion is a sum of monomial sums, each of which is a product of 0 to m elements of Γ. The row that is multiplied by a_k will contribute $\binom{\ell}{k}$ of each monomial with ℓ factors. For example, the three-factor monomial $\gamma_a\gamma_b\gamma_c$ (i.e., $\ell = 3$) can be generated in the a_2 row (i.e., $k = 2$) by

$$\gamma_a\gamma_b(1-\gamma_c)\ldots, \qquad \gamma_a\gamma_c(1-\gamma_a)\ldots, \qquad \text{or} \qquad \gamma_b\gamma_c(1-\gamma_a)\ldots,$$

and so the a_2 row will contribute $\binom{3}{2} = 3$ of this monomial that each have a weight of $-a_2$. The expression in Eq. (C.9b) results from summing the elements of each column of the expansion above. ☐

Remark (Coordinate transformation) The relationship in Eq. (C.9c) is a coordinate transformation from $(a_1, a_2, \ldots, a_m)$ in SOBP space to $(b_1, b_2, \ldots, b_m)$ in SOMS space.

Proposition C.4. (Lower bound) *Let $\mathcal{X} \subseteq [0, 1]$ in Eq. (C.2). If $a_k \geq 0$ for all $k \in \{1, \ldots, |\Gamma|\}$, then both Eqs (C.9a) and (C.9b) are greater than or equal to $\min\{a_k : k \in \{0, 1, \ldots, |\Gamma|\}\}$.*

Proof. Let $a \triangleq \min\{a_k : k \in \{0, 1, \ldots, |\Gamma|\}\}$, S be the series in Eq. (C.9a) with a_k as given, and let Y be the series in Eq. (C.9a) with $a_k = a$. Because $\gamma_i \in [0, 1]$ for all $i \in \mathcal{I}$, then $Y \leq S$. Moreover, by Theorem C.1, the bounding series Y can be written in the form of Eq. (C.9b) with

$$b_\ell = \overbrace{a\sum_{k=0}^{\ell}\binom{\ell}{k}(-1)^k = a(1+(-1))^\ell}^{\text{By Proposition C.1 (i.e., binomial theorem)}} = a \times 0^\ell = \begin{cases} a & \text{if } \ell = 0,\\ 0 & \text{if } \ell > 0.\end{cases}$$

So $Y = a$ and, by Theorem C.1, Eq. (C.9a) and the corresponding Eq. (C.9b) are both greater than or equal to $Y = a$. ☐

Proposition C.5. (Upper bound) *Let $\mathcal{X} \subseteq [0, 1]$ in Eq. (C.2). If $a_k \geq 0$ for all $k \in \{1, \ldots, |\Gamma|\}$, then both Eqs (C.9a) and (C.9b) are less than or equal to $\max\{a_k : k \in \{0, 1, \ldots, |\Gamma|\}\}$.*

Proof. Let $a \triangleq \max\{a_k : k \in \{0, 1, \ldots, |\Gamma|\}\}$, S be the series in Eq. (C.9a) with a_k as given, and let Y be the series in Eq. (C.9a) with $a_k = a$. Because $\gamma_i \in [0, 1]$ for all $i \in \mathcal{I}$, then $Y \geq S$. Moreover, by Theorem C.1, the bounding series Y can be written in the form of Eq. (C.9b) with

$$b_\ell = \overbrace{a\sum_{k=0}^{\ell}\binom{\ell}{k}(-1)^k = a(1+(-1))^\ell}^{\text{By binomial theorem}} = a \times 0^\ell = \begin{cases} a & \text{if } \ell = 0,\\ 0 & \text{if } \ell > 0,\end{cases}$$

where the replacement is justified by Proposition C.1 (i.e., the binomial theorem). So $Y = a$ and, by Theorem C.1, Eq. (C.9a) and the corresponding Eq. (C.9b) are both less than or equal to $Y = a$. □

Proposition C.6. (SOBP lower bound) *Let* $\mathcal{X} \subseteq [0,1]$ *in* Eq. (C.2)*. For* $\Gamma \subseteq \mathcal{I}$ *and* $g \in \mathbb{N}$, $\mathrm{SOBP}_g(\Gamma) \geq 1/(g+m)$ *where* $m \triangleq |\Gamma|$.

Proof of Proposition C.6. Apply Proposition C.4 with $a_k \triangleq 1/(g+k)$ for all $k \in \{0, 1, \ldots, |\Gamma|\}$. □

Proposition C.7. (SOBP upper bound) *Let* $\mathcal{X} \subseteq [0,1]$ *in* Eq. (C.2)*. For* $\Gamma \subseteq \mathcal{I}$ *and* $g \in \mathbb{N}$, $\mathrm{SOBP}_g(\Gamma) \leq 1/g$.

Proof of Proposition C.7. Apply Proposition C.5 with $a_k \triangleq 1/(g+k)$ for all $k \in \{0, 1, \ldots, |\Gamma|\}$. □

C.3 Translating SOBP to SOMS

Proposition C.8. (Regrouping by monomials) *Let* $\Gamma \subseteq \mathcal{I}$, $m \triangleq |\Gamma|$, *and* $g \in \mathbb{N}$*. Then* $\mathrm{SOBP}_g(\Gamma)$ *is equal to*

$$\underbrace{\frac{1}{g}}_{b_0} - \underbrace{\sum_{k=0}^{1} \frac{(-1)^k}{k+g}}_{b_1} \sum_{i \in \Gamma} \gamma_i + \underbrace{\sum_{k=0}^{2} \frac{(-1)^k}{k+g} \binom{2}{k}}_{b_2} \sum_{\{i,j\} \subseteq \Gamma} \gamma_i \gamma_j - \ldots + - \ldots + (-1)^m \underbrace{\sum_{k=0}^{m} \frac{(-1)^k}{k+g} \binom{m}{k}}_{b_m} \prod_{i \in \Gamma} \gamma_i \tag{C.10}$$

where

$$b_\ell \triangleq \sum_{k=0}^{\ell} \frac{(-1)^k}{k+g} \binom{\ell}{k} \tag{C.11}$$

is the underbraced coefficient corresponding to the sum of monomials with $\ell \in \{0, 1, \ldots, m\}$ *factors.*

Proof of Proposition C.8. Apply Theorem C.1 with $a_k \triangleq 1/(k+g)$ for all $k \in \{0, 1, \ldots, m\}$. □

Proposition C.9. (General SOBP weight expression) *For* $\ell \in \mathbb{W}$ *and* $g \in \mathbb{N}$,

$$\sum_{k=0}^{\ell} \frac{(-1)^k}{k+g} \binom{\ell}{k} = \frac{\ell!(g-1)!}{(g+\ell)!}.$$

Proof of Proposition C.9.

$$\begin{aligned}
\sum_{k=0}^{\ell} \frac{(-1)^k}{k+g} \binom{\ell}{k} &= \sum_{k=0}^{\ell} \frac{(-1)^k}{k+g} \frac{\ell!}{k!(\ell-k)!} = \sum_{k=0}^{\ell} \frac{\ell!}{(k+g)k!(\ell-k)!} (-1)^k \\
&= \sum_{k=0}^{\ell} \frac{\ell!}{(k+g)(k+g-1)!(\ell-k)!} \frac{(k+g-1)!}{k!} (-1)^k \\
&= \frac{\ell!}{(\ell+g)!} \sum_{k=0}^{\ell} \frac{(\ell+g)!}{(k+g)!(\ell+g-(k+g))!} \frac{(k+g-1)!}{k!} (-1)^k
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\ell!}{(\ell+g)!} \sum_{k=0}^{\ell} \binom{\ell+g}{k+g} \frac{(k+g-1)!}{k!} (-1)^k \\
&= \frac{\ell!}{(\ell+g)!} \sum_{k=0}^{\ell} \binom{\ell+g}{k+g} \frac{(k+g-1)!}{k!} x^k \Bigg|_{x=-1} \\
&= \frac{\ell!}{(\ell+g)!} \sum_{k=0}^{\ell} \frac{(\ell+g)!}{(k+g)!(\ell+g-(k+g))!} \frac{(k+g-1)!}{k!} x^k \Bigg|_{x=-1} \\
&= \frac{\ell!}{(\ell+g)!} \sum_{k=g-1}^{\ell+g-1} \frac{(\ell+g)!}{(k+1)!(\ell+g-(k+1))!} \frac{k!}{(k-(g-1))!} x^{k-(g-1)} \Bigg|_{x=-1} \\
&= \frac{\ell!}{(\ell+g)!} \sum_{k=g-1}^{\ell+g-1} \binom{\ell+g}{k+1} \frac{k!}{(k-(g-1))!} x^{k-(g-1)} \Bigg|_{x=-1} \\
&= \frac{\ell!}{(\ell+g)!} \frac{\mathrm{d}^{g-1}}{\mathrm{d}x^{g-1}} \sum_{k=0}^{\ell+g-1} \binom{\ell+g}{k+1} x^k \Bigg|_{x=-1} \\
&= \frac{\ell!}{(\ell+g)!} \frac{\mathrm{d}^{g-1}}{\mathrm{d}x^{g-1}} \left(\sum_{k=-1}^{\ell+g-1} \binom{\ell+g}{k+1} x^k - \binom{\ell+g}{0} \frac{1}{x} \right) \Bigg|_{x=-1} \\
&= \frac{\ell!}{(\ell+g)!} \left(\frac{\mathrm{d}^{g-1}}{\mathrm{d}x^{g-1}} \frac{1}{x} \sum_{k=0}^{\ell+g} \binom{\ell+g}{k} x^k - \frac{\mathrm{d}^{g-1}}{\mathrm{d}x^{g-1}} \frac{1}{x} \right) \Bigg|_{x=-1} \\
&= \frac{\ell!}{(\ell+g)!} \left(\frac{\mathrm{d}^{g-1}}{\mathrm{d}x^{g-1}} \frac{(1+x)^{\ell+g}}{x} - (-1)^{g-1} \frac{(g-1)!}{x^g} \right) \Bigg|_{x=-1}
\end{aligned}
$$

which is justified by Proposition C.1 (i.e., the binomial theorem). The right-hand side of this equation is equal to

$$\frac{\ell!(g-1)!}{(\ell+g)!}. \qquad \square$$

Theorem C.2. (Transformation of SOBP) *Let* $\Gamma \subseteq \mathcal{I}$, $m \triangleq |\Gamma|$, *and* $g \in \mathbb{N}$. *Then* $\mathrm{SOBP}_g(\Gamma)$ *is equal to*

$$\underbrace{\frac{1}{g}}_{b_0} - \underbrace{\frac{1}{(g+1)g}}_{b_1} \sum_{i\in\Gamma} \gamma_i + \underbrace{\frac{2}{(g+2)(g+1)g}}_{b_2} \sum_{\{i,j\}\in\Gamma} \gamma_i\gamma_j - \ldots +- \ldots + (-1)^m \underbrace{\frac{m!(g-1)!}{(g+m)!}}_{b_m} \prod_{i\in\Gamma} \gamma_i. \tag{C.12}$$

