Monte Carlo fictitious play for finding pure Nash equilibria in identical interest games

Seksan Kiatsupaibul
Department of Statistics, Chulalongkorn University, Bangkok 10330, Thailand, seksan@cbs.chula.ac.th

Giulia Pedrielli
School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, Arizona 85281, gpedriel@asu.edu

Christopher Thomas Ryan
UBC Sauder School of Business, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z2, chris.ryan@sauder.ubc.ca

Robert L Smith
Industrial and Operations Engineering, University of Michigan, Ann Arbor, Michigan 48109, rlsmith@umich.edu

Zelda B Zabinsky
Department of Industrial and Systems Engineering, University of Washington, Seattle, Washington 98195, zelda@u.washington.edu

Computing equilibria in large-scale games is an important topic in many areas. One approach is to define a dynamic procedure such as Fictitious play (FP) that converges to a mixed Nash equilibrium (NE) in identical interest games (among other classes) but suffers from exponential iteration complexity. Recent variants of FP reduce the computational burden, but many still do not guarantee convergence to a pure NE. We analyze a procedure—Monte Carlo Fictitious Play (MCFP)—that overcomes these limitations and efficiently discovers a pure NE in finite time with probability one in identical interest games. We also show a variant of MCFP finds a pure NE with optimal utility with probability one. Numerical results demonstrate the comparative performance of several variants of MCFP.

Key words: Game theory, equilibrium computation, game-theoretic learning algorithms, fictitious play, optimal equilibria

History:

---

1. Introduction

Devising simple dynamic procedures that converge to equilibria in large-scale games is an important topic. Concrete applications are plentiful, including routing and motion planning (Dolinskaya et al. 2016, Swenson et al. 2018), pedestrian flow (Ma et al. 2017), and dynamic pricing (Masuda and Whang 1999). Moreover, these simple dynamic procedures can be used as a basis to solve distributed learning and control problems (Marden and Shamma 2015, Swenson et al. 2018, 2015). In these scenarios, multiple agents with their own individual utilities achieve a coordinated effort to minimize an overall objective by communicating with each other to arrive at a game-theoretic equilibrium. See Marden and Shamma (2015) for an accessible overview of the approach. In particular, we discuss in detail how our methods apply to the drone coordination problem in Swenson et al. (2018) that employs this type of approach. These methods are also used to solve large-scale optimization problems (Lambert et al. 2005, Garcia et al. 2007, Swenson et al. 2018, **?**, Lei and Shanbhag 2020, Lei et al. 2020). Large-scale optimization problems can be cast as identical interest games by assigning subsets of the decision variables to players and set each player's utility function equal to the same overall objective. In this context, a pure-strategy Nash equilibrium (what we refer to as a pure NE throughout) serves as a kind of locally optimal solution, since players cannot improve the objective function by changing the variables that they have been assigned.

Known procedures for identifying equilibria have their inherent benefits and drawbacks. Fictitious play (FP), introduced in (Brown 1951), has been shown to converge to a Nash equilibrium (NE) in a growing number of classes of games including identical interest games (Monderer and Shapley 1996a), potential games (Monderer and Shapley 1996b), and 2-player games with 2 rows and n columns (2 by n games) (Berger 2005) (for a unified approach to convergence see (Shamma and Arslan 2004)). Unfortunately, this NE may be mixed, which is undesirable in many applications. In addition, the per iteration complexity of FP grows exponentially fast in the number of players. This motivated innovations to maintain the convergence properties of FP but ease its computational burden (Abernethy et al. 2019). Sampled fictitious play (SFP) (Lambert et al. 2005) greatly reduces the amount of work performed in each iteration of FP by eliminating the need to compute empirical expectations in each iteration. Best replies are computed using *samples* of plays drawn independently from history. However, SFP still suffers from a growing number of samples at each iteration. (Swenson et al. 2017) reduce this computational burden via single sample fictitious play (SSFP) algorithm to only requiring a single sample per iteration *although, unlike SFP (Dolinskaya et al. (2016)),* this algorithm must tune parameters appropriately. Importantly, Hannan consistency for the sampled fictitious play mechanism was proved in (Li and Tewari 2018) under Bernoulli sampling.

These improvements on FP maintain its attractive property of converging to mixed Nash equilibria in identical interest games. However, these algorithms (including SSFP) are not guaranteed to find pure NE, but only mixed NE, which are impractical in some applications. Moreover, the iterates of the algorithm converge only to a subset of mixed equilibria (with no single one delivered), even in the limit as the number of iterations grows.

Algorithms such as FP can sometimes be adapted to find pure NE, at the expense of introducing additional parameters and computational challenges. For instance, fictitious play with limited memory and inertia (Young 2004) and a joint strategy fictitious play with inertia (Marden et al.

2009) revise dynamic procedures that hone in on pure Nash equilibria. The iterates of these algorithms converge to pure NE, but users must select tuning parameters in order to run the procedures.

Beyond this, even if a pure NE is eventually found, not all pure NE have the same utility. In identical interest games, equilibria can be ordered according to the utility they deliver to each player. In fact, in identical interest games, a pure NE is a local optimum with respect to a neighborhood system consisting of translations along coordinate axes. These local optima are ordered by their payoffs where the most preferred NE is a global optimal solution we call a pure optimal NE.

We study an implementation of fictitious play called Monte Carlo Fictitious Play (MCFP) that overcomes many of the limitations of previous variants of FP. (Dolinskaya et al. 2016) originally developed this algorithm to deliver optimal solutions in finite time with probability one for deterministic dynamic programs. However, when applied *directly* to identical interest games in strategic form, its performance may still suffer from a lack of convergence to a pure equilibrium as with FP, SFP, and SSFP.

Our innovation is that we define an auxiliary tree game and prove that MCFP, applied to the auxiliary tree game, is guaranteed to find a pure NE in finite time with probability one. The auxiliary game modifies the extensive-form tree description of the original strategic-form game to remove all non-singleton information sets by having different players at each node in the tree, called tree players. It is here where the value of the auxiliary tree structure for convergence is evident. Whereas fictitious play algorithms applied to strategic form games can get "stuck" in cycles of unilateral best responses that do not converge to a pure NE, the auxiliary tree structure allows exploration of unilateral best responses among tree players that are not unilateral best responses in the original game. It is precisely the randomization induced in "off equilibrium paths" (which can become equilibrium paths in the auxiliary tree formulation) which allows the MCFP algorithm to determine a pure NE. Another benefit of MCFP applied to the auxiliary tree game is that an *optimal*, pure NE is guaranteed to be discovered in finite time with probability one, although confirmation of the global optimum is not computationally practical. An optimal pure NE in an identical-interest game is a pure NE that maximizes the shared utility function of all players.

In summary, we establish the following attractive features when applying MCFP to the auxiliary tree game formulation of identical interest games: (i) it finds a pure NE for the original game in finite time with probability one, (ii) if allowed to continue instead of stopping at the first pure NE found, it will find an *optimal* pure NE for the original game in finite time with probability one, (iii) each iteration of MCFP can be executed in polynomial time in the number of strategic game players and the maximum number of actions per player, and (iv) it is efficient and empirically outperforms other known algorithms (e.g., Young's FP with inertia (Young 2004)).

We should acknowledge that our algorithms require each agent to communicate with a central coordinator that broadcasts random draws to all players at each iteration of the algorithm. This is in contrast to recent papers that focus on settings where communication is restricted (see, for instance, Young (2009), Pradelski and Young (2010), Marden et al. (2014)). These papers must settle for weaker notions of convergence than what we achieve here.

The rest of the paper is organized as follows. Section 2 introduces identical interest games in their strategic form. In Section 3, we develop the auxiliary tree game for an identical interest game. Section 4 describes our application of the MCFP algorithm concept to the auxiliary tree game. Section 5 includes a proof that MCFP delivers a pure NE in finite time with probability one. Section 6 contains the results of our numerical experiments that demonstrate the practical advantages of our approach, including an application to the drone assignment problem posed in (Swenson et al. 2018).

2. Identical interest games in strategic form

Let Ξ be a finite game in strategic form with the set of players $\mathcal{N} = \{1, \ldots, n\}$. Let the finite set of pure strategies (actions) of player $i \in \mathcal{N}$ be $\mathcal{X}_i$ with $x_i \in \mathcal{X}_i$ a specific action. Also, let $m_i = |\mathcal{X}_i|$ be the cardinality of $\mathcal{X}_i$ and let $m = \max_{i \in \mathcal{N}} m_i$. For simplicity, we denote the elements of actions sets as $\mathcal{X}_i = \{1, 2, \ldots, m_i\}$ for all i unless specified otherwise. Let $\mathbf{x} = (x_1, \ldots, x_n)$ be an *action profile* and let $\mathcal{X} = \Pi_{i=1}^n \mathcal{X}_i$ be the *set* of all *action profiles*. Let the utility function of player $i \in \mathcal{N}$ be $u_i : \mathcal{X} \to \mathbb{R}$. We consider the case where the utility functions are *identical*, i.e., $u_i(x_1, \ldots, x_n) = u(x_1, \ldots, x_n)$ for $i = 1, \ldots, n$.

Our objective is to find a *pure NE* for this identical interest game. An action profile $\mathbf{x} = (x_1, \ldots, x_n)$ is a pure NE if no player has anything to gain by changing only their own action. Symbolically, $\mathbf{x}$ is a pure NE if, for every player i, given the actions $\mathbf{x}_{-i} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \cdots, x_n)$ of the remaining players, $u(x_i, \mathbf{x}_{-i}) \geq u(a_i, \mathbf{x}_{-i})$ for $a_i \in \mathcal{X}_i$, where $u(x_i, \mathbf{x}_{-i}) := u(x_1, x_2, \ldots, x_n)$ and $u(a_i, \mathbf{x}_{-i}) := u(x_1, x_2, \ldots, x_{i-1}, a_i, x_{i+1}, \ldots, x_n)$.

We also consider finding an optimal solution, denoted $\mathbf{x}^*$, that maximizes utility as follows:

$$u^* := \max_{\mathbf{x} \in \mathcal{X}} u(\mathbf{x}). \tag{1}$$

An optimal solution exists since $\mathcal{X}$ is finite. Observe that $\mathbf{x}^*$ is a pure NE with optimal utility value $u^* = u(\mathbf{x}^*)$.

In this paper, we often give special attention to the class of identical interest *coordination* games. In a coordination game, all players have the same action set; i.e., there exists a set $\mathcal{Z}$ such that

$\mathcal{X}_i = \mathcal{Z}$ for all $i \in \mathcal{N}$. Players get positive utility if and only if players "coordinate" by taking the same action in $\mathcal{Z}$. Thus, we can assign a utility $u_z = u(z, z, \ldots, z) > 0$ to each action $z \in \mathcal{Z}$ and set

$$u_i(x_1, x_2, \ldots, x_n) = \begin{cases} u_z & \text{if } x_i = z \text{ for all } i \in \mathcal{N} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Admittedly, the class of identical interest coordination games is quite simple. Finding an optimal pure NE simply amounts to finding the largest u_z over $z \in \mathcal{Z}$, which takes $O(m)$ time. However, general algorithms for solving identical interest games cannot easily identify that a game is an identical interest coordination game. Indeed, verifying that a game is a coordination game is essentially as difficult as finding an equilibrium in the game since, in the worst case, you must enumerate all action profiles.

