A Decentralized Approach to Solve Group AHP with Agreements by Consensus

M. Rebollo, A. Palomares, and C. Carrascosa

Universitat Politècnica de València,
Camino de Vera s/n 46022 Valencia (Spain)
{mrebollo, apalomares, carrasco}@dsic.upv.es

Abstract. The analytical hierarchical process (AHP) is a multi-criteria, decision-making process that has demonstrated to be of a high utility to achieve complex decisions. This work presents a method to apply it in grupal decisions, where the weights that each user assigns to the criteria are different and private. A combination of consensus process and gradient ascent is used to reach a common agreement that optimizes the utility of the decision using the information exchanged in the local neighborhood exclusively.

The AHP problem is modeled through a multilayer network. Each one of the criteria are negotiated by consensus with the direct neighbors on each layer of the network. Furthermore, each node performs a transversal gradient ascent and corrects locally the deviations from the personal decision to keep the best option. The process locates the global optimal decision, taking into account that this global function is never calculated nor known by any of the participants. If there is not a global optimal decision where all the participants have a not null utility, but a set of suboptimal decisions, they are automatically divided into different groups that converges into these suboptimal decisions.

Keywords: complex networks, consensus, gradient descent, analytical hierarchical process, agreement

1 Introduction

The Analytic Hierarchical Process (AHP) is a muli-objective optimization method. The decision makers provide subjective evaluations regarding to the relative importance of the different criteria and the preference of each alternative for each criteria [12]. The result is a ranking of the considered alternatives that includes the relative score assigned to each one of these alternatives. The main advantage of this process is that it allows (i) to organize the information in a efficient and clear way, even for complex problems; and (ii) synthesize and visualize the effects of changes in the levels or preferences. Furthermore, it is possible to measure the consistence of the model, since a perfect consistency is very difficult to be achieved due to the subjectivity introduced to judge the relative importance of each criteria.

The AHP can be used for a single used to take a decision, but also for a group of people, such as a committee or a group of experts, to achieve a common agreement. There are works that extends the original AHP problem. But these approaches assume

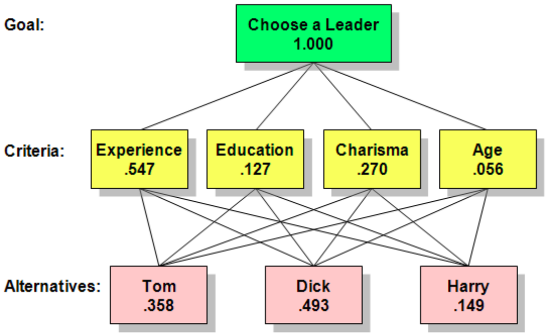

Fig. 1: Example of criteria hierarchy for a AHP

that all the actors are able to exchange information. This work proposes a method for group decision making based on AHP, where the participants are connected though a network and they interact exclusively with their direct neighborbors. A combination of consensus [18] and gradient ascent is used as optimization method [14].

The proposed solution considers each criterion as a layer in a multiplex network. A consensus process is performed in each layer, trying to achieve a common decision for the corresponding criteria for all the participants. Simultaneously, a gradient ascent is executed across the layers, trying to keep the preferred value for each one of the participants in the decision. This joint process converges to the desired, agreed decision. This decision is the optimal decision of the group if some conditions are fulfilled.

The rest of the paper is organized as follows. Section 2 explains the related techniques that have been combined and used to define the final proposed method to solve AHP in a decentralized and distributed way. The method is detailed and analyzed in Section 3 and, finally, Section 4 shows the results. Section 5 closes this work with the conclusions.

2 Related works

2.1 The AHP process

The AHP begins with the definition of the criteria used to evaluate the alternatives, organized as a hierarchy. The importance of each criteria is defined through its weight $w^{\alpha} \in [0, 1]$. For example, let's assume that a new leader has to be chosen among three candidates: Tom, Dick and Harry. To evaluate them, their age, experience, education and charisma are going to be considered. The criteria hierarchy and the weights associated to each criterion α are show in Figure 1.

