Methods

Tatiana Tatarenko* and Jan Zimmermann

Distributed optimization methods for *N*-cluster games

Verteilte Optimierungsmethoden für N-Cluster Spiele

https://doi.org/10.1515/auto-2021-0137
Received September 20, 2021; accepted February 4, 2022

Abstract: This work provides methodological approaches to solve convex optimization problems arising in multi-agent systems which can be reformulated in terms of a so called *N*-cluster game. We consider different settings of information available to each agent in the system. First, we present a centralized algorithm, which requires a central coordinator having full access to information about agents' actions and gradients of their cost functions, to demonstrate how the standard gradient descent method can be applied to achieve an optimal output in *N*-cluster games. After that we relax the full information setting and assume that only partial information is available to each agent. Focus lies on the following two cases. In the first case, the agents have access to their gradient functions and are allowed to exchange information with their local neighbors over a communication graph that connects the whole system. In the second case, the agents do not know the functional form of their objectives/gradients and can only access the current values of their objective functions at some query point. Moreover, the agents are allowed to communicate only with their local neighbors within the cluster to which they belong. For both settings we present the convergent optimization procedures and analyse their efficiency in simulations.

Keywords: multi-agent systems, distributed optimization, game theory, discrete-time methods

Zusammenfassung: Diese Arbeit stellt methodische Herangehensweisen zur Lösung von konvexen Optimierungsproblemen in Multi-Agenten-Systemen, formuliert als sogenannte Multi-Cluster Spiele, vor. In diesem Zusammenhang beschäftigen wir uns mit unterschiedlichen Aufteilungen von Informationen auf die Agenten. Zunächst stellen wir einen zentralen Algorithmus vor, der einen zentralen Koordinator mit uneingeschränktem Zugang zu den Aktionen der Agenten und den Gradienten ihrer Kostenfunktionen benötigt. Mit diesem Algorithmus soll demonstriert werden, wie die bekannte Methode des Gradientenabstiegs angewendet werden kann, um ein optimales Ergebnis bezüglich des N-Cluster Spiels zu erzeugen. Anschließend relaxieren wir die Annahme von uneingeschränkter Information und nehmen an, dass jedem Agenten nur ein Teil der Gesamtinformationen zur Verfügung steht. Hierbei liegt der Fokus auf den folgenden zwei Fällen. Im ersten Fall haben die Agenten Zugang zu den Gradienten ihrer eigenen Funktionen und Informationen können über einen das gesamte System vernetzenden Kommunikationsgraphen mit den direkten Nachbarn ausgetauscht werden. Im zweiten Fall kennen die Agenten die funktionale Form ihrer eigenen Zielfunktionen/Gradienten nicht und können den aktuellen Wert ihrer Zielfunktion nur an bestimmten Punkten abfragen. Zusätzlich ist es den Agenten nur erlaubt, Informationen mit den Agenten des eigenen Clusters auszutauschen. Für beide Fälle stellen wir konvergierende Optimierungsprozesse vor und analysieren deren Effizienz in Simulationen.

Schlagwörter: Multi-Agenten-Systeme, verteilte Optimierung, Spieltheorie, zeitdiskrete Methoden

Dedicated to the 60th birthday of Prof. Dr.-Ing. Jürgen Adamy.

1 Introduction

In our technical world with its increasing complexity there are numerous systems consisting of many individuals, which can be considered as independently operating subsystems. Such so called multi-agent systems are, for example, robot teams or swarms, communication/computation facilities in wireless networks, or energy sources and energy consumers in energy networks, such as electrical

***Corresponding author: Tatiana Tatarenko,** Control Methods and Robotics Lab, Technical University of Darmstadt, Darmstadt, Germany, e-mail: tatiana.tatarenko@rmr.tu-darmstadt.de
Jan Zimmermann, Control Methods and Robotics Lab, Technical University of Darmstadt, Darmstadt, Germany, e-mail: jan.zimmermann@rmr.tu-darmstadt.de

or gas networks. Formally, any multi-agent system consists of several agents with definite objectives that are to be reached by appropriately chosen local actions/control laws of agents. These objectives can be expressed either by a *common goal* to be achieved by the system or by *individual profit maximization*. Methods from optimization theory need to be applied in the former case of the so called cooperative tasks, whereas selfish and competitive behaviour of agents in the latter case is to be analyzed by means of non-cooperative game theory.

However, cooperative and competitive tasks coexist in many practical situations, such as cloud computing, hierarchical optimization in Smart Grid, and adversarial networks [7, 9, 12]. To deal with such a combination in terms of a single model, so called *N-cluster games* have recently been considered in the literature [11, 8, 16, 21, 22, 23, 25, 26]. In such N-cluster games, each cluster corresponds to a player with the goal to minimize its own cost function. However, each cluster in the N-cluster game is represented not by an actual decision-maker but by a group of agents belonging to this cluster. Each of these agents has its own local cost function, which depends on the actions of agents in the whole system, i. e., of all clusters. The cluster's objective, in turn, is the sum of the local cost functions of the agents within the cluster. Therefore, in such models, each agent intends to find a strategy to solve the resulting N-cluster game, i. e., the competitive tasks, and, thus, to minimize the cooperative cluster's cost function equal to the sum of the individual ones, i. e., the cooperative task.

N-cluster games arising in multi-agent systems can be solved centrally. However, for a centralized solution, a central controller or central computing unit is required to collect the whole information about the system in order to solve the optimization problem under consideration. This approach has limitations. Firstly, systems with such settings are sensitive to the failure of the central unit. Secondly, the information exchange is costly, since agents need to transmit their local information to the central unit and to receive the instructions from it. Moreover, due to a large system's dimension, the optimization problem can become computationally intractable for the central controller. Finally, there may be no excess resources to incorporate a central computing unit into the system. Thus, optimization methods, in which agents can only use their *locally available information* about problem formulation as well as system's states, need to be developed and applied to N-cluster games. We further refer to such methods as *distributed optimization methods*. Continuous time distributed optimization algorithms to find a Nash equilibrium in multi-cluster games were proposed in [16, 21, 22, 23]. The paper [21] solves an unconstrained multi-cluster game by using gradient-based algorithms, the recent paper [16] deals with communication-based approach to games in which players have coupled constraints, whereas the works [22] and [23] propose gradient-free algorithms, based on zero-order information, for computing Nash and generalized Nash equilibria respectively. Since one faces the problem of discretization for practical implementation of the methods formulated in continuous time, especially in the case when communication takes place, in this work, we focus on discrete time algorithms, where each optimization variable is updated according to an iterative procedure.[1] In this domain, the work [11] presents a leader-follower communication-based algorithm, which can solve unconstrained multi-cluster games in linear time. The paper [25] augments this setting by introducing a special constraint type to the problem. The authors in [26] extend these results to the case of leaderless communication architecture. All papers [11, 25, 26] prove linear convergence in games with strongly monotone mappings and first-order information, meaning that agents can calculate gradients of their cost functions and use this information to update their states. The main idea in those works is the combination of a consensus dynamic, which is possible to set up due to existing communication, with a gradient-tracking technique, enabling a constant step size for the corresponding optimization procedure [15]. However, in several applications, each player might not know the functional form of its objective. For example, travel times of edges in a traffic network, market outcomes in an auction, or energy tariff functions (see [20]) are unknown a priori and depend in non-trivial ways on actions and objectives of other players. In game-theoretic problems, a player can have access to the so-called payoff information which does not contain reliable gradient values. That is why, in order to be able to make a gradient step, players need to estimate their gradients based on experienced values of their local cost functions. To do so, they can use existing techniques for smooth approximation of the gradient functions [3]. These techniques have been used in online and distributed optimization [1, 6, 17] as well as in game-theoretic problems [4, 10, 18] with merely payoff information in systems. In the context of N-cluster games, the work [8] deals with a gradient-free approach and, thus, uses payoff information to set up the corresponding optimization procedure. The gradient estimations are constructed based on two points: the currently played action and its

