2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Information Broadcasting Algorithm for Finite-Time Reaching-at-Risk Consensus with Application to Weapon-Target Assignment

N. Léchevin, C.A. Rabbath, and Y. Zhang

Abstract— We propose a randomized algorithm aimed at reaching, in finite time, exact consensus among a set of agents that are linked though a connected, possibly time-varying, graph. The information exchanged among neighboring agents is limited in size, by randomly selecting the content from an agent internal state. The time needed to reach the agreement is a random variable, whose empirical cumulative distribution function is utilized to determine a stopping rule that ensures consensus is achieved with a prescribed confidence level. Simulation results show that the consensus-reaching time obtained with a prescribed confidence level compares relatively well to local-averaging-based algorithms. The algorithm is shown to be robust, to some extent, to lossy communications and is demonstrated on a weapon-target assignment problem.

I. Introduction

Motivated by the desire to implement distributed cooperative optimization schemes in realistic operating conditions, an agreement-reaching algorithm is proposed to ensure consistency of the state variables exchanged among agents despite limited communications bandwidth. Distributed cooperative optimization may be needed for weapon-target assignment strategies and multiagent path planning [1]. Typically, such strategies require that each agent solves a local optimization problem using information obtained from neighboring agents. Inter-agent communications are needed to reach a so-called agreement or consensus. Consensus ensures that the action taken by any agent in the team is consistent with that of its neighbors. Algorithms that enable such consensus reaching for a set of agents belong typically to two classes: Distributed Averaging-Based Algorithms (DABAs) [2], [3], [4], and probabilistic counting approaches, such as randomized gossip mechanisms [5]. Applications of consensus-reaching algorithms include coordination of groups of vehicles [6], and belief propagation in network of sensors [7]. See [2] for an extended bibliography on the literature on consensus.

Most algorithms yield asymptotic convergence to a consensus value. Yet, three approaches have been recently proposed to ensure finite-time convergence to consensus. The first one is based on normalized and signed gradient descent flows [8]. It leads to nonsmooth dynamics. Discrete-time implementation of such algorithms may thus exhibit chattering when the states are in the vicinity of the consensus value. A second approach is derived by calculating weights in DABAs so that the observability matrix of the network is contained in some defined space [9]. Such strategy is time optimal in fixed-topology communication graphs. Weight computation necessitates that the topology of the graph be fixed and that an initialization phase be carried out before the consensus reaching phase. Robustness to intermittent losses of communication links and of computing nodes is not guaranteed, however. A third approach leverages properties of de Bruijn's graph and block Kronecker product to show that the average consensus can be reached in finite time if the number of nodes is an exact power of the maximum in-degree of the graph [10].

We propose a discrete-time, probabilistic algorithm, labeled as Finite-Time, Reaching-at-Risk Consensus (FTRaRC), that allows reaching, in finite time and in a distributed manner, the exact value of an average-based agreement among connected agents with a prescribed level of confidence (LoC). Our results have to be contrasted with those of [8]. Indeed, the proposed FTRaRC is not a discretization of a continuous-time algorithm, the real-time implementation of FTRaRC is free of chattering, and the performance of FTRaRC is independent of the sampling period. We show by means of simulations that FTRaRC is able to reach consensus even though the communication graph is temporarily unconnected. Contrary to DABAs, FTRaRC does not require agents to update and communicate a local estimate of the consensus, but rather entails aggregation of the information of all agents and then computes the consensus once the integral information is available to every agent. The amount of data sent by each agent is kept constant during each communication event, and is selected randomly from an internal state, which stores the information relevant to the mission of interest. Such randomness motivated us to employ the expression reaching-at-risk. The proposed FTRaRC, however, requires additional memory when compared to DABAs, which makes it unsuitable for relatively large networks of agents, and for consensus that involves interagent exchanges with high-dimension matrices.

II. Definitions, Assumptions, and Objectives

A set $\mathcal{F}$ of $p \in N^+$ agents, F_i, where $i \in \{1, ..., p\} = B$, is considered. Let $\mathcal{E}_B$ represent the set of all pairs of

N. Léchevin is with Numerica Technologies, Inc., Québec, Qc, Canada, and with the Dpt of MIE, Concordia University, Montréal, Qc, Canada. nicolas.lechevin.numerica@drdc-rddc.gc.ca.

C.A. Rabbath is with DRDC Valcartier, 2459 Pie-XI N., Québec, Qc, Canada G3J 1X5. camille-alain.rabbath@drdc-rddc.gc.ca.

Y. Zhang is with the Dpt of MIE, Concordia University, Montréal, Qc, Canada. ymzhang@encs.concordia.ca.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

agents F_i and F_j such that F_i communicates with F_j, and vice versa. A neighbor of F_i is thus defined as the agent F_j such that $(F_i, F_j) \in \mathcal{E}_B$. The communication graph $\mathcal{G}_c = (\mathcal{F}, \mathcal{E}_B)$ is assumed time invariant, undirected, and connected, although not necessarily complete. The terms agent and graph node are taken as synonyms in this paper. $\mathbf{0}_p$ and $\mathbf{1}_p$ stand for the $1 \times p$ zero and unitary vectors, respectively.

