Network Games: Condensation of the Graph as a Hierarchical interpretation of the Game

Giacomo Casadei, Carlos Canudas de Wit

Network Games: Condensation of the Graph as a Hierarchical interpretation of the Game

G. Casadei [*] **C. Canudas De-Wit** [*]

[*] *Univ. Grenoble Alpes, CNRS, Inria, GIPSA-lab, F-38000 Grenoble, France (e-mail: giacomo.casadei@gipsa-lab.fr, carlos.canudas-de-wit@gipsa-lab.fr).*

Abstract: In this paper we investigate the problem of optimal games played over networks and focus our attention on the importance of the topology of communication between the agents. We consider a set of agents which are connected via a directed communication graph: each agent in the network has to optimize a local cost function which depends on the agent's decision and on the decision taken by the set of its neighbors, giving rise to a Network Game. We show that, by condensing the strongly connected components of the control graph into *super-nodes*, it is possible to give a hierarchical interpretation to the Network Game. Then we apply the proposed architecture to the case of a large scale network which takes inspiration by traffic networks application.

Keywords: Control of Networks, Network Games, Graph Theory, Aggregation over Networks

1. INTRODUCTION

Control and optimization over large population networks have become a popular topic within the control community. The main reason is that modern applications require multiple systems to communicate and interact with each other to fulfill the desired task. For instance power networks, sensor networks and social networks are solid examples in which is fundamental to control different parts of the network to achieve a global desired behavior.

In the recent years, the control community has largely focused on cooperative approaches to networks. In this framework the agents in the network are *willing* to collaborate and find an agreement between each other in such a way that they *coordinate their motion.* This approach is represented by the huge literature of consensus/synchronization problems (Isidori et al. (2014), Arcak (2007)) and its application to robot-swarm coordination (Sepulchre et al. (2008), Olfati-Saber (2006)), power networks (Dhople et al. (2014), Xiang and Hill (2014)) and social networks (Mirtabatabaei and Bullo (2012), Blondel et al. (2010)). However, not in all the frameworks and not in all the situations, it is possible to consider a cooperative approach. In several scenarios, the nodes are *selfish* and in competition with the others to pursue their goal. This leads to a non-cooperative interaction between the agents and thus to games played over networks. Notable examples of this scenario can be found in traffic networks and more in general in network congestion control (Pisarski and Canudas de Wit (2015), Barrera and Garcia (2015)) and smart grids (Ma et al. (2014), Mohsenian-Rad et al. (2010)).

When the number of nodes in the network is *large*, it becomes analytically impossible to use conventional game theoretic tools to find a solution to the problem. This motivated researchers to define a new type of games, named *aggregative*, where the response of an agent depends, rather than on each other players decision, on the *aggregation* of all the other agents action (see Jensen (2006), Jensen (2010)). The problem of games over networks can be seen as an extension of aggregative game, where the aggregation map depends indeed on the topology of a communication graph. In other words the payoff of each players depends only on the set of neighbors (see Jackson and Zenous (2014) for an overview on the topic). In Parise et al. (2015), the authors considered a refined typology of networks games in which the aggregate information is depending on a directed communication graph and showed that under a certain number of conditions the players reach a Nash Equilibrium. In this paper we consider a similar game setting, where a certain number of nodes representing physical systems are influencing each other through a graph. First, we study the influence of this graph topology on the structure of the game and show that the *condensation* of the graph leads to a hierarchical interpretation of the game and thus to a *quasi-sequential* architecture of optimization. Then, we introduce the concept of *physical graph* and *control graph* in flow networks, and show that the condensation of the control graph helps in determining the equilibrium the agents will reach.

The paper is organized as follows. In Section 2, we present the theoretical problem we aim to solve. In Section 3, we analyze the graph topology and provide a level-of-priority interpretation of the game. Then, in Section 4 we specify the problem to a generic large scale *flow networks* and

[⋆] The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement N° 694209).

[⋆⋆]The authors would like to thank Rémi Piotaix for the help and support.

introduce the notion of control graph. In Section 5 we present some interesting simulation results.

