Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 51-32 (2018) 367–372

Satisfying the Cost Constraints in a Network System Operating by the Consensus Protocol with Different Task Priorities ⋆

Natalia Amelina * **Oleg Granichin** * **Olga Granichina** ** **Yury Ivanskiy** * **Yuming Jiang** ***

* *Faculty of Mathematics and Mechanics, Saint Petersburg State University 198504, Universitetskii pr. 28, St. Petersburg, Russia, Institute of Problems of Mechanical Engineering, the Russian Academy of Sciences, 199178, Bolshoy pr. 61, V.O., St. Petersburg, Russia (e-mail: natalia_amelina@mail.ru, oleg_granichin@mail.ru, ivanskiy.yuriy@gmail.com).*
** *Herzen State Pedagogical University, 191186, Moika Emb. 48, St.Petersburg, Russia)*
*** *Norwegian University of Science and Technology, NO-7491, Trondheim, Norway (e-mail: jiang@item.ntnu.no)*

Abstract: A multi-agent network system of different computing nodes processing tasks of different priority levels is considered. Agents balance their loads for each priority level by achieving consensus of their load values. Agents operate by local voting protocol and exchange information about their states in presence of noise in communication channels in the system with switching topology. The network usage for task exchange is limited by the constraints on average cost of utilized links. A way to meet the constraints by randomization of link usage is considered. Simulation illustrating the considered approach is provided.

Keywords: Cost constraints, Multi-Agent Systems, Consensus achievement.

1. INTRODUCTION

Recently the consensus approach has been widely applied for solving various practical problems such as cooperative control of multivehicle networks Ren et al. (2007); Granichin et al. (2012), distributed control of robotic networks Bullo et al. (2009), flocking problem Yu et al. (2010a); Virágh et al. (2014), optimal control of sensor networks Kar and Moura (2010) and others. An interesting application of the consensus approach is adaptation of airplane's "feathers" in a turbulence flow Granichin et al. (2017). A possible way to utilize consensus approach in the task of load balancing in computer, production, transport, logistics, and other service networks is to formulate the problem of load balancing as the consensus achievement among nodes' loads across the network Amelina et al. (2015). In works Ren and Beard (2007); Chebotarev and Agaev (2009); Li and Zhang (2009); Yu et al. (2010b); Huang (2012); Proskurnikov (2013); Lewis et al. (2014); Olfati-Saber and Murray (2004) the authors considered the conditions for achieving consensus in multi-agent network systems.

In the networks processing tasks with several priority levels to equalize agents' loads different priority levels should be treated separately. In order to balance the load across the network system via consensus protocol the consensus should be targeted for each class separately since the consensus values of agents' loads could differ for separate priority levels. This calls for differentiated consensuses problem setting i.e. achieving the consensus for each priority level in the network with tasks of different priorities Amelina et al. (2014a,b).

In practice it might be important to limit communication between the agents in the network. The limitations could be caused by necessity to save the battery charge for autonomous agents. The need to constrain the task exchange among the nodes could also be caused by the network traffic prioritization Jiang et al. (2002). In case of limited resources they are utilized to serve the high priority tasks first while lower priority tasks wait to be processed. A way to limit the agent communication by satisfying the imposed averaged cost constraints on utilized netwotk topolgy by randomization of communication links usage was suggested in Amelina et al. (2014b).

In this paper we consider the approach to meet the averaged cost constraints on the utilized network introduced in Amelina et al. (2014b) in more detail.

The paper is organized as follows. Notation used in the paper and the problem formulation are given in Section 2. The control protocol for achieving the consensus is introduced in Section 3. In Section 4 the main assumptions and

⋆ This work was supported by Russian Foundation for Basic Research (projects 16-07-00890 and 17-51-53053).

2405-8963 © 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.11.411

main results are presented. Simulation results are given in Section 5. Section 6 contains conclusion remarks.

2. PROBLEM STATEMENT

We consider a dynamic network system of n agents, which exchange information among themselves during tasks processing. Tasks of m different classes may come to different agents of the system in different discrete time instants $t = 0, 1, \ldots$. Agents process incoming tasks in parallel. Tasks can be redistributed among agents based on a feedback.

Without loss of generality, agents in the system are numbered. Assume $N = \{1, \ldots, n\}$ denotes the set of agents in the network system, E denotes the set of edges of topology graph. Let $i \in N$ be the number of an agent. The network topology may switch over time. Let the dynamic network topology be modeled by a sequence of digraphs $\{(N, E_t)\}_{t\geq 0}$, where $E_t \subset E$ denotes the set of edges at time t of topology graph (N, E_t). The corresponding adjacency matrices are denoted as $A_t = [a_t^{i,j}]$, where $a_t^{i,j} > 0$ if agent j is connected with agent i and $a_t^{i,j} = 0$ otherwise. Here and below, an upper index of agent i is used as the corresponding number of an agent (not as an exponent). Denote $\mathcal{G}_{A_t}$ as the corresponding graph.