Once the criteria are defined, a pairwise matrix is created, assigning a relative judgement or preference value to each pair of alternatives. The value a_{ij} represents the preference of the alternative i over the alternative j for the considered criteria, and $a_{ij} = 1/a_{ji}$.

	Tom	Dick	Harry	Priority (l_i^α)
Tom	1	1/4	4	**0.217**
Dick	4	1	9	**0.717**
Harry	1/4	1/9	1	**0.066**

Candidate	Exp	Edu	Char	Age	Goal
Tom	0.119	0.024	0.201	0.015	*0.358*
Dick	0.392	0.010	0.052	0.038	**0.492**
Harry	0.036	0.093	0.017	0.004	*0.149*

Table 1: (left) Local priority matrix with the relative importance of each candidate regarding to their experience. (right) Final priorities for the selected candidates. Dick is selected candidate, with the higher global value

From this pairwise matrix, the local priority l_i^α is calculated, which defines the preference of the alternative i for the criterium α. The local priority is calculated as the values of the principal right eigenvector of the matrix.

Finally, all the local priorities are synthesize across all the criteria in order to calculate the final, global priority p_i for each alternative. There exist many methods to calculate the priorities. The most usual ones are the mean of the rows of he pairwise matrix to calculate l_i^α, and the weighted average $p_i = \sum w^\alpha l_i^\alpha$ for the global priority,

There are approaches to extend AHP into grupal decision problems, but they are centralized solutions and use complete information. In this work, the participants are connected through a network that bounds the possible information exchanges. An agreement in the final decision is reached through a combination of a consensus process and a gradient ascent (see Figure 2)

2.2 Consensus on Networks

Consensus means reaching an agreement on the value of a variable which might represent, for example, a physical quantity, a control parameter, or a price. Agents are connected through an acquaintances network whose topology constraints the possible interaction between them. This is one of the most promising research subjects in the MAS area that is currently emerging [8,9,11,13,21].

The theoretical framework for solving consensus problems in agent networks was formally introduced by Olfati–Saber and Murray [17, 18]. Let G be a graph of order n with the set of entities E as nodes. Let (G, X) be the state of a network, where $X = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ and x_i is a real value that is associated with the node $e_i \in E$. A consensus algorithm is an interaction rule that specifies the information exchange between the agents and all of their neighbors in the network in order to reach the agreement. Consensus is reached in the network when $x_1 = \ldots = x_n$. It has been demonstrated that a convergent and distributed consensus algorithm in discrete-time exists and it converges to the average of their initial values.

$$x_i(t+1) = x_i(t) + \varepsilon \sum_{j \in N_i} (x_j(t) - x_i(t)) \tag{1}$$

where N_i denotes the set formed by all nodes connected to the node i (neighbors of i) and ε is the step size, $0 < \varepsilon < \min_i 1/d_i$, being d_i the degree of node i. This expression, when is executed by the agents, converges to the average of their initial values.

An interesting modification of the consensus introduces weights in the agents, which represent their importance in the system. Let $w = (w_1, w_2, \ldots, w_n)^T$ be a vector with the weight associated to each node. The following algorithm (see [17], p. 225) can be used to obtain the value of the weighted average consensus

$$x_i(t+1) = x_i(t) + \frac{\varepsilon}{w_i} \sum_{j \in N_i} (x_j(t) - x_i(t)) \tag{2}$$

where N_i denotes the set formed by all nodes connected to the node i (neighbors of i) and ε is the step size. The algorithm converges to the weighted average of the initial values of the state of each agent $x_i(0)$ if $\varepsilon < \min_i d_i/w_i$, being d_i the degree of node i [19].