1.1 Notation and basic concept on Graph Theory

Denoting by $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ the set of N nodes of the network, a topology is described by a *directed communication graph* given by the following objects:

- $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of *edges* that models the interconnection between nodes, according to the following convention: (v_k, v_j) belongs to $\mathcal{E}$ if there is a flow of information from node j to node k. It is assumed that there are no self-loops, i.e. that $(v_k, v_k) \notin \mathcal{E}$.
- for each $(v_k, v_j) \in \mathcal{E}$ the flow of information from node j to node k in the i-th topology is *weighted* by the (k, j)-th entry $a_{kj} \geq 0$ of the so-called *adjacency matrix* $A \in \mathbb{R}^{N \times N}$.

The set of *neighbors* of node v_k is the set $\mathcal{N}_k = \{v_j \in \mathcal{V} : a_{kj} \neq 0\}$. A *path* from node v_j to node v_k is a sequence of r distinct nodes $\{v_{\ell_1}, \ldots, v_{\ell_r}\}$ with $v_{\ell_1} = v_j$ and $v_{\ell_r} = v_k$ such that $(v_{i+1}, v_i) \in \mathcal{E}$.

Theorem 1. A graph $\mathcal{G}$ is said to be *connected* if there is a node v such that, for any other node $v_k \in \mathcal{V}\backslash\{v\}$, there is a path from v to v_k. A graph is said to be strongly connected if for every node $v_i \in \mathcal{V}$, there exists a path from v_i to all $v_k \in \mathcal{V} \setminus \{v_i\}$.

The property of connectedness just introduced can be formulated also for portion of the graph, namely subgraphs.

Definition 1. A strongly connected component (SCC) of a di-graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, A\}$ is a maximal subgraph $\tilde{\mathcal{G}} = \{\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, \tilde{A}\}$ subject to being strongly connected.

2. PROBLEM FORMULATION

In this paper we investigate the problem of games over networks. Our main goal is to give an immediate understanding of the hierarchical dependency between different *parts* of the network.

A N-player game is defined as $\Gamma = (J_i, U_i)$ with $U_i \subseteq \mathbb{R}^n$ the finite dimensional convex strategy set of player i, $U = \prod_i U_i$ the joint strategy set and $J_i : S \to \mathbb{R}$ the upper semi-continuous utility function of player i. Given a strategy profile $u_{-i} \in U_{-i}$ for all agents except agent i, the best-response correspondence for agent i is

$$R(u_{-i}) = \arg\min_{u_i} \{J_i(u_i, u_{-i})\}$$

In the aggregative games scenario, the utility function of each agent can be written as $J_i\big(u_i, \sigma_i(u_{-i})\big)$ where σ_i is the *so-called* aggregate information. The aggregate function is in general defined as

$$\sigma_i = \sum_{j \neq i} u_j$$

so as the sum of all other players decision. In our setting, we are interested in considering a different class of aggregative games in which the interaction term depends only on the set of neighbors: in particular, given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, A\}$, we define $\sigma_i : U_{\mathcal{N}_i} \to \mathbb{R}$, with $\mathcal{N}_i$ the set of neighbors of agent i, as

$$\sigma_i = \sum_{j=1}^{N} a_{ij} u_j \tag{1}$$

where a_{ij} are the elements of the adjacency matrix $A \in \mathbb{R}^{N \times N}$ describing the communication between the players. In accordance with the definition of σ_i in (1), we consider a set of local utility functions defined as

$$J_i\big(u_i, \sigma_i\big) = u_i^T Q^i u_i + (\sigma_i^T + c^i) R^i u_i \tag{2}$$

with $Q^i, R^i \in \mathbb{R}^{n \times n}$ matrices of weights. The cost function is indeed composed by two terms: the first is a quadratic term in u_i while the second term penalizes the interaction with the neighbors through σ_i. The local optimization problem is formulated as

$$u_i^\star = \arg\min_{u_i \in U_i} J_i\big(u_i, \sigma_i\big) \tag{3}$$

One of the most important concepts in games is the Nash equilibrium. A Nash equilibrium is a strategy set $\{u_i^\star\}_{i=1}^N$ in which no agent has any advantage in deviating from its own strategy given the other players'. In this framework, the aggregative Nash equilibrium can be formulated as follows.

Definition 2. Given a N-player game $\Gamma = (J_i, U_i)$ and a communication graph $\mathcal{G}$, a set of strategies $\{u_i^\star \in U_i\}_{i=1}^N$ is said to be a Nash equilibrium for Γ if

$$J\big(u_i, \sigma_i(u_{\mathcal{N}_i})\big) = \min_{u_i \in U_i} J_i\big(u_i, \sigma_i(u_{\mathcal{N}_i})\big) \tag{4}$$

for all $i = 1, \ldots, N$.

