Using Graph Properties and Clustering Techniques to Select Division Mechanisms for Scalable Negotiations

Ivan Marsa-Maestre, Catholijn M. Jonker, Mark Klein and Enrique de la Hoz

Abstract This paper focuses on enabling the use of negotiation for complex system optimisation, which main challenge nowadays is scalability. Our hypothesis is that analysing the underlying network structure of these systems can help divide the problems in subproblems which facilitate distributed decision making through negotiation in these domains. In this paper, we verify this hypothesis with an extensive set of scenarios for a proof-of-concept problem. After selecting a set of network metrics for analysis, we cluster the scenarios according to these metrics and evaluate a set of mediation mechanisms in each cluster. The validation experiments show that the relative performance of the different mediation mechanisms change for each cluster, which confirms that network-based metrics may be useful for mechanism selection in complex networks.

1 Introduction

A wide range of real world systems can be modelled as dynamic sets of interconnected nodes [13, 21]. The adequate management of complex networked systems is becoming critical for industrialized countries, since they keep growing in size and complexity. An important sub-class involves autonomous, self-interested enti-

I. Marsa-Maestre (✉) · E. de la Hoz
University of Alcala, Madrid, Spain
e-mail: ivan.marsa@uah.es

E. de la Hoz
e-mail: enrique.delahoz@uah.es

C.M. Jonker
Technical University of Delft, Delft, The Netherlands
e-mail: c.m.jonker@tudelft.nl

M. Klein
Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: m_klein@mit.edu

K. Fujita et al. (eds.), *Modern Approaches to Agent-based Complex Automated Negotiation*, Studies in Computational Intelligence 674,
DOI 10.1007/978-3-319-51563-2_5

ties (e.g. drivers in a transportation network). The self-interested nature of the entities in the network causes the network to deviate from socially-optimal behaviour. This leads to problems related to unavailability and inefficient use of resources, such as severe traffic jams or casualties in evacuations. New techniques are needed to manage these exponentially growing complex self-interested networks (CSIN) that form the social infrastructures we rely on for progress and welfare. Different fields of research are working on these challenges, but, so far, with only mixed success. Optimization techniques are especially suited to address large-scale systems with an underlying network structure, usually with a divide and conquer approach [27, 32]. However, their performance severely decreases as the complexity of the system increases [23], and with the presence of autonomous entities which deviate from the globally optimal solution, thus harming the social goal. Negotiation techniques are known to be useful to handle self-interested behaviour, but scale poorly with problem size and the intricacies of interdependencies [14]. We focus on distributed, mediated solutions, where a mediator first divides the problem into interconnected subproblems, and then the agents interact (by means of negotiation techniques) to evolve into a solution by themselves. However, given the wide variety of CSIN domains and the inherent variability of CSIN scenarios even within a single domain, intending to find a one-size-fits-all mechanism is unrealistic. Instead, our hypothesis is that the underlying network structure may be used to characterize CSIN scenarios, and to select the most adequate mechanism for each scenario. In this paper, we contribute to test this hypothesis in the following way:

- We propose a proof-of-concept domain for CSINs (“chessboard-evacuation”), and generate a number of scenarios in different categories for it (Sect. 3).
- We select a set of metrics based on graph theory to analyze these scenarios (Sect. 4).
- We cluster the scenarios according to the aforementioned metrics, and then apply a collection of distributed, mediated division approaches to each cluster. Experiments show how the relative performance of the different mediation mechanisms change for each cluster (Sect. 5).

2 Complex Self-interested Networks (CSIN)

Network models are a suitable way to represent many real-world systems [22, 24]. This paper focuses on a particular set of networked systems, where network structure and element behaviour may change dynamically, where there is a social goal or desired behaviour for the network as a whole, and where there are autonomous elements (agents) with individual objectives (also called preferences or utility spaces), usually in conflict with the social goal or among themselves. There are a great number of real-world problems fitting into this category, like electricity grids, transportation infrastructures or cellular communication systems. We call these systems Complex Self-Interested Networks (CSIN). The problem of achieving efficient behaviours in these systems in terms of both social and individual goals is what we call CSIN behaviour optimization. Techniques potentially suited for CSIN include auctions, optimization techniques, and negotiation protocols. Combinatorial auctions [26, 35] can enable large-scale collective decision-making in nonlinear domains, but only of limited type (i.e., negotiations consisting solely of resource/item allocation decisions). Multi-attribute auctions [1, 4, 12] are also aimed only at purchase negotiations and require full revelation of preference information. Constraint-based and other optimization tools [3, 18, 33] offer good solutions with interdependent issues, but are not equipped to deal with self-interested parties. The distributed, adaptive, and self-interested nature of CSIN suggests the use of negotiation techniques. The negotiation research literature offers solutions for problems with one issue (typically price) or a few independent issues [1, 7, 9, 25]. However, these solutions are demonstrably sub-optimal for negotiations with multiple interdependent issues [14]. Attempts to address this challenge [11, 16, 36] face serious limitations in terms of outcome optimality, strategic stability and scalability. These three criteria are key performance indicators for the success of optimization systems in real-world CSIN infrastructures, due to their continuous increase in network size, structural complexity and dynamics [28]. Promising results in overcoming the aforementioned limitations were achieved by developing negotiation mechanisms suited for complex system optimization. These negotiation mechanisms build on techniques from computer science (e.g. nonlinear optimization, complexity analysis), which enable a clever search of the agent utility spaces. This allows to reduce the combinatorial size of the problem and to increase the optimality of the negotiation outcomes (agreements). However, due to the high diversity of complex negotiation scenarios, the different approaches are specifically tailored for particular domains, and it is very unlikely to find an approach which can be used to tackle any arbitrary complex system [17]. However, real world CSIN problems are not arbitrary; they have an underlying network structure. Our hypothesis is that this structure can be exploited to select the most adequate mediation mechanisms from a library of available approaches.