1 An iterative procedure describes a rule to obtain x_{k+1} from x_k at each iteration (time step) k.

shifted counterpart. Thus, such estimations require additional coordination between the agents, which is the first shortcoming of the proposed approach. Moreover, the authors in [8] use a constant step size in the optimization iterations to accelerate convergence of the proposed method. However, under this choice, convergence can only be guaranteed to a neighborhood of the equilibrium, as any constant step size implies a non-diminishing disturbance in the procedure, caused by inaccuracy of the gradient estimations. In this work, we overcome these limitations of the paper [8] and present a payoff-based algorithm which uses only actual values of cost functions to estimate local gradients and is provably convergent to a Nash equilibrium in the game under consideration.

Moreover, this work aims to provide methodological approaches to solve optimization problems in N-cluster games by taking into account information available in the system. Focus lies on convex optimization and *discrete time algorithms*. We start with a centralized algorithm to demonstrate how the standard gradient descent can be applied to achieve an optimal output in N-cluster games. After that, we consider the following two information settings: communication- and payoff-based information. In the former case, the agents have access to their gradient functions (so called *first-order oracle*) and are allowed to exchange information with their local neighbors over a given communication graph in the whole system. As it has been mentioned above, in many practical situations the agents do not know the functional form of their objectives/gradients and can only access the current values of their objective functions at some query point. In such cases, the information structure is referred to as *zero-order oracle*. Thus, the payoff information setting assumes the agents to be able to only observe the values of their local costs at a query point and communicate with their direct neighbors within the corresponding cluster to which they belong. Under some technical assumptions for both information settings, we present an adapted version of the centralized gradient-based procedure which is guaranteed to converge to a stable solution (Nash equilibrium) in the N-cluster game under consideration. We comment on the procedures' rates and provide a numerical example to emphasize influence of the information settings on efficiency (convergence speeds) of the optimization algorithms.

In the next section we provide a formal definition of the N-cluster game and formulate the problem which the agents intend to solve in this game. Section 3 deals with the classical gradient-based optimization procedure which can be used to solve the N-cluster games in the case of the full information setting. Section 4 relaxes the full information setting and presents methods applicable to the case of the partial information exchanged between the agents by means of communication. Section 5 considers even more general information set up, in which agents have no knowledge about the closed form of their costs and corresponding gradients and just observe the payoffs given the current joint action. In Section 6, we provide some numerical experiments for the presented optimization methods. Section 7 concludes the paper.

Notations. The set $\{1,\dots,n\}$ is denoted by $[n]$. For any function $f : K \to \mathbb{R}$, $K \subseteq \mathbb{R}^n$, $\nabla_i f(x) = \frac{\partial f(x)}{\partial x_i}$ is the partial derivative taken in respect to the i-th coordinate of the vector variable $x \in \mathbb{R}^n$. We consider a real normed space E, which is the space of real vectors, i. e., $E = \mathbb{R}^n$. We use (u,v) to denote the inner product in E and $\|\cdot\|$ to denote the Euclidean norm induced by this inner product. We use $\mathbb{B}_r(p)$ to denote a ball with radius $r \geq 0$ and center $p \in E$ and $\mathbb{S}$ to denote the unit sphere with its center in $0 \in E$. We use $\mathrm{Proj}_\Omega[v]$ to denote the projection of $v \in E$ to a set $\Omega \subseteq E$. The mathematical expectation of a random value ξ is denoted by $\mathrm{E}\{\xi\}$.

2 Nash equilibrium in *N*-cluster games

We consider a non-cooperative game between N clusters. Each cluster $i \in [N]$ itself consists of n_i agents. Let J_i^j and $\Omega_i^j \subseteq \mathbb{R}$ denote[2] respectively cost function and feasible action set of agent j in cluster i. We denote the joint action set of agents in cluster i by $\Omega_i = \Omega_i^1 \times \ldots \times \Omega_i^{n_i}$. Each function $J_i^j(x_i, x_{-i})$, $i \in [N]$, depends on $x_i = (x_i^1,\dots,x_i^{n_i}) \in \Omega_i$, which represents the joint action of agents within cluster i, and $x_{-i} \in \Omega_{-i} = \Omega_1 \times \ldots \times \Omega_{i-1} \times \Omega_{i+1} \times \Omega_N$, denoting the joint action of agents from all clusters except cluster i. The cooperative cost function in cluster $i \in [N]$ is, thus,

$$J_i(x_i, x_{-i}) = \frac{1}{n_i} \sum_{j=1}^{n_i} J_i^j(x_i, x_{-i}). \tag{1}$$

We denote the joint action set of all agents in the game by $\Omega = \Omega_1 \times \ldots \times \Omega_N$.

Let us denote the game between the clusters introduced above by $\Gamma(N, \{J_i\}, \{\Omega_i\})$. We make the following assumptions regarding the game Γ:

2 All results below are applicable for games with different dimensions $\{d_i^j\}$ of the action sets $\{\Omega_i^j\}$. The one-dimensional case is considered for the sake of notation simplicity.

Assumption 1. *The N-cluster game under consideration is* strongly convex. *Namely, for all $i \in [n]$, the set Ω_i is convex, the cost function $J_i(x_i, x_{-i})$ is continuously differentiable in x_i for each fixed x_{-i}. Moreover, the game mapping, which is defined as*

$$\boldsymbol{F}(x) \triangleq [\nabla_1 J_1(x_1, x_{-1}), \ldots, \nabla_N J_N(x_N, x_{-N})]^T \tag{2}$$

is strongly monotone *on Ω, namely:*

$$(\boldsymbol{F}(x) - \boldsymbol{F}(y), x - y) \geq \mu \|x - y\|^2 \tag{3}$$

for some $\mu > 0$ and any $x, y \in \Omega$.

