A Graph-based Approach for Distributed Parameter Coordination in Wireless Communication Networks

Igor M. Guerreiro*, Dennis Hui†, Jiann-Ching Guey‡ and Charles C. Cavalcante*
*Wireless Telecommunications Research Group (GTEL), Universidade Federal do Ceará
Fortaleza, Brazil, Email:{igor,charles}@gtel.ufc.br
† Ericsson Research, San José, California, USA, Email: dennis.hui@ericsson.com
‡Industrial Technology Research Institute (ITRI), Hsinchu, Taiwan, Email: jcguey@itri.org.tw

***Abstract*—This paper addresses distributed parameters coordination methods for wireless communication systems founded on message-passing algorithms on graphs. This work provides evaluations on the potential of such an approach in a wireless communication network, and compares its performance and convergence properties with those of a baseline selfish/greedy approach. Simulation results for an example application, i.e. frequency reuse planning, is presented and discussed. The results show that graph-based techniques generally have higher probability of reaching (near) optimal solution than the greedy approach, though its rate of convergence tends to be slower and the message size is typically larger. Methods of improving the rate of convergence of graph-based distributed coordination techniques and reducing the associated message size are therefore important topics for future studies.**

I. Introduction

In a cellular network, there are many occasions in which each cell needs to set a parameter value, such as reference signal, transmit power, beam direction, or scheduled user. This is preferably performed in such a way that each cell's setting is compatible with the settings of its neighboring cells in order to achieve a certain notion of optimality of the entire network, such as maximizing the average system or user throughput. The choice made by one cell on a local parameter often affects the interference level experienced by its immediate neighbors and hence their respective choices made on their local parameters, which in turn would influence the choices made by their neighbors' neighbors. The simplest example is perhaps the classical frequency reuse problem where the frequency band in which each cell transmits or receives is preferably different from those of its immediate neighboring cells to avoid mutual interference.

In some cases, such as the frequency reuse problem, the parameter is of relatively static nature and can be optimally planned and set before the network deployment. However, in other cases, such as the transmit power control or user/beam selection problems, the parameter is more dynamic and requires coordination to be continually performed. Therefore, a systematic methodology for coordinating the choices of any parameters across the network is desired. Moreover, in order to facilitate flexible, dense deployment of small base-stations in future cellular networks, there is also an increased interest in methods of performing the coordination of parameters among neighboring cells in an autonomous and distributed fashion without a central controller, as any unplanned addition (or removal) of base-stations can substantially alter the system topology and thus the preferred settings.

Factor graphs and the associated sum-product algorithm have been widely used in probabilistic modeling of the relationship among inter-dependent (random) variables or parameters. There are numerous successful applications [1] including, most notably, various fast-converging algorithms for decoding low-density parity check (LDPC) codes and turbo codes, generalized Kalman filtering, fast Fourier Transform, etc. Similar (but different) applications of factor graphs have also been recently proposed for the problem of fast beam coordination among base-stations in [2], [3]. In this work, we propose to investigate the performance of some graph-based methods for coordination of discrete parameters in a wireless communication network. The basic idea here is to model the relationship between the local parameters to be coordinated among different communication nodes of a network and their respective performance metrics or costs using a factor graph [1]. Based on such a graph, a variant of the sum-product algorithm [1], namely the min-sum algorithm, can then be applied in order for all nodes, through iterative message passing with their respective neighbor nodes, to decide upon the best set of local parameters that can collectively maximize a global performance metric across the network. The algorithm allows each communication node to be indecisive of its own decision until sufficient information about how its decision would affect the overall network performance is accumulated.

The paper is organized as follows. Section II describes the system model and the problem considered in this work, as well as discusses some classical solutions and the proposed algorithm. In Section III, we formulate the optimization problem of an application considering the proposed algorithm. In Section IV, we report some numerical results obtained which highlight the outcomes of our algorithm. Finally, we state our conclusions in Section V.