3 Proof-of-Concept Domain: Chessboard Evacuation

To test our hypothesis, we have devised a proof-of-concept domain for a preliminary validation experiment. This is what we have called the chessboard evacuation problem, which has some resemblances with the coordinating pebble motion on graphs problem [5, 15], and with cooperative robot path finding [29]. An example chessboard evacuation scenario is shown in Fig. 1.

Let us assume that pawns want to evacuate the chessboard as quickly as possible through the exit represented with an arrow. Pawns can move one square per time unit in the vertical or horizontal axis, and they lose time when they collide. This problem has the main characteristics of CSIN:

- The coordinated pebble motion and the cooperative robot path finding problems are known to be NP-hard [10, 29].

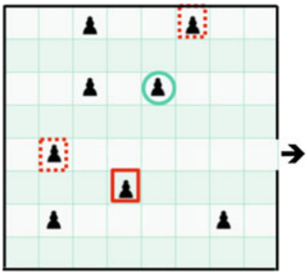

Fig. 1 Chessboard evacuation scenario

- Once a solution has been found, some pawns may disagree on having to wait or to take longer than optimal paths to avoid conflicts, which may make them deviate from the solution, causing collisions and inefficient behaviours.

3.1 Formalisation of the Problem

An instance of the chessboard evacuation problem can be seen as a tuple $\langle B, P, g\rangle$, where:

- $B = \langle N, E\rangle$ is a graph representing the board, where each node $n \in N$ is a space in the board, and each edge $e \in E$ connects two adjacent spaces. Furthermore, for any two nodes $n, m \in N$, $e(n, m)$ denotes an edge between n and m.
- P is the set of pawns. Each pawn p is characterized by its initial position $n_{p,0} \in N$, which refers to a node in the graph B.
- $g \in N$ is the goal, representing the square from which pawns will evacuate the board.

A potential solution to the problem would be a set of routes $R = \{r_p | p \in P\}$, where each route $r_p = \{n_{p,t} \in N | t = 0, \dots, \tau_p\}$ represents the sequence of positions occupied by pawn p at each time t. The evacuation time for pawn p is denoted by τ_p. For a solution R to be valid, all pawns need to travel a continuous path from their initial position to the goal and no pawns can be allowed to be at the same position at the same time, that is:

- $n_{p,\tau_p} = g \forall p$
- Consecutive positions in a route correspond either to the same node or to connected nodes in the graph, that is, nodes which are connected by an edge:

$$\forall t, p : e(n_{p,t}, n_{p,t+1}) \in E \text{ or } n_{p,t} = n_{p,t+1} \tag{1}$$

- $\forall t, \forall p, p' \in P : r_{p,t} \neq r_{p',t}$

For any solution R, the time $\tau(R) = \max_{p \in P} |\tau_p|$ is called evacuation time. A solution R_i is assumed to be better than another solution R_j if $\tau(R_i) < \tau(R_j)$.

3.2 Modeling Agent Self-interests

Of course, in complex self-interested networks, solutions are not implemented either in a centralized or in a one-shot manner. Even if solutions are computed in a centralized way, agents are self-interested and autonomous, and can deviate at any point from any externally imposed plan. To represent this in the chessboard problem, we model agent decision making as follows:

- Pawns give a value v to each node $n \in N$, equal to the shortest path length from this node to the goal g. Therefore, pawns self-interest is normally to move to the lowest value adjacent node.
- Pawns are considered to be in conflict if the values of the nodes corresponding to their current positions are equal, since that means those pawns would necessarily collide in their way to the goal if they take their optimal paths and do not wait.
- At any time t, pawns which are asked (by imposition or negotiation) to make a move which is suboptimal (i.e., does not minimize value of the next node), will follow their optimal path (i.e., they will deviate from the proposed solution) with probability
$$\pi = 1 - \frac{1}{n_c}, \tag{2}$$
  where n_c is the expected number of conflicts the pawn would be in in the next time unit if it followed the suggested route. This models pawn risk aversion, in the sense that being in a high number of conflicts increases the expected evacuation time. Note that, if the proposed route does not involve conflicts for the pawn, it will follow it without any problem.

Since pawns are still self-interested as defined above, they may produce collisions. If two or more pawns collide in the same node, they are all sent back to the node they came from. Collisions propagate backwards, so any pawns trying to enter a node where another pawn has been sent back due to a collision would also suffer a collision.

3.3 Categories of Scenarios

For the verification of the hypothesis to be significant enough, it has to be evaluated in a wide variety of scenarios. In order to achieve this variety, we have considered three different categories of scenarios, according to the properties of the underlying graph B:

- *Chessboards (CB)*: in this category, the graph is a 8-by-8 square lattice (resembling a chessboard), where we randomly add a number of obstacles (non-transitable nodes) for diversity.

