Holonification of a Network of Agents Based on Graph Theory

Monireh Abdoos, Ahmad Esmaeili, and Nasser Mozayani

Computer Engineering Department, Iran University of Science and Technology, Tehran, Iran

Abstract. A multi-agent system consists of a group of interacting autonomous agents. The key problem in such a system is coordination and cooperation, i.e. how to ensure that individual decisions of the agents result in jointly optimal decisions for the overall system. This problem becomes even more serious when the number of the agents is large. Holonic model is an effective method to manage large scale problems. In holonic approaches, the formation of the initial holons is very critical and has a great influence on their performance and effectiveness. In this paper, we use a graph based modelling approach to group a population of agents with a greedy method, driven by a very simple and effective quality measure. The proposed method is evaluated by applying it to an urban traffic problem as a case study and it is shown the proposed method produces better results.

1 Introduction

Multi-Agent Systems (MASs) have often been effective tools for simulation and modelling of distributed applications [1]. Most of real world applications are complex and large scale in which we usually find a great number of interacting entities. The problems modelled by multi-agent systems can be formulated as decision making problems where an agent needs to decide which action to execute in order to maximize its objective function. Optimization of the decision making process in multi-agent systems is very challenging due to the fact that each agent needs to take into account other agents in the system. An approach to cope with large scale systems is to organize agents towards a common goal where each agent interacts with the other agents according to a network topology [2]. In the organization paradigm interactions among the agents are optimized and reduced effectively which makes it a very popular technique to manage the complexity of large systems. In multi-agent systems, an organization is defined as the collection of roles, relationships and authority structures which govern the behaviour of the system. There are many models proposed for the concept of organizations in multi-agent system. The most commonly ones used are: hierarchies, holarchies, coalitions, teams, congregation, societies, federations, markets, matrix organization and compound organizations [3]. In this paper, we will focus on holarchies and holonic systems.

Every holonic system needs a holonification process for formation of structure of the holons [4]. In other words, the aim in holonification is to specify the way that agents organize inside of a holon and make different levels in the holarchy. Holonification

G. Jezic et al. (Eds.): KES-AMSTA 2012, LNAI 7327, pp. 379–388, 2012.

is considered very critical in design of holonic multi-agent systems and an improper method may increase the complexity of the system and decrease the efficiency.

This paper presents a new holonification method that enables the agents to find and form proper holons. In the proposed method, we have made use of graphs to model the population of agents in the holonification process; in addition, a quality measure is introduced for estimating the quality of the created holons. The proposed method is applied to urban traffic problem where an attempt is made to cooperatively control traffic signals.

This paper is organized as follows: in section two we present a brief description of holonic multi-agent system. Section three explains our graph-based holonification algorithm in detail. A case study of traffic signal control using holonic approach is presented in section four. Finally, section five gives a conclusion and future works.

2 Holonic Multi-Agent System

The term "holonic" introduced by Koestler (1967) which is a combination from the Greek holos = whole, with the suffix -on which means a particle or part [5]. A holonic organization is a hybrid, recursive and hierarchical structure which is able to generate dynamic linkages to control structure. A holon is characterized by some features: being stable, having the capability of autonomy, being capable of cooperation [6]. In a holonic system, the concept of a holarchy refers to architecture of self-organizing open hierarchical systems [5]. A holarchy can be modelled using whole-part relationships and it is managed in a distributed manner by system elements or holons. Fig. 1 shows a three-level holarchy. In Holonic Multi-Agent System (HMAS) the structure can be

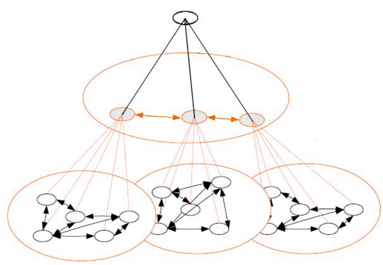


Fig. 1. Holarchy

seen as a set of hierarchical levels, where the agents can interact only with other agents at the same level or at the level immediately below or above. In these systems, individual holons define their activities based on their local knowledge and decide about their behaviour by means of the permitted interactions. Holonic organizations have proven to be effective solutions to several problems associated with hierarchical and self organizing structures [7] and have been successfully applied in a wide range of complex systems. For instance, we can mention the works done in manufacturing systems [8], transportation [9], adaptive mesh problem [10], health organizations [11].