Proof of Theorem C.2. Apply Proposition C.9 to Proposition C.8. $\square$

Proposition C.10. (Seed-1 case) *Let* $\Gamma \subseteq \mathcal{I}$. *Then*

$$\mathrm{SOBP}_1(\Gamma) = \mathrm{SOMS}_1(\Gamma). \tag{C.13}$$

Proof of Proposition C.10. By applying Theorem C.2 with $g = 1$,

$$\mathrm{SOBP}_1(\Gamma) = 1 - \frac{1}{2} \sum_{i\in\Gamma} \gamma_i + \frac{1}{3} \sum_{\{i,j\}\subseteq\Gamma} \gamma_i\gamma_j - \ldots +- \ldots + (-1)^m \frac{1}{m+1} \prod_{i\in\Gamma} \gamma_i = \mathrm{SOMS}_1(\Gamma). \quad \square$$

Proposition C.11. (Seed-1 SOBP derivative) *Let $\Gamma \subseteq \mathcal{I}$ and $k \in \Gamma$. Then*

$$\frac{\partial}{\partial \gamma_k} \operatorname{SOBP}_1(\Gamma) = -\operatorname{SOMS}_2(\Gamma \setminus \{k\}). \tag{C.14}$$

Proof of Proposition C.11. Let $m \triangleq |\Gamma|$. By Proposition C.10,

$$\begin{aligned}
&\frac{\partial}{\partial \gamma_k} \operatorname{SOBP}_1(\Gamma) \\
&\qquad = \frac{\partial}{\partial \gamma_k} \left(1 - \frac{1}{2} \sum_{i \in \Gamma} \gamma_i + \frac{1}{3} \sum_{\{i,j\} \subseteq \Gamma} \gamma_i \gamma_j - \ldots +- \ldots + (-1)^m \frac{1}{m+1} \prod_{i \in \Gamma} \gamma_i \right) \\
&\qquad = - \left(\frac{1}{2} \sum_{i \in (\Gamma \setminus \{k\})} \gamma_i - \frac{1}{3} \sum_{\{i,j\} \subseteq (\Gamma \setminus \{k\})} \gamma_i \gamma_j + \ldots -+ \ldots - (-1)^{m-1} \frac{1}{2 + (m-1)} \prod_{i \in (\Gamma \setminus \{k\})} \gamma_i \right) \\
&\qquad = -\operatorname{SOMS}_2(\Gamma - \{k\})
\end{aligned}$$

□

C.4 Bounding SOMS

Proposition C.12. (SOMS weight expression) *For $\ell \in \mathbb{W}$ and $h \in \mathbb{N}$,*

$$\sum_{k=0}^{\ell} \binom{\ell}{k} (-1)^k \frac{k!(h-1)!}{(k+h)!} = \frac{1}{\ell + h}. \tag{C.15}$$

Proof of Proposition C.12.

$$\begin{aligned}
\sum_{k=0}^{\ell} \binom{\ell}{k} (-1)^k \frac{k!(h-1)!}{(k+h)!} &= \sum_{k=0}^{\ell} \frac{\ell!}{k!(\ell-k)!} (-1)^k \frac{k!(h-1)!}{(k+h)!} \\
&= \ell!(h-1)! \sum_{k=0}^{\ell} \frac{(-1)^k}{(k+h)!(\ell-k)!} \\
&= \ell! \frac{h!}{h} \left(\frac{1}{h!} \frac{1}{\ell!} - \frac{1}{(h+1)h!} \frac{1}{(\ell-1)!} + \ldots -+ \ldots + \frac{(-1)^\ell}{(\ell+h)!} \right) \\
&= \left(\frac{1}{h} - \frac{\ell}{(h+1)h} + \frac{\ell(\ell-1)}{(h+2)(h+1)h} - \ldots +- \ldots + \frac{(-1)^\ell \ell!(h-1)!}{(\ell+h)!} \right) \\
&= \left(\frac{1}{h} - \frac{\ell}{h} \left(\frac{1}{h+1} - \frac{\ell-1}{h+1} \left(\frac{1}{h+2} - \cdots \right) \right) \right) \\
&= f_0
\end{aligned}$$

where

$$f_k \triangleq \frac{1}{h+k} - \frac{\ell-k}{h+k} f_{k+1}. \tag{C.16}$$

Clearly, $f_\ell = 1/(h+\ell)$. By Eq. (C.16), if $f_k = 1/(h+\ell)$ for some $k \in \{1, \ldots, \ell\}$, then

$$\begin{aligned}
f_{k-1} &= \frac{1}{h+k-1} - \frac{\ell-k+1}{h+k-1} f_k = \frac{1}{h+k-1} - \frac{\ell-k+1}{h+k-1} \frac{1}{h+\ell} \\
&= \frac{(h+\ell) - (\ell-k+1)}{(h+k-1)(h+\ell)} = \frac{h+k-1}{(h+k-1)(h+\ell)} = \frac{1}{h+\ell}.
\end{aligned}$$

By induction, Eq. (C.16) is true for all $k \in \{0, 1, \ldots, \ell\}$, and so $f_0 = 1/(\ell + h)$. □

Theorem C.3. (Transformation of SOMS) *Let* $\Gamma \subseteq \mathcal{I}$*,* $m \triangleq |\Gamma|$*, and* $h \in \mathbb{N}$*. Then* $\mathrm{SOMS}_h(\Gamma)$ *is equal to*

$$\begin{aligned}
&\frac{1}{h} \prod_{i \in \Gamma} (1-\gamma_i) \\
&+ \frac{1}{(h+1)h} \sum_{i \in \Gamma} \left(\gamma_i \prod_{j \in (\Gamma \setminus \{i\})} (1-\gamma_j) \right) \\
&+ \frac{2}{(h+2)(h+1)h} \sum_{\{i,j\} \subseteq \Gamma} \left(\gamma_i \gamma_j \prod_{k \in (\Gamma \setminus \{i,j\})} (1-\gamma_k) \right) \\
&+ \cdots \\
&+ \frac{m!(h-1)!}{(h+m)!} \prod_{i \in \Gamma} \gamma_i.
\end{aligned}$$

Proof of Theorem C.3. By definition,

$$\begin{aligned}
\mathrm{SOMS}_h(\Gamma) \triangleq \frac{1}{h} - \frac{1}{h+1} \sum_{i \in \Gamma} \gamma_i + \frac{1}{h+2} \sum_{\{i,j\} \subseteq \Gamma} \gamma_i \gamma_j \\
- \ldots +- \ldots + (-1)^\ell \frac{1}{h+\ell} \sum_{\substack{\mathcal{C} \subseteq \Gamma \\ |\mathcal{C}| = \ell}} \left(\prod_{i \in \mathcal{C}} \gamma_i \right) - \ldots +- \ldots \\
+ (-1)^m \frac{1}{h+m} \prod_{i \in \Gamma} \gamma_i,
\end{aligned}$$

but, by Proposition C.12,

$$\frac{1}{h+\ell} = \sum_{k=0}^{\ell} \binom{\ell}{k} (-1)^k \frac{k!(h-1)!}{(k+h)!},$$

and so Theorem C.1 applies with

$$b_\ell = \sum_{k=0}^{\ell} \binom{\ell}{k} (-1)^k \underbrace{\frac{k!(h-1)!}{(k+h)!}}_{a_k}$$

for each $\ell \in \{0, 1, \ldots, m\}$ and $a_k \triangleq k!(h-1)!/(k+h)!$ for each $k \in \{0, 1, \ldots, m\}$. □

Proposition C.13. (Coefficient montonocity) *Take* $m \in \mathbb{W}$ *and* $h \in \mathbb{N}$*. Let*

$$a_k \triangleq \frac{k!(h-1)!}{(h+k)!} = \frac{k!}{(h+k)(h+k-1)\cdots(h+1)(h)} = \frac{1}{h} \prod_{i=1}^{k} \frac{i}{h+i}.$$

Then $a_0 > a_1 > a_2 > \cdots > a_m > 0$.

Proof of Proposition C.13. Because $h \geq 1$,

$$a_1 = \frac{1}{h} \frac{1}{h+1} < \frac{1}{h} = a_0.$$

Assuming that $a_k < a_{k-1}$ for some $k \in \{1, 2, \ldots, m\}$, then

$$a_{k+1} = a_k \frac{1}{h+k} < a_k$$

because $h \geq 1$ and $k \geq 1$. Hence, $a_k \geq a_{k+1}$ for all $k \in \{0, 1, \ldots, m-1\}$ by induction. Further, because a_m is a product of strictly positive factors, $a_k > 0$ for all $k \in \{0, 1, \ldots, m\}$. □

Proposition C.14. (SOMS lower bound) *Let* $\mathcal{X} \subseteq [0, 1]$ *in Eq. (C.2). For* $\Gamma \subseteq \mathcal{I}$ *and* $h \in \mathbb{N}$,

$$\mathrm{SOMS}_h(\Gamma) \geq \frac{m!(h-1)!}{(h+m)!} = \frac{m!}{(h+m)(h+m-1)\cdots(h+1)h} = \frac{1}{h}\prod_{k=1}^{m}\frac{k}{h+k}$$

where $m \triangleq |\Gamma|$.

Proof of Proposition C.14. Apply Theorem C.3 to $\mathrm{SOMS}_h(\Gamma)$ and then, using the greatest lower bound implied from Proposition C.13, apply Proposition C.4 with $a_k \triangleq k!(h-1)!/(h+k)!$ for all $k \in \{0, 1, \ldots, |\Gamma|\}$. □

Proposition C.15. (SOMS upper bound) *Let* $\mathcal{X} \subseteq [0, 1]$ *in Eq. (C.2). For* $\Gamma \subseteq \mathcal{I}$ *and* $h \in \mathbb{N}$, $\mathrm{SOMS}_h(\Gamma) \leq 1/h$.

Proof of Proposition C.15. Apply Theorem C.3 to $\mathrm{SOMS}_h(\Gamma)$ and then, using the upper bound implied from Proposition C.13, apply Proposition C.5 with $a_k \triangleq k!(h-1)!/(h+k)!$ for all $k \in \{0, 1, \ldots, |\Gamma|\}$. □

D Parallel and distributed computation

Unless otherwise noted, the following results and definitions are either taken from, based upon, or highly influenced by Bertsekas and Tsitsiklis [8].

D.1 Vector spaces

In a vector space $\mathcal{V} \subseteq \mathbb{R}^n$, $a \in \mathbb{R}$ is a scalar and $\underline{v} \in \mathcal{V}$ is a vector with elements that might be shown in coordinate notation as $(v_1, v_2, \ldots, v_n)$ or in vector notation as $[v_1, v_2, \ldots, v_n]^\top$ where $\top$ indicates an element-wise row–column transposition. In the case where vector space $\mathcal{V}$ is a Cartesian product of other vector spaces, the elements (or coordinates) of vector $\underline{v} \in \mathcal{V}$ may themselves be vectors (e.g., $\underline{v}_1 \in \mathcal{V}_1$ where $\mathcal{V} = \mathcal{V}_1 \times \mathcal{V}_2$).

The topology of each vector space (i.e., the definition of its open sets) is induced from a metric (i.e., a measure of distance between points) that is induced from a norm (i.e., a measure of the length of a vector). The standard 1-norm, 2-norm, and maximum norms are used. Any other norms will be defined as necessary.