Before proposing our variant of fictitious play (FP), let us recall standard FP. In fictitious play, each player i believes all opponents are playing mixed strategies given by the empirical distribution of their historical actions. That is, for every action $x_j \in \mathcal{X}_j$, let $w_j(x_j)$ denote the number of times opponent j took action x_j. Then, player i believes opponent j will take action x_j with probability $P_j(x_j) = w_j(x_j)/\sum_{x \in \mathcal{X}_j} w_j(x)$. Player i then best replies to the mixed strategies represented by the probabilities $P_j(x_j)$ for each opponent j. It was shown in (Monderer and Shapley 1996a) that if all players best reply in this way, their beliefs converge to the set of mixed NE. To illustrate this, let us take the very simple scenario of an identical interest coordination game with two players.

Table 1 Game A in its strategic form Ξ.

		Player 2	
		U	D
Player 1	U	1	0
	D	0	1

EXAMPLE 1. Let **Game A** be the two-person identical interest coordination game with the strategic form Ξ shown in Table 1. Suppose the initial actions are $x_1 = U$ and $x_2 = D$. Then, player 1 forms a belief that player 2 will take action D with probability 1. In this case, player 1 best responds with action D. Similarly, player 2 forms a belief that player 1 will take action U with probability 1 and so best responds with action U. The empirical distributions in the second round of fictitious play are thus both discrete uniform distributions: each player believes the other will take action U with probability 0.5 and action D with probability 0.5. In that scenario, the action that maximizes expected utility is tied. Assuming ties are broken randomly, as fictitious play iterates, the empirical distribution converges to the mixed NE of each player equally likely playing U or D. In other words, the procedure, breaking ties in this way, does not converge to either of the pure Nash equilibria (U, U) or (D, D). $\triangleleft$

We are now ready to state a variant of Monte Carlo Fictitious Play (MCFP) algorithm applied directly to the original game in strategic form. In each iteration k of the algorithm, we maintain a vector S_i^k that tracks the best replies of player i. That is, for all $i = 1, 2, \ldots, n$, we have $S_i^k = (S_i^k(x_i) : x_i \in \mathcal{X}_i)$ where $S_i^k(x_i)$ is the number of times player i best replies with action $x_i \in \mathcal{X}_i$ through iteration k.

MCFP on the original strategic form game (MCFP-O)

Step O.1 *Initialization.* For each player $i \in \mathcal{N}$, set $S_i^0 \leftarrow (0, 0, \ldots, 0)$. Set $k \leftarrow 1$.

Step O.2 *Draw an action profile.* For each player $i \in \mathcal{N}$, draw action $x_i \in \mathcal{X}_i$ with probability $S_i^{k-1}(x_i)/(k-1)$ (if $k = 1$, draw uniformly at random from $\mathcal{X}_i$) to form the drawn action profile $\mathbf{p}_D = (x_1, x_2, \ldots, x_n)$.

Step O.3 *Compute a best-reply action profile.* For each player $i \in \mathcal{N}$, compute a best reply x_i^* to p_D, breaking ties uniformly at random, to form a best-reply action profile $\mathbf{p}_R = (x_1^*, \ldots, x_n^*)$.

Step O.4 *Stopping Condition.* If $\mathbf{p}_R$ is a pure NE then return $\mathbf{p}_R$ and terminate. Otherwise, go to **Step O.5**.

Step O.5 *Update.* For all $i \in \mathcal{N}$, update $S_i^k(x_i^*) \leftarrow S_i^{k-1}(x_i^*) + 1$; and for $x_i \neq x_i^*$, $S_i^k(x_i) \leftarrow S_i^{k-1}(x_i)$. Update $k \leftarrow k + 1$ and go to **Step O.2**.

Table 2 **Game B in its strategic form Ξ.**

		Player 2	
		U	D
Player 1	U	1	0
	D	0	2

Table 3 **Applying MCFP-O to Game B.**

Iter. k	Draw p_D 1	2	Best reply p_R 1	2	Utility $u(p_R)$	History S_i^k 1	2
0						$(0,0)$	$(0,0)$
1	**U**	**D**	D	U	0	$(0,1)$	$(1,0)$
2	D	U	U	D	0	$(1,1)$	$(1,1)$
3	**D**	**D**	D	D	2	$(1,2)$	$(1,2)$

Actions in bold indicate a nondeterministic choice that was selected randomly for purposes of illustration.

Example 2. In Table 3, we apply MCFP-O algorithm to **Game B** given in strategic form in Table 2. Suppose the first draw from **Step O.2** is $\mathbf{p}_D = (U, D)$. Based on this drawn profile, the best reply is $\mathbf{p}_R = (D, U)$ and histories update to $S_1^1 = (0, 1)$ and $S_2^1 = (1, 0)$. The second iteration is now entirely deterministic, resulting in $S_1^2 = (1, 1)$ and $S_2^2 = (1, 1)$. Now, the draw for each player is uniform with probability 0.50 of drawing either U or D. In the illustration in Table 3, we took $\mathbf{p}_D = (D, D)$. This was a "lucky" draw since it results in terminating the algorithm.

Observe that this pass of MCFP-O resulted in the optimal pure NE (D, D) with a utility of 2. There is no guarantee that MCFP-O finds an optimal pure NE even if allowed to continue after finding its first pure NE. Suppose the first draw was $\mathbf{p}_D = (U, U)$. The players will best reply by (U, U) and the algorithm terminates. Even if the algorithm were allowed to continue, the players would take action U in every iteration. Therefore, there is no opportunity for them to switch to the optimal pure NE (D, D). Indeed, the algorithm is absorbed in the nonoptimal pure NE (U, U). $\triangleleft$

Interestingly, in identical interest coordination games, the MCFP-O algorithm finds a pure (potentially non-optimal) NE in *finite time with probability one*. To make this notion of convergence precise, we make the following formal definition.

DEFINITION 1. Let F_k denote the event that p_R is a pure NE in **Step O.3** in iteration k of the MCFP-O algorithm. Let F denote the union of all F_k; that is, $F = \bigcup_{k=1}^{\infty} F_k$. Then we say MCFP-O *finds a pure NE in finite time with probability one* if the probability of event F is one. Indeed, if the event F occurs with probability one, then this means, with probability one, there exists a positive integer k such that F_k occurs. In other words, with probability one there exists a k such that the algorithm terminates after k iterations.

PROPOSITION 1. *MCFP-O, when applied to an identical interest coordination game, finds a pure NE in finite time with probability one.*

Proof of Proposition 1. Let $\mathcal{X}^*$ denote the "coordinated" action profiles; that is, $\mathcal{X}^* = \{(z, z, \ldots, z) \in \mathcal{X} : z \in \mathcal{Z}\}$. Let p_D be a drawn profile on iteration k and let p_R denote a best reply to p_D. At iteration k, one of the following holds:

(i) $\mathbf{p}_D \in \mathcal{X}^*$,

(ii) $\mathbf{p}_D$ can be adjusted in one player's action to yield a coordinated action profile in $\mathcal{X}^*$, or

(iii) $\mathbf{p}_D$ must be changed in an at least two players' action to yield a coordinated action profile in $\mathcal{X}^*$.

If case (i) is ever reached in any iteration k then the algorithm terminates in iteration k since $\mathbf{p}_R = \mathbf{p}_D$ when $\mathbf{p}_D \in \mathcal{X}^*$. In other words, the event F_k in Definition 1 occurs. Indeed, there is no possibility for ties in best replies in an identical interest coordination game since $u_z > 0$ for all $z \in Z$ and so the only possible choice for $\mathbf{p}_R$ is $\mathbf{p}_D$. This is because any deviation would lead to non-coordinated outcome (i.e., element not in $\mathcal{X}^*$), yielding a payoff of zero for the deviating player.

Moreover, if case (ii) or (iii) produce a best reply in $\mathcal{X}^*$ for any iteration k, the algorithm terminates with a pure NE and event F_k has occurred. Thus, it suffices to show that the probability of the event that cases (ii) and (iii) are visited infinitely often, and a best reply in $\mathcal{X}^*$ is not chosen, has probability zero. This establishes that the event F in Definition 1 occurs with probability one and the proof is done.

First, consider the setting where $n = 2$. Observe that case (iii) cannot happen when $n = 2$, and so the only way the algorithm has not reached case (i) (and terminated) is if it has only found itself in (ii) up until that point. In particular, in the first iteration where case (ii) occurs, $\mathbf{p}_D = (z_1, z_2)$ where $z_1 \neq z_2$. Then, when considering the best reply step, player 1 will best reply with action z_2 and player 2 will best reply with action z_1. Action z_2 is included in player 1's history and action z_1 is included in player 2's history. Thus, in the next round, player 1 will draw action z_2 and player 2 will draw action z_1. But then, in this round, the best reply will be player 1 taking action z_1 and player 2 taking action z_2. Thus, the only possible best replies vectors are (z_1, z_2) or (z_2, z_1). Due to symmetry, the probability of player i drawing action z_i approaches $1/2$ as the number of times case (ii) is reached approaches infinity. Thus, the probability that case (ii) is reached $k_{(ii)}$ times before termination is $(1/2)^{k_{(ii)}}$ for $k_{(ii)}$ sufficiently large. Thus the probability that case (ii) is reached infinitely often (and results in no best replies in $\mathcal{X}^*$) is zero.

Next, we consider $n > 2$. Consider the setting where case (iii) is visited infinitely often. Then, all actions in $\mathbf{p}_R$ are selected uniformly at random from $\mathcal{Z}$ because all unilateral deviations yield a utility of 0. Thus, $\mathbf{p}_R \in \mathcal{X}^*$ with probability at least $(1/m)^n$. This probability is irrespective of the iteration number k, so the probability that $\mathbf{p}_D \in \mathcal{X}^*$ after $k_{(iii)}$ visits to case (iii) is less than $((1/m)^n)^{k_{(iii)}}$. Since case (iii) is visited infinitely often, this probability converges to 0 as $k_{(iii)} \to \infty$. Thus the probability that case (iii) is reached infinitely often (and results in no best replies in $\mathcal{X}^*$) is zero.

Thus, we are only left to consider the event that case (ii) is visited infinitely often when $n > 2$. When case (ii) is reached, all but one player, say player i, chooses their action randomly from $\mathcal{Z}$ when determining $\mathbf{p}_R$. Hence, there is at least a $(1/m)^{n-1}$ chance (irrespective of k) that all other players best reply with the action of player i, resulting in $x_R \in \mathcal{X}^*$. The probability this does not happen after $k_{(ii)}$ iterations is at most $((1/m)^{n-1})^{k_{(ii)}}$, which converges to 0 as $k_{(ii)} \to \infty$. This completes the proof. ■

It is an open question whether MCFP-O terminates with probability one when applied to a more general identical interest game (that is, non-coordination game) in strategic form.