To introduce some properties of the network topology, the following definitions from the graph theory will be used. Define *the weighted in-degree* of node i as the sum of i-th row of matrix A: $deg_i^-(A) = \sum_{j=1}^n a^{i,j}$; $\mathcal{D}(A) = \mathrm{diag}\{deg_1^-(A), \ldots deg_n^-(A)\}$ is the corresponding diagonal matrix; $deg_{max}^-(A)$ is the maximum in-degree of graph $\mathcal{G}_A$. Let $\mathcal{L}(A) = \mathcal{D}(A) - A$ denote the *Laplacian* of graph $\mathcal{G}_A$; $\cdot^{\mathrm{T}}$ is a vector or matrix transpose operation; $||A||$ is the Frobenius norm: $\|A\| = \sqrt{\sum_i \sum_j (a^{i,j})^2}$; $Re(\lambda_2(A))$ is the real part of the second eigenvalue of matrix A ordered by the absolute magnitude; $\lambda_{\max}(A)$ is the maximum eigenvalue of matrix A. $\mathbb{E}$ is a mathematical expectation symbol.

It is said that digraph $\mathcal{G}_B$ is a subgraph of a digraph $\mathcal{G}_A$ if $b^{i,j} \leq a^{i,j}$ for all $i, j \in N$.

Digraph $\mathcal{G}_A$ is said to contain a *spanning tree* if there exists a directed tree $\mathcal{G}_{tr} = (N, E_{tr})$ as a subgraph of $\mathcal{G}_A$ which includes all vertices of $\mathcal{G}_A$.

We suppose that tasks (jobs) belong to different classes $k = 1, \ldots, m$ and every agent has m queues — one for each task class.

The behavior of agent $i \in N$ is described by characteristics of two types:

- lengths of m queues of tasks of each class k at time instant t: $q_t^{i,k}$, $k = 1, \ldots, m$,
- average productivity: p_{av}^i or average amount of tasks of all priorities (i.e. $p_{av}^i = \mathbb{E}(p_t^i) = \mathbb{E}(\sum_{k=1}^m p_t^{i,k}) = \sum_{k=1}^m p_{av}^{i,k}$) processed by agent i during certain time interval.

Here and further $p_t^{i,k}$ stands for number of tasks of priority k processed by agent i at time instant t and $p_{av}^{i,k} = \mathbb{E}p_t^{i,k}$. Each agent should distribute its own productivity among all task classes in such a way that, on one hand the priorities for task classes are provided and on the other hand the "starvation problem" is taken into account i.e. tasks of the lower priority classes do not wait for execution for too long. This is achieved by making use of the probabilistic priority discipline Jiang et al. (2002). Each task class is given a productivity fraction P_k, $k = 1, \ldots, m$ which is the same for a certain class k on every agent in the system. On each agent the tasks from their queues are chosen for execution randomly according to the following formula:

$$\tilde{p}_t^{i,k} = \begin{cases} \dfrac{P_k}{\sum_{q_t^{i,l}>0} P_l}, & \text{if } q_t^{i,k} > 0; \\ 0, & \text{otherwise,} \end{cases}$$

where $\tilde{p}_t^{i,k}$ is the probability of choosing a task of class k for execution on agent i at a time instant t. Therefore the bigger fraction P_k corresponds to the higher chance of that task of class k to be executed. Thus the agent's productivity is distributed among all classes of tasks in the following way: $\mathbb{E}p_t^{i,k} = p_{av}^{i,k} = \tilde{p}_t^{i,k} p_{av}^i$. Here $p_t^{i,k}$ is number of operations allocated for tasks of class k on agent i at time instant t if the productivity p_{av}^i means the whole number of operations which agent i is able to proceed during the time from t till $t+1$. Note that according to the definition of $\tilde{p}_t^{i,k}$ if at certain time instant t' the queue of tasks of class k' on the agent i' is empty, no operations would be allocated for tasks of class k'. Instead $p_{t'}^{i',k'}$ operations would be distributed among other task classes in proportions of their productivity fractions P_k, $k \neq k'$.

For all $i \in N$, $t = 0, 1, \ldots$, the dynamics of the network system in a vector form is as follows

$$\mathbf{q}_{t+1}^i = \mathbf{q}_t^i - \mathbf{p}_t^i + \mathbf{z}_t^i + \mathbf{u}_t^i, \tag{1}$$

where $\mathbf{q}_t^i = [q_t^{i,k}]$ is a vector whose kth element is defined by the amount of tasks of kth class; $\mathbf{p}_t^i = [p_t^{i,k}]$, and $\mathbf{z}_t^i = [z_t^{i,k}]$ is an m-vector whose kth element $z_t^{i,k}$ is the amount of new tasks of class k, which came to the system and were received by agent i at time instant t; $\mathbf{u}_t^i \in \mathbb{R}^m$ is a vector of control actions (redistributed tasks of class k to agent i at time instant t), which could (and should) be chosen based on some information about queue lengths of neighbors $\mathbf{q}_t^j$, $j \in N_t^i$, where N_t^i is the set $\{j \in N : a_t^{i,j} > 0\}$.