Other works have extended the consensus algorithm for its application in large-scale systems [5], for its usage as a clustering technique [15], for treating problems derived from a failure in communications [10], or for applications in arbitrary directed graphs [7]. However, the application of the consensus algorithm to dynamic networks, where participants may enter and leave during the consensus process, is still an open issue.

2.3 Distributed Gradient Descent

Consensus leads to the average value of the network. But agreement processes frequently involve the optimization of some global utility function. Centralized methods usually require data fusion and distribution along the network, which supposes a high computational and communication cost when the systems scale. Decentralized approaches take advantage of scalability, adaptation to dynamic network topologies and can handle data privacy. Coupled optimization problems can be solved using a variety of distributed algorithms. A classical way is to iteratively refine an estimate of the optimizer using incremental subgradient methods [1]. It is used in static networks, where the topology does not change during the process. Matei [16] studies how the degree distribution in random networks affects the optimal value deviation, defining some metrics to evaluate the quality of the approximated solution. One way of accelerating the consensus process has been proposed by Pereira [20]. This new method is applied to random sensor networks. It is based on the study of the network spectral radius, requiring a complete view of the network to obtain that radius. The relation among the connection probabilities in a random network and the convergence speed has also been studied [20]. This relation also determines the optimal ε value that minimizes the convergence time. The work of F. Zanella [2] applies the Newton–Raphson method to distributed convex optimization problems. To minimize the optimization function, it uses a consensus process that converges to the exact solution in contrast to the subgradient–based methods. This last work has been extended to consider asynchronous transmission [3] and the multi-dimensional case in order to optimize an n-dimensional function [4].

The combination of consensus and gradient models can be expressed as a two step process [14]

$$x_i(t+1) = \sum_j w_{ij} x_j(t) - \alpha \nabla f_i(x_i(t)) \tag{3}$$

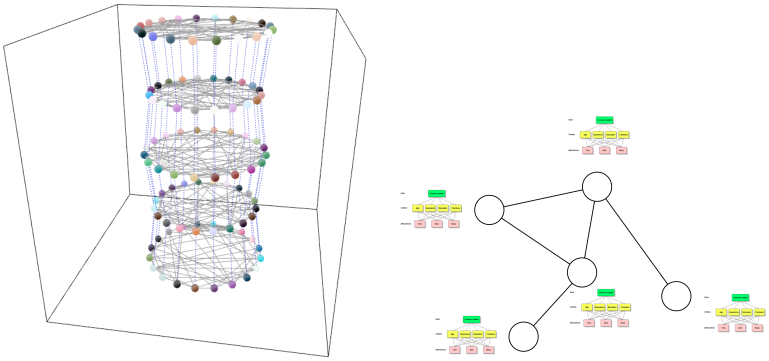

Fig. 2: (Left) Multilayer network example with 20 agents and 5 layers. (Right) Example of network, where each agent has its own values for the criteria an a preferred option.

where $W = [w_{ij}]$ is a symmetric, double stochastic matrix (note that it has the same properties demanded to the consensus process to converge) and $\nabla f_i(x_i(t))$ performs a gradient descent to minimize a cost function.

2.4 Multilayer networks

Multilayer networks are a recent formalism created to model the phenomena that appears in complex networks in a more realistic way. Usually, relations do not occur isolated in one network and notions such as network of networks, multilayer networks, multiplex networks or interdependent networks are defined. In multilayer networks, links of different type exist among the nodes. For example, in a group of people, links representing friendship, working relations or family ties can be defined. Or in a communication model, different media, such as phone and mail, can be considered. Each one of this different links form a network in one layer. The interdependence among layers is defined through cross links between the nodes that represent the same entity in each network. These cross links models the transference of information that passes from one layer to the others.