Assumption 1 (Information available to F_i) At the initial time, $M_{1,i,1}$ and $M_{2,i,1}$ are available to F_i, $i \in \{1, ..., p\}$. $M_{2,i,1}$ is either a scalar, a vector or a matrix, which is utilized to compute the consensus $(1/p)\sum_{i=1}^{p} M_{2,i,1}$. $M_{1,i,1}$ represents an identity number, to which $M_{2,i,1}$ is associated.

Assumption 2 (Communications). The time is discretized as t_k, where $t_1 < ... < t_k < t_{k+1} < ...$ for all $k \in N^+$. During time interval $[t_k, t_{k+1})$, which is not necessarily periodic, each F_i updates and communicates $\{M_{1,i,k}, M_{2,i,k}\}$ to its neighbors. To comply with usual communication bandwidth constraints, it is assumed that the amount of data exchanged between any pair of agents over $[t_k, t_{k+1})$ during the entire consensus-reaching process has a fixed size given by the dimensions of $M_{1,i,k}$ and $M_{2,i,k}$.

Although we assume a time-invariant $\mathcal{G}_c$, numerical simulations include intermittent losses of communications. The results of Section IV indicate that FTRaRC displays a certain level of robustness to intermittent losses of communications.

Information Consensus Objective. From Assumptions 1 and 2, find a consensus reaching-at-risk mechanism that is distributed over the set of agents F_i, $i \in \{1, ..., p\}$, and that relies on the exchange of information $\{M_{1,i,k}, M_{2,i,k}\}$ through $\mathcal{G}_c$ such that, given a prescribed LoC, every agent $F_i \in \mathcal{F}$ is able to compute, in a finite number of iterations, the exact consensus value defined as $(1/p)\sum_{i=1}^{p} M_{2,i,1}$. Note that although NEWTAS motivated us to focus on average consensus, the proposed FTRaRC remains valid for consensus that involves any function of $M_{2,i,1}$.

III. Consensus Algorithm

Most information consensus algorithms proposed in the last few years ensure asymptotic convergence, in the time domain, of the variables shared through a connected, possibly time-varying, graph $\mathcal{G}_c$ [2], [3]. The reader is referred to Fig. 1(a). Information state $I_{i,k}$ of F_i at iteration step k is usually updated by means of the following rule (see (15) in [2])

$$I_{i,k} = I_{i,k-1} + \varepsilon \sum_{l \in N_i} (I_{l,k-1} - I_{i,k-1}), \tag{1}$$

where N_i denotes the set of neighbors of i, and ε is the sample period associated with the discrete-time consensus algorithm. Such rules enable each node's information state to approach $(1/p)\sum_{i=1}^{p} I_{i,1}$ as $t \to \infty$.

FTRaRC is based on the following observation. Assume that the size of data communicated among each pair of adjacent nodes in $\mathcal{G}_c$ is allowed to vary, as opposed to Assumption 2. An agent stores, at each t_k, the data communicated from its neighbors and publishes, at t_{k+1}, the content of its internal memory. It is straightforward to notice that the connectedness of the communication graph implies that a consensus will be reached after a few iterations since all agents are in possession of the same content. To see this, consider the graph of Fig. 1(a), which is composed of 8 agents. Assume that one aims at agreeing about the mean value of a variable I_i indexed by $i = \arg(F_i)$, which is the identity of F_i. At iteration step $k = 0$, F_i's internal memory, M_i, is set to i. At $k > 0$, F_i communicates to its neighbors the content of M_i and the set of all I_*, where index $*$ belongs to M_i. It can be easily shown that consensus is reached after four steps; i.e., $M_i = B$ is obtained, at $k = 4$, for all $i \in \{1, ..., 8\}$. Each agent is thus able to compute $(1/8)\sum_{k=1}^{8} I_k$. To satisfy Assumption 1, we propose to constrain the size of the amount of data that is communicated by F_i to its neighbors, and to randomly select such information from M_i, as explained in the sequel. Let $u_{1,i,k}^{j}$ and $u_{2,i,k}^{j}$ be the input variables of F_i communicated by $F_j \in N_i$ at $t_k \in N^+$, where N_i stands for the set of neighbors of F_i. The input variable $u_{1,i,k}^{j}$ carries the identity of only one of the p agents selected randomly, whereas $u_{2,i,k}^{j}$ carries the state that is eventually used to compute the agreement. FTRaRC (Fig. 1(b)) can be described as a 3-step algorithm as follows.

Step 1. (Initialization) For each agent F_i set, at time t_1,

$$M_{1,i,1} = i \ (= \arg F_i), \ M_{2,i,1} = I_{i,1}. \tag{2}$$

$I_{i,1}$ denotes the information available to F_i at t_1.