Remark 1. *Note that strong monotonicity of the game mapping $\boldsymbol{F}(x)$ implies strong convexity of cluster i's cost function $J_i(x_i, x_{-i})$ in x_i for fixed x_{-i} for all $i \in [N]$. Indeed, let us consider two points $x = (x_i, x_{-i}) \in \Omega$ and $y = (y_i, x_{-i}) \in \Omega$. Then* (3) *implies*

$$(\nabla_i J_i(x_i, x_{-i}) - \nabla_i J_i(y_i, x_{-i}), x_i - y_i) \geq \mu \|x_i - y_i\|^2.$$

Assumption 2. *The mapping $\boldsymbol{F}$ is Lipschitz continuous on Ω, namely:*

$$\|\boldsymbol{F}(x) - \boldsymbol{F}(y)\| \leq L \|x - y\| \tag{4}$$

for some $L > 0$ and any $x, y \in \Omega$.

Let us notice that, given Assumptions 1 and 2, $\frac{\mu}{L} \leq 1$. Note also that the assumptions above are standard in the literature on both game-theoretic and zero-order optimization [4].

Example of an N-cluster game. As an example of an N-cluster game, consider a Smart Grid consisting of N microgrids that are connected to a main grid. Each microgrid $i \in [N]$ contains $n_{g,i}$ dispatchable, decentralized power generation units and $n_{c,i}$ consumers, where the latter have a certain demand for energy. Thus, $[n_i] = [n_{g,i}] \cup [n_{c,i}]$. The goal of each microgrid is to satisfy the demand of its consumers and minimize the cost of doing so. To reach that goal, given the total demand $d_i = \sum_{j=1}^{n_{c,i}} d_i^j$ of the consumers $j \in [n_{c,i}]$ of the microgrid i, this microgrid can buy power of the amount $p_i = \sum_{j=1}^{n_{c,i}} p_i^j$ from the main grid or produce power with its local power generation units. Here, each p_i^j is the amount of energy assigned to the user $j \in [n_{c,i}]$. We assume that microgrids are operated by different companies, which are in competition with each other. Furthermore, the price each microgrid has to pay for a unit of power from the main grid depends on the demand of the whole system. In this example, we define the cost function for buying power from the main grid for each consumer $j \in [n_{c,i}]$ as follows:

$$J_i^j(p_i, p_{-i}) = q\left(\sum_{i=1}^{N} p_i\right) p_i^j, \tag{5}$$

where q is some positive factor set by the main grid operator. Note that these cost functions are coupled, as the actions of other microgrids have an influence on the price of the main grid. Next to the main grid costs, each microgrid has an expense for operating the decentralized generation units, which are dependent on the amount of power g_i^j produced by each $j \in [n_{g,i}]$. Summation over all local generators leads to $g_i = \sum_{j=1}^{n_{g,i}} g_i^j$. We formalize this cost as the following quadratic equation for each generator $j \in [n_{g,i}]$:

$$J_i^j(g_i^j) = a_i^j (g_i^j)^2 + b_i^j g_i^j + c_i^j \tag{6}$$

with some constant, positive cost coefficients a_i^j, b_i^j and c_i^j. Note that the quadratic approximation of fuel-based power plants operation costs is common for economic dispatch or power management problems, see, for example, [2, 24]. Of course, each generator is subject to lower and upper bounds on its production, i. e., $\underline{g}_i^j \leq g_i^j \leq \overline{g}_i^j$.

Combining the two functions, the cost of each microgrid i, i. e., the cluster in the game, has to pay for satisfying the demand takes the form

$$J_i(x_i, x_{-i}) = \frac{1}{n_{c,i} + n_{g,i}} \left[\sum_{j \in [n_{c,i}]} J_i^j(p_i, p_{-i}) + \sum_{j \in [n_{g,i}]} J_i^j(g_i^j) \right] \tag{7}$$

with $x_i = [p_i, g_i]$ and $x_{-i} = [p_{-i}, g_{-i}]$. Therefore, each microgrid $i \in [N]$ aims to solve the optimization problem

$$\min_{x_i} J_i(x_i, x_{-i}) \tag{8a}$$

$$\text{s.t. } p_i + g_i = d_i \tag{8b}$$

$$\underline{g}_i^j \leq g_i^j \leq \overline{g}_i^j, \forall j. \tag{8c}$$

Here, the equality constraint in (8b) enforces the balance between bought/produced power and the demand of the consumers inside the microgrid, where the latter is assumed to be fixed. As p_i and g_i describe the sum of the local strategies regarding buying power from the main grid and power production in the generators, constraint (8b) represents a coupling point of the cluster i's decision variables. The first sum in the cost function (7) can be regarded as the coupling element of the microgrids as this term is influenced by actions of other microgrids, creating a non-cooperative game situation, in which the microgrids act as

players. The generators can be regarded as agents that belong to the microgrid i, each possessing its own local cost function. As all microgrids aim to minimize the sum of the costs of all generators in the respective microgrid, a cooperative optimization problem arises within each microgrid.

One of the stable solutions in any game Γ corresponds to a Nash equilibrium defined below.

Definition 1. *A vector* $x^* = [x_1^*, x_2^*, \cdots, x_N^*]^T \in \Omega$ *is called a* Nash equilibrium *if for any* $i \in [N]$ *and* $x_i \in \Omega_i$

$$J_i(x_i^*, x_{-i}^*) \leq J_i(x_i, x_{-i}^*).$$

Note that Assumption 1 guarantees equivalence between the set of Nash equilibria of Γ and the solution set of the following variational inequality [13]:

$$\text{Find } x^* \colon (\boldsymbol{F}(x^*), x - x^*) \geq 0 \text{ for all } x \in \Omega. \tag{9}$$

Moreover, based on this equivalence it can be demonstrated that strong monotonicity of $\boldsymbol{F}$ in Assumption 1 implies existence and uniqueness of a Nash equilibrium in the game Γ [14].

In this work, we are interested in *computing a Nash equilibrium* in any N-cluster game $\Gamma(N, \{J_i\}, \{\Omega_i\})$, for which Assumptions 1–2 hold, by taking into account available information for each agent $j \in [n_i]$ from the cluster $i \in [N]$.

3 Centralized approach

We start with the full information setting to provide some intuition on how the standard gradient descent can solve N-cluster games under consideration. In the subsequent sections this procedure will be adapted to some more restrictive information settings (see Section 4 and 5).