A multilayer network (see Figure 2, left) is formally defined [6] as a pair $M = (G, C)$ where $G = \{G^1, \ldots, G^p\}$ is a family of graphs $G^\alpha = (E^\alpha, L^\alpha), \forall \alpha \in [1, p]$ called layers, and $C = \{L^{\alpha\beta} \subseteq E^\alpha \times E^\beta, \forall \alpha, \beta \in [1, p], \alpha \neq \beta\}$ is the set of connections between two different layers G^α and G^β. The elements of each L^α are called *intralayer* connections and the elements of C are the *interlayer* ones or *crossed layers*. The characteristic of the multilayer network is that all the layers have the same set of nodes $E^1 = \ldots = E^p = E$ and the cross layers are defined between equivalent nodes $L^{\alpha\beta} = \{(e^\alpha, e^\beta), \forall e \in E; \alpha,\beta \in [1, p]\}$.

In the present work, multilayer networks are used to represent the different criteria that form the decision. Each criterion will be negotiated in one layer.

3 Decentralized AHP using Consensus in Multiplex Networks

Lets consider the participants connected in an undirected network. The topology is not relevant, but all the nodes must be connected in one component. Lets consider only the criteria that are the leafs of the hierarchy defined for the AHP problem, with $\sum w^\alpha = 1$, different and private for each one of the participants. Lets create a multilayer network, where each layer represents one of the final criteria. Each layer is weighted using the weight defined for the criteria. For example, the problem exposed in Fig. 1 has 4 criteria: experience, education, charisma and age. Therefore, a network with 4 layers is created. Furthermore, the weights associated to each one of them are 0.547, 0.127, 0.270 and 0.056 respectively. An utility function can be defined for each preference of the participants using a gaussian function with mean l_i^α and standard deviation $1 - w_i^\alpha$ (see Section 3.1). This function is used by the participant to perform the gradient ascent, trying to keep as near as possible to its preferred distribution.

Each participant has its own criteria and the goal of the system is to agree the best candidate according to all the agents involves in the decision. Therefore, a consensus process is executed in each layer in order to find the weighted average. But this process considers the criteria as independent and it does not converge in the value that optimize the decision. The combination of the consensus process with a gradient ascent, as it is defined in Eq. 4, corrects the deviation produced by the consensus and each participant tries to maintain the decision that maximizes its own local utility. This decentralized process leads to a consensus value near to the global optimum, considered as the sum of the local utility functions. Observe that this global utility function is never calculated and the participants reach this value exchanging information with their direct neighbors.

$$x_i^\alpha(t+1) = x_i^\alpha + \overbrace{\frac{\varepsilon}{w_i^\alpha}\sum_{j\in N_i^\alpha}\left(x_j^\alpha(t) - x_i^\alpha(t)\right)} + \underbrace{\varphi\nabla u_i(x_i^1(t),\ldots,x_i^p(t))} \tag{4}$$

The result of the process is a common and agreed priority for the alternative evaluated in each layer. All the alternatives can be evaluated at the same time using independent consensus process if a vector of preferences is exchanged instead one alternative at a time.

If the global utility function is a smooth one and all the participants have an utility $u_i > 0$ for any final decision, the proposed method converges to the optimal decision for the group. But if there is no point in which all the participants have a positive utility, the resulting global utility function will have one (or more that one) local maximum that may alter the convergence process. In those cases, we allow the nodes to break the links with those neighbors that are pulling them to an undesired area. To do that, it is enough with breaking the communications and stopping exchange information with them. In this case, the network can be split in several groups and each one of them will reach a different decision.

The advantage of this distributed approach is that avoids the bottlenecks problems that arise in mediated solutions. Individual agents are not conscious of a final, global solution, but of the convergence to an agreed compromise among its near neighbors.

Furthermore, the system is scalable since new nodes can be added without additional notifications to the rest of the network.