3. An auxiliary tree-game

Our method for finding pure Nash equilibria in general identical interest games analyzes an auxiliary game to Ξ (denoted Γ), which we call the tree game. We construct Γ in two steps. First, we write Ξ in its equivalent extensive form $\widetilde{\Xi}$. We represent the extensive form game $\widetilde{\Xi}$ by a tree $(\mathcal{V} \cup \mathcal{W}, \mathcal{A})$ where $\mathcal{V} \cup \mathcal{W}$ is the set of nodes and $\mathcal{A}$ is the set of arcs. The node set is partitioned into two subsets $\mathcal{V}$ and $\mathcal{W}$. The subset $\mathcal{V}$ is the union of subsets $\mathcal{V}_1, \ldots, \mathcal{V}_n$ where subset $\mathcal{V}_i, i = 1, \ldots, n$ (what we often call simply Stage i) is the information set of player i of the original game. The special subset $\mathcal{W}$ is reserved for the terminal representation of utilities. The set of arcs $\mathcal{A}$ is partitioned into subsets $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$. For $i = 1, 2, \ldots, n-1$, every arc in $\mathcal{A}_i$ is directed from a node in $\mathcal{V}_i$ to a node in $\mathcal{V}_{i+1}$. The arcs in $\mathcal{A}_n$ are directed from nodes in $\mathcal{V}_n$ into $\mathcal{W}$. For all i, each node v in $\mathcal{V}_i$ has out-degree m_i (one for each action of player i). For $i = 2, \ldots, n$, each node in $\mathcal{V}_i$ has in-degree 1. The nodes in $\mathcal{V}_1$ have in-degree zero, while the nodes in $\mathcal{W}$ have in-degree 1 and out-degree 0. Taken together, this implies that for $i = 2, \ldots, n$, $\mathcal{V}_i$ has $m_1 m_2 \cdots m_{i-1}$ nodes with in-degree 1.

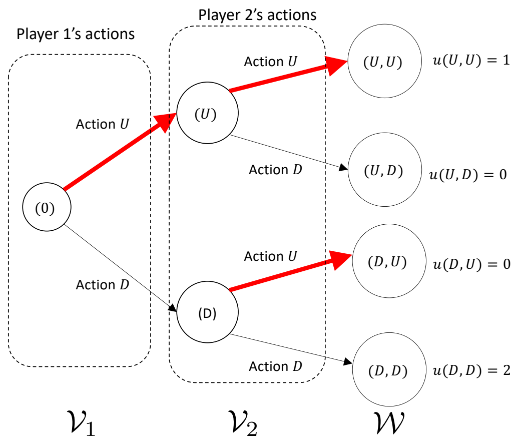
In the second step, convert $\tilde{\Xi}$ into the tree game Γ as follows. Each player in Γ corresponds to a node in $\mathcal{V} = \mathcal{V}_1 \cup \cdots \cup \mathcal{V}_n$ and is called a *tree player*. The tree game now has complete information: each player has an information set that consists of a single node in the tree.

For each Stage i, the action set $\mathcal{Y}_j$ available for each tree player $j \in \mathcal{V}_i$ is equal to the set of actions $\mathcal{X}_i$. Thus, all tree players in the same stage have the same action set. We denote the nodes in the tree according to the path of actions taken to reach that node from the unique node in $\mathcal{V}_1$. That is, for $i = 2, \ldots, n$, the node labels in Stage i represent the actions taken by players in Stage 1 to Stage $i-1$ leading to that node, with the default label (0) for the player in Stage 1. These node labels capture the actions taken by preceding players to reach each node.

The space of all strategies of tree players in the tree game Γ is then $\mathcal{Y} = \Pi_{j \in \mathcal{V}} \mathcal{Y}_j$. We call the strategy $\mathbf{y} \in \mathcal{Y}$ in the tree game Γ a *tree policy* since it provides an action for each player in the tree. This is also to distinguish it from the terminology "action profile" that we reserve for speaking about the original game Ξ. Each tree policy $\mathbf{y}$ contains a unique complete path starting from node (0) to a terminal node in $\mathcal{W}$. A tree player that is on the complete path is said to be a *path* player (or *in-play*). The remaining tree players are said to be *non-path* players (or *not-in-play*).

We define a *projection* π as a mapping from $\mathcal{Y}$ to $\mathcal{X}$ where $\pi(\mathbf{y})$ denotes the actions of the in-play tree players of tree policy $y \in \mathcal{Y}$. Thus, the projection of a tree policy in Γ is an action profile in Ξ. We say that $y \in \mathcal{Y}$ is an *extension* of $\mathbf{x} \in \mathcal{X}$ if $\pi(\mathbf{y}) = \mathbf{x}$. Note that many possible extensions of an action profile $\mathbf{x} \in \mathcal{X}$ exist. We define the utility function $v(\mathbf{y})$ of a tree policy $\mathbf{y} \in \mathcal{Y}$ as the utility at the terminal node on the complete path contained in y, i.e., $v(\mathbf{y}) = u(\pi(\mathbf{y}))$, for all $\mathbf{y} \in \mathcal{Y}$. Intuitively, the utility function of the tree game Γ is the utility of the path players playing the original game Ξ. Accordingly, there is a connection—but not a correspondence—between equilibria in Γ and Ξ.

Figure 1 The tree game Γ corresponding to Game B.



Example 3. Consider the identical interest coordination game **Game B** with strategic form Ξ shown in Table 2. Figure 1 illustrates the auxiliary tree game Γ corresponding to **Game B**. The tree game Γ has three tree players: (0), (U), and (D). Tree players (U) and (D) have the same action set. The heavy arcs in Figure 1 indicate a tree policy $\mathbf{y} = (U,U,U)$ corresponding to tree players (0), (U), and (D) that play U, U, and U, respectively. Observe that there is one complete path – the uppermost path – ending at the terminal node with $u(U,U) = 1$. Therefore, tree players (0) and (U) are path players (or in-play), while tree player (D) is a non-path player (or not-in-play). The utility of tree policy $\mathbf{y}$ is $v(\mathbf{y}) = u(\pi(U,U,U)) = u(U,U) = 1$. $\triangleleft$

Proposition 2. *Let Ξ be a strategic form identical interest game and let Γ be its corresponding tree game. Every pure NE action profile $\mathbf{x}$ in Ξ can be extended to a pure NE tree policy in Γ. If $\mathbf{x}^*$ is an optimal pure NE in Ξ (i.e., $\mathbf{x}^*$ solves* (1)*) then every extension of $\mathbf{x}^*$ is an optimal pure NE in Γ and, conversely, if $\mathbf{y}^*$ is an optimal pure NE in Γ with $v(\mathbf{y}^*) = u^*$, then the projection of $\mathbf{y}^*$ is an optimal pure NE in Ξ.*

Proof of Proposition 2. Given a pure NE $\mathbf{x} = (x_1, \ldots, x_n)$ of Ξ, we construct a tree policy $\mathbf{y} \in \mathcal{Y}$ and show that it is a pure NE tree policy. For all tree players $j \in \mathcal{V}_i$, we let $y_j = x_i$, $i = 1, \ldots, n$ so that all tree players in the same stage have the same action (such a construction is found in Figure 1). It is clear from this construction that $\pi(\mathbf{y}) = \mathbf{x}$.

Since $\mathbf{x}$ is a pure NE in Ξ, its utility cannot be improved by any unilateral deviation $\mathbf{x}'$, i.e., $u(\mathbf{x}) \geq u(\mathbf{x}')$ for every $\mathbf{x}'$ that is a unilateral deviation of $\mathbf{x}$. Let $\mathbf{y}'$ be a unilateral deviation of the $\mathbf{y}$ constructed in the previous paragraph. Since $\mathbf{y}$ is constructed such that all tree players in the same stage have the same action, if any tree player switches actions to form a unilateral deviation $\mathbf{y}'$, then the projection of $\mathbf{y}'$ is a unilateral deviation in Ξ, i.e., $\pi(\mathbf{y}') = \mathbf{x}'$. Therefore, we have

$v(\mathbf{y}) = u(\pi(\mathbf{y})) = u(\mathbf{x}) \geq u(\mathbf{x}') = u(\pi(\mathbf{y}')) = v(\mathbf{y}')$, where $\mathbf{x}' = \pi(\mathbf{y}')$ is the unilateral deviation in $\mathbf{x}'$ corresponding to the unilateral deviation $\mathbf{y}'$ in Γ. Therefore $\mathbf{y}$ is a pure NE tree policy in Γ.

Let $\mathbf{x}^*$ be an optimal pure NE in Ξ, with $u(\mathbf{x}^*) = u^*$. Then every extension $\mathbf{y}$ of $\mathbf{x}^*$ has the utility $v(\mathbf{y}) = u(\pi(\mathbf{x}^*)) = u^*$ and so is automatically a pure NE since no deviation (unilateral or otherwise) can improve on a utility of u^* in Γ. Conversely, suppose that $\mathbf{y}^*$ is an optimal pure NE in Γ with $v(\mathbf{y}^*) = u^*$. This means that $v(\mathbf{y}^*) = u(\pi(\mathbf{y}^*)) = u^*$, and thus the projection $\mathbf{x} = \pi(\mathbf{y}^*)$ is an optimal pure NE in Ξ. ■

Remark 1. It is important to note that not every extension of a pure NE action profile $\mathbf{x}$ in Ξ is a pure NE in Γ nor does every pure NE tree policy in Γ project to a pure NE action profile in Ξ. See the counter-examples in Examples 4 and 5 for illustrations.

Example 4. Consider again **Game B**, whose tree game form is given in Figure 1, and consider the tree policy $\mathbf{y} = (U, U, D)$. Tree players (0) and (U) are path players and the tree policy $\mathbf{y}$ projects to the action profile (U, U) in the original game. Observe that the projection $\pi(\mathbf{y}) = (U, U)$ is a pure NE in Ξ, but $\mathbf{y}$ is *not* a pure NE in the tree game. Indeed, tree player (0) has a profitable deviation to take action D, resulting in improving the utility from 1 to 2. This unilateral deviation in action from U to D for tree player (0) (i.e., comparing (U, U, D) to (D, U, D)) results in tree players (0) and (D) becoming path players, and projects to (D, D) in the original game. Notice that (D, D) is not a unilateral deviation of (U, U) in the original game.