3. CONDENSATION OF THE GRAPH: A SEQUENTIAL ARCHITECTURE

In general, solving an optimization problem over a complex graph is a really difficult task. In case of loops, for instance, a *trade* between the various agents has to take place. Furthermore, the agents might take decisions in different moments (*i.e* not being *synchronous*). One could wonder if there are nodes with *high priority* from which to start the computation and then propagate the result to the others.

In this section we introduce a methodology to interpret the network and obtain a hierarchical representation of the graph. Through this hierarchy, it will be possible to understand clearly the inter-dependence of the nodes and to set a *sequential* procedure of optimization. First, we introduce the concept of *condensation* of a graph.

Definition 3. The condensation $\mathcal{G}^\star$ of a graph $\mathcal{G}$ is a reduced graph in which:

- strongly connected components (SCC) of $\mathcal{G}$ are contracted to a single node
- (SCC_i, SCC_j) is an edge in $\mathcal{G}^\star$ if and only if there exists a $v_i \in SCC_i$ and a $v_j \in SCC_j$ such that (v_i, v_j) is and edge in $\mathcal{G}$

Furthermore, we recall a fundamental result about directed graphs.

Theorem 2. (Th. 3.6, Harary et al. (1965)). For any directed graph $\mathcal{G}$, the condensation $\mathcal{G}^\star$ of $\mathcal{G}$ is a directed acyclic graph (DAG).

As a matter of fact $\mathcal{G}^\star$ can be seen as hierarchical representation of the optimization problem over the original graph

$\mathcal{G}$. In graph theory, a level assignment is a procedure in which for every node v_i, we define an integer n_i: a level ordering is a level assignment in which for every edge connecting v_i to v_j, either we have $n_i < n_j$ or $n_i > n_j$ (ascending and descending level assignment respectively). In the following, we consider only ascending level assignment without loss of generality.

The definition of the DAG $\mathcal{G}^\star$ allows us to describe the optimization over the original network $\mathcal{G}$ as a sequential procedure by sorting the nodes into levels. Hence, we define a *quasi-level assignment* (in which nodes belonging to a strongly connected component have the same level) based on the longest path to reach a node: given node v_i, its level n_i is equal to the length of the the longest path from any root of $\mathcal{G}^\star$ plus 1. Then, the level assignment $n_i \in [1, \ell+1]$ where ℓ is the length of the longest path in $\mathcal{G}^\star$. Based on this level assignment, we define a sequential procedure for solving the optimization process on the graph, in which nodes with the smallest level compute first and then propagate their decision to the higher levels in a cascade-fashion. This recursive procedure is illustrated in Table 1.

Algorithm for the DAG	
Initialize	set $k = 1$
Iterate	all nodes $v_j \in$ level k, compute their optimum value based on the result of the previous level $u^{(k-1)^\star}$: • if $v_j \in \operatorname{lev} k$ is simple node, $u_j^\star = \arg\min_{u_j \in U_j} J_j\big(u_j, \sigma_j\big)$ • if $v_j \in \operatorname{lev} k$ belongs to a super node, iterate $u_j^\star = \arg\min_{u_j \in U_j} J_j\big(u_j, \sigma_j\big)$ Then $u^{k^\star}(k) = \operatorname{col}\{u_j^\star\}$ for all $j : v_j \in \operatorname{lev} k$ If $k = \ell + 1$ stop

Table 1. Sequential procedere of optimization: a hierarchical structure.

In order to solve the problem of optimization *inside* the super-nodes we make use of game-theory tools and show that, given the information from the *father-nodes* (namely the nodes belonging to a *previous* level, if present), each super-node eventually reaches a Nash-equilibrium.