3.1 Utility Function

Utility functions have some common properties in any optimization problem: independence, completeness, transitivity and continuity. As we propose a model with cooperative agents, we'll assume that the utility functions have a maximum and this maximum will be the starting point for all the agents. Furthermore, the function must be a decreasing one. The normal distribution fulfills all this properties. Therefore, it has been the selected one for the utility function u_i of the agents. We can assume that agents are initially situated in its maximum value, which corresponds with the mean value of the utility function. The weight assigned to the term can be used in the dispersion measure. An agent does not desire changes in its more relevant term. Therefore, any change in its value must decrease drastically its utility. On the other hand, the agents would allow changes in terms with low importance, which might slightly decrease their utilities. In the case of a normal distribution, the standard deviation is the parameter that rules this behavior. If we use $\sigma_i^\alpha = 1 - w_i^\alpha$ we obtain the desired behavior. The utility function is defined as follows:

$$u_i^\alpha(x_i^\alpha) = e^{-\frac{1}{2}\left(\frac{x_i^\alpha - l_i^\alpha}{1-w_i^\alpha}\right)^2} \tag{5}$$

All this individual functions are combined in one unique utility function for the agent.

$$u_i(x_i) = \prod_\alpha u_i^\alpha(x_i^\alpha) \tag{6}$$

This definition corresponds to a renormalized multi-dimensional gaussian distribution such that the maximum utility for the agent i is $u_i(x_i(0)) = 1$.

The global utility of the system is the sum of the individual utilities of the agents. This value is never calculated in the system directly and the function is known by none of the participants in the agreement.

$$U = \sum_i u_i(x_i) \tag{7}$$

4 Application example

Lets consider a group of 9 agents that are going to take a decision using AHP. A bi-dimensional example has been chosen to be able to represent it graphically, so just 2 criteria will be considered. Figure 3 shows the utility function calculated from the initial preferences of each participant.

Figure 4 shows the initial and final status of the process. When the combined process stops, all the participants have reached the same point, which corresponds to the common decision agreed by the agents. For this solution to exist, the only condition is that all the participants have a positive utility $u_i > 0$ along the complete solution space.

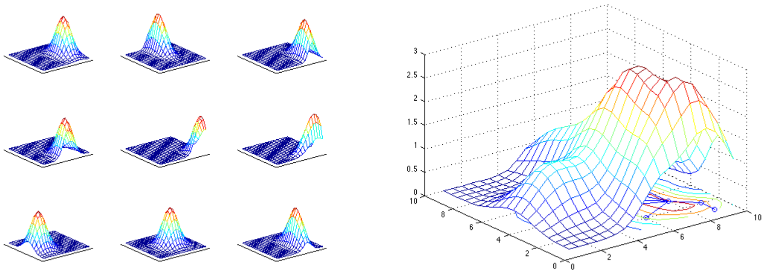

Fig. 3: (Left) Local utilities $u_i(x_i)$ as defined in Eq. 6 from the AHP criteria for each of one the 9 participants. (Right) Final global utility function U (Eq. 7) to locate the optimal decision, defined as the sum of the individual, local utility functions. These functions is not calculated, nor known by the participants, but the process converges to the maximum of this function.

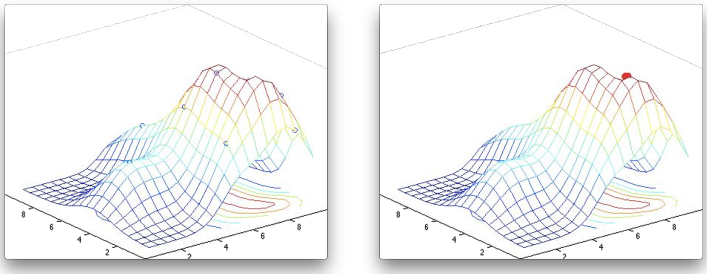

Fig. 4: Initial and final states for a decision in a group of 9 participants using two criteria. An agreed solution exists and it is located correctly by the group using local information only.

Figure 5 shows the evolution of the value for each criterion (left and right) for each one of the participants (in a different colour) along the process. It converges to the final decision. If these values are considered as the x and y coordinates, it matches with the point that corresponds to the solution in Figure 4.