Let us assume there is a central unit who has access to the full information in the system, namely to the action sets, the cost functions and their gradients. Equivalently, we can assume that in this case each agent knows the partial derivatives of the users' costs within the cluster as well as the joint action of all users in the system. Thus, at each moment in time t the following iterations can be applied for computing a Nash equilibrium in $\Gamma(N, \{J_i\}, \{\Omega_i\})$. Let $x(0) \in \Omega$ be an arbitrary initial joint action. Given $x(t)$ obtained at time t, the central unit runs the following updates for each $i \in [N]$ at time $t+1$:

$$x_i(t+1) = \text{Proj}_{\Omega_i}[x_i(t) - \gamma \nabla_i J_i(x(t))], \tag{10}$$

where $\gamma > 0$ is a step size parameter. Note that the procedure (10) is represented by a standard gradient descent step for each cluster i with respect to its local cost J_i. If we assume full information, as described above, is available for each agent in the system, then the updates of local action x_i^j are as follows:

$$x_i^j(t+1) = \text{Proj}_{\Omega_i^j}[x_i^j(t) - \gamma \nabla_{i,j} J_i(x(t))], \tag{11}$$

where $\nabla_{i,j} J_i(x(t)) = \frac{\partial J_i(x(t))}{\partial x_i^j}$ is the jth coordinate of the vector $\nabla_i J_i(x(t))$. The following theorem provides sufficient conditions for the procedure (10) to converge to the solution of the game.

Theorem 1. *Let Assumptions 1 and 2 hold in the game* $\Gamma(N, \{J_i\}, \{\Omega_i\})$. *Let the step size* γ *in the algorithm* (10) *be chosen as* $\gamma = \frac{\mu}{L^2}$. *Then the algorithm* (10) *converges geometrically fast to the unique Nash equilibrium in* Γ *with the following rate:*

$$\|x(t) - x^*\|^2 \leq \exp\left\{-\frac{t}{L^2/\mu^2}\right\} \|x(0) - x^*\|^2.$$

Proof. By summarizing the update rules of the algorithm, we conclude that the joint action $x(t)$ evolves as follows:

$$x(t+1) = \text{Proj}_{\Omega}[x(t) - \gamma \boldsymbol{F}(x(t))].$$

Therefore, as x^* solves the variational inequality (9), the following well-established result, see, for example, [13], can be applied:

$$x^* = \text{Proj}_{\Omega}[x^* - \gamma \boldsymbol{F}(x^*)].$$

Thus, we can use the non-expansiveness of the projection operator to obtain

$$\begin{aligned}
&\|x(t+1) - x^*\|^2 \leq \|x(t) - \gamma(\boldsymbol{F}(x(t)) - \boldsymbol{F}(x^*)) - x^*\|^2 \\
&= \|x(t) - x^*\|^2 - 2\gamma(\boldsymbol{F}(x(t)) - \boldsymbol{F}(x^*), x(t) - x^*) \\
&\qquad + \gamma^2 \|\boldsymbol{F}(x(t)) - \boldsymbol{F}(x^*)\|^2 \\
&\leq (1 - 2\mu\gamma + \gamma^2 L^2) \|x(t) - x^*\|^2,
\end{aligned}$$

where in the last inequality we used (3) and (4). By choosing $\gamma = \frac{\mu}{L^2}$ we conclude that

$$\|x(t+1) - x^*\|^2 \leq \left(1 - \frac{\mu^2}{L^2}\right)^t \|x(0) - x^*\|^2.$$

Finally, noticing that $\left(1 - \frac{\mu^2}{L^2}\right)^t \sim \exp\left\{-\frac{t}{L^2/\mu^2}\right\}$ for large t, we conclude the proof. □

As it has been mentioned above, the full information setting in the system is a decisively restrictive assumption. That is why, in the following sections, we relax it by assuming that just partial information is available for each agent in the system.

4 Communication-based distributed algorithm

In contrast to the full information setting of the previous section, we now assume that the information describing the optimization problem is distributed among the agents in the network and that there exists no agent or central unit that has access to all information of the problem. This means in particular that the structure of the cost function J_i^j of agent j in cluster i is only known by this agent and not by any other agent in the network. Furthermore, we assume that each cluster's constraint set Ω_i is known by every agent j in cluster i but not by any other agent that is not part of this cluster. In such a setup, the clusters represent only virtual players, which means that the clusters only define the group affiliation of the agents but do not perform any actions themselves. However, the agents inside a specific cluster aim to minimize the cooperative cost function defined in (1), i. e., aim to achieve the social welfare optimum inside the clusters, while reacting to the actions of agents from other clusters. Therefore, each agent needs an update direction of its action in order to achieve the social welfare optimum of the group. In contrast to the approach in the preceding section, we present in this section an approach from [26] that is based on the direct calculation of local gradients. Furthermore, we intuitively extend the results of [26], which are limited to unconstrained N-cluster games, to the constrained case. Note that we do not provide an explicit convergence proof for the constrained case in this paper but rather argument with the simulation results.

The approach of [26] is based on a gradient-tracking technique. The main idea is to "mimic" the centralized gradient descent step in (10) by constructing the estimation of the sum of the local cost functions' gradients within each cluster as this information is not available under the current setting. This sum is equal to $\sum_{j=1}^{n_i} \nabla_i J_i^j(x_i, x_{-i})$ and corresponds to the gradient of the cluster cost function $J_i(x_i, x_{-i})$. In order to perform such an estimation, the agents need to be able to exchange information with each other over a specific communication architecture. This communication architecture is represented by a communication graph, in which the agents correspond to the nodes of the graph while the communication channels between the agents are the edges. The gradient-tracking approach of [26] assumes two kinds of directed communication graphs: Firstly, a global graph $\mathcal{G}([N], \mathcal{A})$, which contains all agents as nodes and connects these agents through the arcs from the set $\mathcal{A}$ regardless of their cluster membership. And secondly, N local graphs $\mathcal{G}_i([n_i], \mathcal{A}_i)$, one for each cluster i, which contain the agents of the respective cluster i as nodes and connect only these agents through the arcs from the set $\mathcal{A}_i$, such that no inter-cluster communication takes place. In accordance with [26], we make the following assumptions regarding these graphs:

Assumption 3. *The underlying directed communication graph $\mathcal{G}([N], \mathcal{A})$ is strongly connected. The associated non-negative mixing matrix $R = [r_{kj}] \in \mathbb{R}^{n \times n}$ defines the weights for the directed arcs and is row-stochastic, such that $r_{kj} > 0$ if and only if $(k, j) \in \mathcal{A}$ and $\sum_{k=1}^{n_i} r_{kj} = 1$, $\forall j \in [n]$.*

The underlying directed communication graphs $\mathcal{G}_i([n_i], \mathcal{A}_i)$ are strongly connected for all $i = 1, ..., N$. The associated non-negative mixing matrices $C_i = [c_{kj}^i] \in \mathbb{R}^{n_i \times n_i}$ define the weights of the directed arcs and are column-stochastic, such that $c_{kj}^i > 0$ if and only if $(k, j) \in \mathcal{A}_i$ and $\sum_{j=1}^{n_i} c_{kj}^i = 1$, $\forall k \in [n_i]$, $\forall i$.