II. System Model

A. Problem Description

Consider a communication network with N communication nodes. A communication node described here may represent any communication device in general in a wireless communication network; for examples, a base-station (BS), an access

978-1-4673-4941-3/12/$31.00 ©2012 IEEE

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on November 25,2022 at 17:37:42 UTC from IEEE Xplore. Restrictions apply.

node, or a user equipment (UE) in a cellular communication system. Let p_i, for $i = 1, 2, \ldots, N$, denote a discrete parameter (or a vector of two or more discrete parameters) of the ith communication node, whose value is drawn from a finite set $\mathcal{P}_i$ of possible parameter values for that node, and let $\mathbf{p} \equiv \begin{bmatrix} p_1 & p_2 & \cdots & p_N \end{bmatrix}^T$ be a vector collecting all the parameters in the network. Each node i is associated with a list $\mathcal{N}_i$ of proper neighboring nodes[1] (i.e. excluding node i) whose choices of parameter values can affect the local performance of node i. For convenience, also let $\mathcal{A}_i \equiv \mathcal{N}_i \cup \{i\}$ denote the "inclusive" neighbor list or just the neighbor list of node i. Let $\mathbf{p}_{\mathcal{A}_i}$ denote the vector of those parameters of nodes in $\mathcal{A}_i$, with its ordering of parameters determined by the sorted indices in $\mathcal{A}_i$. Associated with each node i is a performance metric or cost, denoted by $M_i(\mathbf{p}_{\mathcal{A}_i})$, which is a function of those parameters in the neighbor list $\mathcal{A}_i$ of node i. Each node i is assumed to be capable of communicating with all nodes in $\mathcal{A}_i$. Our goal is for each node i to find, in a distributed fashion, its own optimal parameter p_i^*, which is the corresponding component of the optimal global parameter vector $\mathbf{p}^*$ that minimizes the total (global) performance metric given by

$$M(\mathbf{p}) \equiv \sum_{i=1}^{N} M_i(\mathbf{p}_{\mathcal{A}_i}). \tag{1}$$

B. Centralized Solution

Conceptually, the simplest approach to the optimization problem described above is to solve it jointly at a central location by direct computing

$$\mathbf{p}_C \equiv \begin{bmatrix} p_{C,1} \\ p_{C,2} \\ \cdots \\ p_{C,N} \end{bmatrix} \equiv \arg\min_{\mathbf{p}} \sum_{i=1}^{N} M_i(\mathbf{p}_{\mathcal{A}_i}), \tag{2}$$

which is an optimal solution to the problem by definition. A major issue of this approach is the huge computational complexity for large network size N, as the complexity grows exponentially as the number of communication nodes increases.

C. Selfish/Greedy Solution

Another approach to the optimization problem above is for each communication node to selfishly[2] set its own parameter to optimize its own local performance based on the most recent choices made and given by its neighbors. In this approach, the complexity grows only exponentially with the maximum number of neighboring nodes, but linearly with the total number of nodes N in the network. More precisely, in this approach, the local parameter vector p_i at each communication node is iteratively chosen as

$$p_{S,i}^{(n+1)} \equiv \arg\min_{p_i} M_i(\mathbf{p}_{\mathcal{A}_i}) \Big|_{\mathbf{p}_{\mathcal{N}_i} = \mathbf{p}_{S,\mathcal{N}_i}^{(n)}} \tag{3}$$

[1]Proper neighbors of a given communication node refer to as the communication nodes which directly communicate with that node.

[2]From a game-theoretic perspective, the greedy solution may be seen as a non-cooperative game in which the players are the communication nodes and their actions are the iterative choices of parameters.