Nevertheless, when this condition is not fulfilled when some of the participants has an utility equal to zero in some areas of the solution space. In that case, the shape of the global utility function will show peaks and valleys, with local optimal values. Then, the convergence to the optimal solution is not guaranteed and, as it is shown in Figure 6, the process halts on any value, depending on the initial preferences and the distribution of the utility functions over the solution space.

Our proposal to solve this additional problem is to allow break links among the participants. When a participant detects that the solution guides towards a point with zero-utility, the agent can decide to break the link to those neighbors who are pulling from the preferences. As Figure 7 shows, in this case the network is broken into groups,

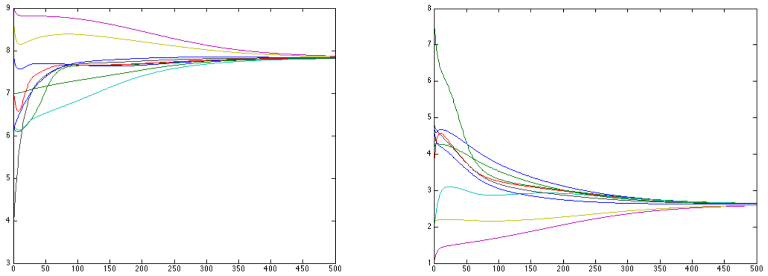

Fig. 5: Evolution of the values for each criteria fora each one of the participants. The convergence is guaranteed if $\forall i \; u_i > 0$ in all the solution space

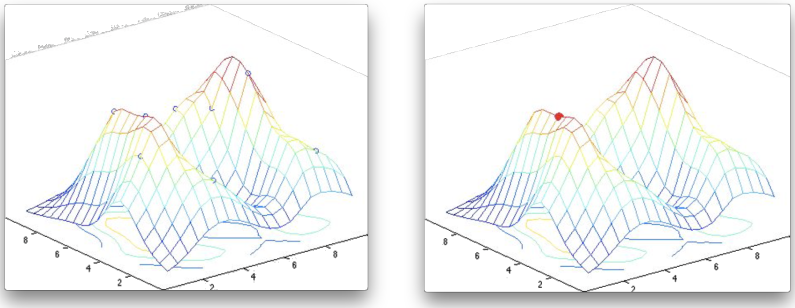

Fig. 6: Example of convergence to a suboptimal solution because participants refuses to move towards the best solution for the group since its has zero-utility for some individual agent.

each one of them converges to a different agreement. The optimal decision is located by the group formed by those participants whose utility function is positive in the best solution. Actually, this solution is reached if the agents with zero-utility are just removed from the system. Despite doing so, we allow this participants to reach another decision forming a separate group.

Figure 8 shows the evolution of the criteria in such a case. It can be clearly observed how more that one decision is taken. In this case, the network is divided into 4 groups: the bigger one arrives to the best decision, and another group formed by two agents arrives to another private agreement. Finally, another two participants remain isolated. The dendrogram of this figure shows the group formation, and the last graphic shows the global utility value, taking into account the sum of the solutions reached by the different groups.

Finally, the performance of the algorithm has been analyzed using networks of different sizes. The obtained results are shown in Figure 9. Experiments were run in a 3.2 GHz Intel Core i5, with 8Gb of RAM. Random networks from 100 to 1000 nodes have

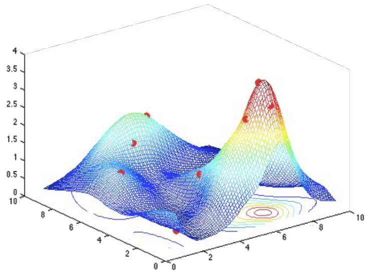

Fig. 7: Initial and final states for an AHP process allowing to break links and reconnect to near neighbors. This solutions guarantees the convergence of a subgroup to the best possible decision, along with another agreements around suboptimal solutions.