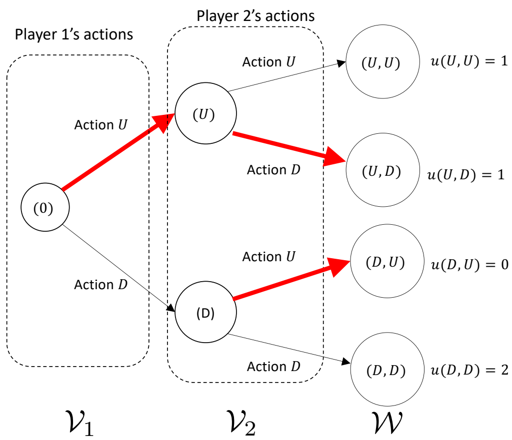
By contrast, consider the tree policy (u, u, u) represented by heavy red arcs in Figure 1. This tree policy is a pure NE in the tree game since no tree player has a profitable unilateral deviation. The tree policy also projects to the same action profile (U, U) in the original game Ξ. ◁

Example 5. Consider **Game C**, a two-person game that is a slight variation of **Game B**. The strategic form Ξ of **Game C** is captured in Table 4, and its associated tree game is captured in Figure 2. Observe that the only difference is that the utility of action profile (U, D) has changed from 0 to 1. Consider the tree policy (U, D, U) represented by heavy arcs in Figure 2. This is a pure NE in the tree game since no tree player has a profitable unilateral deviation. However, this pure NE in the tree game maps to the action profile (U, D) in the original game, which is not a pure NE in Ξ. Observe that shifting from action U to D is a profitable unilateral deviation from (U, D) for player (U) in the original game. However, this outcome cannot be reached by a unilateral deviation in the tree game since a unilateral deviation for tree player (0) (to go "down" instead of "up") projects to the action profile (D, U) in the original game. The reader may note that this issue arises because tree players (U) and (D) are taking different actions. ◁

Table 4 **The strategic form of Game C.**

		Player 2	
		U	D
Player 1	U	1	1
	D	0	2

Figure 2 **The tree game associated with Game C.**



4. MCFP on the auxiliary tree game for general identical interest games

In this section, we adapt MCFP logic to the tree game to find equilibria in the original strategic game Ξ. Whereas we showed that the iterates of MCFP-O only converge to a pure NE in the original game when that game is an identical interest coordination game, we return here to consideration of general (that is, not necessarily coordination) identical-interest games. We present two versions of MCFP applied to the tree game: MCFP-C and MCFP-I. The first is a conceptual algorithm that makes clear the basic operations of the approach but is not implementable in practice because it has the potential for making many unnecessary calculations. This is resolved in MCFP-I where careful attention is paid to when and where calculations are necessary as the algorithm proceeds.

We first discuss MCFP-C. In each iteration k and for each $i \in \mathcal{N}$, we maintain a vector $H_j^k \in \mathbb{Z}^{m_i}$ that tracks the best replies of tree player $j \in \mathcal{V}_i$. That is, for all $i = 1, 2, \ldots, n$ and every tree player j in Stage i, we have $H_j^k = (H_j^k(y_j) : y_j \in \mathcal{X}_i)$ where $H_j^k(y_j)$ is the number of times tree player j best replies with action $y_j \in \mathcal{X}_i$ through iteration k.

Conceptual version of MCFP for the auxiliary tree game (MCFP-C)

Step C.1 *Initialization.* For each tree player $j \in \mathcal{V}$, set $H_j^0 \leftarrow (0,0,\ldots,0)$. Set $k \leftarrow 1$.

Step C.2 *Draw a tree policy.* For each tree player $j \in \mathcal{V}$, draw action y_j from $\mathcal{Y}_j$ with probability $H_j^{k-1}(y_j)/(k-1)$ (if $k=1$, draw uniformly at random from $\mathcal{Y}_j$) to form a drawn tree policy $\mathbf{y}_D = (y_j)_{j\in\mathcal{V}}$.

Step C.3 *Compute a best-reply tree policy.* For each $j \in \mathcal{V}$, compute a best reply y_j^* to $\mathbf{y}_D$, breaking ties uniformly at random, to form a best-reply tree policy $\mathbf{y}_R$.

Step C.4 *Stopping Condition.* If $\mathbf{y}_R$ projects to a pure NE in Ξ then return the projection $\pi(\mathbf{y}_R)$ and terminate. Otherwise, go to **Step C.5**.

Step C.5 *Update.* For each player j, update $H_j^k(y_j^*) \leftarrow H_{y_j^* j}^{k-1}(y_j^*) + 1$; and for $y_j \neq y_j^*$, $H_j^k(y_j) \leftarrow H_j^{k-1}(y_j)$. Set $k \leftarrow k+1$ and go to **Step C.2**.

The algorithm deserves a few words of explanation. In every iteration, **Step C.2** produces an action for each tree player, which provides a drawn tree policy y_D in the tree game (the subscript 'D' connotes "draw"). This determines a unique set of path players and the remaining set of non-path players. In **Step C.3**, all tree players determine their best reply to the actions drawn in **Step C.2**. To calculate best replies, we look at unilateral deviations. For path players, unilateral deviations give rise to a different unique complete path to consider. Indeed, if path player j in node-set $\mathcal{V}_i$ considers an alternate action $a_j \in \mathcal{Y}_j$, $a_j \neq y_j$, this determines a new path of tree players in stages $i+1$ to n. This resulting tree policy y' in the tree game projects to a different action profile x' in the original game and yields a potentially different utility value.

However, for non-path players, unilateral deviations do not change the path or the projection. That is, if the unilateral deviation of a non-path player changes the tree policy in the tree game from $\mathbf{y}$ to $\mathbf{y}'$ then $\pi(\mathbf{y}) = \pi(\mathbf{y}')$ and so $v(\mathbf{y}) = v(\mathbf{y}')$. Thus, each non-path player is indifferent between all of its alternative actions because the outcome is tied, so every alternative action is a best reply. Accordingly, the stipulation in **Step C.3** to break ties uniformly at random makes the best-reply step a uniform random selection for non-path players.

In every iteration, at the end of **Step C.3**, there is a new tree policy $\mathbf{y}_R$ generated in the tree game. **Step C.4** checks if the projection $\mathbf{x} = \pi(\mathbf{y}_R)$ is a pure NE in the original game Ξ. This involves computing the utilities of all unilateral deviations $\mathbf{x}'$ to $\mathbf{x}$ and checking if $u(\mathbf{x}) \geq u(\mathbf{x}')$. We know from Proposition 2 that this check is insufficient for implying that $\mathbf{y}_R$ is a pure NE in the tree game. However, our goal is to find equilibria in the original game. Thus, in principle, there is no loss if a pure NE in the tree game is never found in the course of the algorithm.

Example 6. To illustrate the MCFP-C algorithm, we apply it to the tree game induced by **Game C**. As in Figure 2, the three tree players are represented by nodes (0), (U), and (D). Table 5 shows step-by-step the states of the algorithm, identified by the drawn tree policies and the best replies of the three tree players. We also track the histories of each tree player. In this example, in iteration 1 there is a tie for path player (U), so its best reply is also sampled uniformly from the action set.

In iteration 2, the best reply for tree player (D) is sampled as D, but, with probability 1/2, it could have been sampled as U. If the tie is broken with U, then we get the same path and it is possible to repeat the cycle for a long time. However, with probability one, ties will eventually be broken differently and the algorithm will terminate in finite time with probability one. This is formalized in Theorem 1.

The algorithm stops when the tree policy projects to a pure NE in the original game. At termination, the action profile of the original game (D, D), is not only a pure NE but also achieves the maximum utility of the original game Ξ. $\triangleleft$

Table 5 An example of MCFP-C applied to the tree game associated with Game C

Iteration k	Draw $\mathbf{y}_D$ (0)	(U)	(D)	Best reply of player $\mathbf{y}_R$ (0)	(U)	(D)	Projected policy $\pi(\mathbf{y}_R)$	Utility $u(\pi(\mathbf{y}_R))$	History of player H_j^k (0)	(U)	(D)
0									$(0,0)$	$(0,0)$	$(0,0)$
1	U	D	U	U	**D**	**U**	(U,D)	1	$(1,0)$	$(0,1)$	$(1,0)$
2	U	D	U	U	**D**	**D**	(U,D)	1	$(2,0)$	$(0,2)$	$(1,1)$
3	U	D	D	D	**D**	**D**	(D,D)	2			

Actions in bold indicate a nondeterministic choice selected randomly for illustration purposes.

When looking at the conceptual version of the algorithm, one notices that this algorithm is not efficient computationally in each iteration. Recall that there are a total of $|\mathcal{V}|$ tree players in Γ and so, in principle, **Step C.2** and **Step C.3** need to be executed for each of these $|\mathcal{V}|$ players in every iteration. We now present an implementable version of the algorithm with a far smaller computational burden.

The overall idea of the implementable algorithm is as follows. Only tree players on the unique path of the random draw in **Step C.2** need to "actively" determine a best reply (all non-path players best reply from their full action set uniformly at random). Accordingly, we do not need to maintain an explicit history for tree players that have never been in play. For players that have been in play at least once, we only update their history in iterations in which they are in play.

Similar to the vector H^k in the conceptual algorithm, we maintain a vector B^k that tracks the best replies of path players through iteration k. Specifically, for all $i = 1, 2, \ldots, n$ and every tree player j in Stage i, we have $B_j^k = (B_j^k(y_j) : y_j \in \mathcal{X}_i)$ where $B_j^k(y_j)$ is the number of times tree player j was a path player and best-replied with action $y_j \in \mathcal{X}_i$ through iteration k. We also need to keep track of the number of times a tree player j was a path player and computes a best reply, which is simply the $||\cdot||_1$-norm of the vector B_j^k; that is, $||B_j^k||_1 = \sum_{y_j\in\mathcal{X}_i} B_j^k(y_j)$ when tree player j is in Stage i.

We need to be able to efficiently draw a random action from history at each stage in a way that is stochastically equivalent to the conceptual algorithm, in the following sense.

Definition 2. We say the algorithms MCFP-C and MCFP-I are stochastically equivalent if for each iteration k, the probability of drawing the complete path $(y_1, y_2, \ldots, y_n)$ in the drawn tree policy $\mathbf{y}_D$ in MCFP-C is equal to the probability of drawing the path $\mathbf{p}_D = (y_1, y_2, \ldots, y_n)$ in MCFP-I, and the probability of projecting best-reply tree policy $\mathbf{y}_R$ in **Step C.3** to the action profile $(y_1^*, \ldots, y_n^*)$ in MCFP-C is the same probability as computing the best-reply path $\mathbf{p}_R = (y_1^*, \ldots, y_n^*)$ in MCFP-I on iteration k.

For tree player j in $\mathcal{V}_i$, a *random draw from history* at iteration k uses weighted draws from history, and the whole action set $\mathcal{Y}_j = \mathcal{X}_i$. Specifically, with probability $||B_j^{k-1}||_1/(k-1)$, action $y_j \in \mathcal{X}_i$ is drawn using historical data with probability $B_j^{k-1}(y_j)/||B_j^{k-1}||_1$, and with probability $1-(||B_j^{k-1}||_1/(k-1))$, action q is drawn with probability $1/m_i$ (that is, uniformly from the action set $\mathcal{X}_i$). In summary, the probability of drawing action $y_j \in \mathcal{X}_i$ from history for tree player $j \in \mathcal{V}_i$ at iteration k is:

$$\frac{B_j^{k-1}(y_j)}{k-1} + \left(1 - \frac{||B_j^{k-1}||_1}{k-1}\right)\frac{1}{m_i}. \tag{3}$$

If $k=1$, the probability of drawing action y_j is simply $1/m_i$.