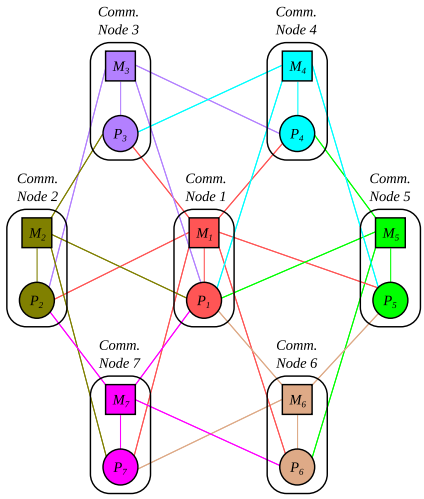

Fig. 1. Factor graph model for a communication network with local parameters and local performance measures.

where n is the iteration index, $p_{S,i}^{(n+1)}$ denotes the choice of p_i made at iteration $n+1$ using this selfish/greedy approach, and $\mathbf{p}_{S,\mathcal{N}_i}^{(n)}$ denotes the vector of those parameter choices made by nodes in $\mathcal{N}_i$ at the nth iteration.

D. Graph-based Solution

In this section, we describe another approach to the problem of minimizing the global metric in (1) by modeling the communication nodes and the associated local performance metrics using a factor graph. A factor graph is a bipartite graph consisting of a set of variable nodes and a set of factor nodes. Each variable node represents a variable and can only be connected to a factor node (but not another variable node) through an edge, while each factor node represents a function of some of the variables. A factor node is connected to a variable node if and only if the corresponding function represented by the factor node depends on that variable.

Specifically, for the problem formulated above, we associate each variable node with the parameter p_i of a communication node and each factor node with its local performance metric $M_i(\mathbf{p}_{\mathcal{A}_i})$. Accordingly, we label a variable node corresponding to p_i as $v(p_i)$ and a factor node corresponding to $M_i(\mathbf{p}_{\mathcal{A}_i})$ as $v(M_i)$. An edge connecting a factor node $v(M_i)$ with a variable node $v(p_k)$ exists if and only if $k \in \mathcal{A}_i$. For example, Fig. 1 shows a hexagon layout of 7 communication nodes, each associated with a factor node representing the local performance metric $M_i(\mathbf{p}_{\mathcal{A}_i})$ and a variable node representing the local parameter p_i. The factor node associated with the local performance metric $M_i(\mathbf{p}_{\mathcal{A}_i})$ of communication node i is connected through edges to the respective variable nodes associated with its own local parameter p_i and those parameters $\{p_i\}_{i \in \mathcal{N}_i}$ of its neighbors upon which the metric $M_i(\mathbf{p}_{\mathcal{A}_i})$ depends. For clarity, we use a different color to represent different communication nodes and the corresponding edges connecting their respective factor nodes with the relevant variable nodes.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on November 25,2022 at 17:37:42 UTC from IEEE Xplore. Restrictions apply.

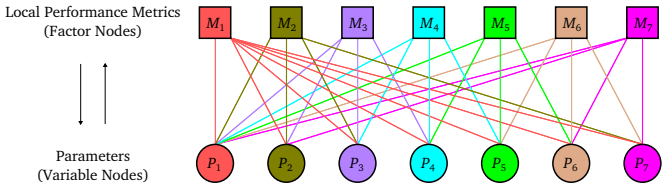

Fig. 2. Re-organized Factor Graph for a Communication Network with Local Parameters and Local Performance Measures.

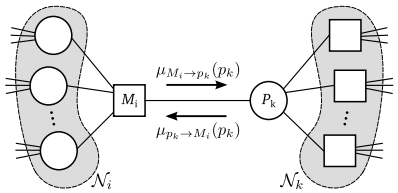

Fig. 3. A factor-graph fragment, showing the message pass between factor node M_i and variable node p_k.