3 Holonification

In this section, we present our graph-based approach for constructing the holarchies. The first step in the holonification process involves modelling different entities of the system and the relations among them. Then, an algorithm is needed to properly extract the structure of the holons based on the model. Furthermore, a measure should be used to evaluate the quality of the extracted holons.

3.1 Graph-Based Modelling

Graph concepts, as one of very old and effective concepts, have always been found useful in the modelling of the relationships and interactions in large and complex systems. Depending on the applications and relationships among the agents, a directed or undirected graph can be utilized. In this paper, an undirected and weighted graph is used in which, the agents are represented by the vertices and the relationships between any two agents is represented by an edge in the graph.

Holonification in the agent's network is very similar to the concept of partitioning in graph theory. In a graph partitioning problem, the aim is to divide the vertices of a graph into a set of partitions in such a way that some criteria are satisfied. Holonification is a process of partitioning the network into communities or holons in such a way that the most related group of the agents belong to the same holon. This process as like the graph partitioning process is NP-hard [12] and an algorithm to approximate a near optimal solution is desirable [13].

Formally, let's denote a graph by the pair $G = < V, E >$ in which V and E are the sets of vertices and edges respectively. Holonification is the problem of finding set of holons, H, such that:

$$H = \{h_1, h_2, ..., h_k\}; h_i = \{a_1^i, a_2^i, ..., a_{n_i}^i\} \mid \cup_i h_i = V \tag{1}$$

In Eq. 1, a_j^i denotes the j-th agent in holon i; n is the total number of agent in h_i and k is the number of holons and $h_i, i = 1, 2, ..., k$ are the holons. Generally, the holons may overlap, i.e. there are some members who belong to more than one holon; However, since many real world complex applications use non-overlapping communities as sub-structures, we assumed that the holons are disjoint, that is: $h_i \cap h_j = \emptyset; i \neq j$. We represent the concept of dependencies in multi-agent systems through the weights of the graph. Below, a detailed explanation of the dependency concept is given. It is also explained how the quality of holons is measured.

Dependency. In a multi-agent system, decision of an agent in the network influences some other agents. It depends on the state that whether an agent acts independently [14]. These influences are context-specific and can be qualified by a *dependency measure*. In our graph-based modelling, each dependency is considered as an edge in the graph and the degree of dependencies is defined by weight value. In literature there are some works pertaining to the learning of these dependencies. Among these works we can mention [14] and [15].

Since the definition of dependency usually varies in different applications and is based on the modelling purpose, we skip to discuss more about this concept and without loss of generality we assume that the dependency of two agents, a_i and a_j is available to the algorithm through a function, $d(a_i,a_j)$.

Quality Measure. Holonification reduces the complexity of the large networks by partitioning the agent's network. It divides the network to much simpler units such that the overall computational decreases, i.e. large scale problem is converted to multiple simple sub-problems and the computational load of the large network is scaled down. In our approach holonification tries to find a proper set of holons that maximizes a quality measure whose definition depends on the application. In this paper, it has been assumed that more dependent agents belong to same holons without any restriction on the number of holons and the number of agents in them. We define the quality measure as:

$$Q(H) = \sum_{h_i \in H} \frac{1}{n_i} \sum_{a^i_j, a^i_k \in h_i} d(a^i_j, a^i_k) \tag{2}$$