Implementable version of MCFP (MCFP-I)

Step I.1 *Initialization.* For each tree player $j \in \mathcal{V}$, set $B_j^0 \leftarrow (0, 0, \ldots, 0)$. Set $k \leftarrow 1$.

Step I.2 *Draw a path.* For tree player (0) in Stage 1, draw the action $y_1 \in \mathcal{X}_1$ using distribution (3) and draw uniformly from $\mathcal{X}_1$ if $k = 1$. Then recursively for Stage $i = 2, 3, \ldots, n$, draw action y_i for tree player $(y_1, y_2, \ldots, y_{i-1})$ in Stage i according to distribution (3) (when $k = 1$ draw uniformly at random from $\mathcal{Y}_{(0)} = \mathcal{X}_1$). Let $\mathbf{p}_D = (y_1, y_2, \ldots, y_n)$ denote the drawn path from tree player (0) to a node in $\mathcal{W}$.

Step I.3 *Compute best replies for path players.* For $i = 1, \ldots, n$, evaluate the alternate actions of tree player $(y_1, y_2, \ldots, y_{i-1})$ in Stage i (or tree player (0) in the case of $i = 1$) as follows. For each action $a \in \{1, \ldots, m_i\}$ compute a path that reaches tree players in Stages $i+1, \ldots, n$, starting with action a as $(a, \tilde{y}_{i+1}^a, \ldots, \tilde{y}_n^a)$. If $a = y_i$, we set $\tilde{y}_h^a = y_h$ for $h = i+1, i+2, \ldots, n$, where the y_h are the drawn actions in **Step I.2**. For $a \neq y_i$, $\tilde{y}_h^a$ for $h = i+1, i+2, \ldots, n$ are drawn randomly according to distribution (3). Choose the best reply $\hat{y}_i$ uniformly at random from the set

$$\arg\max_{a \in \{1, \ldots, m_i\}} u(y_1, y_2, \ldots, y_{i-1}, a, \tilde{y}_{i+1}^a, \ldots, \tilde{y}_n^a).$$

Step I.4 *Compute a path of best replies.* In this step, we form a path of best replies $p_R = (y_1^*, y_2^*, \ldots, y_n^*)$ recursively as follows. The best reply for Stage 1 is $y_1^* = \hat{y}_1$, where $\hat{y}_1$ is as computed in **Step I.3**. If $\hat{y}_1 \neq y_1$, then the best replies for Stages 2 through n must be determined for non-path players, which are sampled uniformly from their action sets. If $\hat{y}_1 = y_1$, then the best reply for Stage 2 is set as $y_2^* = \hat{y}_2$. If $\hat{y}_1 = y_1$ and $\hat{y}_2 \neq y_2$, then the best replies for Stages 3 through n must be determined for non-path players by sampling uniformly from their action sets. If $\hat{y}_1 = y_1$ and $y_2^* = y_2$, then set $y_3^* = \hat{y}_3$ and continue in this fashion. In this manner, the best-reply path, $\mathbf{p}_R = (y_1^*, y_2^*, \ldots, y_n^*)$ is constructed.

Step I.5 *Stopping condition.* If $\mathbf{p}_R$ is a pure NE in Ξ then return that pure NE and terminate the algorithm. Otherwise, go to **Step I.6**.

Step I.6 *Update.* For each tree player $j \in \mathcal{V}$ in the path p_R at Stage i, update $B_j^k(y_i) \leftarrow B_j^{k-1}(\hat{y}_i) + 1$; and for $y_i \neq \hat{y}_i$, $B_j^k(y_i) \leftarrow B_j^{k-1}(y_i)$. Set $k \leftarrow k+1$ and go to **Step I.2**.

A few words of explanation are in order. The draws that occur in **Step I.2** of MCFP-I are simulating a small subset of those that would have occurred in **Step C.2** in the conceptual algorithm. In particular, only a single path is generated through the tree as opposed to a whole tree policy, as in the conceptual algorithm. Having said that, parts of **Step C.2** of the conceptual algorithm need to be executed in **Step I.3** of the implementable algorithm. Indeed, in order to compute best replies for the path players, alternate paths in the tree need to be “drawn” and compared with. In other words, **Step I.3** of MCFP-I includes a combination of computations in **Step C.2** and **Step C.3** of MCFP-C.

Step I.4 provides the portion of the best-reply tree policy $\mathbf{y}_R$ in **Step C.3** of MCFP-C that is equivalent to the projection $\pi(\mathbf{y}_R)$ in **Step C.4**. In MCFP-C, the complete path to project is clear from the tree policy $\mathbf{y}_R$. However, in **Step I.3**, the best replies of the path players computed in this step need not form a complete path. Accordingly, **Step I.4** must be executed in order to construct the path $\mathbf{p}_R$ to be used in **Step I.5**. In particular, **Step I.4** can be seen as part of the original best-reply step (**Step C.3**) in MCFP-C, here executed if a best reply of a path player directs away from the original path. It turns out, however, that the best replies in this step need not be recorded in history since either they are the same as drawn in the previous step or are uniformly selected from the set of actions. This allows for polynomial iteration complexity, as described in Proposition 4 below.

It is also critical to note that **Step I.3** plays a very important role in the convergence properties for the algorithm, even when it produces actions $\hat{y}_i$ that are different from those in the path $\mathbf{p}_R$. Every action choice outside of **Step I.3** is "random". It is only in **Step I.3** that an optimization step needs to be performed to compute the best reply. In other words, **Step I.3** is the "signal" the algorithm uses to make "smart" choices, with other steps aiding future "exploration".

Example 7. To illustrate MCFP-I, we apply it to the tree game associated with **Game C**. Table 6 shows the step by step implementation. Comparing Table 6 for MCFP-I with Table 5 for MCFP-C, we can see that MCFP-I draws a complete path and determines a best-reply path using the two path players (as opposed to three tree players with MCFP-C). We also show the histories of each path player in B_j^k (as opposed to H_j^k in MCFP-C).

On the first iteration, the drawn action for tree player (0) is uniform from $\{U, D\}$ because $k = 1$. The first iteration for MCFP-I is the same as for MCFP-C, with the exception that there is no explicit draw or best reply calculation for tree player (D). In the second iteration, the probability of choosing U for tree player (0) in MCFP-I is the same as for MCFP-C. And the second iteration is also comparable. In the third iteration, the drawn path is again consistent with MCFP-C, and the best reply for tree player (0) is D. Although the random action used in the best reply is not recorded explicitly, the probability that the Stage 2 action is D for MCFP-I is the same probability as MCFP-C. $\triangleleft$

Table 6 An example of MCFP-I applied to the tree game associated with Game C

Iteration k	Draw of player $\mathbf{p}_D$ (0)	(U)	(D)	Best reply of player $\mathbf{p}_R$ (0)	(U)	(D)	Best-reply path path $\mathbf{p}_R$	Payoff off $u(\mathbf{p}_R)$	History of player B_j^k (0)	(U)	(D)
0									$(0,0)$	$(0,0)$	$(0,0)$
1	U	D	-	U	**D**	-	(U,D)	1	$(1,0)$	$(0,1)$	$(0,0)$
2	U	D	-	U	**D**	-	(U,D)	1	$(2,0)$	$(0,2)$	$(0,0)$
3	U	D	-	D	**D**	-	(D,D)	2			

Actions in bold indicate a nondeterministic choice that was selected randomly for purposes of illustration.

We summarize the above discussion in the following result. For brevity, a detailed proof beyond the above discussion is omitted.

PROPOSITION 3. *The algorithms MCFP-C and MCFP-I are stochastically equivalent, in the sense defined in Definition 2. The stopping conditions for both algorithms are also equivalent.*

The next result studies the iteration complexity of the algorithm. Note that the number of tree players is $|\mathcal{V}| = 1 + \sum_{j\in\mathcal{V}} \prod_{k=1}^{j} m_k$, which is on the order of $O(m^n)$ where $m = \max_{i=1,\dots,n}\{m_i\}$.

PROPOSITION 4. *Each iteration of MCFP-I requires $O(mn^2)$ random samples and $O(mn)$ utility function calls.*

Proof of Proposition 4. **Step I.2** entails n draws from history, since only n stages are needed to determine a complete path. Each of these n draws include a random sample from an action set with at most m actions. For each path player, **Step I.3** makes at most m utility evaluations to explore all unilateral deviations, and each unilateral deviation requires at most n random draws from history. Altogether, for n players, **Step I.3** entails $O(mn^2)$ random samples to generate mn alternative paths. Each alternate path requires a utility function to evaluate for deciding a best reply for a total of mn utility function calls. **Step I.4** constructs $\mathbf{p}_R$ and samples random actions at most n times without calling the utility function. Finally, **Step I.5** also makes mn utility function calls in order to check if the projection is a pure NE in the original game Ξ. $\square$

5. Analysis of MCFP

In this section, we analyze the performance of MCFP-C and MCFP-I, as well as a "mixed" algorithm MCFP-M that combines MCFP-O and MCFP-I.

We first adapt the definition of "finite time with probability one" given in Definition 1 to our current setting.

DEFINITION 3. Let F_k denote the event that p_R in **Step I.4** is a pure NE in iteration k of the MCFP-I algorithm. Let F denote the union of all F_k; that is, $F = \bigcup_{k=1}^{\infty} F_k$. Then we say MCFP-I *terminates with a pure NE in finite time with probability one* if the probability of event F is one.

Consider the MCFP-I algorithm where we ignore the stopping condition **Step I.5**. That is, the algorithm continues to run regardless of whether p_R is a pure NE or not. Under this condition, let G_k denote the event that p_R in **Step I.2** is an *optimal* pure NE in iteration k of the MCFP-I algorithm. Then we say MCFP-I *finds an optimal pure NE in finite time with probability one* if the probability of event $G = \bigcup_{k=1}^{\infty} G_k$ is one. ◁

Theorem 1. *Let Ξ be a strategic identical interest game whose corresponding tree game Γ is taken as input to algorithm MCFP-I. Then (i) MCFP-I terminates with a pure NE in finite time with probability one, and (ii) MCFP-I finds an optimal pure NE in finite time with probability one.*

The proof of this result is subsumed by a later result (Theorem 2). We defer the argument until that point.

Remark 2. Observe that (ii) in Theorem 1 implies that the algorithm produces a sequence of utility values that eventually yield the optimal utility. It is important to stress, however, that the algorithm cannot verify that this utility is, in fact, optimal. We know of no simple stopping condition that can certify global optimality. ◁

Remark 3. As argued in Example 2, MCFP-O does not enjoy property (ii) in Theorem 1; namely, that an optimal pure NE is found in finite time with probability one. Even running the algorithm indefinitely may not uncover the optimal pure NE because it gets absorbed in a non-optimal equilibrium. ◁

Theorem 1 has attractive convergence properties. However, in our numerical experiments in Section 6 we still find that MCFP-I can require significant computational effort to find a pure NE, despite it being faster than many other known methods. By contrast, we find in those same numerical experiments that MCFP-O finds a pure NE more rapidly, despite not having a theoretical guarantee of convergence to a pure NE. Moreover, each iteration of MCFP-O requires less computation than an iteration of MCFP-I. In the remainder of this section, we show how to "mix" MCFP-I and MCFP-O to get a "best of both worlds".