Tasks have different priorities and, for each priority, the maximum cost of the network $\{N_t^i, i \in N\}$ that could be used is defined the following way:

$$C(\{N_t^i, i \in N\}) = \max_{i \in N} \sum_{j \in N_t^i} a_t^{i,j}. \tag{2}$$

Let the tasks of class 1 have the highest priority and tasks of class m have the lowest priority. The highest priority tasks should be served faster therefore the network of higher cost should be available for their redistribution among agents. Consider subgraphs of network graph $\mathcal{G}_{A_t}$ and let B_t^k be the adjacency matrices for network subgraphs available for kth class tasks transmission. Since $\mathcal{G}_{B_{av}^k}$ should have the "richest" topology for $k = 1$ and $\mathcal{G}_{B_{av}^m}$ should have the most "poor" one, we could say that the network has the topology decomposition $\{\mathcal{G}_{B_{av}^k}\} : \mathcal{G}_{B_{av}^m} \subseteq$

$\mathcal{G}_{B_{av}^{m-1}} \subseteq \ldots \subseteq \mathcal{G}_{B_{av}^{1}}$, where $\mathcal{G}_{B_{av}^{k}}$ stands for the graph with adjacency matrix $B_{av}^k = \mathbb{E}(B_t^k)$.

***Definition** 1.* We will say that network topology decomposition $\{\mathcal{G}_{B_t^k}\}$ satisfies average cost constraint $\{c_k\}$ if for every priority class k

$$deg^-_{\max}(B_{av}^k) = \mathbb{E}deg^-_{\max}(B_t^k) = \mathbb{E}\max_{i\in N}\sum_{j\in N_t^{i,k}} b_t^{i,j,k} \le c_k, \tag{3}$$

where $N_t^{i,k}$ is the neighbors set of agent i at time t formed in accordance with the topology $\mathcal{G}_{B_t^k}$. In case the maximal in-degree violates the cost constraint we randomize usage of the links between the agents in such a way that average cost of used topology meets the the cost constraint.

Consider the following example. For the undirected ring of 6 agents network topology graph adjacency matrix is

$$A = \begin{pmatrix} 0&1&0&0&0&1\\ 1&0&1&0&0&0\\ 0&1&0&1&0&0\\ 0&0&1&0&1&0\\ 0&0&0&1&0&1\\ 1&0&0&0&1&0 \end{pmatrix}.$$

Suppose cost constraints are $c_1 = 2$, $c_2 = 1$. Since in the example the cost of each link equals 1, each agent is allowed to use two links for exchange of priority 1 tasks and one link per time instant in average for exchange of tasks of priority 2. Agents don't have to randomize the link usage for priority 1 tasks, so the adjacency matrix of network graph for priority 1 task exchange B_t^1 (matrix of redistribution protocol for priority 1) equals A for all $t = 0, 1, \ldots$. So $B_{av}^1 = B_t^1 = A,\ t = 0, 1, \ldots$. To meet cost constraint for priority 2 agents may use the links with probability 0.5 so that in average the cost of utilized network is 1. B_t^2 for certain t may be as follows:

$$B_t^2 = \begin{pmatrix} 0&1&0&0&0&1\\ 1&0&0&0&0&0\\ 0&1&0&1&0&0\\ 0&0&0&0&0&0\\ 0&0&0&0&0&1\\ 1&0&0&0&1&0 \end{pmatrix}.$$

Note that at each time instant topology graph for priority 2 tasks exchange $\mathcal{G}_{B_t^2}$ may not be connected. But if the graph $G_{B_{av}^2}$ corresponding to averaged adjacency matrix B_{av}^2 is connected, load would be balanced across the network operating by protocol (6). Since we aim to meet the averaged cost constraints requirement the view of averaged adjacency matrices B_{av}^1, B_{av}^2 is more interesting than the view of adjacency matrices B_t^1, B_t^2 at certain time instant t. For cost constraints $c_1 = 2$, $c_2 = 1$ corresponding averaged adjacency matrices of utilized network topology are

$$B_{av}^1 = \begin{pmatrix} 0&1&0&0&0&1\\ 1&0&1&0&0&0\\ 0&1&0&1&0&0\\ 0&0&1&0&1&0\\ 0&0&0&1&0&1\\ 1&0&0&0&1&0 \end{pmatrix}, B_{av}^2 = \begin{pmatrix} 0&0.5&0&0&0&0.5\\ 0.5&0&0.5&0&0&0\\ 0&0.5&0&0.5&0&0\\ 0&0&0.5&0&0.5&0\\ 0&0&0&0.5&0&0.5\\ 0.5&0&0&0&0.5&0 \end{pmatrix}.$$

Cost constraints $deg^-_{\max}(B_{av}^1) = 2$ and $deg^-_{\max}(B_{av}^2) = 1$ are satisfied.