Assumption D.1. (Cartesian product assumption) *Without loss of generality, represent the Euclidean space $\mathbb{R}^n$ as the Cartesian product $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \cdots \times \mathbb{R}^{n_m}$ where $n_1 + \cdots + n_m = n$ and $n_i \geq 1$ for each $i \in \{1, 2, \ldots, m\}$. Hence, a vector $\underline{x} \in \mathbb{R}^n$ will be represented as $(\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_m)$ where $\underline{x}_i \triangleq (\underline{x}_{i1}, \underline{x}_{i2}, \ldots, \underline{x}_{in_i}) \in \mathbb{R}^{n_i}$ for each $i \in \{1, \ldots, m\}$. Assume that set $\mathcal{X} \subseteq \mathbb{R}^n$ is the Cartesian product $\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m$, where $\mathcal{X}_i$ is a nonempty subset of $\mathbb{R}^{n_i}$ for each $i \in \{1, \ldots, m\}$. Likewise, a vector $\underline{x} \in \mathcal{X}$ will be represented as $(\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_m)$ where $\underline{x}_i \in \mathcal{X}_i$ for each $i \in \{1, \ldots, m\}$. Assume that subspace $\mathbb{R}^{n_i}$ is endowed with norm $\|\cdot\|_i$ for each $i \in \{1, 2, \ldots, m\}$.*

Definition D.1. (Block-maximum norm) Take Assumption D.1 for granted (i.e., $\mathcal{X} \subseteq \mathbb{R}^n$ is a special Cartesian product of normed spaces). The *block-maximum norm* on $\mathbb{R}^n$ for a vector $\underline{x} \in \mathcal{X}$ is

$$\|\underline{x}\| \triangleq \max\{\|\underline{x}_i\|_i : i \in \{1, 2, \ldots, m\}\}.$$

Definition D.2. (Induced matrix norm for product spaces) Without loss of generality, represent the Euclidean space $\mathbb{R}^n$ as the Cartesian product $\mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \cdots \times \mathbb{R}^{n_m}$ where $n_1 + \cdots + n_m = n$. Hence, a vector $\underline{x} \in \mathbb{R}^n$ will be represented as $(\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_m)$ where $\underline{x}_i \in \mathbb{R}^{n_i}$ for each $i \in \{1, \ldots, m\}$. Assume that subspace $\mathbb{R}^{n_i}$ is endowed with norm $\|\cdot\|_i$ for each $i \in \{1, 2, \ldots, m\}$. For any matrix A of dimension $n_i \times n_j$, the induced matrix norm

$$\|A\|_{ij} \triangleq \max\left\{\frac{\|A\underline{x}\|_i}{\|\underline{x}\|_j} : \underline{x} \in \mathbb{R}^{n_j}, \underline{x} \neq \underline{0}\right\} = \max\left\{\|A\underline{x}\|_i : \underline{x} \in \mathbb{R}^{n_j}, \|\underline{x}\|_j = 1\right\}.$$

This definition matches the general definition for induced matrix norms of arbitrary matrices. However, the norm desired for each vector subspace block is made explicit in the notation. That is, the two subscripts indicate the two different norms to be used.

D.2 Functional analysis

Here, definitions and useful results from basic functional analysis are given. Because results will be used in the context of a subspace of the Euclidean $\mathbb{R}^n$ space that has several norms available to it, the following definitions implicitly assume that a sufficient topology (i.e., defined open sets) can be induced from a metric that is induced from a norm. Results and definitions are taken from Rudin [47] and Bertsekas and Tsitsiklis [8].

Definition D.3. (Differentiable vector-valued functions) For $\mathcal{X} \subseteq \mathbb{R}^n$, if $f : \mathcal{X} \mapsto \mathbb{R}^m$ is a vector-valued function where $f \triangleq (f_1, f_2, \dots, f_m)$, it is called *differentiable* if each component $f_i : \mathcal{X} \mapsto \mathbb{R}$ of f is differentiable. Similarly, a vector-valued function is *continuously differentiable* if each of its components are continuously differentiable.

Definition D.4. (Gradient) For $\mathcal{X} \subseteq \mathbb{R}^n$, the *gradient* of scalar-valued continuously differentiable function $f : \mathcal{X} \mapsto \mathbb{R}$ at a point $\underline{x} \triangleq (x_1, x_2, \dots, x_n) \in \mathcal{X}$ is the column vector

$$\nabla f(\underline{x}) \triangleq \begin{bmatrix} \frac{\partial f(\underline{x})}{\partial x_1} \\ \frac{\partial f(\underline{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\underline{x})}{\partial x_n} \end{bmatrix}.$$

Definition D.5. (Jacobian) For $\mathcal{X} \subseteq \mathbb{R}^n$, the *Jacobian* of vector-valued continuously differentiable function $f : \mathcal{X} \mapsto \mathbb{R}^m$ defined with $f \triangleq (f_1, f_2, \dots, f_m)$ at a point $\underline{x} \in \mathcal{X}$ is the transpose of the matrix

$$\nabla f(\underline{x}) \triangleq \begin{bmatrix} \nabla f_1(\underline{x}) & \nabla f_2(\underline{x}) & \cdots & \nabla f_m(\underline{x}) \end{bmatrix}$$

which is a collection of gradients. Hence, because the Jacobian is $(\nabla f(\underline{x}))^\top$, the entry in its i^{th} row and j^{th} column is the partial derivative $\partial f_i / \partial f_j$ evaluated at the point $\underline{x}$.

Definition D.6. (Hessian) Take $\mathcal{X} \subseteq \mathbb{R}^n$ and scalar-valued continuously differentiable function $f : \mathcal{X} \mapsto \mathbb{R}$ at point $\underline{x} \in \mathcal{X}$. If vector-valued gradient $\nabla f : \mathbb{R}^n \mapsto \mathbb{R}^n$ is continuously differentiable at $\underline{x} \triangleq (x_1, x_2, \dots, x_n) \in \mathcal{X}$, then the *Hessian* of f at $\underline{x}$ is

$$\nabla^2 f(\underline{x}) \triangleq \nabla \nabla f(\underline{x}) = \begin{bmatrix} \frac{\partial^2 f(\underline{x})}{\partial {x_1}^2} & \frac{\partial^2 f(\underline{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\underline{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\underline{x})}{\partial x_1 \partial x_2} & \frac{\partial^2 f(\underline{x})}{\partial {x_2}^2} & \cdots & \frac{\partial^2 f(\underline{x})}{\partial x_n \partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\underline{x})}{\partial x_1 \partial x_n} & \frac{\partial^2 f(\underline{x})}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 f(\underline{x})}{\partial {x_n}^2} \end{bmatrix}.$$

That is, the Hessian is the transpose of the Jacobian of the gradient.

Remark (Symmetric Hessian) The Hessian is defined for a scalar-valued continuously differentiable function whose vector-valued gradient is also continuously differentiable; hence, by continuity of the partial derivatives in the Hessian, the Hessian matrix will be symmetric. So the Hessian is the Jacobian of the gradient.

Definition D.7. (Block gradient in Cartesian product space) Take Assumption D.1 for granted (i.e., $\mathcal{X} \subseteq \mathbb{R}^n$ is a special Cartesian product of normed spaces), and take $f : \mathcal{X} \mapsto \mathbb{R}^n$ to be a continuously differentiable vector-valued function. For $i \in \{1, \dots, m\}$ with $\underline{x} = (\underline{x}_1, \dots, \underline{x}_i, \dots, \underline{x}_m) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_m$ and $\underline{x}_i = (x_{i1}, \dots, x_{in_i})$, the *block gradient matrix*

$$\nabla_i f(\underline{x}) \triangleq \begin{bmatrix} \frac{\partial f_1(\underline{x})}{\partial x_{i1}} & \frac{\partial f_2(\underline{x})}{\partial x_{i1}} & \cdots & \frac{\partial f_n(\underline{x})}{\partial x_{i1}} \\ \frac{\partial f_1(\underline{x})}{\partial x_{i2}} & \frac{\partial f_2(\underline{x})}{\partial x_{i2}} & \cdots & \frac{\partial f_n(\underline{x})}{\partial x_{i2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_1(\underline{x})}{\partial x_{in_i}} & \frac{\partial f_2(\underline{x})}{\partial x_{in_i}} & \cdots & \frac{\partial f_n(\underline{x})}{\partial x_{in_i}} \end{bmatrix}$$

is an $n_i \times n$ matrix.

Remark (Relationship to Jacobian) Assume that f has m vector-valued block component functions so that $f \triangleq (f_1, f_2, \ldots, f_m)$. That is, $f_j : \mathcal{X} \mapsto \mathbb{R}^{n_i}$ with $f_j \triangleq (f_{j1}, f_{j2}, \ldots, f_{jn_j})$ for each $j \in \{1, 2, \ldots, m\}$. For $i, j \in \{1, 2, \ldots, m\}$ with $\underline{x} = (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_i, \ldots, \underline{x}_m) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_m$ and $\underline{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in_i})$, the block gradient matrix

$$\nabla_i f_j(\underline{x}) = \begin{bmatrix} \frac{\partial f_{j1}(\underline{x})}{\partial x_{i1}} & \frac{\partial f_{j2}(\underline{x})}{\partial x_{i1}} & \cdots & \frac{\partial f_{jn_i}(\underline{x})}{\partial x_{i1}} \\ \frac{\partial f_{j1}(\underline{x})}{\partial x_{i2}} & \frac{\partial f_{j2}(\underline{x})}{\partial x_{i2}} & \cdots & \frac{\partial f_{jn_j}(\underline{x})}{\partial x_{i2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{j1}(\underline{x})}{\partial x_{in_i}} & \frac{\partial f_{j2}(\underline{x})}{\partial x_{in_i}} & \cdots & \frac{\partial f_{jn_j}(\underline{x})}{\partial x_{in_i}} \end{bmatrix}$$

is an $n_i \times n_j$ matrix, and the block gradient matrix

$$\nabla_i f(\underline{x}) = \begin{bmatrix} \nabla_i f_1(\underline{x}) & \nabla_i f_2(\underline{x}) & \cdots & \nabla_i f_m(\underline{x}) \end{bmatrix}$$

is an $n_i \times n$ matrix, and the gradient matrix

$$\begin{aligned} \nabla f(\underline{x}) &= \begin{bmatrix} \nabla_1 f(\underline{x}) \\ \nabla_2 f(\underline{x}) \\ \vdots \\ \nabla_m f(\underline{x}) \end{bmatrix} \\ &= \begin{bmatrix} \nabla f_1(\underline{x}) & \nabla f_2(\underline{x}) & \cdots & \nabla f_m(\underline{x}) \end{bmatrix} \\ &= \begin{bmatrix} \nabla_1 f_1(\underline{x}) & \nabla_1 f_2(\underline{x}) & \cdots & \nabla_1 f_m(\underline{x}) \\ \nabla_2 f_1(\underline{x}) & \nabla_2 f_2(\underline{x}) & \cdots & \nabla_2 f_m(\underline{x}) \\ \vdots & \vdots & \nabla_i f_j(\underline{x}) & \vdots \\ \nabla_m f_1(\underline{x}) & \nabla_m f_2(\underline{x}) & \cdots & \nabla_m f_m(\underline{x}) \end{bmatrix} \end{aligned} \tag{D.1}$$

is the transpose of the $n \times n$ Jacobian of f. In the case of a continuously differentiable gradient, the Hessian is symmetric, and so the Hessian will be equivalent to the Jacobian of the gradient. In either case, the block gradient carves out blocks of the Jacobian of vector-valued functions defined on Cartesian product spaces.