where $H = \{h_1, h_2, ..., h_k\}$ is the set of holons; n_i is the number of agents in h_i; a^i_j and a^i_k are two agents that are members of h_i; and $d(a^i_j, a^i_k)$ is the dependency function defined for two agents. As it can be seen from Eq. 2 the quality measure computes the sum of average dependencies over the number of agents inside the holons. If we show the total number of agents in the network by N; the number of agents in holon h_i by n_i; and the number of intra-holon connections by c_i; then minimum quality measure, $Q(H) = 0$, occurs when every agent in the network belongs to a separate disjoint holon, that is: $\forall h_i \in H; h_i = \{a^i_1\}$. In order to compute the maximum value of $Q(H)$, we have:

$$\sum_{h_i \in H} \frac{1}{n_i} \sum_{a^i_j, a^i_k \in h_i} d(a^i_j, a^i_k) \quad \leq \sum_{h_i \in H} \frac{1}{n_i} c_i \times \max_{j,k} [d(a^i_j, a^i_k)] \tag{3}$$

Considering the fact that maximum number of connection is obtained in a complete graph model which yields the maximum value for c_i as $\frac{n_i(n_i-1)}{2}$, we have:

$$\begin{aligned} Q(H) \leq \frac{1}{2} \sum_{h_i \in H} \max_{j,k} d(a^i_j, a^i_k) &\leq \frac{1}{2} \max_i [\max_{j,k} [d(a^i_j, a^i_k)]] \times \sum_{h_i \in H} (n_i - 1) \\ &= \frac{1}{2} \max_{i=1,...,k; h_i} (\max_{j,k} [d(a^i_j, a^i_k)] \times (N-k) \end{aligned} \tag{4}$$

According to Eq. 4 the minimum quality measure, $Q(H) = 0$, happens when $N = k$ and maximum quality measure happens when $k = 1$. It means that there is just one holon that all agents are the member of it. It definitely happens when all holons (sub-graph) are a complete graph. In this case, the output of the proposed holonification method will be one holon for such a network.

3.2 Holon Formation

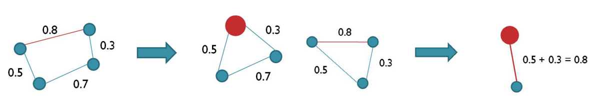
At the beginning of the holonification process, it is assumed that we have a network of agents modelled by $G =< V, E >$ which is a weighted undirected graph constructed according to the procedure described before. The algorithm starts with an empty set of holons and during the algorithm, new holons are added according to following steps: first, the process of choosing the best candidates to form a holon or to join an existing holon. Second, the reformation of the graph to reflect the changes made in first step.

As for the first sub-process, a greedy method is applied. In this method at every step an unvisited edge of the graph G is chosen in such a way that its weight is the maximum among all other unvisited edges. There are three possibilities: (i) the end nodes of the chosen edge do not belong to any existing holon; and (ii) one of the end nodes of the edge belongs to an existing holon and (iii) both end nodes of the edge belongs to an existing holon. Concerning the first case, a new holon is created. Since the overall measurement is always improved, this newly created holon is added to the list of the existing holons and adding the end nodes of the edge to the created holon, they are all marked as being visited.

When the second case occurs, it is assumed that the corresponding holon grows to hold the other end node of the edge. Now the measurement is calculated for the new holon considering its new structure. If it is improved, the holon is changed into its new structure. Finally in case of (iii) two holons corresponding to the two end nodes of the edge are merged. This way, a new and larger holon is created which contains all of the members of the merged holons. Then the measurement is calculated for the new holon. If it is improved, which will cause an improvement in the overall measurement, this new holon replaces with the last two small holons.

When an appropriate decision about the growth of the holons is made in every step, the structure of the graph (agents network) is reformed. In this reformation we attempt to treat the newly created holons as a new node in the graph and as the representative of its member nodes (agents). This is done in order to facilitate the decision making process. For the reformation of the graph we define two general rules as follows: (i) a node is created for each newly constructed holon. now all of the edges pointing to the members of the created holon, are corrected to point to the representative node. (ii) when an existing holon is grown, all of the edges pointing to its members point to the representative node of the grown holon.