Note that the mixing matrix is a weighted adjacency matrix according to the definition in above assumption. By $x^{(j)} = (x_i^{(j)}, x_{-i}^{(j)})$, we describe the estimation made by agent j in cluster i of the global decision vector $x = (x_i, x_{-i})$. These estimations are necessary in order to evaluate the local gradients. Furthermore, we introduce the so called gradient-tracking estimation variable $y_i^{(j)}$, which contains an estimation of the gradient $\sum_{j=1}^{n_i} \nabla_i J_i^j(x_i, x_{-i})$ in cluster i made by agent j. With these definitions, we are able to formalize a communication- and gradient-based solution algorithm as follows:

$$\hat{x}^{(j)}(t) = \sum_{k=1}^{N} r_{kj} x^{(k)}(t), \tag{12a}$$

$$x_i^{(j)}(t+1) = \mathrm{Proj}_{\Omega_i}\left[\hat{x}_i^{(j)}(t) - \alpha y_i^{(j)}(t)\right], \tag{12b}$$

$$x_{-i}^{(j)}(t+1) = \hat{x}_{-i}^{(j)}(t), \tag{12c}$$

$$\begin{aligned} y_i^{(j)}(t+1) = &\sum_{k=1}^{n_i} c_{kj}^i y_i^{(k)}(t) \\ &+ \nabla_i J_i^j\left(x_i^{(j)}(t+1), x_{-i}^{(j)}(t+1)\right) \\ &- \nabla_i J_i^j\left(x_i^{(j)}(t), x_{-i}^{(j)}(t)\right), \end{aligned} \tag{12d}$$

where $\alpha > 0$ is a constant step size. Equation (12a) describes the communication step over the row-stochastic matrix R. The resulting vector $\hat{x}^{(j)}(t)$ is the weighted sum of the estimations of agent j's neighbors and its own estimation. This vector is then split into the components $\hat{x}_i^{(j)}$ and $\hat{x}_{-i}^{(j)}$, which respectively describe the estimations regarding j's own cluster i and the estimation regarding other cluster variables. These values are then differently processed to achieve new estimations: $\hat{x}_{-i}^{(j)}$ is directly taken

as the new estimation value, i. e., $x_{-i}^{(j)}(t+1)$. For the cluster action estimations, first a step with size α is performed in the direction of the current gradient estimation $y_i^{(j)}(t)$ and then the result is projected onto constraint set Ω_i of cluster i. In the last step of the algorithm, namely Equation (12d), the agents exchange information about their gradient estimations with their neighbors and calculate again a weighted sum of the provided estimations. Then, each agent j calculates the gradient of their local cost functions, using the estimation of time $t+1$ and t. The difference between these gradients is then added to the weighted sum and, thus, the gradient-tracking variable is updated. If the gradient-tracking variable is initialized such that $y_i^{(j)}(0) = \nabla_i J_i^j(x^{(j)}(0))$, it can be shown that

$$\sum_{j=1}^{n_i} y_i^{(j)} = \sum_{j=1}^{n_i} \nabla_i J_i^j \left(x_i^{(j)}(t), x_{-i}^{(j)}(t)\right), \forall i. \tag{13}$$

Therefore, if all $y_i^{(j)}$ for $j = 1, ..., n_i$ converge to a consensus, the consensus variable estimates the sum of the gradients of the local cost functions, provided that the action estimations converge as well. The initialization of the action estimations can be random.

As mentioned in the beginning of this section, convergence is not yet proven for a constrained N-cluster game under Algorithm (12). However, in the simulation section, we provide simulation results of applying Algorithm (12) to the constrained microgrid management problem (8). Nevertheless, for unconstrained multi-cluster games, i. e., $\Omega_i = \mathbb{R}$ for all i, convergence was rigorously proven in [26]. The following results summarize Theorem 1 of [26].

Theorem 2. *Let $\Omega_i = \mathbb{R}$ for all i and Assumptions 1, 2, 3 hold. Then there exists a constant step size $\alpha > 0$ such that the estimations of all agents reach a consensus x^c and this consensus is a unique Nash equilibrium of the N-cluster game, i. e.,*

$$\lim_{t\to\infty} x^{(j)}(t) = \lim_{t\to\infty} x^{(k)}(t) = x^c = x^*, \forall j, k, \tag{14}$$

provided that the estimations are updated according to (12) *and properly initialized. Moreover, the convergence rate is geometrically fast.*

Remark 2. *The introduction of the gradient tracking term $y_i^{(j)}(t)$, which uses the difference between the current gradient and the one from the previous step (see* (12d)*) allows for the choice of a constant step size $\alpha > 0$, which guarantees geometrically fast convergence of the algorithm* (12). *More details on the idea of using the gradient tracking technique can be found in [15].*

Remark 3. *Note that, analogously, to the centralized method* (10)*, the convergence rate of the communication-based algorithm* (12) *is linear. However, it can be noticed from Proposition 1 in [26] that the convergence rate of the latter procedure depends in a sophisticated way on the parameters μ and L of the game mapping as well as on the singular number σ_R and σ_{C_i} of the mixing matrices (see Assumption 3). It is naturally to expect that this rate is slower than one of the centralized algorithm. The simulation results in Section 6 demonstrate this fact.*

5 Payoff-based distributed algorithm

In this section, we further restrict the assumption on the information available to the agents in N-cluster games. Now we deal with systems, where there is *no explicit communication between the clusters*. However, as before, the agents within each cluster can interact over an undirected communication graph $\mathcal{G}_i([n_i], \mathcal{A}_i)$, for which the following assumption holds.

Assumption 4. *The underlying undirected communication graph $\mathcal{G}_i([n_i], \mathcal{A}_i)$ is connected for all $i = 1, \dots, N$. The associated non-negative mixing matrix $W_i = [w_{kj}^i] \in \mathbb{R}^{n\times n}$ defines the weights on the undirected arcs such that $w_{kj}^i > 0$ if and only if $(k, j) \in \mathcal{A}_i$ and $\sum_{k=1}^{n_i} w_{kj}^i = 1$, $\forall k \in [n_i]$.*

We consider the following *zero-order* information: No agent has access to the analytical form of any cost function, including its own. Each agent can only observe the value of its local cost function given any joint action of all agents in the system. Formally, given a joint action $x \in \Omega$, each agent $j \in [n_i]$, $i \in [N]$ receives the value $J_i^j(x)$ from a zero-order oracle. The zero-order oracle here is a part of the environment (information setting) in which the game is formulated. For example, in energy management applications, one can consider a central power station, which sets energy tariffs based on demand (actions of users in the grid), such a zero-order oracle. The grid's users do not know the functional form of these tariffs but obtain their values at each time slot from the central power station. Note that the oracle does not intend or is able to coordinate agents in achieving their goals. General zero-order oracles are considered in the related works [4, 17, 8]. No agent has or receives any information about the gradients. That is why to be able to adapt the gradient-based algorithm to this setting, each agent needs to estimate the local gradient by using purely the cost function values (zero-order in-