Step 2. (Iterations)

Task 2.1. The interconnection of F_i's inputs with the output of neighbor $F_j \in N_j$ is described, at $t_k > t_1$, by

$$\begin{aligned} u_{1,i,k}^{j} &= y_{1,j,k-1}, \ U_{1,i,k} = \{u_{1,i,k}^{j}, j \in N_i\}, \\ u_{2,i,k}^{j} &= y_{2,j,k-1}, \ U_{2,i,k} = \{u_{2,i,k}^{j}, j \in N_i\}. \end{aligned} \tag{3}$$

Task 2.2. The internal memory $M_{1,i,k}$ and $M_{2,i,k}$ of F_i are updated as follows

$$\begin{aligned} M_{1,i,k} &= M_{1,i,k-1} \cup U_{1,i,k}, \\ M_{2,i,k} &= M_{2,i,k-1} \cup U_{2,i,k}. \end{aligned} \tag{4}$$

Task 2.3. The fixed size in the amount of data communicated between F_i and the set of its adjacent agents is guaranteed by determining the signals sent by F_i as

$$\begin{aligned} y_{1,i,k} &= \mathrm{rand}(\underbrace{M_{1,i,k} \backslash Y_{1,i,k-1}}_{\mathcal{M}_i}) = i^*, \\ y_{2,i,k} &= I_{i^*,1}, \end{aligned} \tag{5}$$

where $Y_{1,i,k-1} = \{y_{1,i,1}, ..., y_{1,i,k-1}\}$. The randomized selection of the output in (5) is defined as follows

$$\begin{aligned} & y_{1,i,k} = \mathrm{rand}(\mathcal{M}_i) \\ & = \begin{cases} j & \text{if } \mathcal{M}_i = (0, ..., 0, j, 0, ..., 0), \\ i_l & \text{if } \mathcal{M}_i = (i_1, ..., i_{p'}),\ p' \leq p, \\ & \text{and } i_l \neq j, \end{cases} \\ & p(y_{1,i,k} = i_l) = p_{ii_l},\ \forall l \in \{1, ..., p'\}, \\ & \textstyle\sum_{l=1}^{p'} p_{ii_l} = 1,\ p_{ii_l} > 0, \end{aligned} \tag{6}$$

where subscript i refers to F_i.

Step 3. (Termination condition) The consensus is reached at $t_{k'}$, when $k^{'} \in N^+$ is such that the equality

$$M_{1,i,k'} = B \tag{7}$$

is satisfied for all $i \leq p$. Hence, all of the agents agree on the same information state computed as

$$M_{2,i,k} = \frac{1}{p} \sum_{i=1}^{p} I_{i,k'}. \tag{8}$$

Proposition 1. Supposing Assumption 1 and Assumption 2 hold and that $\mathcal{G}_c$ is a connected, time-invariant graph, then there exists $k^{'} \in N^+$ such that the FTRaRC expressed in (3)-(8) satisfies $P(M_{1,i,k'} = B) > 0$ for all $i \in B$; i.e., the probability that consensus is reached in finite time is positive. ⋈

Proof. From Assumption 1, there exists a set $r_{ij}(\kappa) = \{\{F_{i_1}, ..., F_{i_{\kappa-1}}\} \in \mathcal{F}^{\kappa-1};\ (F_i, F_{i_1}) \in \mathcal{E}_B, ..., (F_{i_j}, F_{i_l}) \in \mathcal{E}_B, ..., (F_{i_{\kappa-1}}, F_j) \in \mathcal{E}_B,\ j, l < \kappa - 1\}$, where $\kappa \leq p$. Denote $\kappa^*_{ij} = \underset{r_{ij}(\kappa_{ij}) \neq \emptyset}{\arg\min}\, \kappa_{ij}$ and $r^*_{ij} = r_{ij}(\kappa^*_{ij})$ the set of shortest paths from F_i to F_j, which exists since $\mathcal{G}_c$ is connected. Note that r^*_{ij} is not necessarily a singleton. Therefore, the probability that the identity i $(= M_{1,i,1})$ of F_i at $k = 1$ and $M_{2,i,1}$ reach any node $F_j \in r^*_{ij}$ after κ^*_{ij} iterations of (3)-(8) is given by

$$\begin{aligned} & P(i \in F_j, F_j \in r^*_{ij}) \\ & = 1 - \prod_{(F_{i_1}, ..., F_{i_{\kappa^*_{ij}-1}}) \in r^*_{ij}} (1 - p_{i,i_1} p_{i_1,i_2} ... p_{i_{\kappa^*_{ij}-1},j}), \end{aligned} \tag{9}$$