The first step to construct this "mixing" is to adapt MCFP-O to the tree game. We call this algorithm Structured Monte Carlo Fictitious Play (MCFP-S). MCFP-S mimics MCFP in the original game by controlling the "structure" of the draws and best replies to mimic how they would appear if the algorithm was applied to the original game; namely, where tree players in the same stage have the same history and take the same actions.

Structured Monte Carlo fictitious play (MCFP-S)

Step S.1 *Initialization.* For each Stage $i \in \mathcal{N}$, set $S_i^0 \leftarrow (0, 0, \ldots, 0)$. Set $k \leftarrow 1$.

Step S.2 *Draw a path.* For each Stage i, draw $y_i \in \mathcal{X}_i$ with probability $S_i^{k-1}(y_i)/(k-1)$ (if $k = 1$, draw uniformly at random from $\mathcal{X}_i$), resulting in a drawn path $\mathbf{p}_D = (y_1, y_2, \ldots, y_n)$.

Step S.3 *Compute best replies for path players.* For $i \in \mathcal{N}$ compute the best reply y_i^* for the tree player in p_D in Stage i. If a non-path player j is reached, take that non-path player's action to be y_i (as drawn in **Step S.2**) when $j \in \mathcal{V}_i$. Let $\mathbf{s}_R = (y_1^*, \ldots, y_n^*)$ be the best-reply path.

Step S.4 *Stopping Condition.* If $\mathbf{s}_R$ is a pure NE in Ξ then return $\mathbf{s}_R$ and terminate. Otherwise, go to **Step S.5**.

Step S.5 *Update.* For all $i \in \mathcal{N}$, update $S_i^k(y_i^*) \leftarrow S_i^{k-1}(y_i^*) + 1$; and for $y_i \neq y_i^*$, $S_i^k(y_i) \leftarrow S_i^{k-1}(y_i)$. Update $k \leftarrow k + 1$ and go to **Step S.2**.

The algorithms MFCP-S and MCFP-I differ in how best replies are constructed. In the MCFP-I algorithm, tree players in the same Stage i can draw different actions whereas in MCFP-S there is uniformity across stages. This alters the "alternate paths" that a player experiences when considering unilateral deviations, and thus ultimately can impact their calculation of best replies. In the mixed algorithm below (MCFP-M), iterations execute one of two types of best replies depending on a parameter α. We need to keep track of this history of best replies in order to compute the probability of drawing an action in the draw step. Here we need to track both MCFP-I best replies *and* MCFP-S best replies. The caveat here is that MCFP-I best replies are at the tree player level whereas MCFP-S are at the stage level. As in MCFP-I, we let $B_j^{k_I}$ denote the vector of best-reply counts accrued through executing k_I MCFP-I-style best replies for tree player $j \in \mathcal{V}$ and we let $S_i^{k_S}$ denote the vector of best-reply counts accrued through k_S MCFP-S-style best replies for players in Stage i.

Thus, the probability of drawing an action in the draw step is more complicated than it was in MCFP-I (see formula (3)). Here, the unconditional probability of drawing action $y_i \in \{1, \ldots, m_i\}$ from history for tree player $j \in \mathcal{V}_i$ after $k_I - 1$ calls to MCFP-I-style best replies and $k_S - 1$ calls to MCFP-S-style best replies is:

$$\frac{B_j^{k_I-1}(y_i) + S_i^{k_S-1}(y_i)}{k_I + k_S - 2} + \left(1 - \frac{||B_j^{k_I-1}||_1 + ||S_i^{k_S-1}||_1}{k_I + k_S - 2}\right)\frac{1}{m_i} \tag{4}$$

when $k_I + k_S > 2$ and equal to $1/m_i$ otherwise.

Mixed Monte Carlo Fictitious Play (MCFP-M)

Step M.1 *Initialization.* For each Stage $i \in \mathcal{N}$, set $S_i^0 \leftarrow (0,0,\ldots,0)$ and for each tree player $j \in \mathcal{V}$, set $B_j^0 \leftarrow (0,0,\ldots,0)$. Set $k_I \leftarrow 1$ and $k_S \leftarrow 1$ and input $\alpha \in [0,1]$.

Step M.2 *Draw a path.* For tree player (0) in Stage 1, draw action $y_1 \in \mathcal{X}_1$ using distribution (4). Then, recursively for $i = 2,3,\ldots,n$, draw action y_i for tree player $(y_1, y_2, \ldots, y_{i-1})$ in Stage i according to distribution (4). Let $\mathbf{p}_D = (y_1, y_2, \ldots, y_n)$ denote the drawn path from player (0) to a node in $\mathcal{W}$.

Step M.3 *Mixing step.* With probability α go to **Step M.4**, otherwise, go to **Step M.5**.

Step M.4 *Best reply step of MCFP-I*

Step M.4.1 *Compute a best-reply path.* Execute **Step I.3** and **Step I.4** where draws from history follow (4) instead of (3) to form the best-reply path $\mathbf{p}_R = (y_1^*, \ldots, y_n^*)$.

Step M.4.2 *Stopping condition.* If $\mathbf{p}_R$ is a pure NE in Ξ then return that pure NE and terminate the algorithm. Otherwise, go to **Step M.4.3**.

Step M.4.3 *Update.* For each tree player $j \in \mathcal{V}$ in the path $\mathbf{p}_R$ at Stage i, update $B_j^{k_I}(\hat{y}_i) \leftarrow B_j^{k_I-1}(\hat{y}_i) + 1$; and for $y_i \neq \hat{y}_i$, $B_j^{k_I}(y_i) \leftarrow B_j^{k_I-1}(y_i)$. Set $k_I \leftarrow k_I + 1$ and go to **Step M.2**.

Step M.5 *Best reply set of MCFP-S*

Step M.5.1 *Compute best replies for path players.* Execute steps analogous to **Step S.3** (only now referring to draws in **Step M.2**). Let $\mathbf{p}_R = (y_1^*, \ldots, y_n^*)$ be the resulting best-reply path.

Step M.5.2 *Stopping condition.* If $\mathbf{p}_R$ is a pure NE in Ξ then return $\mathbf{p}_R$ and terminate. Otherwise, go to **Step M.5.3**.

Step M.5.3 *Update.* For all $i \in \mathcal{N}$, update $S_i^{k_S}(y_i^*) \leftarrow S_i^{k_S-1}(y_i^*) + 1$; and for $y_i \neq y_i^*$, $S_i^{k_S}(y_i) \leftarrow S_i^{k_S-1}(y_i)$. Set $k_S \leftarrow k_S + 1$ and go to **Step M.2**.

We are now ready to prove the main result of the paper. The result refers to the definitions in Definition 3, but applied to algorithm MCFP-M instead of algorithm MCFP-I (with the appropriate straightforward changes).

Theorem 2. *Let Ξ be a general identical interest game whose corresponding tree game Γ is taken as input to algorithm MCFP-M with parameter $0 < \alpha \leq 1$. Then (i) MCFP-M terminates with a pure NE in finite time with probability one, and (ii) MCFP-M finds an optimal pure NE in finite time with probability one.*

Proof of Theorem 2. Consider an optimal path of tree nodes, denoted $p_1^*, p_2^*, \ldots, p_n^*$, with associated optimal actions $y_1^*, y_2^*, \ldots, y_n^*$, (i.e. $(y_1^*, y_2^*, \ldots, y_n^*)$ forms an optimal solution to (1)) with a utility value of u^*. Also, suppose that the number of optimal actions at each of these optimal nodes is at most ℓ, thus allowing multiple optima.

Begin MCFP-M by drawing an arbitrary action for each node. If the drawn actions include $y_1^*, y_2^*, \ldots, y_n^*$ for optimal path players $p_1^*, p_2^*, \ldots, p_n^*$, that is, if y_D is an extension of this optimal solution, then a best reply to $\mathbf{y}_D$ (structured or unstructured) also has an optimal utility value of u^*. Its projection is a pure NE for the original game and the algorithm terminates with a pure NE for Ξ. This yields (i). Thus, it suffices to show that actions $y_1^*, y_2^*, \ldots, y_n^*$ can be drawn.

We now show that in each iteration k, the probability of drawing optimal actions $y_1^*, y_2^*, \ldots, y_n^*$ for optimal nodes $p_1^*, p_2^*, \ldots, p_n^*$ is at least $(\alpha/m)^n$ independent of past draws and best replies where m is an upper bound on the number of feasible actions at each node.

Adopt the inductive hypothesis on i that at every iteration k, the probability of drawing optimal actions $y_i^*, y_{i+1}^*, \ldots, y_n^*$ for optimal nodes $p_i^*, p_{i+1}^*, \ldots, p_n^*$ is at least $(\alpha/m)^{n-i+1}$ independently of the past. Note that the inductive hypothesis is satisfied for $i = n$ since before iteration k, either p_n^* was in play and loaded action y_n^* into its history with probability at least $1/\ell$, independent of the past, or it was not in play and loaded action x_n^* into its history with probability at least $1/m$ if its best reply is unstructured which happens with probability α. Therefore, x_n^* gets loaded independently into history for iterations before k with probability at least α/m and therefore is drawn in iteration k with probability at least α/m. Consider now node p_{i-1}^*. At each iteration before k, if p_{i-1}^* was not in play, it best replied randomly with probability α and loaded action y_{i-1}^* into its history with probability at least $1/m$. If it was in play, it best replied with optimal action y_{i-1}^* with probability at least $1/\ell$, if optimal actions $y_i^*, y_{i+1}^*, \ldots, y_n^*$ were drawn for the subsequent optimal nodes $p_i^*, p_{i+1}^*, \ldots, p_n^*$. But this happens with probability at least $(\alpha/m)^{n-i+1}$ independently of the past by the inductive hypothesis. Hence p_{i-1}^* when in play best replies and loads y_{i-1}^* into its history with probability at least $(1/\ell)(\alpha/m)^{n-i+1}$. In either case (in-play or not), p_{i-1}^* loads y_{i-1}^* into its history with probability at least $(\alpha/m)^{n-(i-1)+1}$ thus restoring the inductive hypothesis. By setting $i = 1$, we conclude that the probability of drawing the optimal actions $y_1^*, y_2^*, \ldots, y_n^*$ for optimal nodes $p_1^*, p_2^*, \ldots, p_n^*$ is at least $(\alpha/m)^n$.

We have shown that the probability that we draw an optimal path and consequently the best reply is optimal, at any iteration k is at least $\delta = (\alpha/m)^n > 0$ independent of what occurred in iterations 1 through $k-1$. In the terminology of Definition 3, we have thus shown that the probability of G_k is at least δ. We next show that the event G (in the terminology of Definition 3) has probability 1, completing the proof.