Suppose that the condensation procedure results in c super nodes. Then, for each node v_i belonging to SN_j ($j = 1, \dots, c$), it is possible to split the aggregate information σ_i in two parts: σ_i^f that comes from the upper levels and thus is a *constant*, and σ_i^{sn} that is the aggregate information from the nodes that belongs to the super-node. In other words, we write

$$J_i\big(u_i, \sigma_i(u_{\mathcal{N}_i})\big) = u_i^T Q^i u_i + \sigma_i^{SN^T} R^i u_i + \sigma_i^{f^T} R^i u_i \tag{5}$$

Suppose that the super-node has a level assignment n_j and that the nodes belonging to SN_j are $v_h, \dots, v_k$. Given a set of of initial strategies $\{u_i\}_{i=h}^k$ for the nodes belonging to SN_j and the aggregate information from the father-nodes (*i.e.* the nodes with a level assignment $n_f < n_j$), it is possible to define the super-node best response as

$$\mathbf{u}_j^\star(\boldsymbol{\sigma}_j) = [u_h(\sigma_h^f, \sigma_h^{sn}), \dots, u_k(\sigma_k^f, \sigma_k^{sn})] \tag{6}$$

which collects the strategies computed by each node belonging to SN_j. To describe the aggregate information available at the nodes in SN_j after each round of optimization, we introduce the aggregation mapping $\mathcal{U}_j : \mathbb{R}^{Nn} \to \mathbb{R}^{Nn}$

$$\mathcal{U}_j(\boldsymbol{\sigma}_j) = \begin{bmatrix} \sum_{i=1}^N a_{hi} u_i^\star(\sigma_h^f, \sigma_h^{sn}) \\ \vdots \\ \sum_{i=1}^N a_{ki} u_i^\star(\sigma_k^f, \sigma_k^{sn}) \end{bmatrix} \tag{7}$$

Proposition 1. The aggregation mapping (7) admits at least one fixed point. For each fixed point $\bar{\boldsymbol{\sigma}}$, the set of strategies $\{u_i^\star(\bar{\sigma}_i^f, \bar{\sigma}_i^{sn})\}_{i=h}^k$ is a Nash equilibrium for the game in SN_j.

Proof: The mapping $\mathcal{U}_j(\boldsymbol{\sigma}_j)$ is continuous and compact valued, so it possesses at least one fixed point. The fact that this point is a Nash equilibrium is a direct consequence of the definition of a Nash equilibrium in aggregative games in Definition 2 (see Smart (1974)). $\triangleleft$

Proposition 1 is valid for all c super-nodes SN_j in the network. To guarantee convergence to a Nash equilibrium one might consider an iterative myopic best response policy, in which each agent plays its best response to the neighbors aggregate information at each round. More in general, every strategy that is a contraction and a non-expansive mapping guarantees that the agents reach a Nash equilibrium (a detailed analysis of the conditions and the techniques to ensure that the Nash equilibrium or a near-Nash equilibrium is reached can be found in Jensen (2006), Jensen (2010), Voorneveld and Norde (1997), Kukushkin (2004)).

As far as the full game is concerned, we replicate the result in Proposition 1. We reorder the nodes and super-nodes belonging to the same level so that they are consecutive and define

$$u^i = \operatorname{col}\{u_j\} \qquad \text{for } j : v_j \in \operatorname{lev} i$$

Accordingly, the adjacency matrix A reads as

$$A = \begin{bmatrix} A^1 & & & \\ A^{1,2} & A^2 & & \\ \vdots & \vdots & \ddots & \\ A^{1,\ell} & \cdots & A^{\ell-1,\ell} & A^\ell \end{bmatrix} = \operatorname{blkdiag}(A_1, \dots, A_\ell) + A^{\mathrm{junc}}$$

Then, we can write the aggregate information by levels as

$$\sigma^i = \sum_{j=1}^{i-1} A^{j,i} u^j$$

and define the *best response* mapping

$$\mathbf{u}(\boldsymbol{\sigma}) = \operatorname{col}\big(u^1, u^2(\sigma^2), \dots, u^\ell(\sigma^\ell)\big)$$

As a consequence, the aggregation mapping for the entire game $\boldsymbol{\mathcal{U}} : \mathbb{R}^{Nn} \to \mathbb{R}^{Nn}$ as

$$\boldsymbol{\mathcal{U}}(\boldsymbol{\sigma}) = \begin{bmatrix} 0 \\ A^{1,2}u^{1^\star} \\ \vdots \\ \sum_{j=1}^{\ell-1} A^{j,\ell}u^{j^\star} \end{bmatrix} = (A^{\mathrm{junc}} \otimes I_n)\mathbf{u}(\boldsymbol{\sigma}) \tag{8}$$

Corollary 1. The aggregation mapping (8) admits at lest one fixed point. For each fixed point $\bar{\boldsymbol{\sigma}}$, the set of strategies $\{u^{i^\star}(\bar{\sigma}^i)\}_{i=1}^{\ell}$ is a Nash equilibrium for the entire game.