where p_{i_j,i_l} stands for the probability that variable i $(= M_{1,i,1})$ is communicated to i_l by i_j. By construction of the algorithm, $p_{i_j,i_l} > 0$, which in turn implies that $P(i \in F_j, F_j \in r^*_{ij}) > 0$. Denote $\kappa^*_i = \max_{j \in B, j \neq i} \kappa^*_{ij}$ and $r^*_i = \{r^*_{ij}, j \in B \backslash \{i\}\}$. Thus for all $\kappa' \geq \kappa^*_i$ and for all $F_j \in \mathcal{F} \backslash \{F_i\}$, (9) implies that $P(i \in F_j, F_j \in r^*_i) > 0$. Letting $k^{''} = (p-1) \max_{i \in B} \kappa^*_i$ yields $P(M_{1,i,k''} = B) > 0$ for a given $i \in B$. The factor $(p-1)$ in k'' comes from the communication constraint in Assumption 2. Similarly, letting $k' = pk''$ leads to $P(M_{1,i,k'} = B) > 0$ for all $i \in B$. Expression of k' is conservative since, depending on $\mathcal{G}_c$ topology, $M_{1,i,k''} = B$ and $M_{1,j,k''} = B$ may occur for some $i, j \in B$. ⋈

Next, Proposition 1 is extended to a class of jointly connected time-varying graphs $\mathcal{G}_c(t)$ [6].

Assumption 3. The p agents of B are assumed to be linked together across time intervals $[\tau_\nu, \tau_{\nu+1})$, $\nu \in N^+$where $\tau_{\nu+1} - \tau_\nu \in (0, T]$ for some fixed $T > 0$; i.e., the time-varying communication graph $\mathcal{G}_c(t)$ is a jointly connected for all $t \in [\tau_\nu, \tau_{\nu+1})$. $\mathcal{G}_c(t)$ is jointly connected over $[\tau_\nu, \tau_{\nu+1})$ if for all $m_\nu \in N^+$ time instants $t_k \in [\tau_\nu, \tau_{\nu+1})$, $\mathcal{G}_c(t_k)$ is an element of $\{G_1, ..., G_{m_\nu}; G_i = (\mathcal{F}, \mathcal{E}_i), i = 1, ..., m_\nu\}$, whose corresponding graph union $\mathrm{G} = \{\mathcal{F}, \cup_{i=1}^{m_\nu} \mathcal{E}_i\}$ is connected.

Proposition 2. Supposing that Assumptions 1, 2, and 3 hold, then there exists $K \in N^+$ such that $P(M_{1,i,K} = B) > 0$ for all $i \in B$. ⋈

Proof. The proof is similar to that of Proposition 1. Indeed, when $\mathcal{G}_c$ is time invariant, the transition of the identity variable i between two connected nodes depends on the probability that i is selected, for communication purpose, by one of the two nodes, which can be carried out at least in one iteration step. With Assumption 3, two nodes may be temporarily unconnected. The set of paths linking two nodes over which the transition probability in (9) has to be computed is no longer fixed and thus no longer known a priori but is built through a series of time intervals $[\tau_\nu, \tau_{\nu+1})$. Assumption 3 implies that there exits $\overline{\nu}_{ij} \in N^+$ such that F_i and F_j are linked together over each interval $[\tau_o, \tau_{o+1})$ with $o = 1, ..., \overline{\nu}_{ij} - 1$. Suppose without loss of generality that at least one communication occurs in $[\tau_o, \tau_{o+1})$. Denote $t_{k(o)}$ as the occurrence time of the communication. By virtue of Assumption 3, there is a positive probability $p_{i_{t_{k(o)}}, i_{t_{k(o+1)}}}$ that the identity variable i is communicated from $F_{k(o)}$ to $F_{k(o+1)}$, where $F_{t_{k(o)}}$ is the agent that broadcasts i at $t_{k(o)}$. It is thus possible to find a sufficiently large integer $\overline{\nu}_{ij}$ such that $F_j = F_{t_{k(\overline{\nu}_{ij})}}$ and $p_{i,i_{t_{k(1)}}} p_{i_{t_{k(1)}}, i_{t_{k(2)}}} ... p_{i_{t_{k(\overline{\nu}_{ij}-1)}}, j} > 0$. The probability that i reaches any $F_j \in \mathcal{F} \backslash \{F_i\}$ from F_i in K_{ij} steps that is greater than or equal to $\overline{\nu}_{ij}$ is bounded from below by $p_{i,i_{t_{k(1)}}} p_{i_{t_{k(1)}}, i_{t_{k(2)}}} ... p_{i_{t_{k(\overline{\nu}_{ij}-1)}}, j}$. Letting $K_i = \max_{j \in B \backslash \{i\}} K_{ij}$, there is a positive probability that $i \in F_j$ for all $j \in B \backslash \{i\}$. Adopting $K = p(p-1) \max_{j \in B \backslash \{i\}} K_{ij}$, one obtains $P(M_{1,i,K} = B) > 0$ for all $i \in B$. ⋈

Remark 1. The algorithm in (3)-(8) reaches an exact agreement among the agents in finite time, although at the expense of a requirement on data storage. Indeed, each agent should be capable of storing p values of $M_{1,i,k}$ and $M_{2,i,k}$. While $M_{1,i,k}$ is an integer, storage requirement is likely to become a critical issue when, for instance, $M_{2,i,k}$ is a large matrix.