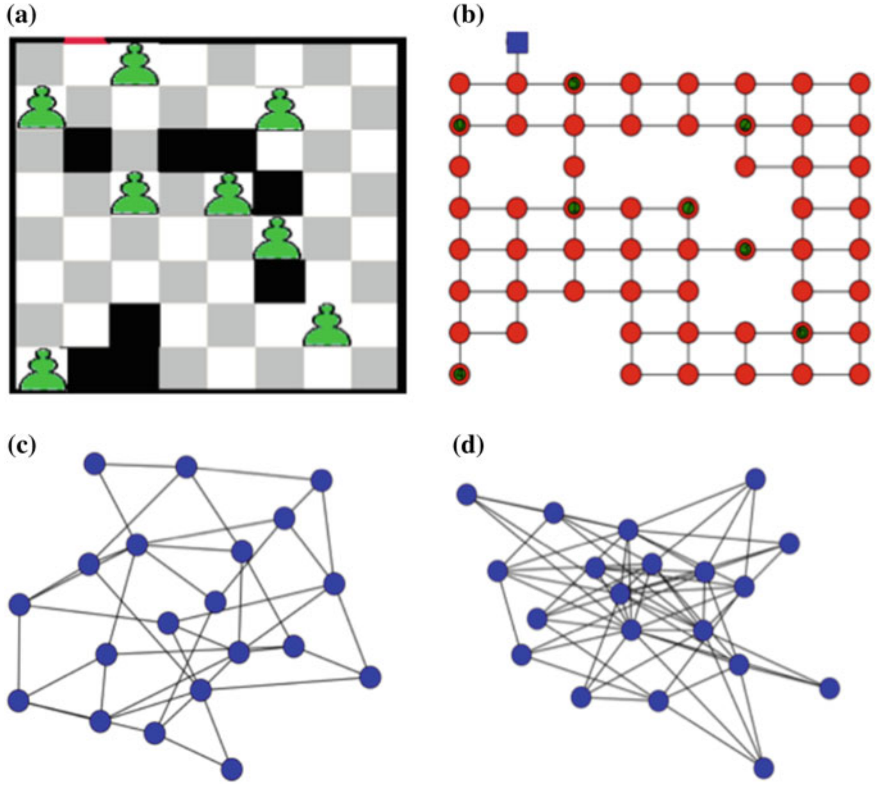

Fig. 2 Examples of each generated graph category: **a** chessboard with 8 obstacles; **b** corresponding lattice graph; **c** ER graph with $p = 0.2$; **d** BA graph with $\alpha = 4$

- *Erdös–Renyi graphs (ER)*: in this category, graphs are generated randomly using the Erdös–Renyi model [20]. In the ER model, every pair of nodes $\{m, n \in N\}$ has a finite probability p of being connected.
- *Barabasi–Albert graphs (BA)*: in this category, graphs are generated randomly using the Barabasi–Albert model [20]. In the BA model, after an initial number of nodes $|N|_0$ has been placed, subsequent nodes are added one-by-one to the graph, with each new node being connected to exactly α of the existing nodes. The probability of a new node connecting to an existing node i is $p_i = \frac{k_i}{\sum_j k_j}$, where k_i is the degree of existing node i. This is called preferential attachment.

For each category of scenarios, different sub-categories were created by varying the number of obstacles, the probability of connection p, and the attachment parameter α. In each scenario, a node is randomly chosen as the goal (in the case of chessboards, one of the peripheral nodes). Figure 2 shows an example for each category of graphs.

4 Graph Metrics for Scenario Characterization

With the model for self-interested agents described above, we run a number of simulations in the aforementioned scenario categories. The details of these experiments are beyond the scope of this paper, and can be found in [31]. The experiments allowed us to select a number of graph metrics from the literature which were significantly correlated to the evacuation times of the simulations. The selected metrics are the following:

- *Graph order*: the number of nodes in the graph.
- *Graph diameter*: the longest distance between any pair of nodes in the graph [19].
- *Wiener index*: gives a measure of graph complexity from the distances in the graph. It is computed as $W(G) = \frac{1}{2}\sum_{i=0}^{|N|}\sum_{j=0}^{|N|} d(n_i, n_j)$, where $d(n_i, n_j)$ is the shortest distance between nodes n_i and n_j [34].
- *Graph density*: the ratio between the number of edges in the graph and the maximum number of edges (that is, if it were a fully-connected graph). For undirected graphs, this density is computed as $D = \frac{2|E|}{|N|(|N|-1)}$.
- *Clustering coefficient*: a measure of the degree to which nodes in a graph tend to cluster together. The cluster coefficient of a graph is computed as the average of the local clustering coefficient of its nodes, which is computed for each node as the ratio between the number of links between its neighbors and the maximum number of links between them (that is, if they were fully connected).
- *Assortativity*: the correlation between the degree (number of neighbors) of adjacent nodes [30].
- *Entropy of betweenness centrality*: Centrality metrics measure the importance of a node within a graph. In particular, betweenness centrality of a node is the ratio of shortest paths in the graph which traverse the node. It is computed as $C_B(n) = \sum_{s,t \in V} \frac{\sigma(s,t|n)}{\sigma(s,t)}$, where $\sigma(s, t)$ is the number of shortest paths between nodes s and t, and $\sigma(s, t|n)$ is the number of such paths where n acts as a bridge. From this metric, we can assess the complexity of a graph using Shannon information theory, by transforming the betweenness centrality metric into a probability function $p(n_i) = \frac{C_B(n_i)}{\sum_{j=0}^{|N|} C_B(n_i)}$, and computing the entropy as $H(G) = -\sum_{i=1}^{|N|} p(n_i)\log(p(n_i))$.

These metrics have been used to divide the scenarios in clusters for the experiments, as described in the following section.

5 Using Graph Metrics for Mechanism Selection

Our hypothesis is that the selected set of metrics may be used as a basis for mechanism selection in CSIN problems, and in particular in the evacuation problem considered in this paper. To verify this, we have devised a number of distributed mediation mechanisms, and we have performed an extensive set of experiments with them in the generated scenarios. We have also clustered the scenarios according to the aforementioned metrics, so that we can see the influence of these metrics in the performance of the different mechanisms.