formation). To do so each agent $j \in [n_i]$, $i \in [N]$, based on its local estimation $x_i^{(j)}$ of the joint action in the cluster i, constructs a feasible query point $x_i'^{(j)} \in \Omega_i$ and sends it to the oracle. As a reply from the oracle, the agent receives the value $J_i^j(x_i'^{(j)}, \tilde{x}'_{-i})$. The vector $\tilde{x}'_{-i}$ here corresponds to the point obtained by some combination of the query vectors sent by the agents from the other clusters. Formally,

$$\tilde{x}'_{-i} = (x_1'^{(j_1)}, \dots, x_{i-1}'^{(j_{i-1})}, x_{i+1}'^{(j_{i+1})}, \dots, x_N'^{(j_N)}), \tag{15}$$

where j_k denotes some agent from the cluster $k \in [N]$, $k \neq i$. Further, each agent $j \in [n_i]$, $i \in [N]$, uses the received value $J_i^j(x_i'^{(j)}, \tilde{x}'_{-i})$ to obtain the random estimation d_i^j of its local cost's gradient $\nabla_i J_i^j$ at the point $(x_i^{(j)}, \tilde{x}_{-i})$, where

$$\tilde{x}_{-i} = (x_1^{(j_1)}, \dots, x_{i-1}^{(j_{i-1})}, x_{i+1}^{(j_{i+1})}, \dots, x_N^{(j_N)}) \tag{16}$$

corresponds to the local estimations of other agents (one for each cluster different from i) based on which query points are obtained. Thus, $d_i^j = d_i^j(J_i^j(x_i'^{(j)}, \tilde{x}'_{-i})) \in \mathbb{R}^{n_i}$. As d_i^j is an estimation of $\nabla_i J_i^j(x_i^{(j)}, \tilde{x}_{-i})$, we represent this vector by the following decomposition:

$$d_i^j = \nabla_i J_i^j(x_i^{(j)}, \tilde{x}_{-i}) + e_i^j, \tag{17}$$

where e_i^j is a random vector reflecting inaccuracy of the obtained estimation, i. e., the estimation error vector. Note that for the joint query point $(x_i'^{(j)}, \tilde{x}'_{-i})$ the oracle is free to choose any combination $\tilde{x}'_{-i}$ of the local queries defined in (15). As one can see, in (17), we use a single point to estimate each gradient. It differs from the two-point estimation technique used in [8], which requires an extra coordination between the agents. There are two main approaches to sample the gradients of the functions based on one-point estimations: a smoothing technique using the Gaussian distribution and a sampling based on the uniform distribution over the unit sphere [3]. To provide an explicit sampling process, let us focus on the latter option. The idea is borrowed from the work [4] dealing with bandit learning in games. To guarantee feasibility of the query point for the gradient sampling, we introduce the following assumption:

Assumption 5. *The action sets Ω_j^i, $j \in [n_i]$, $i \in [N]$, are compact. Moreover, for each cluster i there exists a so called safety ball $\mathbb{B}_r(p) \subseteq \Omega_i$ with $r_i > 0$ and $p_i \in \Omega_i$.*

We assume the safety ball parameters r_i and p_i defined above are known for each agent from the cluster i. To obtain the estimation $d_i^j(t)$ based on the current estimation $x_i^{(j)}(t)$, each agent $j \in [n_i]$ in the cluster i, $i \in [N]$, takes the following steps at time t. The agent samples the vector $z_i^j(t)$ from the uniform distribution on the unit sphere $\mathbb{S} \subset \mathbb{R}^{n_i}$. The query direction is defined by $w_i^{(j)}(t) = z_i^j(t) - r_i^{-1}(x_i^{(j)}(t) - p_i)$. Then, the query point at which the oracle calculates the local cost function value is

$$\begin{aligned}\hat{x}_i^{(j)}(t) &= x_i^{(j)}(t) + \sigma_t w_i^{(j)}(t)\\ &= (1 - \sigma_t r_i^{-1}) x_i^{(j)}(t) + \sigma_t (z_i^j + r_i^{-1} p),\end{aligned} \tag{18}$$

where σ_t is the query radius chosen such that $\sigma_t r_i^{-1} < 1$. Note that, given $x_i^{(j)}(t) \in \Omega_i$, the query point $\hat{x}_i^{(j)}(t)$ above is feasible, i. e., $\hat{x}_i^{(j)}(t) \in \Omega_i$ (see [4] for more details). The gradient estimation itself is obtained as follows:

$$d_i^j(t) = \frac{n_i}{\sigma_t} J_i^{(j)}(\hat{x}_i^{(j)}(t), \hat{\tilde{x}}_{-i}(t)) \cdot z_i^j, \tag{19}$$

where $\hat{\tilde{x}}_{-i}(t)$ is defined as in (15).

Now we are ready to formulate the distributed optimization algorithm for Nash equilibrium computation given the information setting under consideration. Starting with an arbitrary estimation $x_i^{(j)}(0) \in \Omega_i$, each agent j updates the local estimation vector $x_i^{(j)}$, $j \in [n_i]$, $i \in [N]$, as follows:

$$x_i^{(j)}(t+1) = \mathrm{Proj}_{\Omega_i}\left\{\sum_{l=1}^{n_i} w_{jl}^i x_i^{(l)}(t) - \alpha_t d_i^j(t)\right\}, \tag{20}$$

where w_{jl}^i corresponds to the element of the communication matrix W_i in the cluster i, $d_i^j(t)$ is defined in (19), and $\alpha_t > 0$ is the step size. Notice that in contrast to the procedures in the previous sections, a time-dependent step size needs to be introduced here. Moreover, this parameter has to fulfill the following conditions:

$$\begin{aligned}&\sum_{t=0}^{\infty} \alpha_t = \infty, \ \sum_{t=0}^{\infty} \alpha_t^2 < \infty,\\ &\sum_{t=0}^{\infty} \alpha_t \mathrm{E}\{\|e_i^j((t))\| | \mathcal{F}_t\} < \infty,\\ &\sum_{t=0}^{\infty} \alpha_t^2 \mathrm{E}\{\|e_i^j((t))\|^2 | \mathcal{F}_t\} < \infty \text{ almost surely,}\end{aligned} \tag{21}$$

where $e_i^j(t)$ corresponds to the current gradient estimation error defined in (17) and $\mathcal{F}_t$ is the σ-algebra generated by the estimations $\{x_i^{(j)}(m)\}_{m=0}^t$ up to time t, $j \in [n_i]$, $i \in [N]$. The condition $\sum_{t=0}^{\infty} \alpha_t = \infty$ guarantees that the gradient step possesses enough energy to reach the solution, whereas the condition $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$ keeps the "energy" of the procedure bounded, which is required for the convergence. The last two conditions in (21) allow for a diminishing contribution of the error term e_i^j in the gradient estimation (see (17)).

The statement below provides the main convergence result regarding the procedure (20) and repeats the results of [19].