Let $G_{\leq k}$ denote the event that the algorithm finds an optimal path within k iterations. That is, $G_{\leq k} = \bigcup_{j=1}^{k} G_j$. Let $\bar{G}_k$ denote the complement event of G_k, and so we know $P(\bar{G}_k) \leq 1 - \delta$. Now, consider the event $\bar{G}_{\leq k} = \bar{G}_1 \cap \bar{G}_2 \cap \cdots \cap \bar{G}_k$ that the algorithm does not terminate in the first k

iterations. That is, $P(\bar{G}_{\leq k}) = P(\bar{G}_1 \cap \bar{G}_2 \cap \cdots \cap \bar{G}_k) = P(\bar{G}_1)P(\bar{G}_2)\cdots P(\bar{G}_k) \leq (1-\delta)^k$. From here we have

$$P(G_{\leq k}) = 1 - P(\bar{G}_{\leq k}) \geq 1 - (1-\delta)^k \tag{5}$$

Observe that the event $G_{\leq k}$ is contained in the event G. So, in particular, $P(G_{\leq k}) \leq P(G)$. Now, suppose that $P(G) = \beta < 1$. This implies that $P(G_{\leq k}) \leq P(G) = \beta$ for all k. However, this contradicts (5) because there exists a $k(\beta)$ such that $P(G_{\leq k(\beta})) > \beta$. Contradiction. That is, we eventually find an optimal path in finite time with probability one. ■

Observe that Theorem 1 is a direct consequence of the above result taking $\alpha = 1$. The proof of Theorem 2 includes, as part of its argument, intermediaries that are similar in spirit to the proofs found in Section 4.2 of (Dolinskaya et al. 2016).

6. Numerical experiments

In the following section, we explore the practical performance of our algorithms. The measure of "speed" here is in terms of the number of calls to the utility function $u(x)$. Since our algorithms involve random draws and random tie-breaking, performance is averaged over multiple replications (50 instances for the coordination game, and 30 for the drone example). Performance is compared to fictitious play with memory and inertia FP-MI introduced in (Young 2004) and studied more recently in (Swenson et al. 2018).

We describe FP-MI briefly here. *Fictitious play with memory* is a process in which each player chooses the best reply in expected utility based on the empirical distribution of past plays by their opponent(s) where more recent plays receive more weights. We consider two versions of the fictitious play with memory in the next two subsections. In the first subsection, we consider the fictitious play with finite memory. In this version, controlled by the memory size M, the empirical distribution of the plays at iteration k is built considering actions taken by the players at iterations $k-M,\ldots,k-1$. In the second subsection, we consider the fictitious play with fading memory. In the fictitious play with fading memory, the empirical distribution of the plays at iteration k is defined recursively as the convex combination of the latest empirical distribution at iteration $k-1$ and the last play. In particular, let $f_{i,k}$ be the empirical distribution of player i's plays at iteration k and let $\varphi(a_{i,k})$ be the degenerate probability distribution placing mass 1 on player i's action $a_{i,k}$ at iteration k. Controlled by the fading memory parameter γ, the empirical distribution of player i's plays at iteration k is defined recursively as

$$f_{i,k} = (1-\gamma)f_{i,k-1} + \gamma\varphi(a_{i,k}). \tag{6}$$

It has been shown that fictitious play can fail to converge to a pure NE (Young 2004). To avoid such behavior, inertia is introduced. Specifically, assume that a player takes the same action as in

Figure 3 **Av. no. of utility function accesses to obtain first pure NE.**

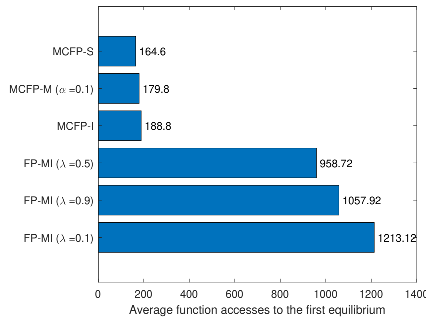
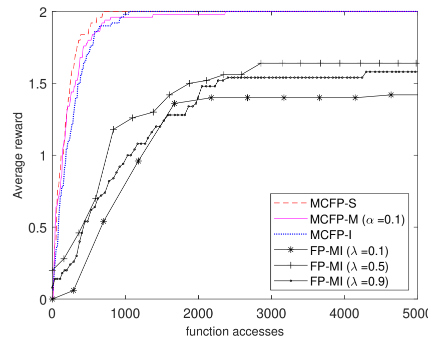


Figure 4 **Best pure NE utility averaged over 50 simulations versus function accesses.**



the previous period with probability $\lambda \in (0,1)$ and chooses the best reply against the product of the empirical distributions with probability $(1-\lambda)$. If the previous action is within the current set of best replies, the player plays it again, so that inertia is used to break the tie. Convergence to a pure NE for FP-MI was proven in (Young 2004). We will use FP-MI generically to refer to both the finite memory and fading memory versions.

In the next two subsections, we show that our algorithms perform favorably in comparison to FP-MI when comparing calls to the utility function to find a first pure NE. We are also interested in the *quality* of the found pure NEs. As discussed in Remark 3, our algorithms can be run without terminating when the first pure NE is reached, if left to run, both MCFP-I and MCFP-M find an optimal pure NE in finite time with probability one. In our numerical investigations, we terminate after a large number of utility function accesses and track the “best” pure NE reached to that iteration.

6.1. Coordination games

We first apply our different implementations of MCFP to find equilibria in identical interest coordination games. In these experiments, we assume each player has two actions; that is, $\mathcal{X}_i = \{U,D\}$ for all $i \in \mathcal{N}$. As a result, there are 2^n possible action profiles and only two possible equilibria: $(U,U,\ldots,U)$ and $(D,D,\ldots,D)$. We assume that $u(U,U,\ldots,U) = 2$ and $u(D,D,\ldots,D) = 1$.

We consider the scenario with 5 players (we also tried 10 players with qualitatively similar results). Figure 3 shows the performance for finding the first pure NE. Each of the algorithms has a stopping rule to terminate once an equilibrium is reached, and so the data in Figure 3 can be viewed as average termination times under the stopping rule. This data suggests that the MCFP variants outperform FP-MI under different parameter specifications. We present three alternate parameter specifications; other choices gave similar results.

The fact that MCFP-I reaches an equilibrium with fewer function calls than FP-MI is evidence that relatively few nodes in the auxiliary tree are ever reached, reaping the benefits of the tree structure without having to process much of its exponential size. Accordingly, this numerical performance in Figure 3 is consistent with the polynomial iteration complexity given in Proposition 4. The relative performance of MCFP-I, MCFP-M, and MFCP-S to one another is also consistent with our theoretical understanding of these algorithms. MCFP-S requires less work in each iteration, which is consistent with the numerical finding that it can find equilibria with fewer utility function accesses. The intermediate number of function calls demonstrated by MCFP-M is also consistent with its construction as a hybrid algorithm. We tried different values of α and M, but only report $\alpha = 0.1$ since other values of α gave qualitatively similar results.

Figure 4 captures the performance of these algorithms in discovering a pure NE with optimal utility (here a utility of 2). We chose $5,000$ as an upper bound on function accesses since this choice demonstrated a pattern where the MCFP variants find a high-quality pure NE, on average, faster than FP-MI. The figure also illustrates that MCFP-S, MCFP-M, and MCFP-I have quite similar performance on this coordination game; all can reach the optimal pure NE with high frequency within the allotted $5,000$ calls. Our results illustrate a slight edge to MCFP-M, which is consistent with the "best-of-both-worlds" design of the algorithm, although the distinctions between the performance of each variant appear to be quite minimal. The fact that MCFP-S quickly tracks towards the optimal equilibria is also consistent with Proposition 1 that guaranteed the convergence of MCFP-O (and thus MCFP-S) to a pure NE.

6.2. A drone coordination problem

We apply our algorithms to the UAV (unmanned aerial vehicle or "drone") target assignment problem proposed in (Swenson et al. 2018). The UAVs can communicate with each other using short-range radio to negotiate a feasible target assignment, resulting in a game, as follows. Suppose there are n UAVs and n target objects. Each UAV is assigned one target and the goal is to cover all targets by assigned UAVs. The action space for each UAV is the set of targets $\{1, 2, \ldots, n\}$. The utility of assigning UAV i to target k (that is, setting $x_i = k$), given the assignment $\mathbf{x}_{-i}$ of the other UAVs, is proportional to the distance $d(i,k)$ from the UAV to the target. Precisely,

$$u_i(x_i = k, \mathbf{x}_{-i}) = d(i,k)^{-1}\mathbf{1}\left(\sum_{j=1}^{n} \mathbf{1}(x_j = k) = 1\right), \tag{7}$$

where $\mathbf{1}$ is the indicator function. Observe that the sum $\sum_{j=1}^{n} \mathbf{1}(x_j = k)$ counts the number of drones that are assigned to target k, and the outer indicator function (with this sum as an argument) means that utility is only assigned when a single drone is assigned to a target.

Let the positions of the objects be equally spaced on a unit circle centered at the origin of a two dimensional plane, i.e., object j, for $j = 1, \ldots, n$ is located at coordinate $(\cos(2\pi j/n),\ \sin(2\pi j/n)$. The location of UAV i, for $i = 1, 3, 5, \ldots, n-1$, is at coordinate $(\cos(2\pi i/n - \pi/16n),\ \sin(2\pi i/n -$

$\pi/16n)$. The location of UAV i, for $i = 2,4,6,\ldots,n$, is at coordinate $\cos(2\pi(i-1)/n + \pi/2n)$, $\sin(2\pi(i-1)/n + \pi/2n)$.

From (7), we can see that the drone assignment problem is not an identical interest game since each player has a different utility function. However, we can recast the problem as an identical interest game with common utility $w(\mathbf{x}) = \sum_{i=1}^{n} u_i(x)$ (see Proposition 5 below) with the overall optimization problem being solved as $\max\{w(\mathbf{x}) : x_i \in \{1,2,\ldots,n\}\}$. Equilibria are the assignments of one drone to one object. Each pure NE is an action of the UAV's to cover all objects. There is one global optimum, when UAV i targets object i for $i = 1,\ldots,n$.