5.1 Distributed, Mediated Division Approaches

As mentioned above, our interest regarding mechanisms focuses on distributed, mediated solutions, where a mediator first divides the problem into interconnected subproblems, and then the agents are let to evolve into a solution by themselves. The two key factors here are, first, how to divide the problem, and then, how to interconnect the different subproblems so that the self-interested agents can autonomously evolve to an emergent, efficient solution. In the particular case of the chessboard evacuation problem, we have chosen to dynamically divide the chessboard graph into subgraphs (lets call them worlds), and let the pawns in each world negotiate the paths they will take. In addition, we make pawns within a world negotiate about where to place the entrance arrows to their world, as seen in Fig. 3a. This entrance arrow placement is the negotiation technique (very simple, in this case) which interconnects the subproblems and guarantees emergence and incentive for cooperation. Since pawns can govern how other pawns enter their worlds, they can ensure that the pawns conceding in a negotiation (i.e., sacrificing their own utility to solve a conflict) are not exposing themselves to further conflicts. In this way, the number of conflicts in a world never increases. This invariant guarantees the progress of the approach, in contrast with the situation we had with the exponential increase due to collisions. An example of this is depicted in Fig. 3b, where pawns within the lower-right world (LR) are in conflict. Any pawn which concedes to the other would be then in conflict with the pawn in the lower-left world (LL). However, they have placed the entrance from LL to LR ensuring that this conflict is no longer possible, since any pawn coming from LL would be further from the goal than any pawn in LR. Therefore, a pawn in LR can agree to wait with the guarantee that its utility loss is bounded (one time unit). This provides an incentive both to accept the proposed problem division and to cooperate within it.

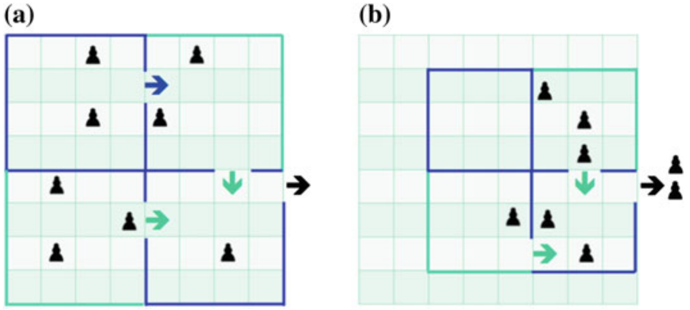

Fig. 3 Distributed, mediated division example

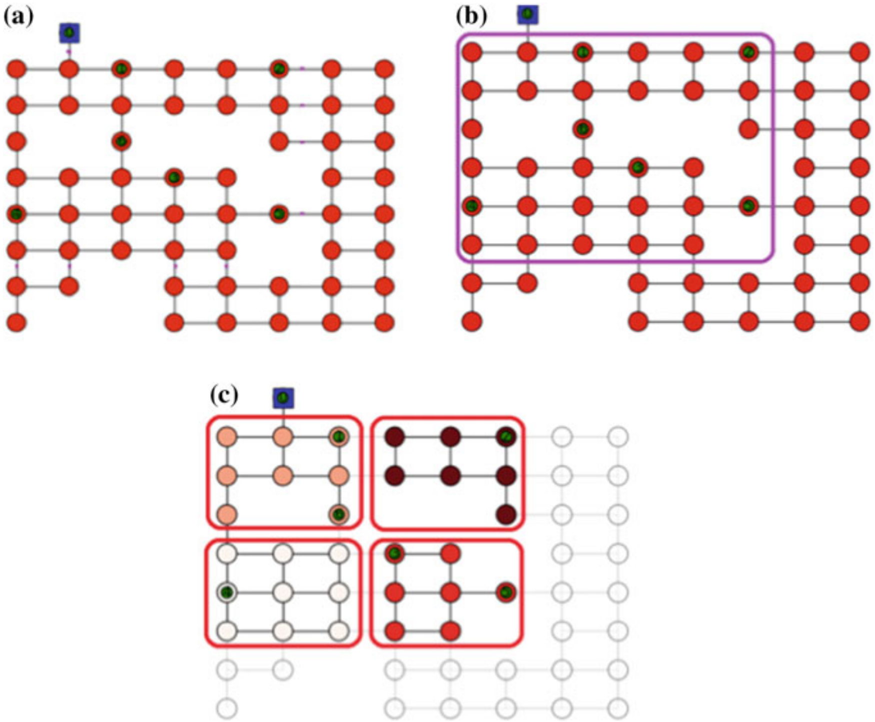

Fig. 4 Example of symmetric division. **a** Chessboard to divide. **b** Minimum square containing all the pawns and the exit. **c** Worlds generated

5.1.1 Symmetric Division

The symmetric division (Fig. 4) is based on dividing the system into four symmetric subsystems. To apply this division method to the chessboard scenarios, first the graph is reduced by getting the minimum square that contains all the pawns and the exit of the system. By using this graph reduction, the evacuation plan is adapted to the congestion state at any given moment. Once the graph has been reduced, the reduced graph is symmetrically divided into four symmetric subgraphs, creating four worlds.

The position of the random graphs' nodes is not fixed, so there can be many different representations of a network. Thus, to apply the symmetric division to random graphs, we need to assign a position on the plane for each node. For that purpose, the Fruchterman–Reingold algorithm is used [8].

5.1.2 Distance Division

Distance division is based on grouping close nodes with similar distances to the exit. Again, we first reduce the graph, in this case by removing those nodes that are further away from the exit than the furthest away pawn (Fig. 5).

Then, the reduced graph is divided into three sections according to the distance of the nodes to the exit. The first section leads to one world. Second and third sections can be too big or even disconnected, so they may have to be divided. To this end, the disconnected components of the sections' graphs are found. Small components are directly mapped to worlds, while big components with more than one node connecting the component to the previous section are split up into two halves. The two halves are created by starting at the two furthest nodes connecting the component to the previous section and then expanding the graph from them (that is, taking adjacent nodes further apart, then adjacent nodes to the new graph, and so on) until all the nodes within the component have been reached by one of the halves.

5.1.3 Pawns Distribution Division

The pawns distribution division (Fig. 6) is based on creating worlds so they have a similar number of pawns. With this division method, there is no need to reduce the system's graph, as the method is adapted to the congestion state by itself.