Denote $p_{av}^{i,k} = \mathbb{E}p_t^{i,k}$ and

$$x_t^{i,k} = \frac{q_t^{i,k}}{p_{av}^{i,k}} \tag{4}$$

the *load* of agent $i \in N$ for priority class $k = 1, \ldots, m$. Though on practice p_{av}^i could be unknown in advance, we consider the problem setting in which the agents' average productivities are known. Assume that $p_{av}^i \ne 0,\ \forall i \in N$ and $P_k \ne 0,\ k = 1, \ldots, m$. In Amelina et al. (2015) it was proven that from all possible options for the redistribution of all tasks the minimum operation time of the system is achieved when loads $x_t^{i,k}$ are equalized throughout the network. Hence, it is important to consider the achievement of the following goal.

It is required to maintain balanced (equal) loads across the network for every priority class and, at the same time, to meet the cost constraint requirement.

At this setting we can consider the consensus problem for states $\mathbf{x}_t^i = [x_t^{i,k}]$ of agents, where $\mathbf{x}_t^i$ is a state vector of agent $i \in N$, consisting of loads $x_t^{i,k}$ for m classes. We use the following definitions.

***Definition** 2.* n agents of a network are said to reach a *consensus* at time t if $\mathbf{x}_t^i = \mathbf{x}_t^j \quad \forall i, j \in N, i \ne j$.

***Definition** 3.* n agents are said to achieve *asymptotic mean square ε-consensus* for $\varepsilon > 0$ when

$$\overline{\lim}_{t\to\infty}\mathbb{E}\|\mathbf{x}_t^i - \mathbf{x}_t^j\|^2 \le \varepsilon.$$

To ensure balanced loads across the network (e.g., in order to increase the overall throughput of the system and to reduce the execution time), it is natural to use a redistribution protocol over time. We assume that to form the control (redistribution) strategy each agent $i \in N$ has noisy observations about its neighbors' states

$$\mathbf{y}_t^{i,j} = \mathbf{x}_t^j + \mathbf{w}_t^{i,j},\ j \in N_t^i, \tag{5}$$

where $\mathbf{w}_t^{i,j}$ is a noise vector.

3. CONTROL PROTOCOL

In Amelina et al. (2015), properties of a control algorithm, called local voting protocol, were studied for stochastic networks in the context of load balancing problem. For each agent the control (amount of redistributed tasks) was determined by the weighted sum of differences between the information about the state of the agent and the information about its neighbors' states. The local voting was used in Amelina et al. (2015) to achieve consensus for single class. In this paper, we consider a multi-class network and we aim to achieve consensus across the network for every class (i.e. we want $x_t^{i,1},\ i = 1 \ldots n$ to reach consensus,... $x_t^{i,k},\ i = 1 \ldots n$ to reach consensus,... $x_t^{i,m},\ i = 1 \ldots n$ to reach consensus, which may be different for different k, $k = 1 \ldots m$). Let us consider a protocol as follows. We define

$$u_t^{i,k} = \gamma^k p_{av}^{i,k} \sum_{j\in \bar{N}_t^i} b_t^{i,j,k}(y_t^{i,j,k} - x_t^{i,k}), \tag{6}$$

where $\gamma^k > 0$ is a step-size of the control protocol and $\bar{N}_t^i \subset N_t^i$ is the neighbor set of agent i (note, that we could use not all the available connections, but some subset of them), $b_t^{i,j,k}$ are protocol coefficients.

Let $B_t^k = [b_t^{i,j,k}], k = 1, \dots, m$ be the matrices of task redistribution protocol for every time instant t. (We set $b_t^{i,j,k} = 0$ when $a_t^{i,j} = 0$ or $j \notin \bar{N}_t^i$.) The corresponding graph $\mathcal{G}_{B_t^k}$ may have the same topology as graph $\mathcal{G}_{A_t}$ of matrix A_t or more poor.

The dynamics of the closed loop system with protocol (6) is as follows: for $k = 1, \dots m$, $i \in N$

$$x_{t+1}^{i,k} = x_t^{i,k} - \tilde{r}_t^{i,k} + \tilde{z}_t^{i,k} + \gamma^k \sum_{j \in \bar{N}_t^i} b_t^{i,j,k}(y_t^{i,j,k} - x_t^{i,k}) =$$

$$x_t^{i,k} - \tilde{r}_t^{i,k} + \tilde{z}_t^{i,k} + \gamma^k \sum_{j \in \bar{N}_t^i} b_t^{i,j,k} x_t^{j,k} - \gamma^k deg_i^-(B_t^k) x_t^{i,k} + \gamma^k \tilde{w}_t^{i,k}, \tag{7}$$

where $\tilde{w}_t^{i,k} = \sum_{j=1}^n b_t^{i,j,k} w_t^{i,j,k}$ and $\tilde{r}_t^{i,k} = p_t^{i,k}/p_{av}^{i,k}, \tilde{z}_t^{i,k} = z_t^{i,k}/p_{av}^{i,k}$.