Remark (Relationship between Hessian and block gradients) Let $g : \mathcal{X} \mapsto \mathbb{R}$ be a scalar-valued continuously differentiable function, and let $f : \mathcal{X} \mapsto \mathbb{R}^n$ be defined as its vector-valued continuously differentiable gradient. That is, let $f(\underline{x}) \triangleq \nabla g(\underline{x})$ for all $\underline{x} \in \mathcal{X}$. Then $f = (f_1, f_2, \ldots, f_m)$ where $f_i(\underline{x}) = \nabla_i g(\underline{x})$ for all $\underline{x} \in \mathcal{X}$. Further, by Eq. (D.1), for $\underline{x} \in \mathcal{X}$, the Hessian

$$\begin{aligned} \nabla^2 g(\underline{x}) = \nabla \nabla g(\underline{x}) &= \begin{bmatrix} \nabla_1 \nabla g(\underline{x}) \\ \nabla_2 \nabla g(\underline{x}) \\ \vdots \\ \nabla_m \nabla g(\underline{x}) \end{bmatrix} \\ &= \begin{bmatrix} \nabla \nabla_1 g(\underline{x}) & \nabla \nabla_2 g(\underline{x}) & \cdots & \nabla \nabla_m g(\underline{x}) \end{bmatrix} \\ &= \begin{bmatrix} \nabla^2_{11} g(\underline{x}) & \nabla^2_{12} g(\underline{x}) & \cdots & \nabla^2_{1m} g(\underline{x}) \\ \nabla^2_{21} g(\underline{x}) & \nabla^2_{22} g(\underline{x}) & \cdots & \nabla^2_{2m} g(\underline{x}) \\ \vdots & \vdots & \nabla^2_{ij} g(\underline{x}) & \vdots \\ \nabla^2_{m1} g(\underline{x}) & \nabla^2_{m2} g(\underline{x}) & \cdots & \nabla^2_{mm} g(\underline{x}) \end{bmatrix} \end{aligned} \tag{D.2}$$

where

$$\nabla^2_{ij} g(\underline{x}) \triangleq \nabla_i \nabla_j g(\underline{x})$$

is the $n_i \times n_j$ block of $\nabla^2 g(x)$ located at the i^{th} row block the j^{th} column block. For example, if each block is a subset of $\mathbb{R}$ (i.e., $m = n$ and $n_i = 1$ for all $i \in \{1, 2, \ldots, m\}$), $\nabla^2_{ij} g$ is the i^{th} row and the j^{th} column of Hessian $\nabla^2 g$.

D.3 Theory of contractions

Convergence analysis of iterative algorithms is simplified when the algorithms *contract* in some way. Here, contraction mappings are defined and theoretical results are given.

Definition D.8. (Contraction mapping and its modulus) Suppose that $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n$, $T : \mathcal{X} \mapsto \mathcal{Y}$, and

$$\|T(\underline{x}) - T(\underline{y})\| \leq \alpha \|\underline{x} - \underline{y}\| \qquad \text{for all } \underline{x}, \underline{y} \in \mathcal{X}$$

where $\|\cdot\|$ is a norm endowed to the corresponding subspace and $\alpha \in [0, 1)$. The mapping T is a *contraction mapping* and α is the *modulus* of T.

Remark (Lipschitz continuity of contraction mappings) Any contraction mapping T is automatically Lipschitz continuous.

Definition D.9. (Block contraction over Cartesian product sets) Take Assumption D.1 for granted (i.e., $\mathcal{X} \subseteq \mathbb{R}^n$ is a special Cartesian product of normed spaces), and assume that $\mathbb{R}^n$ is endowed with the block-maximum norm $\|\cdot\|$. A contraction $T : \mathcal{X} \mapsto \mathcal{X}$ under this block-maximum norm with modulus α is called a *block contraction*.

Definition D.10. (Block component of a block contraction) Take Assumption D.1 for granted (i.e., $\mathcal{X} \subseteq \mathbb{R}^n$ is a special Cartesian product of normed spaces), and assume that $\mathbb{R}^n$ is endowed with the block-maximum norm $\|\cdot\|$. For a block contraction $T : \mathcal{X} \mapsto \mathcal{X}$, a mapping $T_i : \mathcal{X} \mapsto \mathcal{X}_i$ can be defined as the i^{th} *block component* of T. That is, for $\underline{x} \in \mathcal{X}$,

$$T(\underline{x}) \triangleq (T_1(\underline{x}), T_2(\underline{x}), \ldots T_m(\underline{x})).$$

Proposition D.1. (Block component is a contraction) *Take Assumption D.1 for granted (i.e., $\mathcal{X} \subseteq \mathbb{R}^n$ is a special Cartesian product of normed spaces), and assume that $\mathbb{R}^n$ is endowed with the block-maximum norm $\|\cdot\|$. Mapping $T : \mathcal{X} \mapsto \mathcal{X}$ is a contraction with modulus α if and only if the block component $T_i : \mathcal{X} \mapsto \mathcal{X}_i$ is itself a contraction with modulus α for every $i \in \{1, 2, \ldots, m\}$.*

Proof of Proposition D.1 is omitted for brevity. □

Definition D.11. (Contracting iterations) Suppose that $\mathcal{X} \subseteq \mathbb{R}^n$, $T : \mathcal{X} \mapsto \mathcal{X}$ is a contraction mapping, and the sequence $\{\underline{x}(t)\}$ is such that

$$\underline{x}(t+1) = T(\underline{x}(t)) \qquad \text{where } t \in \mathbb{W}. \tag{D.3}$$

The iteration in Eq. (D.3) is a *contracting iteration*.

Definition D.12. (Fixed point) Suppose that $\mathcal{X} \subseteq \mathbb{R}^n$, and let there be a mapping $T : \mathcal{X} \mapsto \mathcal{X}$. Any vector $\underline{x}^* \in \mathcal{X}$ satisfying $\underline{x}^* = T(\underline{x}^*)$ is a *fixed point* of T.

Remark (Algorithm to find fixed points of contractions) The contracting iteration corresponding to contraction T may be viewed as an algorithm for finding the fixed point of T.

Proposition D.2. (Convergence of contracting iterations) *If $\mathcal{X} \subseteq \mathbb{R}^n$ is closed and convex and $T : \mathcal{X} \mapsto \mathcal{X}$ is a contraction with modulus $\alpha \in [0, 1)$, then*

(a) (Existence and uniqueness of fixed points) *The mapping T has a unique fixed point $\underline{x}^* \in \mathcal{X}$.*

(b) (Geometric convergence) *For every initial vector* $\underline{x}(0) \in \mathcal{X}$*, the sequence* $\{\underline{x}(t)\}$ *generated by the contracting iteration* $\underline{x}(t+1) = T(\underline{x}(t))$ *converges to* $\underline{x}^*$ *geometrically with rate* α*. In particular,*

$$\|\underline{x}(t) - \underline{x}^*\| \leq \alpha^t \|\underline{x}(0) - \underline{x}^*\| \qquad \text{for all } t \in \mathbb{W}.$$

Proof of Proposition D.2 is given by Bertsekas and Tsitsiklis [8]. □

D.3.1 Simple linear mapping

The following theorems each take Assumption D.1 for granted (i.e., $\mathcal{X} \subseteq \mathbb{R}^n$ is a special Cartesian product of normed spaces) and provide contracting conditions for the mapping $T : \mathcal{X} \mapsto \mathbb{R}^n$ where, for each $i \in \{1, 2, \ldots, m\}$, i^{th} block-component $T_i : \mathcal{X} \mapsto \mathbb{R}^{n_i}$ is of the form

$$T_i(\underline{x}) \triangleq \underline{x}_i - \sigma G_i^{-1} f_i(\underline{x}) \qquad \text{for all } \underline{x} \in \mathcal{X} \tag{D.4}$$

where $f_i : \mathcal{X} \mapsto \mathbb{R}^{n_i}$ is a function, G_i is a symmetric positive definite matrix, and $\sigma > 0$ is a scalar. Mappings of this form are used in Section D.5.2.

Proposition D.3. (Block-maximum contraction on convex sets) *Take Assumption D.1 for granted (i.e.,* $\mathcal{X} \subseteq \mathbb{R}^n$ *is a special Cartesian product of normed spaces), and assume that* $\mathbb{R}^n$ *is endowed with the block-maximum norm* $\|\cdot\|$*. For each* $i \in \{1, 2, \ldots, m\}$*, let* G_i *be an invertible symmetric matrix of dimensions* $n_i \times n_i$*. Also let* $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ *be a continuously differentiable function defined by* $f \triangleq (f_1, f_2, \ldots, f_m)$ *where* $f_i : \mathbb{R}^n \mapsto \mathbb{R}^{n_i}$ *is a continuously differentiable function for each* $i \in \{1, 2, \ldots, m\}$*. Take* I *to be the identity matrix and* $\sigma \in \mathbb{R}$ *to be some scalar. Assume that* $\mathcal{X}$ *is convex and that there exists a scalar* $\alpha \in [0, 1)$ *such that*

$$\left\|I - \sigma G_i^{-1} (\nabla_i f_i(\underline{x}))^\top\right\|_{ii} + \sum_{\substack{j \in \{1,2,\ldots,n\} \\ j \neq i}} \left\|\sigma G_i^{-1} (\nabla_j f_i(\underline{x}))^\top\right\|_{ij} \leq \alpha \tag{D.5}$$

for all $\underline{x} \in \mathcal{X}$ *and* $i \in \{1, 2, \ldots, n\}$ *where* $\|\cdot\|_{ij}$ *is the induced matrix norm from Definition D.2 acting on matrices of dimension* $n_i \times n_j$*; that is,* $\|A\|_{ij} \triangleq \max\{\|A\underline{x}\|_i : \underline{x} \in \mathbb{R}^{n_j}, \|\underline{x}\|_j = 1\}$*. Then the mapping* $T : \mathcal{X} \mapsto \mathbb{R}^n$ *defined by block component*

$$T_i(\underline{x}) \triangleq \underline{x}_i - \sigma G_i^{-1} f_i(\underline{x}) \qquad \text{for all } i \in \{1, 2, \ldots, m\} \tag{D.6}$$

is a contraction under the block-maximum norm $\|\cdot\|$ *with modulus* α*.*

Proof of Proposition D.3 is given by Bertsekas and Tsitsiklis [8]. □

Proposition D.4. (Block-maximum contraction for scalar blocks of convex sets) *Take Assumption D.1 for granted (i.e.,* $\mathcal{X} \subseteq \mathbb{R}^n$ *is a special Cartesian product of normed spaces), and assume that* $\mathbb{R}^n$ *is endowed with the block-maximum norm* $\|\cdot\|$*. Also assume that:*

(i) $n_i = 1$ *for all* $i \in \{1, 2, \ldots, m\}$ *(i.e., assume* $m = n$*). That is, assume each Cartesian block factor is one dimensional.*

(ii) *The set* $\mathcal{X}$ *is convex and* $f : \mathcal{X} \mapsto \mathbb{R}^n$ *is continuously differentiable.*

(iii) *There exists positive constant* $K > 0$ *such that*

$$\nabla_i f_i(\underline{x}) \leq K \qquad \text{for all } \underline{x} \in \mathcal{X} \quad \text{and} \quad i \in \{1, 2, \ldots, n\}.$$

(iv) *There exists some* $\beta > 0$ *such that*

$$\sum_{\substack{j \in \{1,2,\ldots,n\} \\ j \neq i}} |\nabla_j f_i(\underline{x})| \leq \nabla_i f_i(\underline{x}) - \beta \qquad \text{for all } \underline{x} \in \mathcal{X} \quad \text{and} \quad i \in \{1, 2, \ldots, n\}.$$

Then the mapping $T : \mathcal{X} \mapsto \mathbb{R}^n$ defined by $T(\underline{x}) \triangleq \underline{x} - \sigma f(\underline{x})$ with $0 < \sigma < 1/K$ is a contraction with respect to the maximum norm.