Remark 2. To implement the termination condition in Step 3, $\{M_{1,1,k}, ..., M_{1,i,k}, ..., M_{1,p,k}\}$, should be available to agent F_i. This is not usually the case as only $M_{1,i,k}$ is available to F_i. A possible solution consists of incorporating additional input, state, and output variables to (3)-(8); namely, $u^j_{2,i,k}$, $M_{3,i,k}$, and $y_{3,i,k}$, respectively. To do so, the following termination mechanism is proposed. First, initialize $M_{3,i,1}$ to $\mathbf{0}_p$. Second, set the first entry of $M_{3,i,k}$ to 1, for some $k \in N^+$, when $M_{1,i,k} = B$ or $M_{1,j,k} = B$, where j is such that $F_j \in N_i$. In the latter case, F_j communicates the information that $M_{1,j,k} = B$ by setting its output, $y_{3,j,k}$, to 1; $y_{3,j,k}$ is set to 0 otherwise. Updates of $M_{1,i,k}$ and $M_{2,i,k}$, $i = 1,...,p$, are computed by means of sequential executions of FTRaRC's Step 2. Updates are stopped when $M_{3,i,k} = \mathbf{1}_p$, which means that F_i knows that all of the agents in $\mathcal{F}\backslash\{i\}$ have reached the agreement. This proposed mechanism, however, increases the execution time.

Remark 3. To constrain execution time, set $k^{'}$ introduced in Step 3 to a specified value, which allows reaching agreement with a prescribed LoC δ; i.e., the probability that the agreement is not reached within $k^{'}$ iterations is less than or equal to $1 - \delta$. Stochasticity in the number of iterations necessary to reach the agreement depends on the random selection of $y_{1,i,k}$ in (5).

IV. Numerical Simulations

A. 8-node, 16-node, and 24-node Graphs

We implement (3)-(8) in F_i, $i = 1,...,8$. G_c is shown in Fig. 1(a). At $k = 1$, $M_{2,i,1}$ is initialized to $i\mathbf{1}_p$. The computation p_{ii_l} in (6) can be carried out, in an optimal sense, first, by expressing the objective as a total expected reward over a finite horizon, and then by solving the corresponding Bellman's equation [11], which is not shown due to lack of space. However, this computation is intricate when the communication graph is not trivially small. To avoid analyzing FTRaRC from scratch, Bellman's equation is solved for a simple 3-node tree. This step yields a uniform distribution, which is applied as a randomized, although not an optimal, policy over M_i in (6). The consensus, $M_{2,i,1} = 4.5\mathbf{1}_p$, for all $i \in B$, is reached at $t_{k'}$. The empirical frequency of the number of iterations k' that are necessary to reach consensus is computed after 1000 simulation runs (Fig. 2(a)). The cumulative distribution function (solid line) in Fig. 2(b) shows that a 95% LoC with FTRaRC is achieved with $k^{'} = 50$. Robustness of FTRaRC is tested with intermittent loss of communication links. Edges (F_2, F_5) and (F_3, F_5) are disabled when k is even, whereas edges (F_1, F_4), and (F_6, F_7) are disabled when k is odd. Despite the fact that the deactivation of (F_6, F_7) makes $\mathcal{G}_c$ temporarily unconnected, although $\{\mathcal{G}_{2k}, \mathcal{G}_{2k+1}\}$ is jointly connected, a consensus is reached for every simulation run. As expected, the cumulative distribution function (dashed line) in Fig. 2(b) shows that a 95% LoC with FTRaRC is achieved with a greater number of iterations ($k^{'} = 63$) when compared to the case of healthy communication links. Fig. 3 displays F_i's state $m_{2,i,k}$ ($M_{2,i,k} = m_{2,i,k}\mathbf{1}_p$), which is substituted for $I_{i,k}$ in (1). As shown in Fig. 3, the state of each agent reaches asymptotically the agreement $4.5 = (1/8)\sum_{i=1}^{8} i$. After 142 iterations, the worst relative error between multiagent state and the agreement value is less than 1%. By worst relative error, it is meant the error $F_{i^*} - 4.5$ obtained with F_{i^*}, where $i^* = \underset{i\in\{1,...,p\}}{\arg\max}(m_{2,i,1} - (1/p)\sum_{i=1}^{p} i)$. F_{i^*} corresponds to F_8. FTRaRC with a 95% LoC and a time-invariant graph reaches the consensus almost three times as fast as that obtained with DABA [2] with a 1% worst relative error. To achieve a LoC of 99%, the number of iterations increases to 67 (time-invariant graph) and to 79 (graph with communication losses). Furthermore, the simulations showed that the relative errors obtained with a time-invariant graph, and with 50 and 67 iterations of DABA are 15% and 8%, respectively. The application of FTRaRC, on the same time-invariant graph resulted in worst-case relative errors of 22% and 13% with an empirical frequency of 5% and 1%, respectively. The selection of one particular approach for consensus reaching thus depends on the application's constraints and on the amount of risk one is willing to tolerate. FTRaRC achieves exact consensus reaching with a prescribed LoC. A given LoC implies a certain number of iterations. With DABA, the relative error can be used as a stopping rule. In such case, a given relative error would imply a certain convergence time.