Let us rewrite Eq. (7) in a more compact form. Define the $\mathbb{R}^n$-valued vectors $\mathbf{X}_t^k = [x^{i,k}]$, $\mathbf{R}_t^k = [\tilde{r}_t^{i,k}]$, $\mathbf{Z}_t^k = [\tilde{z}_t^{i,k}]$ and $\mathbf{W}_t^k = [\tilde{w}_t^{i,k}]$. The dynamics of the closed loop system with protocol (6) may be represented as

$$\mathbf{X}_{t+1}^k = \mathbf{X}_t^k + \gamma^k (B_t^k - \mathcal{D}(B_t^k)) \mathbf{X}_t^k - \mathbf{R}_t^k + \mathbf{Z}_t^k + \gamma^k \mathbf{W}_t^k.$$

Due to the view of Laplacian matrices $\mathcal{L}(B_t^k)$ we can rewrite the dynamics of the system in the following vector-matrix form:

$$\mathbf{X}_{t+1}^k = \mathbf{X}_t^k - \gamma^k \mathcal{L}(B_t^k) \mathbf{X}_t^k - \mathbf{R}_t^k + \mathbf{Z}_t^k + \gamma^k \mathbf{W}_t^k. \tag{8}$$

4. MAIN RESULTS

4.1 Assumptions

Let $(\Omega, \mathcal{F}, P)$ be the underlying probability space corresponding to the sample space, the collection of all events, and the probability measure respectively.

Assume that the following conditions are satisfied:

A1 **a)** For all $i \in N, j \in \bar{N}_t^i, k = 1, \dots, m$, observation noises $w_t^{i,j,k}$ are zero-mean, independent identically distributed (i.i.d.) random variables with bounded variances: $\mathbb{E}(w_t^{i,j,k})^2 \le \sigma_{w,k}^2$.

b) Graphs $\mathcal{G}_{B_t^k}$, $k = 1, \dots, m$, $t = 0, \dots$ are i.i.d. (independent identically distributed), i.e. the random events of appearance of "time-varying" edge (j, i) in graph $\mathcal{G}_{B_t^k}$ are independent and identically distributed for the fixed pair (j, i), $i \in N, j \in N_{\max}^i = \cup_t \bar{N}_t^i$. For all $i \in N$, $j \in N_t^i$ weights $b_t^{i,j,k}$ in the control protocol are independent random variables with mean values (mathematical expectations): $\mathbb{E} b_t^{i,j,k} = b_{av}^{i,j,k}$, and bounded variances: $\mathbb{E}(b_t^{i,j,k} - b_{av}^{i,j,k})^2 \le \sigma_{b,k}^2$.

c) For all $k = 1, \dots, m$, $i \in N$, $t = 0, 1, \dots$ random values $z_t^{i,k}$ are independent with expectations: $\mathbb{E} z_t^{i,k} = \bar{z}^k$ which do not depend on i, and bounded variances: $\mathbb{E}(z_t^{i,k} - \bar{z}^k)^2 \le \sigma_{z,k}^2$.

d) For all $i \in N$, $k = 1, \dots, m$, $t = 0, 1, \dots$ random vectors $\mathbf{p}_t^i$ are i.i.d. and consist of independent components. Random values $\tilde{r}_t^{i,k}$, $k = 1, \dots, m$, have expectations: $\mathbb{E} \tilde{r}_t^{i,k} = \bar{r}^k$ and bounded variations: $\mathbb{E}(\tilde{r}_t^{i,k} - \bar{r}^k)^2 \le \sigma_{r,k}^2$ which do not depend on i.

Additionally, all mentioned in Assumption **A1** independent random variables and vectors are mutually independent.

A2 Graphs $\mathcal{G}_{B_{av}^k}$ have a spanning tree (for the consensuses to be achievable throughout the system Chebotarev and Agaev (2009)).

A3 For step-sizes γ^k, $k = 1 \dots m$ of control protocols (6) the following conditions are satisfied:

$$0 < \gamma^k < \frac{1}{deg_{\max}^-(B_{av}^k)}, \ |\delta(\gamma^k)| < 1, \tag{9}$$

where $\delta(\gamma^k) = 1 - 2\gamma^k \lambda_2(\mathcal{L}(B_{av}^k)) + (\gamma^k)^2 \lambda_{max}\left(\mathbb{E}(\mathcal{L}(B_t^k)^T \mathcal{L}(B_t^k))\right)$.

4.2 Consensus achievement

Theorem 1. **If** Assumption **A2** holds **then** for any average cost constraints $\{c_k\}$, $c_k > 0$, there exists network topology decomposition $\{\mathcal{G}_{av}^k\}$ that satisfies the averaged cost constrains $\{c_k\}$ and for which all graphs $\mathcal{G}_{B_{av}^k}$ have spanning trees.

Proof 1. The proof is given in Amelina et al. (2014b).