Proof of Proposition D.4 is given by Bertsekas and Tsitsiklis [8]. □

Remark (Impact of step size on convergence) As shown in the proof of Proposition D.4, a large step size σ will reduce the contraction modulus and lead to faster convergence. The size of σ is limited by the reciprocal of the constant K which is an upper bound for each block gradient. Hence, the steeper the gradients, the faster convergence is possible.

The following theorems handle the simple linear mapping in Eq. (D.4) when matrix G_i is the identity matrix I and each Cartesian factor block is endowed with the quadratic (i.e., ℓ_2) norm.

D.4 Constrained optimization

The following theory is motivated by the problem of minimizing a cost function $F : \mathcal{X} \mapsto \mathbb{R}$ where $\mathcal{X} \subseteq \mathbb{R}^n$ is non-empty, closed, and convex. In most cases, $\mathcal{X} \subset \mathbb{R}^n$, and so F is being minimized subject to *constraints* that are characterized by subset $\mathcal{X}$. Hence, $\mathcal{X}$ will be called the *constraint set*. The orthogonal projection methods discussed in Section D.4.1 allow for the design of iterative algorithms that must work within convex constraint sets.

Proposition D.5. (Optimality conditions)

(a) *If a vector $\underline{x} \in \mathcal{X}$ minimizes F over $\mathcal{X}$, then $(\underline{y} - \underline{x})^\top \nabla F(\underline{x}) \geq 0$ for every $\underline{y} \in \mathcal{X}$.*

(b) *If F is also convex on the set $\mathcal{X}$, then condition (a) is also sufficient for $\underline{x}$ to minimize F over $\mathcal{X}$.*

Proof of Proposition D.5 is given by Bertsekas and Tsitsiklis [8]. □

Remark (Geometric interpretation) Condition (a) in Proposition D.5 states that at the minimum of F on $\mathcal{X}$, displacement and gradient vectors are always pointing in roughly the same direction. It is necessary to climb the gradient in order to move away from the minimum.

D.4.1 Orthogonal projections

To deal with constraints, methods for iterating within the constrained convex set need to be introduced.

Proposition D.6. (Projection theorem)

(a) *For every $\underline{x} \in \mathbb{R}^n$, there exists a unique $[\underline{x}]^+ \in \mathcal{X}$ that minimizes $\|\underline{z} - \underline{x}\|_2$.*

(b) *Given some $\underline{x} \in \mathbb{R}^n$, a vector $\underline{z} \in \mathcal{X}$ is equal to $[\underline{x}]^+$ if and only if $(\underline{y} - \underline{z})^\top (\underline{x} - \underline{z}) \leq 0$ for all $\underline{y} \in \mathcal{X}$.*

(c) *The mapping $f : \mathbb{R}^n \mapsto \mathcal{X}$ defined by $f(\underline{x}) \triangleq [\underline{x}]^+$ is continuous and non-expansive with respect to the ℓ_2-norm. That is, $\|[\underline{x}]^+ - [\underline{y}]^+\|_2 \leq \|\underline{x} - \underline{y}\|_2$ for all $\underline{x}, \underline{y} \in \mathbb{R}^2$.*

Proof of Proposition D.6 is given by Bertsekas and Tsitsiklis [8]. □

Definition D.13. (Orthogonal projection) For a vector $\underline{x} \in \mathbb{R}^n$, the *orthogonal projection* of $\underline{x}$ onto convex set $\mathcal{X}$ is

$$[\underline{x}]^+ \triangleq \arg\min_{z \in \mathcal{X}} \|\underline{z} - \underline{x}\|_2.$$

Remark (Orthogonal projection is well defined) By Proposition D.6, the orthogonal projection is well defined.

Remark (Interpretation) Given a convex set $\mathcal{C} \subseteq \mathcal{X}$, a point $\underline{x} \in \mathcal{X}$, and the orthogonal projection $[\underline{x}]^+$ of $\underline{x}$ onto $\mathcal{C}$,

- $[\underline{x}]^+$ is colinear with a line that is orthogonal to a hypersurface that is tangent to $\mathcal{C}$ at the point $[\underline{x}]^+ \in \mathcal{C}$.
- $[\underline{x}]^+$ minimizes the 2-norm (i.e., "squared") distance between $\mathcal{C}$ and $\underline{x}$.

Definition D.14. (Interval) An *interval* $\mathcal{I} \subseteq \mathbb{R}$ has the property that if $x \in \mathcal{I}$, $y \in \mathcal{I}$, and there exists some $z \in \mathbb{R}$ such that $x \leq z \leq y$ then $z \in \mathcal{I}$.

Proposition D.7. (Projection onto closed interval) *For $x \in \mathbb{R}$, the closed interval $\mathcal{I}$, and the orthogonal projection $[x]^+$ onto $\mathcal{I}$,*

(a) *If $x \in \mathcal{I}$, then $[x]^+ = x$.*

(b) *If $\inf \mathcal{I} = a$ and $x < a$, then $[x]^+ = a$.*

(c) *If $\sup \mathcal{I} = b$ and $x > b$, then $[x]^+ = b$.*

Proof of Proposition D.7 is omitted for brevity. □

Remark (Closed-form projection onto closed interval) By Proposition D.7, for $x \in \mathbb{R}$ and closed interval $\mathcal{I} \triangleq [a, b]$,

$$[x]^+ = \max\{a, \min\{b, x\}\}.$$

Proposition D.8. (Projection onto product space) *Take Assumption D.1 for granted (i.e., constraint set $\mathcal{X} \subseteq \mathbb{R}^n$ is a special Cartesian product), and assume that nonempty subspace $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is closed and convex for each $i \in \{1, \ldots, m\}$. The orthogonal projection $[\underline{x}]^+$ of $\underline{x} \in \mathbb{R}^n$ onto $\mathcal{X}$ is such that*

$$[\underline{x}]^+ = ([\underline{x}_1]_1^+, [\underline{x}_2]_2^+, \ldots, [\underline{x}_m]_m^+), \tag{D.7a}$$

where

$$[\underline{x}_i]_i^+ \triangleq \arg\min_{\underline{z}_i \in \mathcal{X}_i} \|\underline{z}_i - \underline{x}_i\|_2 \tag{D.7b}$$

is the orthogonal projection of $\underline{x}_i \in \mathbb{R}^{n_i}$ onto $\mathcal{X}_i$ for each $i \in \{1, 2, \ldots, m\}$.

Proof of Proposition D.8 is omitted for brevity. □

Remark (Projection onto product of closed intervals) By Proposition D.8, when $\mathcal{X}$ is the Cartesian product of closed intervals $\mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_n$, the projection of $\underline{x} \in \mathbb{R}^n$ onto $\mathcal{X}$ is obtained by using the simple result of Proposition D.7 to project x_i onto $\mathcal{I}_i$ for each $i \in \{1, 2, \ldots, n\}$.

Remark (Parallelization of product projection) Because projection onto a Cartesian product of n intervals can be completed with n simple, separate, and independent scalar projections, computation of $[\underline{x}]^+$ can be done in parallel on n independent agents.

D.5 Variational inequalities and parallel implementation

Constrained and unconstrained optimization, like the motivating problem for the theory in Section D.4, can be formulated as a *variational inequality problem,* and these problems can be easily parallelized under special conditions on the constraint set $\mathcal{X}$, which will be assumed to be nonempty, closed, and convex.

Definition D.15. (Variational inequality) Given a set $\mathcal{X}$ and a function $f : \mathcal{X} \mapsto \mathbb{R}^n$, the *variational inequality problem* $\mathrm{VI}(\mathcal{X}, f)$ finds a vector $\underline{x}^* \in \mathcal{X}$ such that

$$(\underline{x} - \underline{x}^*)^\top f(\underline{x}^*) \geq 0 \qquad \text{for all } \underline{x} \in \mathcal{X}. \tag{D.8}$$

It will be assumed that set $\mathcal{X}$ is nonempty, closed, and convex.

Proposition D.9. (Decomposition lemma) *Take Assumption D.1 for granted (i.e., constraint set $\mathcal{X} \subseteq \mathbb{R}^n$ is a special Cartesian product), and assume that nonempty subspace $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is closed and convex for each $i \in \{1, \ldots, m\}$. Also let $f : \mathcal{X} \mapsto \mathbb{R}^n$ be expressed so that $f(\underline{x}) \triangleq (f_1(\underline{x}), f_2(\underline{x}), \ldots, f_m(\underline{x}))$ where component $f_i : \mathcal{X} \mapsto \mathbb{R}^{n_i}$ for each $i \in \{1, 2, \ldots, m\}$. A vector $\underline{x}^* \in \mathcal{X}$ solves the variational inequality* $\mathrm{VI}(\mathcal{X}, f)$ *if and only if*

$$(\underline{x}_i - \underline{x}_i^*)^\top f_i(\underline{x}^*) \geq 0 \qquad \textit{for all } \underline{x}_i \in \mathcal{X}_i \tag{D.9}$$

for all $i \in \{1, 2, \ldots, m\}$.

Proof of Proposition D.9 is given by Bertsekas and Tsitsiklis [8]. □

D.5.1 Motivation from game theory

Definition D.16. (Nash game) Take Assumption D.1 for granted (i.e., constraint set $\mathcal{X} \subseteq \mathbb{R}^n$ is a special Cartesian product), and assume that nonempty subspace $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is closed and convex for each $i \in \{1, \ldots, m\}$. Consider m players in a game. Each player $i \in \{1, 2, \ldots, m\}$ chooses a strategy $\underline{x}_i \in \mathcal{X}_i$ that either is penalized by an amount equal to $F_i(\underline{x})$ (or, equivalently, is rewarded by an amount equal to $-F_i(\underline{x})$) where $F_i : \mathcal{X} \mapsto \mathbb{R}$ is a continuously differentiable function. A *Nash equilibrium* $\underline{x}^* = (\underline{x}_1^*, \underline{x}_2^*, \ldots, \underline{x}_m^*) \in \mathcal{X}$ is such that

$$F_i(\underline{x}_1^*, \ldots, \underline{x}_{i-1}^*, \underline{x}_i^*, \underline{x}_{i+1}^*, \ldots, \underline{x}_m^*) \leq F_i(\underline{x}_1^*, \ldots, \underline{x}_{i-1}^*, \underline{x}_i, \underline{x}_{i+1}^*, \ldots, \underline{x}_m^*) \qquad \text{for all } \underline{x}_i \in \mathcal{X}_i \tag{D.10}$$

for all $i \in \{1, 2, \ldots, m\}$. In other words, when the m players are in Nash equilibrium, no single player can improve their utility (i.e., reduce their penalty or increase their reward) by unilaterally deviating from the equilibrium.