Proposition 5. *An assignment of drones to targets $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{Z}^n$ is an equilibrium with respect to the game with the original utility functions* (7) *if and only if it is an equilibrium with respect to the identical interest game with common utility function equal to the welfare $w(\mathbf{x}) = \sum_{i=1}^{n} u_i(\mathbf{x})$.*

Proof of Proposition 5. Let $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ be an assignment that is an equilibrium for utility functions (7). Therefore, $\mathbf{x}^*$ is a permutation of $\{1,\ldots,n\}$, and, from the definition of the utility functions, $u_i(\mathbf{x}^*) > 0$ for all i. Fix i and fix $\mathbf{x}_{-i}^*$. Let $\mathbf{x}_i^*$ be replaced by a different x_i', forming an alternative assignment $\mathbf{x}' = (x_1^*, \ldots, x_i', \ldots, x_n^*)$. There is a clash in the assignment, i.e., there exists $j \neq i$ such that $x_j^* = x_i'$. Therefore, $u_i(\mathbf{x}')$ and $u_j(\mathbf{x}')$ become zero, causing $w(\mathbf{x}') < w(\mathbf{x}^*)$. Therefore, $\mathbf{x}^*$ is also an equilibrium with respect to the welfare function. Conversely, consider an assignment $\mathbf{x}' = (x_1', \ldots, x_n')$ that is not an equilibrium with respect to the utility functions. Therefore, $\mathbf{x}'$ is not a permutation of $\{1,\ldots,n\}$. Some objects have no assignment, i.e. there exists k in $\{1,\ldots,n\}$ such that that $x_i' \neq k$ for all i, and some object will have more than one assignment, i.e. there exists l in $\{1,\ldots,n\}$ such that $x_p' = x_q' = l$ for some p,q in $\{1,\ldots,n\}$. Therefore, $u_p(\mathbf{x}') = u_q(\mathbf{x}') = 0$. Fixing x_{-p}', let $x_p' = k$ and form a unilateral reply $\mathbf{x}'' = (x_1', \ldots, x_p' = k, \ldots, x_n')$ by player p. Object k is covered by only player p. By the definition of the utility function, $u_p(\mathbf{x}'') > 0 = u_p(\mathbf{x}')$. As a result, $w(\mathbf{x}'') > w(\mathbf{x}')$. Therefore, this unilateral reply by player p can improve the welfare function. The assignment $\mathbf{x}'$ is not an equilibrium with respect to the welfare function. $\square$

We study the performance of our MCFP variants and FP-MI. We set the fading memory parameter to 0.2 and the inertia parameter to $\lambda = 0.2$, the same parameter set found in (Swenson et al. 2018). For MFCP-M, we consider mixing parameter and $\alpha = 0.1$. We consider the case with 10 drones.

We measure the performance of each algorithm by the relative welfare achieved by each of them against the number of accesses to the utility function (in this case, the welfare function). We apply all of the algorithms until 100,000 welfare function accesses. We perform 30 replications for each algorithm and average the results. Figure 5 shows the number of average welfare function accesses

Figure 5 **Av. no. of welfare function accesses to first pure NE.**

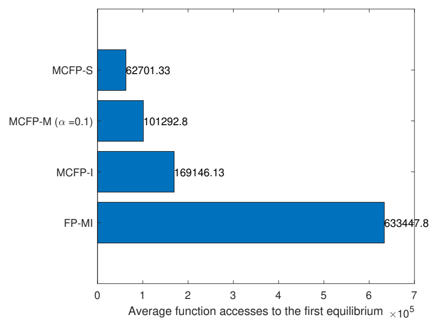
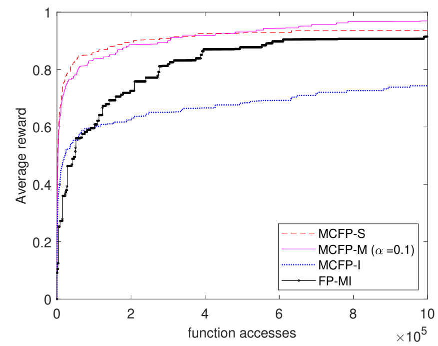


Figure 6 **Best relative welfare versus function accesses.**

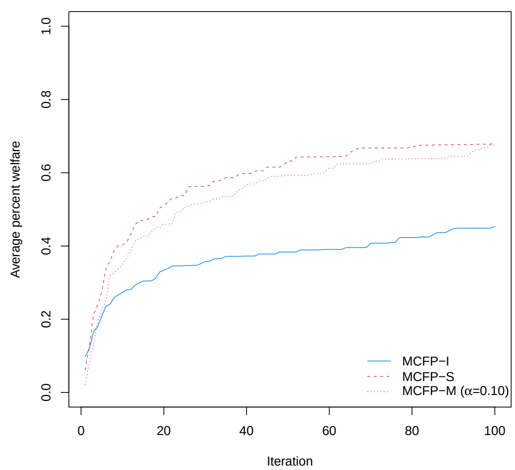


to obtain the first pure NE of the four algorithms we study. Figure 6 shows the relative welfare found up to each welfare function access.

The fact that FP-MI needs many more calls to the welfare function to reach a pure NE (see Figure 5) underscores the speed-up due to a single sampling from history at each iteration that is characteristic of MCFP variants. Among the MCFP variants, Figure 5 also confirms our intuition that MCFP-S can reach a pure NE faster than MCFP-I and the mixed algorithm MCFP-M modulates their performance. Theorem 2 guarantees that MCFP-M eventually uncovers a pure NE with maximal welfare, and this is reflected in the fact that the MFCP-M curve overtakes the MCFP-S curve in terms of average percent of welfare in Figure 6 around halfway through the simulation. We should note that FP-MI appears to outperform MCFP-I in terms of progress towards finding an optimal equilibrium given the iteration count (as seen in Figure 6); however, this simulation tracks the utilities of the best performing iterates, and these iterates need not be equilibria. As we can see in Figure 5, FP-MI progresses slowly towards equilibria.

Finally, the action profiles from the initial iterations of the MCFP algorithms (MCFP-I, MCFP-S, MCFP-M) can sometimes serve as estimates for NE (Nash Equilibrium). At specific stages of each algorithm, namely Step I5 of MCFP-I, Step S4 of MCFP-S, and Step M4.2 and M5.2 of MCFP-M, the algorithms verify the stopping criteria. If these criteria are met, the corresponding action profiles are indeed NE's. In the case of the drone coordination problem, from the 30 simulations, the first NE can be identified as early as the 7-th iteration for MCFP-M ($\alpha = 0.1$), the 21-st iteration for MCFP-S, and the 27-th iteration for MCFP-I. However, when the stopping criteria are not satisfied, the action profile at the end of each iteration from any of these algorithms cannot be considered an estimate for NE. Nevertheless, in these non-NE scenarios, the so-far best common interest utility can be seen as the lower bound of the globally optimal NE. Figure 7 illustrates the so-far best common interest utility (welfares), averaged over 30 simulations, up to 100 iterations for the drone coordination problem.

Figure 7 Best relative welfare, averaged over 30 simulations, up to 100 iterations.



7. Conclusion

In this paper, we have developed several variants of a fictitious-play algorithm that sample history in determining how players best reply as the algorithm proceeds. These algorithms (MCFP-O, MCFP-C, MFCP-I, MCFP-S, and MCFP-M) each have their advantages and disadvantages. MCFP-O (equivalent to MCFP-S) focuses on likely equilibria in the underlying game, giving rise to rapid convergence empirically, but may not converge to a pure NE as the algorithm proceeds. MCFP-C is easy to work with theoretically and can identify a pure NE with probability one, but suffers from operating on the whole tree $(\mathcal{V} \cup \mathcal{W}, \mathcal{A})$ at each iteration. MCFP-I enjoys the theoretical convergence properties of MCFP-C but with less of a computational burden. The mixed algorithm MCFP-M balances the benefits of MCFP-S (lower computational burden) with MCFP-I (nice convergence properties). An open question is whether the MCFP-O algorithm applied to the original game converges to a pure NE in a general identical interest game.

There remain several unanswered questions about these MCFP algorithms that could be the subject of further investigation. First, although we can show that MCFP-C identifies a pure NE with probability one, one may theoretically ask how many iterations are expected before "absorption" into a pure NE. There seems some hope that an analysis using Markov chains with absorbing states might provide some insight, possibly on subclasses of identical interest games (for example, coordination games). Second, one could ask whether other classes of games, beyond identical interest and potential games, are amenable to MCFP-methods for finding pure NE. An extension to other games where fictitious play is known to converge (say the 2 by n games of (Berger 2005)) seems plausible, although other classes may be possible. Third, one may search for special classes of identical interest games where MCFP methods perform particularly well in comparison to other known algorithms.

References

Abernethy JD, Lai KA, Wibisono A (2019) Fictitious play: Convergence, smoothness, and optimism. *CoRR* abs/1911.08418, URL `http://arxiv.org/abs/1911.08418`.

Berger U (2005) Fictitious play in 2 × n games. *J Econ Theory* 120(2):139–154.

Brown GW (1951) Iterative solution of games by fictitious play. *Activity Analysis of Production and Allocation* 13(1):374–376.

Dolinskaya IS, Epelman MA, Şişikoğlu Sir E, Smith RL (2016) Parameter-free sampled fictitious play for solving deterministic dynamic programming problems. *J Opt Theory and App* 169:631–655.

Garcia A, Patek SD, Sinha K (2007) A decentralized approach to discrete optimization via simulation: Application to network flow. *Operations Research* 55(4):717–732.

Lambert TJI, Epelman MA, Smith RL (2005) A fictitious play approach to large-scale optimization. *Operations Research* 53(2):477–489.

Lei J, Shanbhag UV (2020) Asynchronous schemes for stochastic and misspecified potential games and nonconvex optimization. *Operations Research* 68(6):1742–1766.

Lei J, Shanbhag UV, Pang JS, Sen S (2020) On synchronous, asynchronous, and randomized best-response schemes for stochastic nash games. *Mathematics of Operations Research* 45(1):157–190.

Li Z, Tewari A (2018) Sampled fictitious play is hannan consistent. *Games and Economic Behavior* 109:401–412.

Ma WC, Huang DA, Lee N, Kitani KM (2017) Forecasting interactive dynamics of pedestrians with fictitious play. *IEEE Conf on Comp Vision and Pattern Recog*, 774–782.

Marden JR, Arslan G, Shamma JS (2009) Joint strategy fictitious play with inertia for potential games. *IEEE Trans. Autom. Control* 54(2):208–220.

Marden JR, Shamma JS (2015) Game theory and distributed control. *Handbook of Game Theory with Economic Applications*, volume 4, 861–899 (Elsevier).

Marden JR, Young HP, Pao LY (2014) Achieving Pareto optimality through distributed learning. *SIAM Journal on Control and Optimization* 52(5):2753–2770.

Masuda Y, Whang S (1999) Dynamic pricing for network service: Equilibrium and stability. *Manag Sci* 45(6):857–869.

Monderer D, Shapley LS (1996a) Fictitious play property for games with identical interests. *J Econ Theory* 68(1):258–265.

Monderer D, Shapley LS (1996b) Potential games. *Games and Economic Behavior* 14(1):124–143.

Pradelski B, Young HP (2010) Efficiency and equilibrium in trial and error learning. *University of Oxford, Working Paper* .

Shamma JS, Arslan G (2004) Unified convergence proofs of continuous-time fictitious play. *IEEE Trans. Autom. Control* 49(7):1137–1141.

Swenson B, Eksin C, Kar S, Ribeiro A (2018) Distributed inertial best-response dynamics. *IEEE Trans. Autom. Control* 63(12):4294–4300.

Swenson B, Kar S, Xavier J (2015) A computationally efficient implementation of fictitious play in a distributed setting. *Euro Signal Process Conf*, 1043–1047 (IEEE).

Swenson B, Kar S, Xavier J (2017) Single sample fictitious play. *IEEE Trans. Autom. Control* 62(11):6026–6031.

Young HP (2004) *Strategic Learning and its Limits* (Oxford University Press, Oxford).

Young HP (2009) Learning by trial and error. *Games and Economic Behavior* 65(2):626–643.