The graph in Fig. 1 can be also re-organized as in Fig. 2, which clearly shows the bipartite property of the graph with factor nodes connected only to variable nodes through the respective edges. A message-passing algorithm, namely the sum-product algorithm, can then be executed on such a graph. Specific messages are computed and passed iteratively along each edge of the graph. Those messages that are passed on edges connecting factor and variable nodes of different colors correspond to information exchange between two neighboring communication nodes, while those messages that are passed on edges that connect nodes of the same color represent internal communications within each communication node. Each message depends only on the variable whose associated variable node is a vertex of the edge over which the message is passed along. More precisely, each message is simply a table of values with each entry corresponding to one of the possible values of the variable, as described in more details below.

The sum-product algorithm [1] can be applied whenever the variables and functions associated with the factor graph are defined on a commutative semi-ring whose elements satisfy the distributive law. For our problem at hand, we apply the variant of sum-product algorithm that is based on the min-sum commutative semi-ring[3]. In this case, the sum-product algorithm is also called the min-sum algorithm.

More specifically, let $\mu_{M_i \to p_k}(p_k)$ denote the message to be passed from a factor node $v(M_i)$ to a variable node $v(p_k)$, and let $\mu_{p_k \to M_i}(p_k)$ denote the message to be passed from the variable node $v(p_k)$ to the factor node $v(M_i)$. Fig. 3 shows the two kind of messages passing on in a fragment of a factor graph. The min-sum algorithm, when applied to our problem at hand, simply iterates between the following two kinds of message computations and exchanges:

$$\mu_{M_i \to p_k}(p_k) = \min_{\mathbf{p}_{\mathcal{A}_i \setminus \{k\}}} \left\{ M_i(\mathbf{p}_{\mathcal{A}_i}) + \sum_{j \in \mathcal{A}_i \setminus \{k\}} \mu_{p_j \to M_i}(p_j) \right\}, \quad (4)$$

$$\mu_{p_k \to M_i}(p_k) = \sum_{j \in \mathcal{A}_k \setminus \{i\}} \mu_{M_j \to p_k}(p_k). \quad (5)$$

The algorithm may begin with each factor node $v(M_i)$ computing outgoing message $\mu_{M_i \to p_k}(p_k)$ to $v(p_k)$ for each $k \in \mathcal{A}_i$ using (4) with all incoming messages $\mu_{p_k \to M_i}(p_k)$ from connected variable nodes initialized to random values close to zero[4]. Upon receipt of the message $\mu_{M_i \to p_k}(p_k)$, each variable node $v(p_k)$ then compute outgoing message $\mu_{p_k \to M_i}(p_k)$ to $v(M_i)$ for each $i \in \mathcal{A}_k$. The algorithm then iterates until certain stopping criteria, such as a pre-determined maximum number of iteration, is reached, in which case, the parameter for communication node i is determined at its variable node $v(p_i)$ by

$$p_i^* = \arg\min_{p_i} \left\{ \sum_{j \in \mathcal{A}_i} \mu_{M_j \to p_i}(p_i) \right\}. \quad (6)$$

Note that both messages computed in (4) and (5) depend only on the value of p_k. Since $p_k \in \mathcal{P}_k$ and $\mathcal{P}_k$ is assumed to be discrete and finite, each of the messages can be represented by a table of $|\mathcal{P}_k|$ entries, where $|\mathcal{P}_k|$ denotes the cardinality of $\mathcal{P}_k$. In particular, the computation in (5) is just adding up the corresponding entries of multiple tables of the same size together.

When the factor graph contains no cycle[5], it can be shown [1] that the message passing algorithm described above will yield the exact optimal solution that minimizes (1). However, when the graph contains cycle, as it often does in many practical applications such as our problem of interest, (6) typically yields good approximations to the true optimal solution [1].

Note that to reduce computational complexity, the minimization operation in (4) can be computed through the standard alternating-coordinate optimization technique, i.e. starting with an arbitrary choice of $\mathbf{p}_{\mathcal{A}_i \setminus \{k\}}$ and optimizing each parameter in $\mathbf{p}_{\mathcal{A}_i \setminus \{k\}}$ one at a time while holding others fixed.