Theorem 3. *Let Assumptions 1, 2, 4, and 5 hold and* $x_i^{(j)}(t)$, $j \in [n_i]$, $i \in [n]$, *be updated according to* (20), *where* $d_i^j(t)$ *is sampled as defined by* (19). *Moreover, let the step size* α_t *be chosen to satisfy the following conditions:* $\sum_{t=0}^{\infty} \alpha_t = \infty$, $\sum_{t=0}^{\infty} \alpha_t < \infty$, $\sum_{t=0}^{\infty} \alpha_t \sigma_t < \infty$, $\sum_{t=0}^{\infty} \frac{\alpha_t^2}{\sigma_t^2} < \infty$, *where* σ_t *is the sampling radius. Then conditions* (21) *hold and the joint action* $x(t) = (x_1(t), \dots, x_n(t))$ *converges almost surely to the unique Nash equilibrium* x^* *in the game* $\Gamma(n, \{J_i\}, \{\Omega_i\})$, *i. e.,* $\Pr\{\lim_{t\to\infty} \|x(t) - x^*\| = 0\} = 1$.

Remark 4. *Note that the set of the parameters* α_t *and* σ_t *satisfying the condition in the theorem above is non-empty. Indeed, an example of an appropriate choice is* $\alpha_t = \frac{\alpha_0}{t^a}$, $\sigma_t = \frac{\sigma_0}{t^b}$ *with* $\frac{1}{2} < a \leq 1$, $b \geq 0$, $a + b > 1$, $2a - 2b > 1$.[3] *For such choices of the parameters and given Assumption 1, one can apply the Chung's lemma (see [5]) to the inequality (15) in [19] to conclude that the procedure in* (20) *possesses a sublinear convergence rate.*[4] *Thus, the optimization method for the case of merely payoff information in the system possesses the slowest convergence speed to the Nash equilibrium.*

6 Simulation results and discussion

In this section, we present simulation results of the three solution approaches, i. e., the centralized approach of Section 3, communication-based approach of Section 4, and the payoff-based approach of Section 5. All three algorithms are evaluated on the same problem setup, which is a version of the N-microgrid problem, described by the example provided in Section 2. The goal of each microgrid or cluster is to solve their own optimization problem (8) by taking into account the actions of other clusters. A stable solution of the resulting game is the Nash equilibrium x^*, from which no microgrid has a reason to deviate if all other grids keep their decisions. As a specific setup, we implement three microgrids. The first microgrid contains four generators, while each of the other two microgrids contain three generators. Altogether, there are 10 generators and therefore 10 agents in the agent system. For every generator, a one-dimensional power output needs to be chosen. Additionally, the power bought from the main grid needs to be determined by every microgrid. Assuming a single consumer, the combined strategy vector has a dimension of 13. We choose the step sizes $\gamma = 0.01$, $\alpha = 0.025$, $\alpha_t = 0.04/(t^{0.04})$ and query radius $\sigma_t = 18/(t^{0.2})$ for the respective algorithms. The parameters are tuned by grid searches to provide optimized performances. For the communication architectures of the global and local graphs of Algorithm (12), we choose undirected, connected graphs in order to create comparability to the payoff-based algorithm, which is restricted to undirected communication links.

In Figure 1, the relative error norms $||x - x^*||_2/||x^*||_2$ between the Nash equilibrium x^* and the decisions at iteration t are plotted over the iterations. It can be observed that the centralized approach converges at the fastest rate and achieves a relative error of 0.1 already after $t = 35$ iterations. The communication-based algorithm exhibits slower convergence, achieving the same error after $t = 1096$ iterations. The payoff-based algorithm, however, shows by far the slowest convergence. Although always converging on average towards the Nash equilibrium, even after $t = 1 \cdot 10^5$ iterations the remaining relative error norm is approximately at 11.5.

This result is in accordance with the information structure of the different algorithms. The centralized algorithm of Section 3 assumes that all information about the problem is gathered at a single, central node, such that the gradients of the cost functions are available to the optimizing unit and that there is no need to estimate the actions of other clusters or agents. In the distributed setup of the communication-based algorithm of Section 4, however, communication graphs are established between and inside the clusters such that the estimation of actions is possible. This additional estimation process, necessary for achieving consensus among the estimations, together with the sparsity of information is responsible for slower

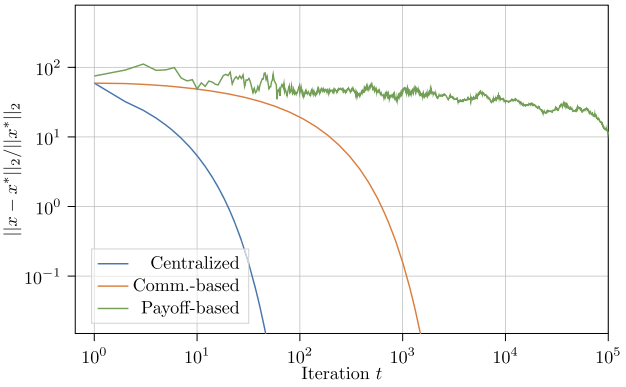

Figure 1: Evolution of the relative error norms for the three algorithms under consideration.

3 One possible parameter set is $a = 1$, $b = \frac{1}{3}$.

4 Here the expected distance to the solution is measured. Namely, one can demonstrate that $\mathrm{E}\|x(t) - x^*\| = O\left(\frac{1}{t^c}\right)$ for some $c > 0$.

convergence. Nevertheless, a constant step size can be employed to update the actions based on the cluster-gradient estimation. The payoff-based approach of Section 5 is based on a similar distributed information setup. In contrast to the communication-based approach, no communication is allowed between the clusters and the consensus dynamic is restricted to inner-cluster states. The updates are performed using an estimated gradient. These estimations in their turn are obtained based on the cost functions' values at specific query points. Thus, less information about the direction of the steepest descent is provided to the agents in comparison to the gradient-tracking method of the communication-based approach. This, together with the randomness of the procedure, prolongs the convergence time. Another disadvantage of this algorithm is the necessity for a declining step size sequence α_t in combination with an additional sequence σ_t for the query radius. It is a known fact that optimization methods, especially of the distributed nature, demonstrate slow convergence rates when the step size sequences are time-dependent. Furthermore, finding an optimal parameterization for these sequences is a non-trivial task, as it is problem-dependent and the optimal choice cannot be ensured. However, it should be emphasized that for the payoff-based method the clusters do not need to reveal their actions to their competitors, as it is sufficient to submit a query point to the oracle in order to receive information about the update step. In contrast to that, for the communication-based method, each agent estimates the actions of all clusters. As a consensus on the estimations is reached with time, each agent knows the actions of their competitors. This fact can be regarded as an intrusion in the privacy of other clusters as only the cost functions and constraints remain private in this setup.

In summary, the convergence speeds of the proposed methods reflect the assumed restriction of information in the system. Whether scarcity of information is preferred over convergence speed needs to be evaluated for each particular problem and the corresponding game design.