The proof is an straightforward consequence of Proposition 1. After each iteration of the algorithm in Table 1, we can express the update of the aggregate information as

$$\boldsymbol{\sigma}(k+1) = \boldsymbol{\mathcal{U}}\big(\boldsymbol{\sigma}(k)\big)$$

which, through the junction matrix A^{junc}, clearly shows the *cascade* propagation of decision between nodes with higher priority to nodes with lower priority in the *structure* of the equilibrium.

4. APPLICATION TO LARGE SCALE FLOW NETWORKS

In this section we apply the analysis presented in Section 3 to the case of large scale flow networks. Flow networks are of particular interest because they represent several field of application, such as pipe networks, power networks and traffic networks. A fundamental aspect about flow networks is to distinguish between two possible *interpretation* of the network itself:

- on one hand, the *physical representation* stands for the real connection between the nodes of the network in terms of flow
- on the other, the *control representation* stands for the dependence of the nodes on the others from the control point of view

The two representation do not necessarily coincide in the sense that the *control directionality* might not be the same as the *physical directionality* of the network. With P-Graph $\mathcal{G}^P = \{\mathcal{V}, \mathcal{E}, A\}$ we represent the physical interconnection between the N nodes in the network and thus the physical flow of *information*. Traffic networks can be seen as *flow networks* where the goal is to control the density of cars and avoid congestion. In this framework, when the density of vehicles is *low*, a certain section is considered in free flow, while when the density is *high*, the same section is in congestion. By partitioning the road into sections, the state of each section has two admissible configuration, namely:

- F (green): representing a *free* evolution of the section
- C (red): representing a *congested* evolution of the section

Based on the P-Graph $\mathcal{G}^P$, we also define a some limitation on the nodes state evolution. First, the nodes are allowed to change at discrete time instants $T_1, T_2, \ldots$, with periodicity T. Furthermore, given a certain configuration of free and congested nodes, the nodes are allowed to change configuration according to the following Assumption.

Assumption 1. Given a P-Graph $\mathcal{G}^P$, we assume that:

i) nodes which are free and have no outgoing edge towards congested node cannot change
ii) nodes which are congested and have no incoming edge from free nodes are not allowed to change

To each node v_i which is allowed to change, we associate a *probability* which is associated to the result of the optimization problem

$$\begin{aligned} P\big(v_i(T_{n+1}) = C | v_i(T_n) = F\big) &= P_i(u_i) \\ P\big(v_i(T_{n+1}) = F | v_i(T_n) = C\big) &= 1 - P_i(u_i) \end{aligned} \tag{9}$$

with $P_i(u_i)$ to be specified later.

As mentioned before, the control graph is an abstraction of the control *dirctions* and its derived from the physical graph, based on the state of each node in the network. As pointed out in Pisarski and Canudas de Wit (2015), traffic networks control is highly influenced by the congested/free configuration of the road. Based on the configuration of each section, namely congested/free, the congestion waves can propagate upstream or downstream. When a certain section is congested, in order to control the downstream propagation of the traffic wave, we have to act in the upstream direction: on the other hand, when a section is in free-flow, the control is applied following the downstream flow of the physical graph. The rules to determine the C-Graph $\mathcal{G}^C$ in case of traffic networks (note that they are equivalent to the case of pipe networks) are summarized in Tab 2.

State	$\mathcal{G}^P$	$\mathcal{G}^C$
FF	$\rightarrow$	$\rightarrow$
CC	$\rightarrow$	$\leftarrow$
FC	$\rightarrow$	$\leftrightarrow$
CF	$\rightarrow$	

Table 2. Table of rules to define the C-Graph.