E. Message-passing Scheduling

Regarding cases with loopy graphs, the iterative process described above must follow a schedule. That is, nodes are activated to participate in the message pass over the iterations. In this work, two sorts of node scheduling were considered: *i*) simultaneous scheduling, where 100% of the nodes participate in the message pass at the same time over all the iterations; and *ii*) random scheduling, where each node has

[3]Recall that a semi-ring is a mathematical structure (e.g. a set) equivalent to a ring without an additive inverse. In a general way, the binary operations addition and multiplication can be replaced with others as long as the distributed law still holds. In this sense, a min-sum semi-ring simply replaces the addition operation with the minimum operation and the multiplication operation with the addition operation.

[4]Depending on the individual performance metrics, initializing incoming messages to zero may lead nodes to compute outgoing messages with equal entries (e.g. entries equal to zero). In such cases, nodes are not able to iteratively find the best parameters as all the entries return the same cost.

[5]In this context, the term cycle refers to a closed path in a graph.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on November 25,2022 at 17:37:42 UTC from IEEE Xplore. Restrictions apply.

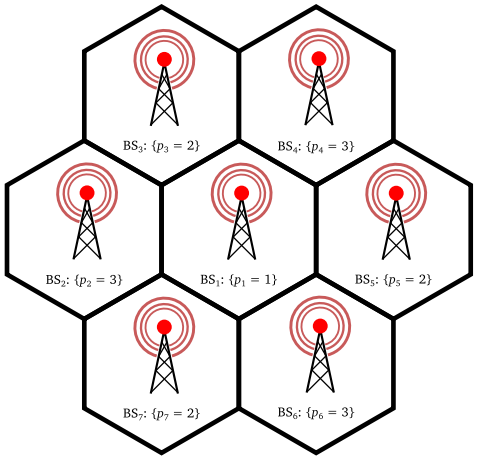

Fig. 4. An example of a frequency reuse planning in a 7-cell grid. The parameter vector $\mathbf{p}^* = [1\ 3\ 2\ 3\ 2\ 3\ 2]^T$ is one of the optimal solutions, where $\mathcal{P}_i = \{1, 2, 3\}$ for $i = 1, 2, \ldots, N$. That is, no collision is observed among the BSs, which are the communication nodes.

a certain probability of being activated to participate in the message pass at each iteration .

In the random scheduling example, "90% Random" means that 90% of the nodes (in average) are activated and participate in the iterative process. In other words, it may be seen as a "coin flipping", that is, each node has two possibilities at the beginning of each iteration: active or inactive. If a node is inactive, no message passes on its associated edges. Note that "100% Random" is equivalent to simultaneous scheduling.

III. Example of Application

As said in Section I, there are many applications in wireless communications in which parameters need to be coordinated. However, one classic example motivated the most: the prisoners dilemma [4]. In a few words, the two prisoners find an equilibrium point if both greedily betray. However, this is not a good solution because they can individually decrease their sentences if both of them stay silent, which is the best solution and it is reached only if there is a sort of communication or centralization. In this context, the graph-based approach reaches the optimal solution by applying the message-passing algorithm.

A possible application of the algorithm described above in wireless communications is the frequency reuse problem, which is what we focus on in this paper. For this problem, each parameter p_i may represent an index to a frequency band out of a set $\mathcal{P}_i$ of all possible frequency bands. The goal here is to use a distributed algorithm to solve the frequency reuse planning problem by minimizing the total number of "collisions" in terms of using the same frequency band in adjacent cells or BSs. Fig. 4 shows an example of such an application where a possible optimal solution (no collision) is reached. In this case, the local performance metric may be chosen as

$$M_i\left(\mathbf{p}_{\mathcal{A}_i}\right) = \sum_{j \in \mathcal{N}_i} \delta\left(p_i, p_j\right), \tag{7}$$

TABLE I
Parameters of Simulation.