The method begins by dividing the system into three sections. The first section is created by starting at the exit and expanding the graph from it until 1/5 of the pawns are reached. The second section is generated starting at the nodes connecting the section to the previous section and expanding the graph from them until 2/5 of the pawns are reached. The third section contains the remaining nodes. Once the sections have been generated, the first section is mapped to a world, while the second and third sections are divided. As in the previous method, first the disconnected components of the sections are found. Those components with less than half the number of pawns of the corresponding section are directly mapped to worlds, while components with more pawns and more than one node connecting the component to the previous section are divided into two worlds, each one containing half the pawns of the component.

5.1.4 Community Division

In graph theory, a community is a set of nodes such that there are many connections between nodes of the same community and few connections with nodes outside the community [19]. The community division consist on reducing the graph by removing those nodes further from the exit than the furthest pawns, and, afterwards, using the Louvain algorithm [2] to find the graph communities, map each of these communities to a world (Fig. 7).

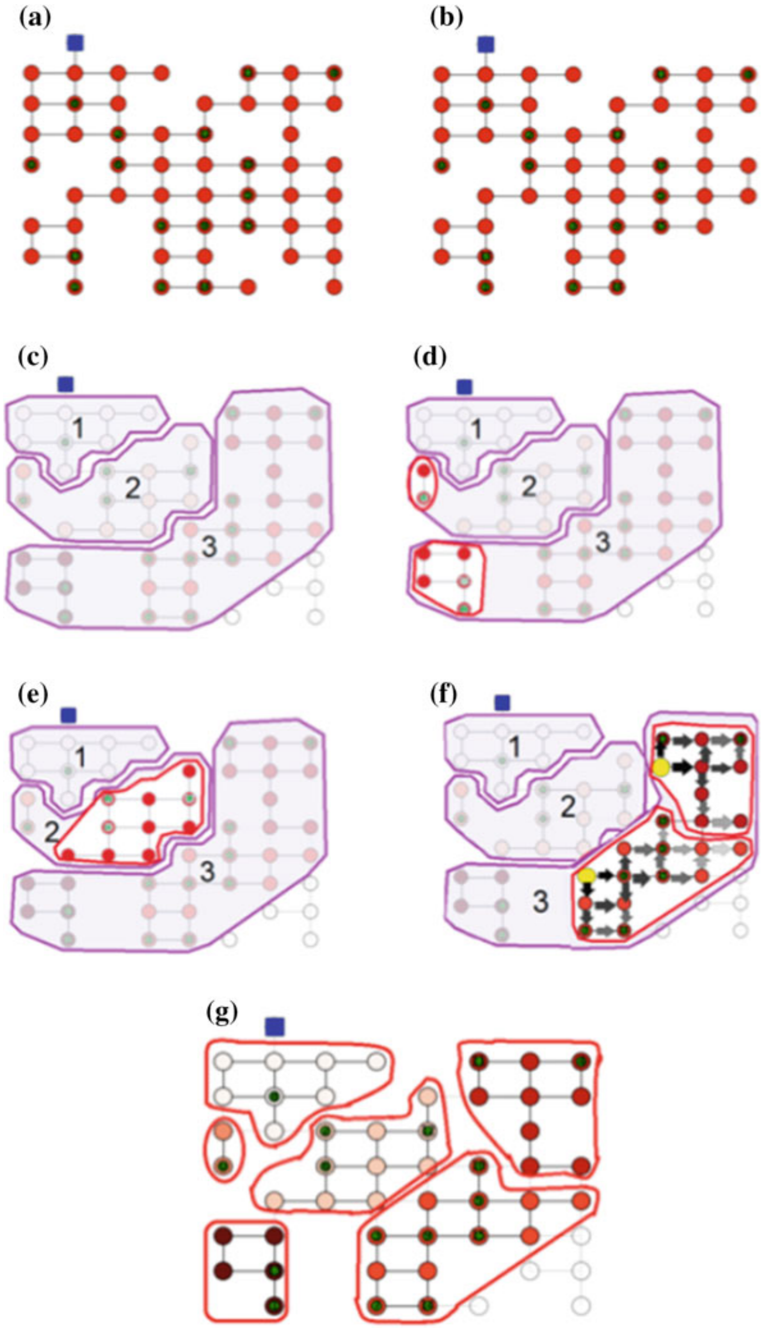

Fig. 5 Example of distance division. **a** Chessboard to divide. **b** Reduced graph. **c** Sections generated. **d** Worlds created from small components. **e** Wolds created from components with just one node connecting the component with the previous section. **f** Division of big components with more than one node connection the component with the previous section. **g** Worlds generated