Remark (Comparing Nash equilibria to optimization) A Nash equilibrium represents a balance among the conflicting interests of all players. In principle, each player may be able to achieve better than the utility at a Nash equilibrium, but that increase is not possible without another player choosing to make a play that returns less utility. A Nash equilibrium represents the outcome of independent players choosing to do as well as possible without any explicit coordination with other players.

Remark (Significance of Cartesian product requirement) A Nash game requires each player to be able to make independent actions, and so the game space is a Cartesian product. If players are able to communicate and move in tandem, the game cannot be described using Nash equilibria unless groups of tandem players are disjoint; in that case, each group must be considered to be a larger player with a higher dimensional play space.

Proposition D.10. (Nash game as variational inequality) *Take Assumption D.1 for granted (i.e., constraint set $\mathcal{X} \subseteq \mathbb{R}^n$ is a special Cartesian product), and assume that nonempty subspace $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is closed and convex for each $i \in \{1, \ldots, m\}$. The corresponding m-player Nash game is a variational inequality* $\mathrm{VI}(\mathcal{X}, f)$ *with*

$$f(\underline{x}) \triangleq (f_1(\underline{x}), f_2(\underline{x}), \ldots, f_m(\underline{x}))$$

where

$$f_i(\underline{x}) \triangleq \nabla_i F_i(\underline{x})$$

for each $\underline{x} \in \mathcal{X}$ and each $i \in \{1, 2, \ldots, m\}$.

Proof of Proposition D.10. By the optimality conditions from Proposition D.5, for player $i \in \{1, 2, \ldots, m\}$ whose play is independent of other play j with $j \neq i$, play $\underline{x}_i^* \in \mathcal{X}_i$ is optimal over convex set $\mathcal{X}_i$ if and only if

$$(\underline{x}_i - \underline{x}_i^*)^\top \nabla_i F_i(\underline{x}^*) \qquad \text{for all } \underline{x}_i \in \mathcal{X}_i$$

where $\nabla_i F_i(\underline{x})$ is the block gradient from Definition D.7. This form matches Eq. (D.9) from Proposition D.9 (i.e., the decomposition lemma). □

D.5.2 Projection algorithm

Because of the non-expansive and continuous properties of the orthogonal projection discussed in Proposition D.6(c), the projection algorithm is a special case of the simple linear mapping discussed in Section D.3.1.

Definition D.17. (Projection iteration) The *projection iteration*, defined by

$$\underline{x}(t+1) = T_p(\underline{x}(t)) \triangleq [R_p(\underline{x}(t))]^+ \triangleq [\underline{x}(t) - \sigma f(\underline{x}(t))]^+ \qquad \text{for all } t \in \mathbb{W} \tag{D.11}$$

where $[\cdot]^+$ is the orthogonal projection onto set $\mathcal{X}$ and *step size* scalar $\sigma > 0$, will be used to computationally find solutions to the variational inequality problem $\mathrm{VI}(\mathcal{X}, f)$.

Proposition D.11. (Fixed point characterization of solutions) *Suppose scalar $\sigma > 0$. A vector $\underline{x}^* \in \mathcal{X}$ is a solution of $\mathrm{VI}(\mathcal{X}, f)$ if and only if $T_p(\underline{x}^*) = \underline{x}^*$ where T_p is the mapping defined in Eq. (D.11).*

Proof of Proposition D.11 is given by Bertsekas and Tsitsiklis [8]. □

D.6 Totally asynchronous iterative distributed algorithms

Here, sufficient conditions for the convergence of a totally asynchronous iterative algorithm are given.

Assumption D.2. (Distributed block topology) *Let $\mathcal{X}$ be the Cartesian product of the $m \in \mathbb{N}$ given nonempty sets $\mathcal{X}_1$, $\mathcal{X}_2$, …, $\mathcal{X}_m$. That is,*

$$\mathcal{X} \triangleq \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m$$

so that for each $\underline{x} \in \mathcal{X}$,

$$\underline{x} \triangleq (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_m)$$

where $\underline{x}_i \in \mathcal{X}_i$ for each $i \in \{1, 2, \ldots, m\}$. Assume that $\mathcal{X}$ has an appropriate notion of convergence defined (e.g., it is a Hausdorff topological space).

Definition D.18. (Totally asynchronous distributed iterations) Take Assumption D.2 for granted. Let $f : \mathcal{X} \mapsto \mathcal{X}$ be a function with i^{th} block component $f_i : \mathcal{X} \mapsto \mathcal{X}_i$ so that

$$f(\underline{x}) \triangleq (f_1(\underline{x}), f_2(\underline{x}), \ldots, f_m(\underline{x}))$$

for all $\underline{x} \in \mathcal{X}$. Assume that there is an element $\underline{x}^* \in \mathcal{X}$ that is a fixed point of f. That is,

$$\underline{x}^* = f(\underline{x}^*) \quad \text{and} \quad \underline{x}_i^* = f_i(\underline{x}^*) \quad \text{for all } i \in \{1, 2, \ldots, m\}.$$

Let $\mathcal{T} \triangleq \mathbb{W}$ be the indices of a sequence of physical times. The *system state* trajectory $\underline{x}(t) \triangleq (\underline{x}_1(t), \underline{x}_2(t), \ldots, \underline{x}_n(t))$ is defined for all $t \in \mathcal{T}$. For each $i \in \{1, 2, \ldots, m\}$, there is a subset

$\mathcal{T}^i \subseteq \mathcal{T}$ representing indices of physical times corresponding to when block i computes its next iteration. Additionally, for each $i, j \in \{1, 2, \dots, m\}$ and $t \in \mathcal{T}$, there is an index

$$\tau^i_j(t) \in \mathcal{T} \quad \text{such that} \quad 0 \leq \tau^i_j(t) \leq t \tag{D.12a}$$

of the least-outdated version of system state block $\underline{x}_j(t)$ available to block i at time t. Hence, each block $i \in \{1, 2, \dots, m\}$ has access to an *outdated state estimate*

$$\underline{x}^i(t) \triangleq \left(\underline{x}^i_1(t), \underline{x}^i_2(t), \dots, \underline{x}^i_m(t)\right) \triangleq \left(\underline{x}_1(\tau^i_1(t)), \underline{x}_2(\tau^i_2(t)), \dots, \underline{x}_m(\tau^i_m(t))\right) \tag{D.12b}$$

for each $t \in \mathcal{T}$. So, for all $t \in \mathcal{T}$, the system state trajectory sequence $\{\underline{x}(t)\}$ is generated by the *totally asynchronous distributed iteration (TADI)*

$$\underline{x}_i(t+1) = \begin{cases} f_i(\underline{x}^i(t))) & \text{if } t \in \mathcal{T}^i, \\ \underline{x}_i(t) & \text{if } t \notin \mathcal{T}^i \end{cases} \tag{D.12c}$$

where $\underline{x}(t) = (\underline{x}_1(t), \underline{x}_2(t), \dots, \underline{x}_n(t))$.

Assumption D.3. (Total asynchronism) *Take Assumption D.2 for granted. For each $i \in \{1, 2, \dots, m\}$,*

(i) *The set $\mathcal{T}^i$ used in Eq. (D.12c) is infinite (i.e., $|\mathcal{T}^i| = |\mathcal{T}| = |\mathbb{N}|$).*

(ii) *If the sequence $\{t_k\}$ generated by taking $t_k \in \mathcal{T}^i$ is such that $\lim_{k\to\infty} t_k = \infty$, then $\lim_{k\to\infty} \tau^i_j(t_k) = \infty$ for all $j \in \{1, 2, \dots, m\}$.*

Remark (Inferior limit of update times is infinite) Under Assumption D.3, for all $i, j \in \{1, 2, \dots, m\}$,

$$\liminf_{t\to\infty} \tau^i_j(t) = \infty,$$

and thus

$$\limsup_{t\to\infty} \tau^i_j(t) = \liminf_{t\to\infty} \tau^i_j(t) = \lim_{t\to\infty} \tau^i_j(t) = \infty.$$

Proposition D.12. (TADI limits are fixed points) *Take Assumptions D.2 and D.3 for granted and let function $f : \mathcal{X} \mapsto \mathcal{X}$ be as in Definition D.18. Assume that:*

(i) *For each $i \in \{1, 2, \dots, m\}$, there is a sequence of nonempty sets $\{\mathcal{X}_i(k)\}_{k\in\mathbb{W}}$ where*

$$\cdots \subset \mathcal{X}_i(k+1) \subset \mathcal{X}_i(k) \subset \cdots \subset \mathcal{X}_i(0) \subseteq \mathcal{X}_i. \tag{D.13}$$

Hence, for all $k \in \mathbb{W}$, there exists nonempty product set

$$\mathcal{X}(k) \triangleq \mathcal{X}_1(k) \times \mathcal{X}_2(k) \times \cdots \times \mathcal{X}_m(k), \tag{D.14}$$

and

$$\cdots \subset \mathcal{X}(k+1) \subset \mathcal{X}(k) \subset \cdots \subset \mathcal{X}(0) \subseteq \mathcal{X}.$$

(ii) *For all $k \in \mathbb{W}$,*

$$f(\underline{x}) \in \mathcal{X}(k+1) \qquad \text{for all } \underline{x} \in \mathcal{X}(k). \tag{D.15}$$

Additionally, if $\{\underline{y}^k\}$ is a sequence such that $\underline{y}^k \in \mathcal{X}(k)$ for every $k \in \mathbb{W}$, then every limit point of $\{\underline{y}^k\}$ is a fixed point of f.