Cell Layout	Hexagon
Number of Nodes	61
Number of Frequency Bins	3
Parameter Initialization	Random
Max. Number of Iterations per Run	100
Number of Simulation Runs	1000
Message-passing Scheduling Type	Simultaneous or Random

where $\delta\left(p_i, p_j\right) = 1$ if $p_i = p_j$, and otherwise $\delta\left(p_i, p_j\right) = 0$.

Other possible choices of the local performance metric may lead to a faster convergence of the algorithm. One could count all "collisions" between neighboring communication nodes within the neighborhood defined by $\mathcal{A}_i$. However, this approach is beyond the scope of this work.

IV. Simulation Results

In this study, the global performance metric represented by (7) in the frequency reuse planning problem was investigated in order to evaluate how it behaves statistically in terms of the cumulative distribution function (CDF). The optimal solution, which occurs when the sum of collisions equals zero, was used as a performance benchmark for both graph-based and greedy techniques. Besides, the speed of convergence of both approaches in terms of the CDF was also assessed.

A hexagon layout with 61 cells and a single communication node in each cell was adopted. The parameters to be coordinated are 3 frequency bins for the 61 cells and the parameter initialization is at random, i.e. nodes pick one of the frequency bins randomly at the beginning of each simulation run. The maximum number of iterations in each simulation run is 100 and a total of 1000 runs was conducted for statistical purposes. These parameters were adopted for both simultaneous and random message-passing schedulers. Table I lists the simulation parameters.

The graph-based technique has different behaviors for different types of message-passing scheduling. Fig. 5 shows that simultaneous scheduling yields a worse performance than the random ones. In this case, nodes may experience a ping-pong effect[6], i.e. the messages computed by factor nodes alternate between two or more values. Also, the chance of reaching optimality (zero collision) is less than 30% and the maximum number of collisions is reached in almost 50% of the cases. With respect to the random scheduling, the random 90% case reaches optimality with a 73% probability, but the maximum number of collisions is close to 300. The random 60% and the random 30% have similar behaviors and yield the best performance. The former reaches optimality with a 70% probability and the maximum number of collisions is less than 25. The latter reaches optimality with a 61% probability and the maximum number of collisions is also less than 25.

[6]Ping-pong effect refers to the alternation between two (or more) possible values of a variable ad infinitum. To avoid the continuity of this state, some perturbation may be inserted in the system, such as the addition of noise or forcing an unexpected value of some variables. In the context of this work, the randomness of the random scheduling acts as a perturbation in the system.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on November 25,2022 at 17:37:42 UTC from IEEE Xplore. Restrictions apply.

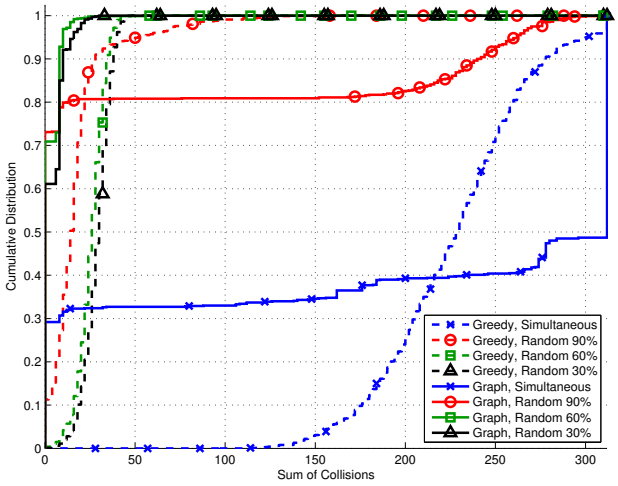

Fig. 5. Performance analysis of the different techniques and message-passing scheduling types based on the sum of collisions.