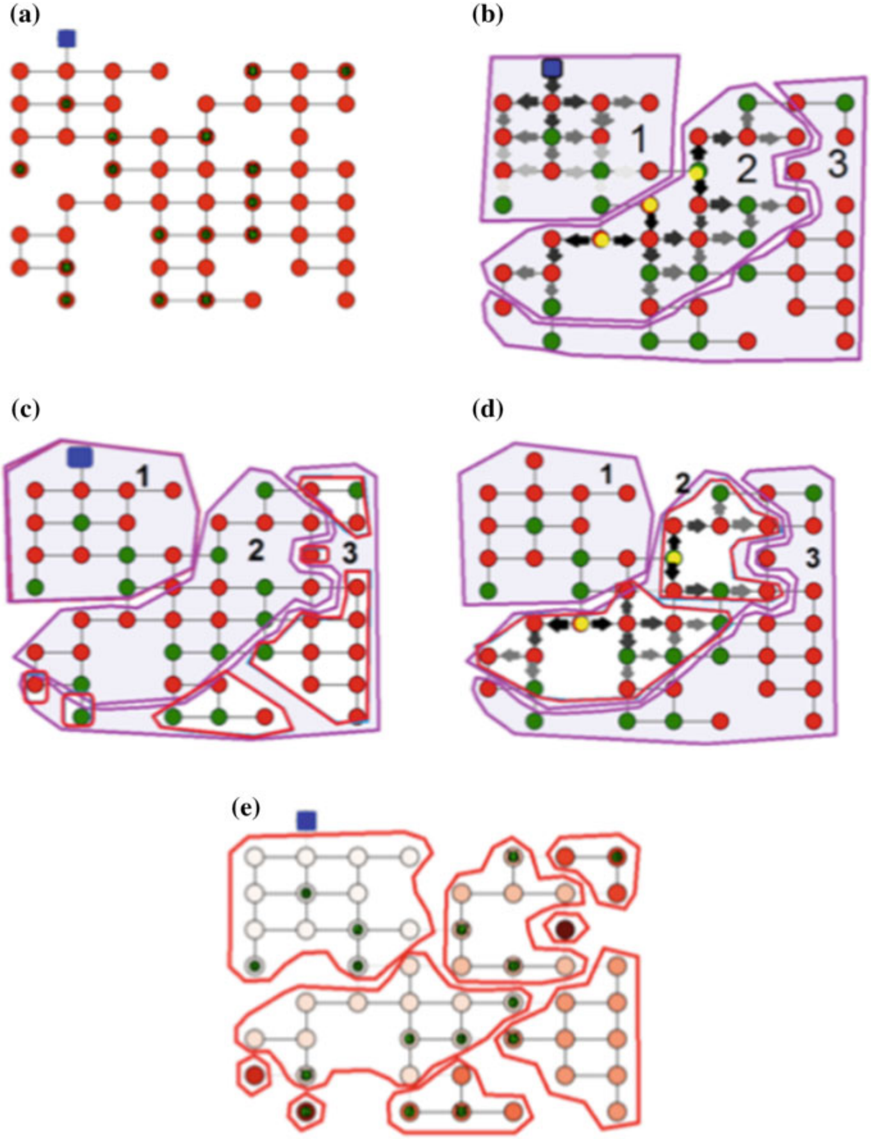

Fig. 6 Example of pawn-based division. **a** Chessboard to divide. **b** Generation of the three sections. **c** Worlds generated from components with few pawns. **d** Division of components with many pawns. **e** Worlds generated

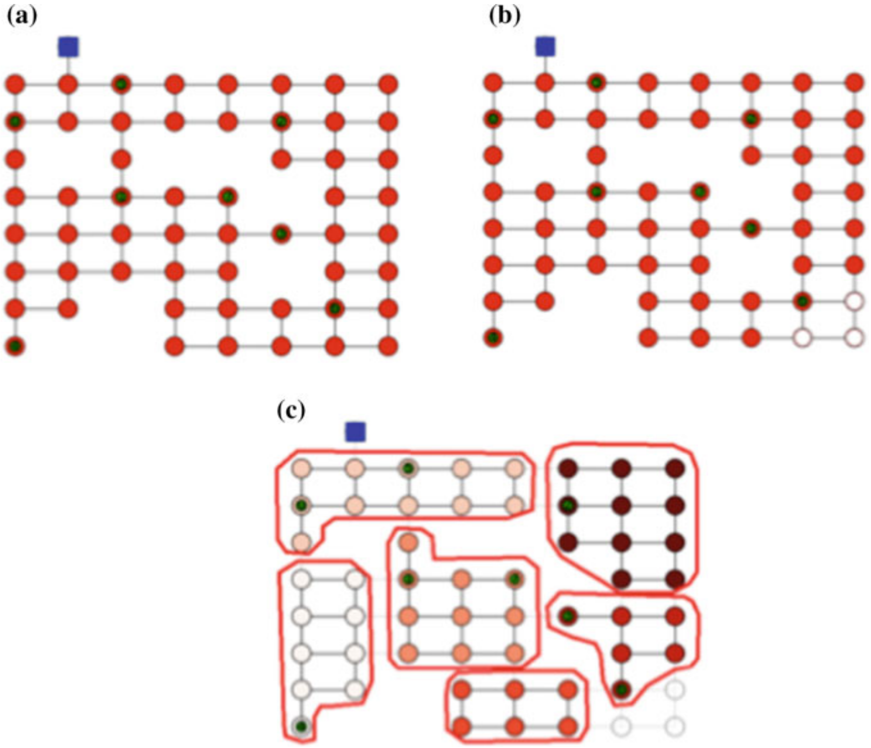

Fig. 7 Example of community division. **a** Chessboard to divide. **b** Reduced graph. **c** Worlds generated

5.2 Negotiation Between Agents

Once the division has been made, agents need to negotiate about where to place the doors between worlds. Since each possible door-placing schema is a binary vector (with one bit per boundary square, which can be set to 1 or 0 depending on whether there should be a door or not in it, respectively), we have used the approach in [14], which is a negotiation mechanism specifically designed for negotiation of complex contracts with binary issues.

5.3 Experimental Setting

We have generated a total of 1600 scenarios for the categories described in Sect. 3. In particular, we generated 50 scenarios for each combination of the following parameters per category:

Table 1 Comparison of the evacuation times (in minutes) of the different mediation approaches in the three detected clusters

Scenario cluster	Reference	Sym.	Dist.	Pawn	Com.
C1	48.56	29.98	36.63	42.56	**27.44**
C2	240.51	163.94	**129.75**	144.92	156.20
C3	264.91	206.42	288.62	**180.60**	220.66

- Chessboards with number of obstacles in $|O| = \{8, 12, 16, 20\}$.
- ER graphs with $|N| \in \{52, 56\}$ (same number of nodes as the chessboards with 8 and 12 obstacles) and $p \in \{0.6, 0.8, 0.10, 0.12\}$.
- BA graphs with $\alpha \in \{1, 2, 3, 4\}$.

For each scenario, we ran 50 simulations of the evacuation problem for each of the considered approaches, along with a reference approach (where no mediation mechanism is used), for a total 80,000 runs per approach (400,000 total simulations). At each simulation, we recorded the evacuation time τ. Finally, we clustered all scenarios according to the metrics described in Sect. 4, using an implementation of DBSCAN [6].