7 Conclusion

This paper presents optimization methods applicable to the combination of the competitive and cooperative tasks arising simultaneously in multi-agent systems. Three methods for different information settings are provided. Their efficiencies are formulated by theoretical statements as well as validated by simulations. In future work, we will further investigate possible information settings in the system as well as consider more realistic assumptions on the costs (non-strongly monotone game mappings) and constraints (coupling and time-dependent constraints for which projection operator is not straightforward to calculate). Moreover, we will aim to estimate lower bounds for optimization algorithms in games with different information structures and present corresponding methods with optimal rates for a given class of problems.

Funding: This work was funded by the German research association (Deutsche Forschungsgemeinschaft / DFG) – priority programme 1984.

References

1. Bach, F. R. and V. Perchet. 2016. Highly-smooth zero-th order online optimization. In: *Conference on Learning Theory (COLT)*, pp. 257–283.
2. Belgioioso, G., W. Ananduta, S. Grammatico and C. Ocampo-Martinez. 2020. Energy management and peer-to-peer trading in future smart grids: A distributed game-theoretic approach. In: *2020 European Control Conference (ECC)*, pp. 1324–1329.
3. Berahas, A. S., L. Cao, K. Choromanski and K. Scheinberg. 2021. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *Foundations of Computational Mathematics*. https://doi.org/10.1007/s10208-021-09513-z.
4. Bravo, M., D. Leslie and P. Mertikopoulos. 2018. Bandit learning in concave n-person games. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18. Red Hook, NY, USA, Curran Associates Inc., pp. 5666–5676.
5. Chung, K. L. Sep 1954. On a stochastic approximation method. *Ann. Math. Statist.* 25(3): 463–483.
6. Flaxman, A. D., A. T. Kalai and H. B. McMahan. 2005. Online convex optimization in the bandit setting: Gradient descent without a gradient. In: *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'05. USA, Society for Industrial and Applied Mathematics, pp. 385–394.
7. Gharesifard, B. and J. Cortés. 2013. Distributed convergence to nash equilibria in two-network zero-sum games. *Automatica* 49(6): 1683–1692.
8. Hu, G. and Y. Pang. 2020. Nash equilibrium seeking in n-coalition games via a gradient-free method. arXiv preprint: 2008.12909.
9. Jarrah, M., M. Jaradat, Y. Jararweh, M. Al-Ayyoub and A. Bousselham. 2015. A hierarchical optimization model for energy data flow in smart grid power systems. *Information Systems* 53: 190–200.
10. Kamgarpour, C. and G. Tatarenko. 2020. Bandit online learning of nash equilibria in monotone games. arXiv preprint: 2009.04258.
11. Meng, M. and X. Li. 2020. On the linear convergence of distributed nash equilibrium seeking for multi-cluster games under partial-decision information. arXiv preprint: 2005.06923.

12. Niyato, D., A. V. Vasilakos and Z. Kun. 2011. Resource and revenue sharing with coalition formation of cloud providers: Game theoretic approach. In: *2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 215–224.
13. Pang, J.-S. and F. Facchinei. 2003. *Finite-dimensional variational inequalities and complementarity problems: vol. 1*. Springer series in operations research. Springer, New York, Berlin, Heidelberg.
14. Scutari, G., D. Palomar, F. Facchinei and J. S. Pang. 2010. Convex optimization, game theory, and variational inequality theory. *IEEE Signal Processing Magazine* 27(3): 35–49.
15. Shi, W., Q. Ling, G. Wu and W. Yin. 2015. EXTRA: An Exact First-Order Algorithm for Decentralized Consensus Optimization. *SIAM Journal on Optimization* 25(2): 944–966.
16. Sun, C. and G. Hu. 2021. Distributed generalized nash equilibrium seeking of n-coalition games with full and distributive constraints. arXiv preprint: 2109.12515.
17. Tang, Y. and N. Li. 2019. Distributed zero-order algorithms for nonconvex multi-agent optimization. In: *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 781–786.
18. Tatarenko, T. and M. Kamgarpour. 2019. Learning nash equilibria in monotone games. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 3104–3109.
19. Tatarenko, T., J. Zimmermann and J. Adamy. 2021. Gradient play in n-cluster games with zero-order information. arXiv preprint: 2107.12648.
20. Tellidou, A. C. and A. G. Bakirtzis. Nov 2007. Agent-based analysis of capacity withholding and tacit collusion in electricity markets. *IEEE Transactions on Power Systems* 22(4): 1735–1742.
21. Ye, M., G. Hu and F. L. Lewis. 2018. Nash equilibrium seeking for N-coalition noncooperative games. *Automatica* 95: 266–272.
22. Ye, M., G. Hu and S. Xu. 2020. An extremum seeking-based approach for Nash equilibrium seeking in N-cluster noncooperative games. *Automatica* 114: 108815.
23. Zeng, X., J. Chen, S. Liang and Y. Hong. 2019. Generalized Nash equilibrium seeking strategy for distributed nonsmooth multi-cluster game. *Automatica* 103: 20–26.
24. Zhao, C, J He, P Cheng and J Chen. 2017. Consensus-Based Energy Management in Smart Grid With Transmission Losses and Directed Communication. *IEEE Transactions on Smart Grid* 8(5): 2049–2061.
25. Zhou, J., Y. Lv, G. Wen, J. Lv and D. Zheng. 2021. Distributed nash equilibrium seeking in consistency-constrained multi-coalition games. arXiv preprint: 2106.10513.
26. Zimmermann, J., T. Tatarenko, V. Willert and J. Adamy. 2021. Solving leaderless multi-cluster games over directed graphs. *European Journal of Control* 62: 14–21.

Bionotes

Tatiana Tatarenko
Control Methods and Robotics Lab,
Technical University of Darmstadt,
Darmstadt, Germany
tatiana.tatarenko@rmr.tu-darmstadt.de

Tatiana Tatarenko received the Dipl.-Math. degree in mathematics with focus on Statistics and Stochastic Processes from Lomonosov Moscow State University, Moscow, Russia, in 2011, and the Ph. D. degree from the Technical University of Darmstadt, Germany, in 2017. She is currently a leader of the research group “Distributed Optimisation and Game Theory” at the Control Methods and Robotics Lab under the direction of Prof. Dr.-Ing. Jürgen Adamy, Department of Electrical Engineering and Information Technology, TU Darmstadt. Her main fields of research are applied mathematics, distributed optimisation and game theory.

Jan Zimmermann
Control Methods and Robotics Lab,
Technical University of Darmstadt,
Darmstadt, Germany
jan.zimmermann@rmr.tu-darmstadt.de

Jan Zimmermann received the degree Master of Science in Electrical Engineering and Information Science from the Technical University of Darmstadt, Germany in 2018. The same year he joined the Control Methods and Robotics Lab led by Prof. Dr. Jürgen Adamy as a Ph. D. student and started working in the research team of Dr. Tatiana Tatarenko. His current research interests include multi-agent-systems, distributed optimization, game theory and application of these fields in the context of smart grids.