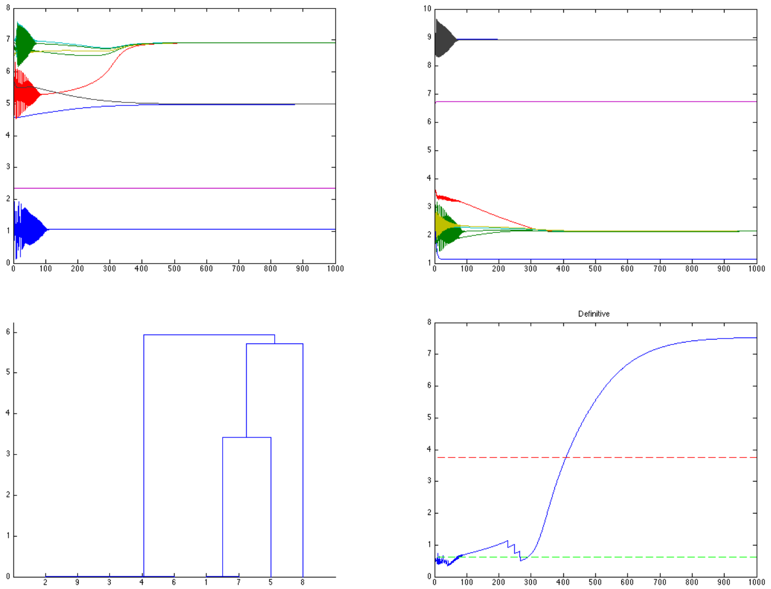

Fig. 8: (Top) evolution of the criteria and convergence into separated groups. (Bottom) Group division and global utility obtained by this process

been generated, with 10 repetitions of each size. The AHP process has been executed over these networks and the obtained execution time has been averaged. The execution time takes into account the AHP process exclusively. The time needed to create the network and define the individual weights for the different criteria and alternatives are not included. The experiments show a quadratic cost for the algorithm in the studied network sizes. Bigger networks need to be analyzed. The main drawback of the current implementation is that the calculation of the φ parameter (see Equation 4) to guarantee the convergence of the method is a centralized one (the φ parameter is related with the value of the Lipschitz constant for each utility function) and the cost is too high to be calculated in bigger networks (beyond 4 magnitude orders with respect to the execution time).

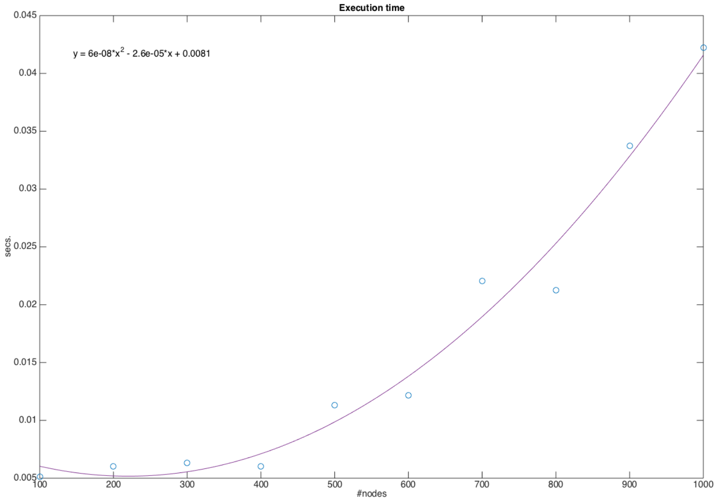

Fig. 9: Execution time of the algorithm with different network sizes

5 Conclusions

This work has presented a method based on a combination of consensus and gradient ascent to solve group AHP in a decentralized environment, where the participants in the decision making process exchanges their preferences with their direct neighbors to reach an agreement that allows the team to select the alternative with the highest utility for the group. This work can be easily extended to the case of having networks of preferences (ANP) or the case of changes in the local priorities or the weights of the criteria during the process.