Consider a more complex case. Denote $\mathcal{G}_c^{(3)} = \{\mathcal{G}_c^{(2)}, \mathcal{G}_c\}$ and $\mathcal{G}_c^{(2)} = \{\mathcal{G}_c, \mathcal{G}_c\}$, where connection between $\mathcal{G}_c^{(2)}$ and $\mathcal{G}_c$ results from the bidirectional connection of F_1 in $\mathcal{G}_c$ and the only one-degree vertex F_8 of $\mathcal{G}_c^{(2)}$. The same applies to connection between $\mathcal{G}_c$ and $\mathcal{G}_c$. Table 1 summarizes the results obtained with FTRaRC and DABA applied to time-invariant graphs $\mathcal{G}_c$, $\mathcal{G}_c^{(2)}$, and $\mathcal{G}_c^{(3)}$, where $\mathcal{G}_c$ is depicted in Fig. 1(a). The average degree of $\mathcal{G}_c$, $\mathcal{G}_c^{(2)}$, and $\mathcal{G}_c^{(3)}$ are 2.25, 2.37, and 2.41, respectively. FTRaRC allows reaching in finite time the exact value of agreement with a prescribed LoC of 95% and of 99% for which the number of iterations is, in all three cases, less than that obtained with DABA [2], [3]. The ratio of consensus-reaching time (local averaging with respect to FTRaRC) is maintained within $[2.8, 3.0]$.

B. 100-node Ring Graph

A 100-node graph is taken from [2]. For such graph, each agent is connected to its two nearest neighbors. Execution of DABA yields the state evolution shown in Fig. 4(a), where 10498 and 6421 iterations are required so that the worst relative error is under 1% and 5%, respectively. The cumulative distribution function obtained with FTRaRC (Fig. 4(b)) shows that 5457 and 6012 iterations are necessary to achieve a LoC of 95% and 99%, respectively, without communications failures. Robustness of FTRaRC is tested with a periodic loss of a single communication link, which is the same throughout the simulation. The link is disabled whenever k is odd. Note that Assumption 3 holds. A zero-order-hold device is applied to DABA whenever the communication between two adjacent nodes is lost. With a time-varying graph, the cumulative distribution function is shifted to the right (Fig. 4(b)). To achieve a LoC of 95%, 6135 iterations are required, whereas 6575 iterations are necessary to achieve a LoC of 99%. When compared to the non-faulty case, such numbers correspond to increases of 12% (LoC of 95%) and 9% (LoC of 99%). Robustness of DABA (not shown) is more pronounced than with FTRaRC since the rate of convergence and the consensus value of DABA have marginally changed ($< 0.1\%$). It is to be noted that the worst performance obtained with FTRaRC (LoC of 99%) when communication is unreliable is similar to that of DABA (relative error of 5%) when performed with an ideal ring graph. As noticed in Remark 1, the price to pay for a reduced number of iterations with FTRaRC, when compared to DABA, is an increased requirement on the storage capability. Each agent should be capable of storing 100 values of $M_{2,i,k}$ and $M_{1,i,k}$; i.e., $[M_{2,1,1},...,M_{2,100,1}]$ and $[M_{1,1,1},...,M_{1,100,1}]$.

C. 100-node Random Geometric Graph

100 nodes are distributed uniformly at random on a plane. Two nodes are connected if their distance is less than or equal to $r > 0$. As shown in Fig. 5(a) with $r = 0.15$, this graph captures the realness of many applications such as a network of communication stations [12]. The average degree of the graph is 6.44 compared to 2 with a ring graph. The cumulative distribution function obtained with FTRaRC shows (Fig. 5(b)) that 1053 and 1346 iterations are necessary to achieve LoC of 95% and 99%, respectively, when internode communications are operative, compared to 131 (5% relative error) and to 312 (1% relative error) with DABA. Keeping in mind the results obtained with the ring graph, it is clear that DABA outperforms FTRaRC when the average degree of the graph increases. To test the robustness to random loss of nodes, at each iteration $k \in N^+$, a node is selected uniformly at random and disabled during time interval $[t_k, t_{k+1})$. Intermittent loss of a node represents malfunction of onboard electronic devices, which temporarily disables a node considered here as a communication station. 1149 and 1377 iterations are necessary to achieve a LoC of 95% and 99%, respectively, representing increases over the non-faulty communications cases equal to 9% and 2%, respectively. The same experiment has been conducted with the ring graph where consensus reaching has been noticed only when the failure is periodically triggered with a period T_{loss} greater than $15p$ iterations, where $p \in N^+$. While Assumption 3 still holds, the number of iteration steps K in Proposition 2 needed to obtain a positive probability of consensus is likely to be prohibitively high when $T_{\text{loss}} < 15p$. A possible explanation of the observed variability of FTRaRC's performance sensitivity regarding the type of graph may be attributed to the high average degree of the graph. The ring graph is unconnected at each iteration step, thus preventing consensus reaching for small K.