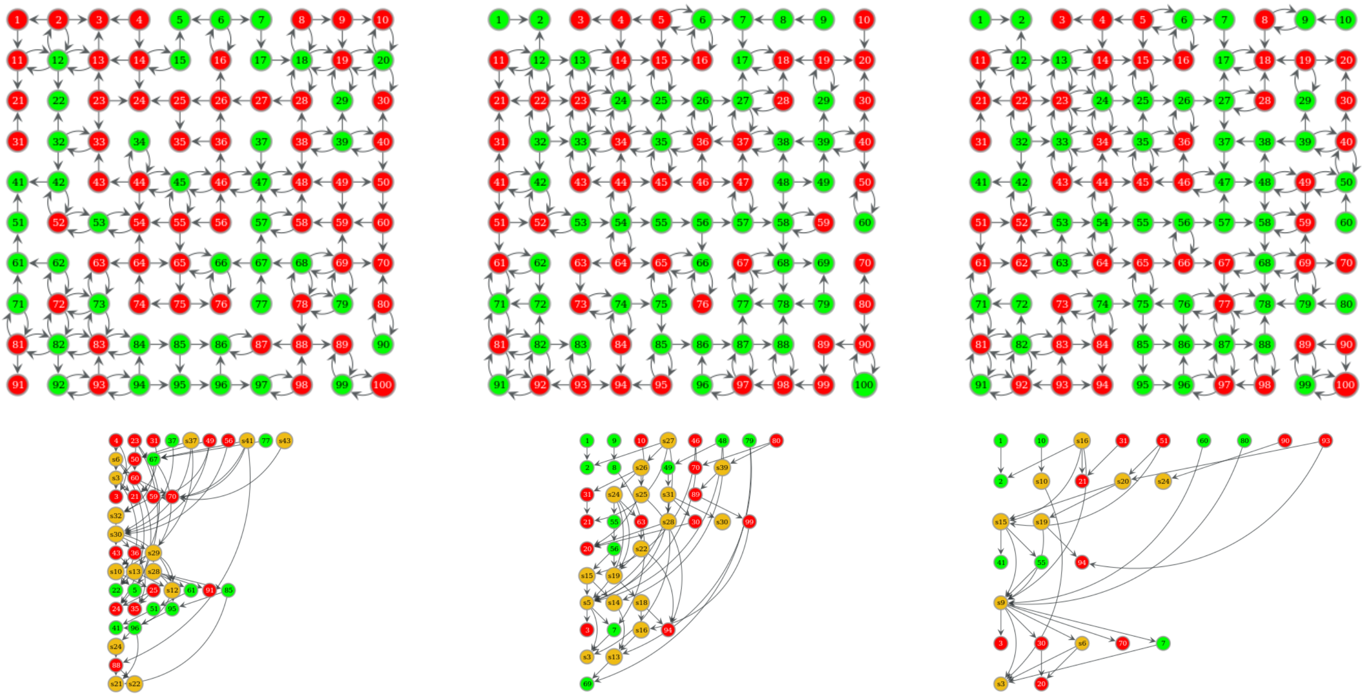
In our framework, nodes can be seen as sections of roads which can be controlled. As a consequence, the local optimization problem (3) is formulated according to the control graph, as

$$u_i^\star = \arg\min_{u_i \in U_i} J_i\big(u_i, \sigma_i\big)\,, \qquad \sigma_i(u_{-i}) = \sum_{j=1}^{N} a_{ij}^c u_j$$

with a_{ij}^c element of the Adjacency matrix $A^C \in \mathbb{R}^{N \times N}$ of $\mathcal{G}^C$.

5. IMPLEMENTATION OF THE PROPOSED ARCHITECTURE AND SIMULATION RESULTS

Following the probelm setting introduced in Section 4, we now display the fundamental steps to implement the approach presented in Section 3. The first step to be performed is the condensation of the graph described in Section 3. For large scale networks, one may think that the computation of the condensation $\mathcal{G}^{C\star}$ can be critical: however, it is important to stress that such a task can be computed in linear time. Several algorithms have been proposed in literature, for instance the *Tarjan's strongly connected components algorithm*, whose complexity is indeed proportional to the number of nodes and



(a) (b) (c)

Figure 1. Example of the evolution of the graph with 100 nodes: from the control graph we obtain the direct acyclic graph determining the structure of the game and the associated Nash equilibrium.

edges, *i.e.* $\mathcal{O}(|\mathcal{V}|, |\mathcal{E}|)$ [1] (see Tarjan (1972)). On the other hand, the *computation payload* substantially consists in the sequential optimization/game described in Table 1. In our framework, the optimization problem is decomposed into a sequential *father-to-son* best-response architecture: in other words, the optimization is a sequence of $\ell + 1$ optimization problems in which, the agents belonging to a certain level i compute in parallel their best response to level $i - 1$ and then pass their result to level $i + 1$.