Theorem 2. **If** Assumptions **A1**–**A3** hold **then** for averaged squared difference $\nu_t^k = \mathbf{X}_t^k - \mathbf{X}_t^{\star,k}$ of trajectory of closed-loop system (8) and $\mathbf{X}_t^{\star,k} = \mathbf{1}_n \otimes \frac{1}{n}\sum_{i=1}^n \mathbf{X}_t^{i,k}$ the following inequality is satisfied:

$$\mathbb{E}\|\nu_t^k\|^2 \le \frac{J_k(\gamma^k)^2 + K_k}{S_k\gamma^k - V_k(\gamma^k)^2} +$$
$$+ \left(\nu_0^k - \frac{J_k(\gamma^k)^2 + K_k}{S_k\gamma^k - V_k(\gamma^k)^2}\right)(1 - S_k\gamma^k + V_k(\gamma^k)^2)^t, \tag{10}$$

i.e. **if** additionally $\nu_0^k < \infty$, **then** the asymptotic mean square ε^k-consensus in (7) is achieved with $\varepsilon^k = \frac{J_k(\gamma^k)^2 + K_k}{S_k\gamma^k - V_k(\gamma^k)^2}$. Here $\mathbf{1_n} \in \mathbb{R}^n$, consisting of 1 at all places and $\otimes$ is Kronecker product, $J_k = \sigma_{w,k}^2 \|B_{av}^k\|^2$, $K_k = (n-1)(\sigma_{z,k}^2 + \sigma_{r,k}^2)$, $S_k = 2\lambda_2(\mathcal{L}(B_{av}^k))$, $V_k = \lambda_{max}\left(\mathbb{E}(\mathcal{L}(B_t^k)^T \mathcal{L}(B_t^k))\right)$.

Proof 2. The proof is similar to the proof of Theorem 1 in Amelina et al. (2014a).

Theorem 2 shows that queues with different priorities achieve m different consensus levels separately. This behavior is termed as *differentiated consensuses.*

Remark 1. To achieve the system convergence certain assumptions have to be met **A1**–**A3**. Assumptions **A1** bound mathematical expectations and variances of the random variables in order to make the resulting divergence ε^k between agents' states bounded. If assumption **A2** is not met consensus would not be achieved among all agents in the network. (But only within connected components of network topology graph.) Theorem 2 gives a conservative estimate of the divergence among agents' states. For the system to converge step-size value γ^k has to be bounded (**A3**). For the proposed estimate to converge the $|\delta(\gamma^k)|$ has to be bounded (**A3**). $\delta(\gamma^k) = (1 - S_k\gamma^k + V_k(\gamma^k)^2)$ in (10) and if **A3** is met $(\delta(\gamma^k))^t \xrightarrow[t\to\infty]{} 0$.

Theorem 3. **If** Assumptions **A1**–**A3** hold **then** optimal step-size $\gamma^{\star,k}$ of control protocol (6) can be calculated by formula:

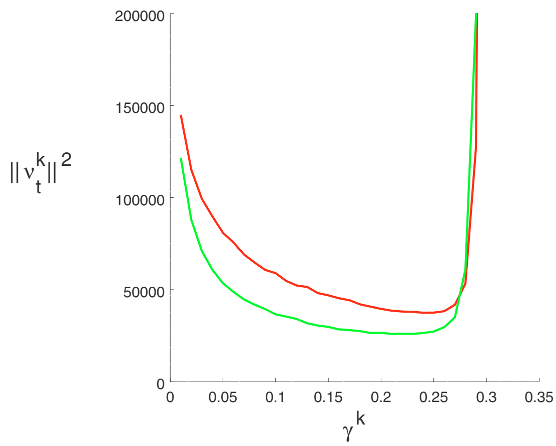

Fig. 1. Dependence of $\|\nu_t^k\|^2 = \|\mathbf{X}_t^k - \mathbf{X}_t^{\star,k}\|^2$ on γ^k in the system with cost constraints $c_1 = 2, c_2 = 2$.

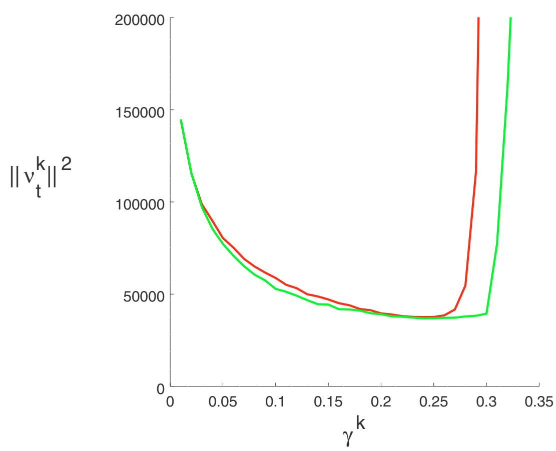

Fig. 2. Dependence of $\|\nu_t^k\|^2 = \|\mathbf{X}_t^k - \mathbf{X}_t^{\star,k}\|^2$ on γ^k in the system cost constraints $c_1 = 2, c_2 = 1$.

$$\gamma^{\star,k} = -\frac{K_k V_k}{J_k S_k} + \sqrt{\frac{(K_k V_k)^2}{(J_k S_k)^2} + \frac{K_k}{J_k}} \tag{11}$$

Proof 3. Formula (11) can be obtained by taking the derivative of $\varepsilon^k = \frac{J_k(\gamma^k)^2 + K_k}{S_k\gamma^k - V_k(\gamma^k)^2}$ with respect to γ^k.