D. Application to NEWTAS

Consider a network of formations of unmanned combat vehicles (UCVs) which have to visit a set of targets so that a global utility function is maximized while complying to constraints on the autonomy of the UCVs. Each target has to be visited at least once. Formations of vehicles are allowed to split into smaller formations or to merge into larger formations in order to manage risk of being killed. The states of the formations are governed by a probabilistic attrition model. Decision making is the solution to a combinatorial optimization problem. We proposed in [1] a parallelized Cross-Entropy-Based Algorithm (CEMBA) to provide the UCVs with fast, near-optimal decision making. It consists of the parallelization of the cross-entropy method applied to a vehicle multirouting problem. Fig. 6 shows the architecture of the distributed target assignment, which is obtained by duplicating in every formation the process detailed in F_i's block. Every formation F_i executes FTRaRC in (3)-(8) and CEMBA sequentially. FTRaRC becomes the first step of CEMBA, whose transition matrix is computed iteratively and then broadcast as the first step of FTRaRC to F_j, where $j \in N_i$. The fast-rate subsystem is composed of FTRaRC operating at h_f. The slow-rate subsystem corresponds to CEMBA. The formation communicates at the fast rate. Once FTRaRC has reached a consensus about the entries of the transition matrix, CEMBA computes a new policy. Distributed weapon-target assignment complies with Assumption 2 since the consensus variable is a Markov chain transition matrix of the entire set of formations. Distributed NEWTAS is applied to networked formations whose connection graph is defined by $\{\{F_1, F_2, F_3, F_4\}, \mathcal{E}_B\}$, where $\mathcal{E}_B = \{(F_4, F_1),\ (F_1, F_2),\ (F_2, F_3)\}$. Each formation comprises 16 vehicles, with an autonomy of 240 units. The opposing team is composed of 15 targets randomly located over a square urban theater with edge length of 100 units. The utility assigned to each target is uniformly randomized over $[0, 1]$. The peak of the global utility is 7.86. Ground units are located randomly. NEWTAS is implemented on a 3.2-GHz Pentium 4 processor with 2 GB RAM. The execution time of the distributed algorithm implemented with Matlab compiled code is 70 seconds. 31 iterations are needed for NEWTAS to converge. The length of the routes followed by F_1, F_2, F_3, and F_4 are 226, 225, 174, and 219, respectively. At the end of the mission, the average number of healthy vehicles within F_1, F_2, F_3, and F_4 are 3.9, 4.6, 11.5, and 6.5, respectively, which gives a total average number of 26.5 healthy vehicles. The extremum of the global utility is reached. NEWTAS is compared numerically with the solution to a Traveling Salesman Problem (TSP), where the 64 vehicles constitute a single formation yielding a total route length of 309 units, which means that each vehicle has to travel over 309 units of distance. The number of healthy vehicles that reach the last target is 22 with the solution to TSP, representing a 26% increase in route length with respect to the worst route length (226) obtained with NEWTAS, and a loss of 17% more vehicles with respect to NEWTAS's total average number of vehicles that remain healthy until the end of the mission.

References

[1] N. Léchevin, C.A. Rabbath, and M. Lauzon, "Distributed,

Near-Optimal Network Enabled Weapon-Target Assignment for Combat Formations," 8th Int. Conf. on Coop. Control and Optim., Gainesville, Florida, 2008.

[2] R. Olfati-Saber, J. Alex Fax, and R. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems", Proceedings of the IEEE, 95, 1, 2007.

[3] W. Ren, R.W. Beard, and E.M. Atkins, "Information Consensus in Multivehicle Cooperative Control," IEEE Control Systems Magazine, 27, 2, April 2007, 71-82.

[4] C.C. Moallemi, and B. Van Roy, "Consensus Propagation," IEEE Trans. on Inform. Theory, 52, 11, 4753-4766, 2006.

[5] D. Mosk-Aoyama, and D. Shah, "Computing Separable Functions via Gossip," in Proc. 25th ACM Symposium on Principles of Distributed Computing, Denver, CO, 113-122, 2006.

[6] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules," IEEE Trans. Aut. Cont., 48, 6, June 2003, 988-1001.

[7] R. Olfati-Saber, E. Franco, E. Frazzoli, and J.S. Shamma, "Belief Consensus and Distributed Hypothesis Testing in Sensor Networks," in Networked Embedded Sensing and Control, LNCIS 331, Springer-Verlag, 169-182, 2006.