We consider a network composed by 100 nodes. The physical network is randomly generated and the initial congested/free configuration is assigned arbitrarily to each nodes. The cost function at each node is defined according to (2) as

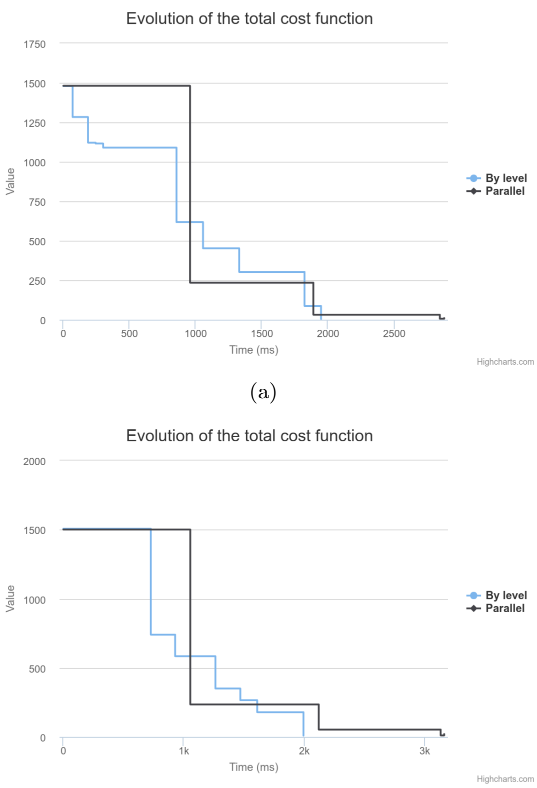
$$J_i(u_i, \sigma_i) = (u_i - \bar{u}_i)^2 + \Big(\sum_{j=1}^{N} \ell_{ij} u_i\Big)^2 \qquad (10)$$

where $\bar{u}_i \in \mathbb{R}$ is a local target value and ℓ_{ij} are elements of the Laplacian matrix associated to the control graph $\mathcal{G}^C$. In other words, (10) is composed by a term which penalizes the displacement with respect to the local target value $\bar{u}_i$ and by another term which penalizes the displacement with respect to the aggregate information σ_i. The N local target values are chosen such that $\bar{u}_i \in [0, 1]$ according to a *pseudo-normal* distribution with mean $\mu = 0.5$ and standard deviation $\varepsilon = 0.2$. Then, the probability to switch between the two configurations $P_i(u_i)$ in (9) is defined as $P_i = |u_i^\star - \bar{u}_i|$, namely the distance between the local target value $\bar{u}_i$ and the result of the optimization $u_i^\star$. This implies that the agents who are capable of getting closer to their target are more likely to become *free*. In Figure 1, three possible configurations of the network are considered and the relative condensation is shown. Starting from the control graph configuration 1(a), we derive the corresponding condensed graph: then, at $T = 5$ sec, the nodes evolves according to the result of the optimization. We obtain the control graph in 1(b) and its associated DAG, and in a similar fashion 1(c). Videos of various examples of the evolution of the nodes and of the optimization procedure is available at the following link: http://scale-freeback.eu/wp-content/uploads/2017/04/videos.zip.

We also compare the case in which agents are sorted by levels and the case in which the agents computes in *parallel*. With *parallel* we mean the case in which the agents are thought to be synchronous and thus compute their optimum all at the same time, without being reorganized: nevertheless, for computational constraints the computation is not *fully* decentralized, in the sense that there is not a separate computational facility for each node. We stress however that, for a *large scale* network, it is unfeasible to think that all the agents are synchronous.

In order to compare the two results, in the implementation of our architecture, we take into account the time necessary to perform the condensation of the graph and the level assignment. Despite the necessary *delay* to perform this computation, the architecture takes advantage of this reordering to compute in a hierarchical sequence the optimal solution to the local problems. Simulation results are shown in Figure 2(a) and 2(b), for a network of 100 and 10000 nodes respectively. The two procedure converge indeed to the same equilibrium, but with rather different computation times. For our architecture, it is possible to see the *level-by-level* decrements of the cost functions while in the case the of the parallel computation the decrements

[1] Here $|\mathcal{A}|$ stands for the cardinality of the set $\mathcal{A}$.