5.4 Experimental Results

The DBSCAN algorithm yielded three scenario clusters (C1 to C3). Table 1 shows the evacuation times for the different mediation approaches, averaged for each of the scenario clusters. We have also represented the average times for the reference approach. In general, we can see that all mediation approaches outperform the reference unmediated approach (except for a slight disadvantage in using the distance-based approach in the C3 cluster). However, we can see there are significant differences in the relative performance of the approaches for each cluster. In the C1 cluster, symmetric and community based divisions work better than the other approaches (above 20% difference). In the C2 cluster, distance-based division improves over 11% with respect to the next top approach, and in the C3 cluster, pawn-based division outperforms the other approaches by 12.6%. We can conclude that graph metrics can help to select the appropriate mechanism to use when facing a given scenario, providing a significant advantage on average evacuation times than choosing a different strategy or no strategy at all.

6 Discussion and Conclusions

The main hypothesis of our work is that underlying structural properties in Complex Self-Interested Networks (CSIN) can be used to decide which mechanisms to use to enhance the performance of such networks. To validate this hypothesis, in this paper we present a chessboard evacuation problem as a proof of concept domain for CSINs. For this domain, we systematically generate a wide variety of scenarios in three categories of graph structures (lattices, Erdös–Renyi graphs and Barabasi–Albert graphs), and characterize them according to a set of metrics selected from graph theory. Then we consider a number of mediated, distributed approaches to facilitate reaching efficient solutions to the evaluation problems, and run an extensive set of simulations with all of them. We analyse the simulation data by clustering the scenarios using the aforementioned metrics. Experiments show that the relative performance of the

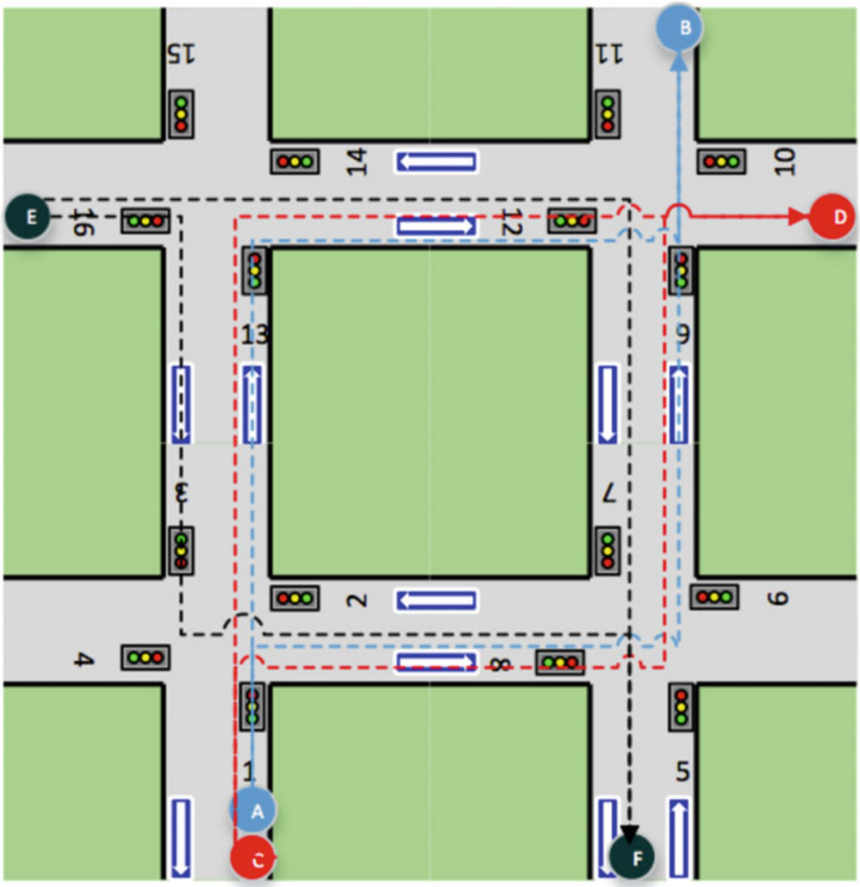

Fig. 8 Traffic management scenario

different mediation approaches significantly change in each cluster, which backs up the idea that analysing the graph properties of a problem can help choosing a suitable mechanism to address it. Though our experiments yield promising results, there is still plenty of work to be done in this area. We are in the process to validate the suitability of the metrics to perform a priori mechanism selection, by training classifiers (e.g. random trees) in large sets of scenarios and using them to predict which would be the best approach to use when confronted with new scenarios. We also want to get more diverse scenario sets, since in this case the clusters detected by DBSCAN where populated in their majority (about 90%) by scenarios in a single generation category (either chessboards, Erdös–Renyi or Barabasi–Albert graphs), which demonstrates a significant generation bias in the experiment set. Finally, we are interested in generalizing the approach to other domains out of the chessboard proof-of-concept. For instance, we are working on a transportation management scenario (Fig. 8), where GPS navigator agents within a given area automatically negotiate about the state of the traffic lights allowing entrance to their area, so that congestion is mitigated.

Acknowledgements This research has been partially supported by two research grants from the Spanish Ministry of Economy and Competitiveness (grants TEC2013-45183-R and TIN2014-61627-EXP).