Remark 2. In formula (11) all elements have K_k which depends on noise variance $\sigma^2_{w,k}$ in numerator and J_k in denominator which depends on variance of amount of incoming tasks $\sigma^2_{z,k}$. Therefore higher noise variance calls for smaller step-size value to achieve the system convergence. On the other hand a larger step-size value is needed to redistribute incoming tasks among the agents and keep the system balanced if variance of amount of incoming tasks is high. Optimality in Theorem 3 is understood in the sense of choosing such value of step-size that consensus in the system is reached with minimal mean-squared error under given conditions.

5. SIMULATION RESULTS

Let us consider a network of $n = 16$ agents connected as a undirected circle. The number of tasks coming to the system at time instant t is a Poisson random variable and distributed with parameter $\sigma_z = 200,\ k = 1, \ldots, m,\ m = 2.$

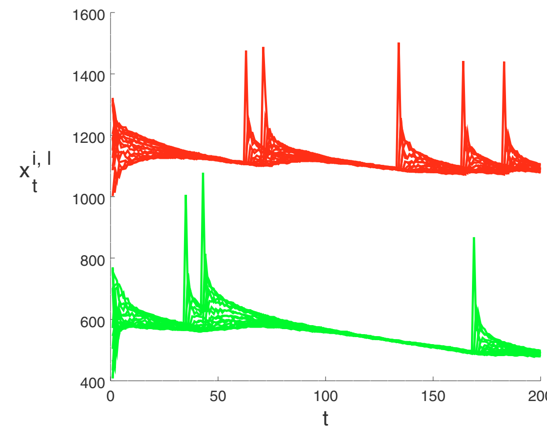

Fig. 3. Consensus achievement in the system cost constraints $c_1 = 2, c_2 = 2$.

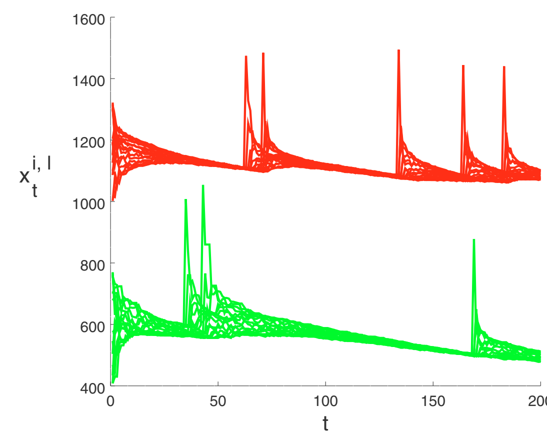

Fig. 4. Consensus achievement in the system cost constraints $c_1 = 2, c_2 = 1$.

Time between tasks arrival to the system is exponentially distributed random variable with parameter 50. The tasks are supposed to be atomic and unitary i.e. complexity of all tasks equals 1 computation unit. Agent average productivities $p^i_{av},\ i = 1 \cdots n$ are constant and have values distributed uniformly in interval $[0.5, 1.5]$. Productivity fractions are 1/2 and 1/2 for tasks of the first and the second priorities. Noise occurring during information exchange between agents $w_t^{i,j,k}$ is a random variable with uniform distribution on interval $[-0.1, 0.1]$. At time instant t_0 all agents have queue lengths distributed evenly in interval $[500, 700]$ for priority 1 tasks and $[200, 400]$ for tasks of priority 2. Step-size γ^k was chosen equal 0.2 for both priorities. Fig. 1, 2, 3, 4 show the results of simulation of two scenarios with different imposed cost constraints, $c_1 = 2, c_2 = 2$ (Fig. 1, 3) and $c_1 = 2, c_2 = 1$ (Fig. 2, 4). Since the network topology graph is undirected ring, each node has two incident edges and $deg^-_{\max} A = 2$. Consequently, in the first simulation scenario the cost constraints equal the maximum cost of the network and the agents are not restricted in usage of the network links. In the second scenario agents are restricted in exchange of tasks of priority 2. If the links are used with probability 1/2 the cost constraint is met.

Fig. 1, 2 illustrate the dependence of consensus achievement precision on step-size value for different priorities. The graphs show the value of squared difference $\|\mathbf{X}_t^k - \mathbf{X}_t^{\star,k}\|^2$ in the system operating by protocol (6) after 20 iterations. When the system is under the cost constraints the agents' loads converge to the consensus value with lower precision. That corresponds to higher position of the green graph on Fig. 4. Also feasible step-size interval slightly broadens. (In case of changing or unknown parameters of system a search stochastic approximation algorithms with input randomization could be used to choose the value of step-size Granichin and Amelina (2015); Amelina et al. (2016)). Fig. 3, 4 show the consensus achievement in the system. Agents' loads converge to average value. New tasks coming to one of the agents disrupt the balance until agents' loads reach the new consensus value. In second case the rate of consensus achievement among agents' loads is smaller since some of the available links are not used in order to meet the cost constraint.