(a)

(b)

Figure 2. Game over a network and comparison between parallel computation and computation by hierarchy levels with: (a) with 100 nodes, (b) with 10000 nodes.

are relative to a full round of optimization over all the nodes.

6. CONCLUSION

In this paper we considered a class of network games and studied the influence of the networks topology in the equilibrium the nodes reach. Based on the topology, we defined a hierarchical architecture which solves the optimization problem in a cascade fashion. The approach has been tested on a simplified scenario mimicking flow networks. The first natural extension of this paper is to apply the proposed architecture to a real case scenario, in particular for controlling complex traffic networks. The idea of *clusterizing* certain parts of the networks has promising perspective in this direction. Furthermore, the analysis of the topology in network games allows also to consider different aspects to the problem: for instance one could think to treat the super-nodes problem with a different approach than simple nodes, or to consider the condensation of the graph as a *controllability* property of the game .

REFERENCES

Arcak, M. (2007). Passivity as a design tool for group coordination. *IEEE Trans. Aut. Contr.*, 52(8), 1380–1390.

Barrera, J. and Garcia, A. (2015). Dynamic incentives for congestion control. *IEEE Trans. on Automatic Control*, 60(2), 299–310.

Blondel, V.D., Hendrickx, J.M., and Tsitsiklis, J.N. (2010). Continuous-time average-preserving opinion dynamics with opinion-dependent communications. *SIAM Journal on Control and Optimization*, 48(8), 5214–5240.

Dhople, S.V., Johnson, B.B., Dörfler, F., and Hamadeh, A.O. (2014). Synchronization of nonlinear circuits in dynamic electrical networks with general topologies. *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 61(9), 2677–2690.

Harary, F., Norman, R.Z., and Cartwright, D. (1965). *Structural models: An Introduction to the theory of Directed Graphs.* John Wiley & Sons.

Isidori, A., Marconi, L., and Casadei, G. (2014). Robust output synchronization of a network of heterogeneous nonlinear agents via nonlinear regulation theory. *IEEE Trans. Aut. Contr.*, 59(10), 2680–2692.

Jackson, M.O. and Zenous, Y. (2014). *Games on Networks.* Elsevier Science.

Jensen, M.K. (2006). Aggregative games and best-reply potentials. *https://ideas.repec.org/p/bir/birmec/06-10.html.*

Jensen, M.K. (2010). Aggregative games. *Games and Economic Behavior*, 43, 45–66.

Kukushkin, N.S. (2004). Best response dynamics in finite games with additive aggregation. *Games and Economic Behavior*, 94–100.

Ma, K., Hu, G., and Spanos, C. (2014). Distributed energy consumption control via real-time pricing feedback in smart grid. *IEEE Trans. on Control Systems Technology*, 22(5), 1907–1914.

Mirtabatabaei, A. and Bullo, F. (2012). *Opinion dynamics in heterogeneous networks: Convergence conjectures and theorems*, 50(5), 2763–2785.

Mohsenian-Rad, A.H., Wong, V., Jatskevich, J., Schober, R., and Leon-Garcia, A. (2010). Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. *IEEE Trans. on Smart Grid*, 1(3), 320–331.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Contr.*, 51(3), 401–420.

Parise, F., Gentile, B., Grammatico, S., and Lygeros, J. (2015). Network aggregative games: Distributed convergence to nash equilibria. *Proc. in 54th CDC*, 2295–2301.

Pisarski, D. and Canudas de Wit, C. (2015). Nash game based distributed control design for balancing of traffic density over freeway networks. *IEEE Transactions on Control of Network Systems*, 149–161.

Sepulchre, R., Paley, D.A., and Leonard, N.E. (2008). Stabilization of planar collective motion with limited communication. *IEEE Trans. Autom. Control*, 53(3), 706–719.

Smart, D.R. (1974). *Fixed Pont Theorems.* Cambridge University Press.

Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2), 146–160.

Voorneveld, M. and Norde, H. (1997). A characterization of ordinal potential games. *Games and Economic Behavior*, 19(2), 235–242.

Xiang, J.and Li, Y. and Hill, D.J. (2014). Cooperative output regulation of multi-agent systems coupled by dynamic edges. *19-th IFAC World Congress*, 1813–1818.