References

1. M. Bichler, An experimental analysis of multi-attribute auctions. Decis. Support Syst. **29**(3), 249–268 (2000)
2. V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks. J. Stat. Mech.: Theory Exp. **2008**(10), P10008 (2008)
3. A. Chechetka, K. Sycara, No-commitment branch and bound search for distributed constraint optimization, in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems* (ACM, 2006), pp. 1427–1429
4. E. de Oliveira, J.M. Fonseca, A. Steiger-Garção, Multi-criteria negotiation on multi-agent systems, in *Proceedings of the CEEMAS'99* (1999), p. 190
5. B. de Wilde, A.W. ter Mors, C. Witteveen, Push and rotate: cooperative multi-agent path planning, in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems* (International Foundation for Autonomous Agents and Multiagent Systems, 2013), pp. 87–94
6. M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd **96**, 226–231 (1996)
7. S. Fatima, M. Wooldridge, N.R. Jennings, Optimal negotiation of multiple issues in incomplete information settings, in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 3 (IEEE Computer Society, 2004), pp. 1080–1087
8. T.M. Fruchterman, E.M. Reingold, Graph drawing by force-directed placement. Softw. Pract. Exper. **21**(11):1129–1164 (1991)
9. N. Gatti, F. Amigoni, A decentralized bargaining protocol on dependent continuous multi-issue for approximate pareto optimal outcomes, in *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems* (ACM, 2005), pp. 1213–1214
10. O. Goldreich, *Finding the Shortest Move-Sequence in the Graph-Generalized 15-Puzzle is NP-Hard* (Springer, Heidelberg, 2011)

11. K. Hindriks, C. Jonker, D. Tykhonov, Avoiding approximation errors in multi-issue negotiation with issue dependencies, in *Proceedings of the 1st International Workshop on Agent-based Complex Automated Negotiations (ACAN 2008)* (Citeseer, 2008), pp. 1347–1352
12. J. Kalagnanam, D.C. Parkes, Auctions, bidding and exchange design, in *Handbook of Quantitative Supply Chain Analysis* (Springer, New York, 2004), pp. 143–212.
13. R. Kinney, P. Crucitti, R. Albert, V. Latora, Modeling cascading failures in the North American power grid. Eur. Phys. J. B-Condens. Matter Complex Syst. **46**(1), 101–107 (2005)
14. M. Klein, P. Faratin, H. Sayama, Y. Bar-Yam, Negotiating complex contracts. Group Decis. Negot. **12**(2), 111–125 (2003)
15. D. Kornhauser, G. Miller, P. Spirakis, in *Coordinating Pebble Motion on Graphs, the Diameter of Permutation Groups, and Applications* (IEEE, 1984)
16. M. Li, Q.B. Vo, R. Kowalczyk, Searching for fair joint gains in agent-based negotiation, in *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, vol. 2 (International Foundation for Autonomous Agents and Multiagent Systems, 2009), pp. 1049–1056
17. I. Marsa-Maestre, M. Klein, C.M. Jonker, R. Aydoğan, From problems to protocols: towards a negotiation handbook. Decis.Support Syst. **60**, 39–54 (2014)
18. P.J. Modi, W.-M. Shen, M. Tambe, M. Yokoo, Adopt: asynchronous distributed constraint optimization with quality guarantees. Artif. Intell. **161**(1), 149–180 (2005)
19. M. Newman, *Networks: An Introduction* (Oxford University Press, Oxford, 2010)
20. M.E. Newman, D.J. Watts, S.H. Strogatz, Random graph models of social networks. Proc. Natl. Acad. Sci. **99**(suppl 1), 2566–2572 (2002)
21. C. Osorio, M. Bierlaire, Mitigating network congestion: analytical models, optimization methods and their applications, in *90th Annual Meeting*, number EPFL-TALK-196049 (2011)
22. G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**(7043), 814–818 (2005)
23. M. Pelikan, K. Sastry, D.E. Goldberg, Multiobjective estimation of distribution algorithms, in *Scalable Optimization via Probabilistic Modeling*, (Springer, Berlin, 2006), pp. 223–248
24. E. Ravasz, A.-L. Barabási, Hierarchical organization in complex networks. Phys. Rev. E **67**(2), 026112 (2003)
25. F. Ren, M. Zhang, Bilateral single-issue negotiation model considering nonlinear utility and time constraint. Decis. Support Syst. **60**, 29–38 (2014)
26. Y. Sakurai, M. Yokoo, K. Kamei, An efficient approximate algorithm for winner determination in combinatorial auctions, in *Proceedings of the 2nd ACM Conference on Electronic Commerce* (ACM, 2000), pp. 30–37.
27. B. Schaller, New York citys congestion pricing experience and implications for road pricing acceptance in the united states. Transp. Policy **17**(4), 266–273 (2010)
28. S.H. Strogatz, Exploring complex networks. Nature **410**(6825), 268–276 (2001)
29. P. Surynek, An optimization variant of multi-robot path planning is intractable, in *Proceedings of the AAAI* (2010)
30. P. Van Mieghem, H. Wang, X. Ge, S. Tang, F. Kuipers, Influence of assortativity and degree-preserving rewiring on the spectra of networks. Eur. Phys. J. B **76**(4), 643–652 (2010)
31. M. Vega, Aplicacin de teora de grafos a redes con elementos autnomos. Master's thesis, University of Alcala, Spain (2014). (In Spanish)
32. P. Vytelingum, S.D. Ramchurn, T.D. Voice, A. Rogers, N.R. Jennings, Trading agents for the smart electricity grid, in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, vol. 1 (International Foundation for Autonomous Agents and Multiagent Systems, 2010), pp. 897–904
33. H. Wang, S. Liao, L. Liao, Modeling constraint-based negotiating agents. Decis. Support Syst. **33**(2), 201–217 (2002)
34. H. Wiener, Structural determination of paraffin boiling points. J. Am. Chem. Soc. **69**(1), 17–20 (1947)
35. M. Xia, J. Stallaert, A.B. Whinston, Solving the combinatorial double auction problem. Eur. J. Oper. Res. **164**(1), 239–251 (2005)

36. X. Zhang, R.M. Podorozhny, V.R. Lesser, Cooperative, multistep negotiation over a multidimensional utility function, *Artificial Intelligence and Soft Computing* (Springer, Berlin, 2000), pp. 136–142