Acknowledgements

This work is supported by the PROMETEOII/2013/019 and TIN2015-65515-C4-1-R projects of the spanish government.

References

1. Johansson et al., B.: Subgradient methods and consensus algorithms for solving convex optimization problems. In: Proc. of IEEE CDC'08. pp. 4185–4190 (2008)
2. Zanella et al., F.: Newton-raphson consensus for distributed convex optimization. In: Proc. of IEEE CDC-ECC'11. pp. 5917–5922 (2011)
3. Zanella et al., F.: Asynchronous newton-raphson consensus for distributed convex optimization. In: Proc. of IFAC NecSys'12 (2012)
4. Zanella et al., F.: Multidimensional newton-raphson consensus for distributed convex optimization. In: Proc. of ACC'12. pp. 1079–1084 (2012)
5. Askari-Sichani, O., Jalili, M.: Large-scale global optimization through consensus of opinions over complex networks. Complex Adaptive Systems Modeling 1(1), 11 (2013)
6. Boccaletti, S., Bianconi, G., Criado, R., del Genio, C., nes, J.G.G., Romance, M., na Nadal, I.S., Wang, Z., Zanin, M.: The structure and dynamics of multilayer networks. Physics Reports (2014)
7. Cai, K., Ishii, H.: Average consensus on arbitrary strongly connected digraphs with dynamic topologies. In: Proc. of ACC'12. pp. 14–19 (2012)
8. Cavalcante, R., Rogers, A., Jennings, N.: Consensus acceleration in multiagent systems with the chebyshev semi-iterative method. In: Proc. of AAMAS'11. pp. 165–172 (2011)
9. Elhage, N., Beal, J.: Laplacian-based consensus on spatial computers. In: AAMAS. pp. 907–914 (2010)
10. Frasca, P., Carli, R., Fagnani, F., Zampieri, S.: Average consensus on networks with quantized communication. Int J Robust Nonlin 19(16), 1787–1816 (2009)
11. et al. Hu, H.X.: Group consensus in multi-agent systems with hybrid protocol. Journal of the Franklin Institute 350(3), 575 – 597 (2013)
12. Ishizaka, A., Labib, A.: Review of the main developments in the analytic hierarchy process. Expert Systems with Applications 38(11), 14336 – 14345 (2011)
13. Ji, Z., Lin, H., Yu, H.: Leaders in multi-agent controllability under consensus algorithm and tree topology. Systems & Control Letters 61(9), 918 – 925 (2012)
14. K. Yuan, Q.L., Yin, W.: On the convergence of decentralized gradient descent. Tech. Rep. Report 13-61, UCLA CAM (2014)
15. Lancichinetti, A., Fortunato, S.: Consensus clustering in complex networks. CoRR abs/1203.6093 (2012)
16. Matei, I., Baras, J.: Performance evaluation of the consensus-based distributed subgradient method under random communication topologies. IEEE Sig. Proc. 5(4), 754–771 (2011)
17. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE 95(1), 215–233 (2007)
18. Olfati-Saber, R., Murray, R.M.: Consensus problems in networks of agents with switching topology and time-delays. IEEE TAC 49(9), 1520–1533 (2004)
19. Pedroche, F., Rebollo, M., Carrascosa, C., Palomares, A.: On the convergence of weighted-average consensus. CoRR arxiv:1203.6093 [math.OC] (2013)
20. Pereira, S., Pages-Zamora, A.: Consensus in correlated random wireless sensor networks. IEEE Sig. Proc. 59(12), 6279–6284 (2011)
21. Salazar-Ramirez, N., Rodríguez-Aguilar, J.A., Arcos, J.L.: Robust coordination in large convention spaces. AI Communications 23, 357–372 (2010)