If the initial $\underline{x}(0) \in \mathcal{X}(0)$, then:

(a) *For all $k \in \mathbb{W}$, there exists a $t'_k \in \mathbb{W}$ such that $\underline{x}(t) \in \mathcal{X}(k)$ for all $t \geq t'_k$.*

(b) *For all $k \in \mathbb{W}$, there exists a $t_k^* \in \mathbb{W}$ such that $t_k^* \geq t_k'$ and, for all $i \in \{1, 2, \dots, m\}$, $\underline{x}^i(t) \in \mathcal{X}(k)$ for all $t \geq t_k^*$.*

(c) *Every limit point of the sequence $\{\underline{x}(t)\}$ generated by the totally asynchronous distributed iteration in Eq. (D.12) is a fixed point of f.*

Proof of Proposition D.12 motivated by Bertsekas and Tsitsiklis [8]. By the assumption that $\underline{x}(0) \in \mathcal{X}(0)$, then for each $i \in \{1, 2, \dots, m\}$,

$$\underline{x}^i(0) = (\underline{x}_1(\tau_1^i(0)), \underline{x}_2(\tau_2^i(0)), \dots, \underline{x}_m(\tau_m^i(0))) = (\underline{x}_1(0), \underline{x}_2(0), \dots, \underline{x}_m(0)) = \underline{x}(0) \in \mathcal{X}(0). \tag{D.16}$$

Let $t_0 \in \mathcal{T}$, and assume that both $\underline{x}(t) \in \mathcal{X}(0)$ and $\underline{x}^i(t) \in \mathcal{X}(0)$ for all $t \leq t_0$ and all $i \in \{1, 2, \dots, m\}$. Then, by Eqs (D.14) and (D.15),

$$\underline{x}(t_0 + 1) = (\underbrace{f_1(\overbrace{\underline{x}^1(t_0)}^{\in \mathcal{X}(0)})}_{\substack{\in \mathcal{X}_1(1) \\ \subset \mathcal{X}_1(0)}}, \underbrace{f_2(\overbrace{\underline{x}^2(t_0)}^{\in \mathcal{X}(0)})}_{\substack{\in \mathcal{X}_2(1) \\ \subset \mathcal{X}_2(0)}}, \dots, \underbrace{f_m(\overbrace{\underline{x}^m(t_0)}^{\in \mathcal{X}(0)})}_{\substack{\in \mathcal{X}_m(1) \\ \subset \mathcal{X}_m(0)}}) \in \mathcal{X}(1) \subset \mathcal{X}(0).$$

Additionally, for all $i \in \{1, 2, \dots, m\}$,

$$\underline{x}^i(t_0 + 1) = (\underbrace{\underline{x}_1(\overbrace{\tau_1^i(t_0 + 1)}^{\in \{0, \dots, t_0 + 1\}})}_{\in \mathcal{X}_1(0)}, \underbrace{\underline{x}_2(\overbrace{\tau_2^i(t_0 + 1)}^{\in \{0, \dots, t_0 + 1\}})}_{\in \mathcal{X}_2(0)}, \dots, \underbrace{\underline{x}_m(\overbrace{\tau_m^i(t_0 + 1)}^{\in \{0, \dots, t_0 + 1\}})}_{\in \mathcal{X}_m(0)}) \in \mathcal{X}(0)$$

by Eqs (D.14), (D.15), and the assumption about t_0. As shown in Eq. (D.16), the assumption is certainly true for $t_0 = 0$. So, by induction, $\underline{x}(t) \in \mathcal{X}(0)$ and $\underline{x}^i(t) \in \mathcal{X}(0)$ for all $i \in \{1, 2, \dots, m\}$ and all $t \in \mathcal{T}$.

For some $k \in \mathbb{W}$, assume that there is a time $t_k \in \mathbb{W}$ such that for all $t \in \mathcal{T}$ with $t \geq t_k$,

(i) $\underline{x}(t) \in \mathcal{X}(k)$.

(ii) $\underline{x}^i(t) \in \mathcal{X}(k)$ for all $i \in \{1, 2, \dots, m\}$.

Take $i \in \{1, 2, \dots, m\}$. Let $t^i \triangleq \min\{t \in \mathcal{T}^i : t \geq t_k\}$. That is, t^i is the first element of $\mathcal{T}^i$ such that $t^i \geq t_k$. Because of Assumption D.3 (i.e., $\mathcal{T}^i$ is an infinite subset of $\mathcal{T} \triangleq \mathbb{W}$), t^i is well defined. Then, by Eq. (D.15),

$$\underline{x}_i(t^i + 1) = f_i(\underbrace{\underline{x}^i(t)}_{\in \mathcal{X}(k)}) \in \mathcal{X}_i(k + 1).$$

However, if there exists some $t \in \mathcal{T}$ such that $\underline{x}_i(t + 1) \in \mathcal{X}_i(k + 1)$, then

$$\underline{x}_i(t + 2) = \left\{ \begin{array}{ll} f_i(\underbrace{\underline{x}^i(t + 1)}_{\in \mathcal{X}(k+1)}) \in \mathcal{X}_i(k + 2) \subset \mathcal{X}_i(k + 1) & \text{if } t + 1 \in \mathcal{T}^i, \\ \\ \underline{x}_i(t + 1) \in \mathcal{X}(k + 1) & \text{if } t + 1 \notin \mathcal{T}^i \end{array} \right\} \in \mathcal{X}(k + 1).$$

Hence, because $\underline{x}_i(t^i + 1) \in \mathcal{X}_i(k + 1)$, then $\underline{x}_i(t) \in \mathcal{X}_i(k + 1)$ for all $t \in \mathcal{T}$ with $t \geq t^i + 1$. Let $t_k' \triangleq \max\{t^i + 1 : i \in \{1, 2, \dots, m\}\}$. Then, for all $t \geq t_k'$, $\underline{x}_j(t) \in \mathcal{X}_j(k + 1)$ for each $j \in \{1, 2, \dots, m\}$. Further, by Eq. (D.14),

$$\underline{x}(t) = (\underbrace{\underline{x}_1(t)}_{\in \mathcal{X}_1(k+1)}, \underbrace{\underline{x}_2(t)}_{\in \mathcal{X}_2(k+1)}, \dots, \underbrace{\underline{x}_m(t)}_{\in \mathcal{X}_m(k+1)}) \in \mathcal{X}(k + 1) \qquad \text{for all } t \geq t_k'.$$

Additionally, by Assumption D.3 (i.e., $\liminf_{t\to\infty} \tau_j^i(t) = \infty$), there is a sufficiently large $t_k^* \geq t_k'$ such that $\tau_j^i(t) \geq t_k'$ for all $i, j \in \{1, 2, \ldots, m\}$ and all $t \in \mathcal{T}$ with $t \geq t_k^*$. So, for all $i \in \{1, 2, \ldots, m\}$,

$$\underline{x}^i(t) = (\underbrace{\underline{x}_1(\overbrace{\tau_1^i(t)}^{\geq t_k'})}_{\in \mathcal{X}_1(k+1)}, \underbrace{\underline{x}_2(\overbrace{\tau_2^i(t)}^{\geq t_k'})}_{\in \mathcal{X}_2(k+1)}, \ldots, \underbrace{\underline{x}_m(\overbrace{\tau_m^i(t)}^{\geq t_k'})}_{\in \mathcal{X}_m(k+1)}) \in \mathcal{X}(k+1) \qquad \text{for all } t \geq t_k^* \geq t_k'.$$

Hence, by induction, because $\underline{x}(t) \in \mathcal{X}(0)$ (i.e., with $k = 0$) and, for all $t \in \mathcal{T}$ with $t \geq 0$ and all $i \in \{1, 2, \ldots, m\}$, $\underline{x}^i(t) \in \mathcal{X}(0)$, then the assumption is true for all $k \geq 0$. Hence, conditions (a) and (b) are true under these assumptions.

Assume that $\underline{x}(0) \in \mathcal{X}(0)$ and $\underline{x}^*$ is a limit point of the sequence $\{\underline{x}(t)\}$. For each $k \in \mathbb{W}$, let $t_k \in \mathcal{T}$ be the time in which $\underline{x}(t) \in \mathcal{X}_k$ for all $t \geq t_k$. Because $\{\underline{x}(t)\}$ is convergent, the subsequence $\{\underline{x}(t_k)\}$ is also convergent. However, $\{\underline{x}(t_k)\}$ is a sequence such that $\underline{x}(t_k) \in \mathcal{X}(k)$ for all $k \in \mathbb{W}$. Hence, the limit point $\underline{x}^*$ is a fixed point of f by the second half of condition (ii). Hence, condition (c) is true under these assumptions. □

Remark (Interpretation of Proposition D.12) By conditions (a) and (b), if $\{\underline{X}(k)\}_{k\in\mathbb{W}}$ is sequence that converges to $\underline{x}^* \in \mathcal{X}$, then $\{\underline{x}(t)\}_{t\in\mathcal{T}}$ and $\{\underline{x}^i(t)\}_{t\in\mathcal{T}}$ for all $i \in \{1, 2, \ldots, m\}$ are also sequences that converge to $\underline{x}^*$. By condition (c), if $\underline{x}^* \in \mathcal{X}$ is a point on which sequence $\{\underline{X}(k)\}_{k\in\mathbb{W}}$ converges, then $\underline{x}^*$ is a fixed point of f (i.e., $\underline{x}^* = f(\underline{x}^*)$).

Proposition D.13. (Existence of TADI limit point) *Take the assumptions of Proposition D.12 for granted. Additionally, take $\underline{x}^* \in \mathcal{X}$. Assume that for any set $\mathcal{N} \subseteq \mathcal{X}$ such that $\underline{x}^* \in \mathcal{O} \subseteq \mathcal{N}$ where $\mathcal{O}$ is an open set, there exists a $k \in \mathbb{W}$ such that $\mathcal{X}(k) \subseteq \mathcal{N}$. Then, as $t \to \infty$, $\underline{x}(t) \to \underline{x}^*$ and, for all $i \in \{1, 2, \ldots, m\}$, $\underline{x}^i(t) \to \underline{x}^*$. Additionally, the limit point $\underline{x}^*$ is a fixed point of f (i.e., $\underline{x}^* = f(\underline{x}^*)$).*

Proof of Proposition D.13. Take open set $\mathcal{O}$ such that $\underline{x}^* \in \mathcal{O}$. The assumption states that there exists a $k \in \mathbb{W}$ such that $\mathcal{X}(k) \subseteq \mathcal{O}$. However, by Proposition D.12(a), there exists a $t_k^* \in \mathbb{N}$ such that, for all $t \geq t_k^*$, $\underline{x}(t) \in \mathcal{X}(k) \subseteq \mathcal{O}$ and $\underline{x}^i(t) \in \mathcal{X}(k) \subseteq \mathcal{O}$ for all $i \in \{1, 2, \ldots, m\}$. Hence, as $t \to \infty$, $\underline{x}(t) \to \underline{x}^*$ and $\underline{x}^i \to \underline{x}^*$ for all $i \in \{1, 2, \ldots, m\}$. Additionally, by Proposition D.12(c), the limit point $\underline{x}^*$ is a fixed point of f (i.e., $\underline{x}^* = f(\underline{x}^*)$). □

Proposition D.14. (Maximum norm contraction mappings) *Take Assumption D.1 for granted (i.e., $\mathcal{X} \subseteq \mathbb{R}^n$ is a special Cartesian product of normed spaces), and assume that $\mathbb{R}^n$ is endowed with the block-maximum norm $\|\cdot\|$. Suppose that $f : \mathcal{X} \mapsto \mathbb{R}^n$ is a contraction mapping with respect to the block-maximum norm. Convergence of f to its unique fixed point $\underline{x}^* \in \mathcal{X}$ is guaranteed by Propositions D.12 and D.13.*