6. CONCLUSION

The problem of load balancing in the network processing tasks of different priorities could be addressed by achieving a consensus among agents' loads in the system for each task priority. Optimal convergence to to the consensus value could be achieved by choosing the corresponding step-size value. In our previous works we proposed an estimate of divergence of agents' loads values in the system operating by local voting protocol. We also proposed a way to minimize the divergence estimate by choosing the optimal step-size value of the protocol in the system with given parameters. If the imposed averaged cost constraints on the network are satisfied by randomizing the links usage (with the probability corresponding to the cost constraint), the previously obtained results remain correct.

REFERENCES

Amelina, N., Fradkov, A., Jiang, Y., and Vergados, D. (2015). Approximate consensus in stochastic networks with application to load balancing. *IEEE Transactions on Information Theory*, 61(4), 1739–1752.

Amelina, N., Granichin, O., Granichina, O., Ivanskiy, Y., and Jiang, Y. (2014a). Differentiated consensuses in a stochastic network with priorities. In *Intelligent Control (ISIC), 2014 IEEE International Symposium on*, 264–269. IEEE.

Amelina, N., Granichin, O., Granichina, O., Ivanskiy, Y., and Jiang, Y. (2016). Using stochastic approximation type algorithm for choice of consensus protocol step-size in changing conditions. *IFAC-PapersOnLine*, 49(13), 265–269.

Amelina, N., Granichin, O., Granichina, O., and Jiang, Y. (2014b). Differentiated consensuses in decentralized load balancing problem with randomized topology, noise, and delays. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, 6969–6974. IEEE.

Bullo, F., Cortés, J., and Martinez, S. (2009). *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press.

Chebotarev, P.Y. and Agaev, R.P. (2009). Coordination in multiagent systems and laplacian spectra of digraphs. *Automation and Remote Control*, 70, 469–483.

Granichin, O. and Amelina, N. (2015). Simultaneous perturbation stochastic approximation for tracking under unknown but bounded disturbances. *IEEE Transactions on Automatic Control*, 60(5).

Granichin, O., Khantuleva, T., and Granichina, O. (2017). Local voting protocol for the adaptation of airplane's "feathers" in a turbulence flow. In *American Control Conference (ACC), 2017*, 5684–5689. IEEE.

Granichin, O., Skobelev, P., Lada, A., Mayorov, I., and Tsarev, A. (2012). Comparing adaptive and non-adaptive models of cargo transportation in multi-agent system for real time truck scheduling. *Proceedings of the 4th International Joint Conference on Computational Intelligence*, 282–285.

Huang, M. (2012). Stochastic approximation for consensus: a new approach via ergodic backward products. *IEEE Transactions on Automatic Control*, 57(12), 2994–3008.

Jiang, Y., Tham, C.K., and Ko, C.C. (2002). A probabilistic priority scheduling discipline for multi-service networks. *Computer Communications*, 25(13), 1243–1254.

Kar, S. and Moura, J.M. (2010). Distributed consensus algorithms in sensor networks: Quantized data and random link failures. *Signal Processing, IEEE Transactions on*, 58(3), 1383–1400.

Lewis, F., Zhang, H., Hengster-Movric, K., and Das, A. (2014). *Cooperative Control of Multi-Agent Systems: Optimal and Adaptive Design Approaches (Communications and Control Engineering)*. Springer.

Li, T. and Zhang, J. (2009). Mean square average-consensus under measurement noises and fixed topologies: Necessary and sufficient conditions. *Automatica*, 45(8), 1929–1936.

Olfati-Saber, R. and Murray, R.M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9), 1520–1533.

Proskurnikov, A.V. (2013). Average consensus in networks with nonlinearly delayed couplings and switching topology. *Automatica*, 49(9), 2928–2932.

Ren, W. and Beard, R. (2007). *Distributed consensus in multi-vehicle cooperative control: theory and applications*. Springer.

Ren, W., Beard, R., and Atkins, E. (2007). Information consensus in multivehicle cooperative control. *Control Systems, IEEE*, 27(2), 71–82.

Virágh, C., Vásárhelyi, G., Tarcai, N., Szörényi, T., Somorjai, G., Nepusz, T., and Vicsek, T. (2014). Flocking algorithm for autonomous flying robots. *Bioinspiration & biomimetics*, 9(2), 25012–25022.

Yu, W., Chen, G., and Cao, M. (2010a). Distributed leader–follower flocking control for multi-agent dynamical systems with time-varying velocities. *Systems & Control Letters*, 59(9), 543–552.

Yu, W., Chen, G., and Cao, M. (2010b). Some necessary and sufficient conditions for second-order consensus in multi-agent dynamical systems. *Automatica*, 46(6), 1089–1095.