[8] J. Cortes, "Finite-Time Convergent Gradient Flows with Applications to Network Consensus," Automatica, 42, 11, 1993-2000.

[9] S. Sundaram and C. N. Hadjicostis, "Distributed Functional Calculation and Consensus using Linear Iterative Strategies." IEEE Journal on Sel. Areas in Comm., 26, 4, 650-660.

[10] J.-C. Delvenne, R. Carli, and S. Zampieri, "Optimal strategies in the Average Consensus Problem," 46th IEEE CDC, 2007.

[11] D. Bertsekas, Dynamic Programming and Optimal Control-Vol. I , Belmont, MA: 3^{rd} Edition, Athena Scientific, 2005.

[12] M. Penrose, Random Geometric Graphs, Oxford U. P., 2003.

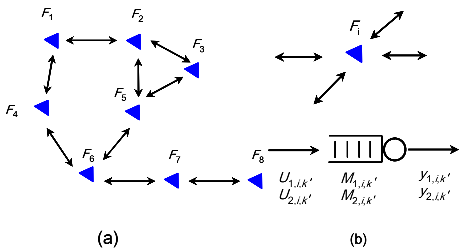

Fig. 1. Consensus: (a) Graph $\mathcal{G}_c$, (b) Information through F_i.

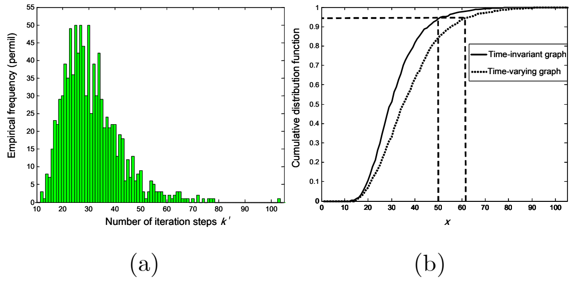

Fig. 2. FTRaRC: (a) Steps for consensus, (b) Cumulative distribution functions.

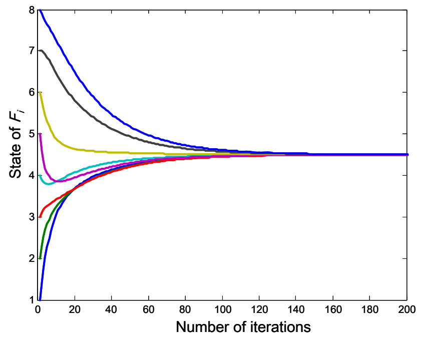

Fig. 3. Time evolution of F_i's state obtained with DABA.

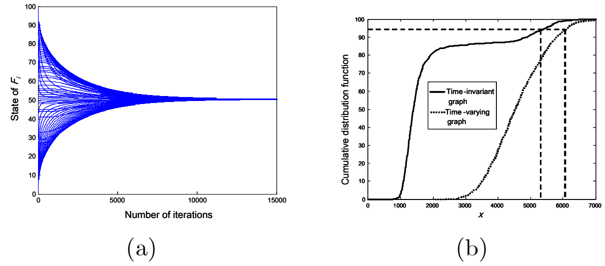

Fig. 4 Agreement in 100-node ring graph: (a) Time evolution of F_i's state (i=1,...,100) with DABA, (b) Cumulative distribution functions $\Pr(k^{'} < x)$ with FTRaRC.

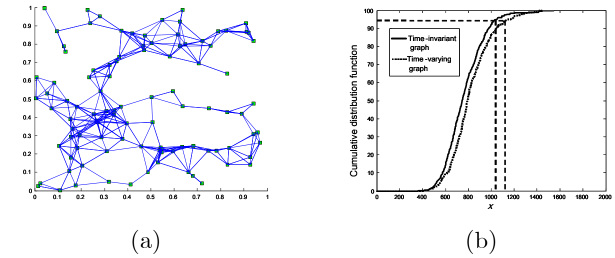

Fig. 5. Agreement in 100-node random geometric graph: (a) Random graph with connections determined by $r = 0.15$, (b) Cumulative distribution functions with FTRaRC.

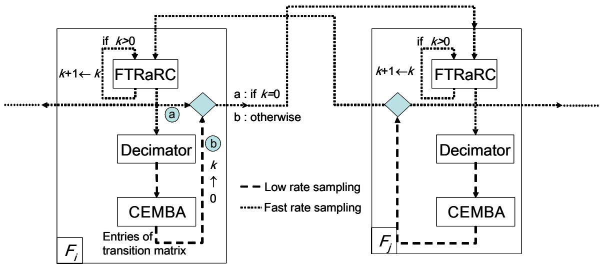

Fig. 6. NEWTAS instantiated in F_i and F_j.

Table 1. Number of iterations to reach agreement.

	$\mathcal{G}_c$	$\mathcal{G}_c^{(2)}$	$\mathcal{G}_c^{(3)}$
FTRaRC LoC 95% | 99%	50|63	210|242	447|543
DABA relative error: 1%	142	599	1334