Proof of Proposition D.14 motivated by Bertsekas and Tsitsiklis [8]. By Proposition D.2(a), there exists a unique fixed point $\underline{x}^* \in \mathcal{X}$ of contraction f (i.e., $\underline{x}^* = f(\underline{x}^*)$). For each $k \in \mathbb{W}$, define the set

$$\begin{aligned}
\mathcal{X}(k) &\triangleq \{\underline{x} \in \mathcal{X} : \|\underline{x} - \underline{x}^*\| \leq \alpha^k \|\underline{x}(0) - \underline{x}^*\|\} \\
&= \{(\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_m) \in \prod_{i=1}^m \mathcal{X}_i : \|\underline{x}_i - \underline{x}_i^*\|_i \leq \alpha^k \|\underline{x}(0) - \underline{x}^*\| \text{ for all } i \in \{1, 2, \ldots, m\}\} \\
&= \prod_{i=1}^m \underbrace{\{\underline{x}_i \in \mathcal{X}_i : \|\underline{x}_i - \underline{x}_i^*\|_i \leq \alpha^k \|\underline{x}(0) - \underline{x}^*\|\}}_{\triangleq \mathcal{X}_i(k)}
\end{aligned}$$

where $\alpha \in [0, 1)$ is the contraction modulus of f and $\underline{x}(0)$ is the initial TADI system state. Because $\alpha \in [0, 1)$, $\mathcal{X}_i(k+1) \subset \mathcal{X}_i(k)$ for all $i \in \{1, 2, \ldots, m\}$ and all $k \in \mathbb{W}$, and so Eq. (D.14) holds. By Proposition D.2(b), if $\underline{x} \in \mathcal{X}(k)$ then $f(\underline{x}) \in \mathcal{X}(k+1)$ for all $k \in \mathbb{W}$, and so Eq. (D.15) holds. Additionally, $\underline{x}^* \in \mathcal{X}(k)$ for all $k \in \mathbb{W}$. Hence, the collection of sets $\{\mathcal{X}(k) : k \in \mathbb{W}\}$ meets the requirements of Proposition D.12. Further, for any open ball $\mathcal{B}_\varepsilon(\underline{x}^*) \triangleq \{\underline{x} \in \mathcal{X} : \|\underline{x} - \underline{x}^*\| < \varepsilon\}$ around $\underline{x}^*$, there exists a $k \in \mathbb{W}$ such that $\mathcal{X}(k) \subseteq \mathcal{B}_\varepsilon(\underline{x}^*)$. So, by Proposition D.13, convergence of f to fixed point $\underline{x}^*$ is guaranteed. □

References

[1] Altman, E., Kherani, A.A., Michiardi, P., Molva, R.: Non-cooperative forwarding in ad-hoc networks. In: Proceedings of Networking. Lecture Notes in Computer Science, vol. 3462, pp. 486–498 (2005)

[2] Altman, E., Kumar, A., Kumar, D., Venkatesh, R.: Cooperative and non-cooperative control in IEEE 802.11 WLANs. In: Proceedings of the 19th International Teletraffic Congress. Beijing (August 29 – September 2, 2005)

[3] Amir, Y., Awerbuch, B., Barak, A., Borgstrom, R.S., Keren, A.: An opportunity cost approach for job assignment in a scalable computing cluster. IEEE Transactions on Parallel and Distributed Systems 11(7), 760–768 (July 2000)

[4] Archer, A., Tardos, É.: Truthful mechanisms for one-parameter agents. In: Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science. pp. 482–491 (October 8–11, 2001)

[5] Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1984)

[6] Bacharach, M.: Economics and the Theory of Games. Macmillan, London (1976)

[7] Başar, T., Olsder, G.J.: Dynamic Noncooperative Game Theory. No. 23 in Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, PA, second edn. (1999)

[8] Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Athena Scientific, Belmont, Massachusetts (1997)

[9] Bredin, J., Maheswaran, R.T., Imer, Ç., Başar, T., Kotz, D., Rus, D.: A game-theoretic formulation of multi-agent resource allocation. In: Proceedings of the Fourth International Conference on Autonomous Agents. Barcelona, Spain (June 3–7, 2000)

[10] Buttyán, L., Hubaux, J.P.: Stimulating cooperation in self-organizing mobile ad hoc networks. Mobile Networks and Applications 8, 579–592 (2003)

[11] Buyya, R.: Economic-based Distributed Resource Management and Scheduling for Grid Computing. Ph.D. thesis, Monash University, Melbourne, Australia (April 2002)

[12] Buyya, R., Abramson, D., Giddy, J.: An economy driven resource management architecture for global computational power grids. In: Arabnia, H.R. (ed.) Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications. CSREA Press, Las Vegas, NV (June 24–29, 2000)

[13] Buyya, R., Abramson, D., Giddy, J.: A case for economy grid architecture for service oriented grid computing. In: Proceedings of the 15th International Parallel and Distributed Processing Symposium. pp. 776–790 (April 23–27, 2001)

[14] Croft, D.P., James, R., Thomas, P.O.R., Hathaway, C., Mawdsley, D., Laland, K.N., Krause, J.: Social structure and co-operative interactions in a wild population of guppies (*Poecilia reticulata*). Behavioral Ecology and Sociobiology 59(5), 644–650 (March 2006)

[15] Cruz, R.L.: A calculus for network delay, part II: network analysis. IEEE Transactions on Information Theory 37(1), 132–141 (January 1991)

[16] Dugatkin, L.A.: Do guppies play TIT FOR TAT during predator inspection visits? Behavioral Ecology and Sociobiology 23(6), 395–399 (December 1988)

[17] Dugatkin, L.A. (ed.): Cooperation Among Animals: An Evolutionary Perspective. Oxford Series in Ecology and Evolution, Oxford University Press, Oxford (1997)

[18] Feigenbaum, J., Shenker, S.: Distributed algorithmic mechanism design: recent results and future directions. In: Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication. pp. 1–13. Atlanta, Georgia, USA (September 28, 2002)

[19] Finke, J., Passino, K.M.: Stable cooperative vehicle distributions for surveillance. Journal of Dynamic Systems, Measurement, and Control 129(5), 597–608 (2007)

[20] Finke, J., Passino, K.M., Sparks, A.G.: Stable task load balancing for cooperative control of networked autonomous air vehicles. IEEE Transactions on Control Systems Technology 14(5), 789–803 (September 2006)

[21] Gil, A.E., Passino, K.M.: Stability analysis of network-based cooperative resource allocation strategies. Automatica 42(2), 245–250 (2005)

[22] Gil, A.E., Passino, K.M., Ganapathy, S., Sparks, A.: Cooperative task scheduling for networked uninhabited air vehicles. IEEE Transactions on Aerospace and Electronic Systems 44(2), 561–581 (April 2008)

[23] Grosu, D., Chronopoulos, A.T.: Algorithmic mechanism design for load balancing in distributed systems. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34(1), 77–84 (February 2004)

[24] Gustafson, R.D., Frisk, P.D., Hughes, J.: College Algebra. Brooks/Cole Publishing, Belmont, CA, tenth edn. (December 2, 2008)

[25] Hamilton, W.D.: The genetical evolution of social behavior. I. Journal of Theoretical Biology 7(1), 1–16 (1964)

[26] Ipakchi, A., Albuyeh, F.: Grid of the future. IEEE Power and Energy Magazine 7(2), 52–62 (March/April 2009)

[27] Iverson, K.E.: A Programming Language. Wiley, New York (1962)

[28] Johns, M.V., Miller, Jr., R.G.: Average renewal loss rates. Annals of Mathematical Statistics 34(2), 396–401 (June 1963)

[29] Knuth, D.E.: Two notes on notation. American Mathematical Monthly 99(5), 403–422 (May 1992)

[30] Lalis, S., Karipidis, A.: JaWS: an open market-based framework for distributed computing over the internet. In: Buyya, R., Baker, M. (eds.) Grid Computing – GRID 2000: 1st IEEE/ACM International Workshop on Grid Computing, Bangalore India, December 17, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1971. Springer, Berlin (December 17–20, 2000)

[31] Lange, D.B.: Mobile objects and mobile agents: the future of distributed computing? In: Jul, E. (ed.) ECOOP'98 – Object-Oriented Programming: 12th European Conference, Brussels, Belgium, July 20–24, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1445, pp. 1–12. Springer, Berlin (July 20–24, 1998)

[32] Lange, D.B., Oshima, M.: Seven good reasons for mobile agents. Communications of the ACM 42(3), 88–89 (March 1999)

[33] Lieberman, E., Hauert, C., Nowak, M.A.: Evolutionary dynamics on graphs. Nature 433(7023), 312–316 (January 20, 2005)

[34] Lygeros, J., Godbole, D.N., Sastry, S.: Multiagent hybrid system design using game theory and optimal control. In: Proceedings of the 35th IEEE Conference on Decision and Control. vol. 2, pp. 1190–1195 (December 1996)

[35] Maheswaran, R.T., Imer, O.Ç., Başar, T.: Agent mobility under price incentives. In: Proceedings of the 38th IEEE Conference on Decision and Control. vol. 4, pp. 4020–4025. Phoenix, Arizona, USA (December 7–10, 1999)

[36] Mas-Colell, A., Whinston, M.D., Green, J.R.: Microeconomic Theory. Oxford University Press, New York (1995)

[37] Milinski, M.: TIT FOR TAT in sticklebacks and the evolution of cooperation. Nature 325(6103), 433–435 (January 29, 1987)

[38] Nicholson, W.: Microeconomic Theory: Basic Principles and Extensions. Dryden Press, Fort Worth, TX, fifth edn. (1992)

[39] Nowak, M.A.: Five rules for the evolution of cooperation. Science 314(5805), 1560–1563 (December 8, 2006)

[40] Nowak, M.A., May, R.M.: Evolutionary games and spatial chaos. Nature 359(6398), 826–829 (October 29, 1992)

[41] Ohtsuki, H., Hauert, C., Lieberman, E., Nowak, M.A.: A simple rule for the evolution of cooperation on graphs. Nature 441(7092), 502–505 (2006)

[42] Oram, A. (ed.): Peer-to-Peer: Harnessing the Power of Disruptive Technologies. O'Reilly & Associates, Sebastopol, CA (2001)

[43] Osborne, M.J., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge, MA (1994)

[44] Perkins, J.R., Kumar, P.R.: Stable, distributed, real-time scheduling of flexible manufacturing/assembly/disassembly systems. IEEE Transactions on Automatic Control 34(2), 139–148 (February 1989)

[45] Qiu, D., Srikant, R.: Modeling and performance analysis of BitTorrent-like peer-to-peer networks. In: Yavatkar, R., Zegura, E.W., Rexford, J. (eds.) Proceedings of the ACM SIGCOMM 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. pp. 367–378. Portland, Oregon, USA (August 30 – September 3, 2004)

[46] Reed, D., Pratt, I., Menage, P., Early, S., Stratford, N.: Xenoservers: accountable execution of untrusted programs. In: Proceedings of the Seventh Workshop on Hot Topics in Operating Systems. pp. 136–141. Rio Rico, Arizona, USA (March 28–30, 1999)

[47] Rudin, W.: Principles of Mathematical Analysis. McGraw-Hill, New York, NY, third edn. (1976)

[48] Smith, R.G.: The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Transactions on Computers 29(12), 1104–1113 (December 1980)

[49] Trivers, R.L.: The evolution of reciprocal altruism. Quarterly Review of Biology 46(1), 35–57 (March 1971)

[50] Waldspurger, C.A., Hogg, T., Huberman, B.A., Kephart, J.O., Stornetta, W.S.: Spawn: a distributed computational economy. IEEE Transactions on Software Engineering 18(2), 103–117 (February 1992)

[51] White, J.E.: Telescript technology: mobile agents. In: Milojičić, D., Douglis, F., Wheeler, R. (eds.) Mobility: Processes, Computers, and Agents, pp. 460–493. ACM Press, New York (1999)