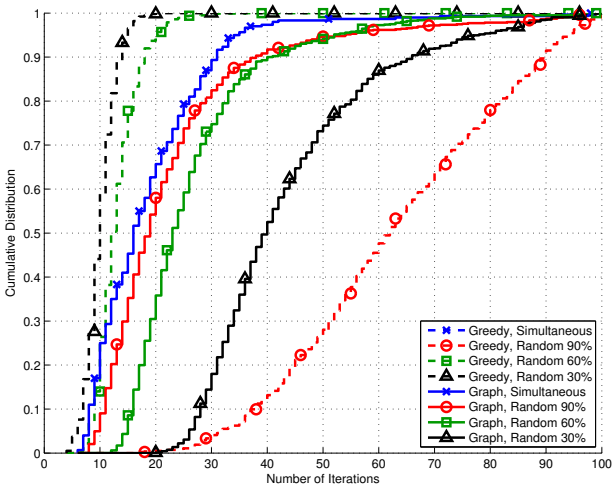

Fig. 6. Speed of convergence of the different techniques and message-passing scheduling types based on the number of iterations until convergence.

Similarly, the greedy technique also has different behaviors for different types of message-passing scheduling. Fig. 5 shows that simultaneous scheduling still provides a worse performance than the random ones. Moreover, the greedy approach with simultaneous scheduling often leads to a ping-pong effect. From the dashed blue curve, the chance of reaching optimality is zero, and a maximum number of collisions of more than 300 is reached in around 5% of the cases. On the other hand, the random 90% case reaches optimality with a 11% probability and the final number of collisions is bounded above by 150 with probability one. Again, the random 60% and the random 30% have similar behaviors. Both of them reach optimality with quite a small probability while the final number of collisions is also bounded by 50 with probability one.

As for speed of convergence, the graph-based technique with simultaneous scheduling converges[7] faster than the random scheduling cases. Fig. 6 shows that the rate of convergence increases with the probability of node activation. However, the simultaneous case converges with 30% probability, which is small compared to the random scheduling cases. In this context, the random 90% case converges with 78% probability, the random 60% case with 99% probability and the random 30% case with 95% probability.

On the other hand, the greedy technique with simultaneous scheduling does not reach a point of convergence. Thus, the number of iterations needed to converge is *infinitum* for this case. Fig. 6 shows that the rate of convergence decreases with the probability of node activation. Then again, the random scheduling cases converge with a higher probability than simultaneous scheduling. The dashed blue curve, which increases at *infinitum*, refers to the simultaneous case and converges with probability zero. The random 90% case converges with 83% probability and both the random 60% and the random 30% cases with 100% probability.

[7]In this context, convergence means that no ping-pong effect was experienced by the nodes.

V. Conclusion

The graph-based approach for distributed parameter coordination considers the impact of nodes decisions on their neighboring nodes. The information (message) exchange is only among neighbors. Such a technique has higher probability of reaching (near) optimal solution than the greedy approach. However, the speed of convergence is slower and messages are larger in size compared to the greedy approach. The use of random scheduling in the frequency reuse planning case tends to enhance the rate of convergence. It is worthwhile to note that the graph-based approach is totally adaptable to any discrete problem of parameter coordination. As for the numerical results, the graph-based technique provides good gains in the global cost mainly with random scheduling. Comparison with the centralized solution may confirm the promising outcomes of the graph-based technique.

Acknowledgment

This work was partially supported by the Innovation Center, Ericsson Telecomunicações S.A., Brazil.

References

[1] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498 –519, feb 2001.

[2] I. Sohn, S. H. Lee, and J. Andrews, "A graphical model approach to downlink cooperative mimo systems," in *Proc. IEEE Globecom 2010*, dec. 2010, pp. 1 –5.

[3] B. L. Ng, J. Evans, S. Hanly, and D. Aktas, "Distributed downlink beamforming with cooperative base stations," *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5491 –5499, dec. 2008.

[4] Z. Han and K. Liu, *Resource Allocation for Wireless Networks: Basics, Techniques, and Applications*. Cambridge University Press, 2008.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on November 25,2022 at 17:37:42 UTC from IEEE Xplore. Restrictions apply.