In the process of reforming the graph and correction of the connections (edges), two cases occur: (i) there are no similar edges concerning their end nodes after the correction of the edges as described above. In this way the weights of the edges are not changed. An example is shown in Fig. 2(a). (ii) there are two or more similar edges corresponding their end nodes. In this case, a new edge represents the similar edges and its weight is the sum of the similar edges. Fig. 2(b) demonstrates this concept. These sets of steps are applied in every phase of the algorithm in which a new holon is identified or an existing one is grown. The holonification algorithm terminates when there is no unvisited edge left in the graph. At the end, for every singleton node that does not belong to any holon, a new holon is created with that node as its only member.



(a) There are no similar edges. (b) There are two or more similar edges.

Fig. 2. Two examples for holonification

It should be noted that, the original structure of the graph, concerning the actual edges (dependencies) between the nodes (agents) is not affected and changed by this algorithm. In other words all of the manipulations described above are applied temporarily to identify the proper holons in the graph. The holonification process is described in the algorithm 1.

```
Holon = ∅;
while there_is_an_unvisited_edge do
    e1 = edge_with_max_weight();
    if holonOf(end_nodes(e1)) = ∅ then
        h1 = newHolon(end_nodes(e1));
        add(Holon, h1);
        rep_node = newRepresentativeNode();
        reform_graph();
    end
    else if e1.onlyOneEndIsInHolon() then
        h1 = connectedHolon(e1);
        newQ = (h1.Q * h1.numOfNodes() + e1.weight)/(h1.numberOfNodes() + 1);
        if(newQ > h1.Q)
        h1.grow(e1);
        reform_graph();
    end
    else if e1.two_ends_are_in_holon() then
        h1 = first_connected_holon(e1);
        h2 = second_connected_holon(e2);
        newQ =
        (h1.Q * h1.numOfNodes() + h2.Q * h2.numberOfNodes() + e1.weight)/
        (h1.numOfNodes() + h2.numberOfNodes());
        if newQ > (h1.Q + h2.Q) then
            Merge(h1, h2);
            Reform_graph();
        end
    end
end
returnHolon;
```

Algorithm 1. The proposed holonification algorithm

4 Case Study: Urban Traffic Network

The problem of intelligent traffic control has been studied for many years. Agent-based urban traffic models have been found as efficient tools for traffic planning. The application of multi-agent systems to urban traffic control has made it possible to create and deploy more intelligent traffic management. As the complexity of traffic control on a network grows, it becomes more difficult to coordinate the actions of a large number of traffic entities that are available in the network. One way of handling this complexity is to divide the coordination problem into smaller coherent sub-problems that can be solved with a minimum number of interactions [16]. A common way in agent-based simulation of urban traffic is to consider each intersection as an agent. So the intersection map forms a network of interacting agents [17],[16], [18]. There are some researches that have used a graph model to represent the traffic network [19], [20]. The problems and simulations related to urban traffic networks can be a good test bed for holonic solutions due to its dynamic nature and possibly large set of entities. In this case study, we will see how our algorithm can be applied for the modelling and holonification of the network.

4.1 Holonification

We model the intersections as the nodes of the graph and the roads as the edges. The resulting graph has the following properties: *Connected*: there is at least one path between two intersections. *Planar* : It is assumed that the network has an intersection wherever two roads cross. *Weighted*: every two adjacent intersections are dependant physically considering their traffic flow. These dependencies are specified by the weights of the graph. Dependency between two nodes is measured by a weight value. *Undirected*: It has been assumed that the dependency is bidirectional.

Let's have a traffic network consisting of 14 intersections and 22 connecting roads between the intersections. Making a graph model of this network using the method described above, we will have a graph with $|V| = 14$ and $|E| = 22$ as shown in Fig. 3(a). The dependency between every two intersections is computed according to the distance between them and the traffic rate flowed to each other. It can be easily seen that there are many ways for putting the nodes (agents) inside of holons. Finding the best holonification that maximizes our quality measure is an NP-complete problem. For this example, we choose three adjacent nodes to make a holon, the resulting set

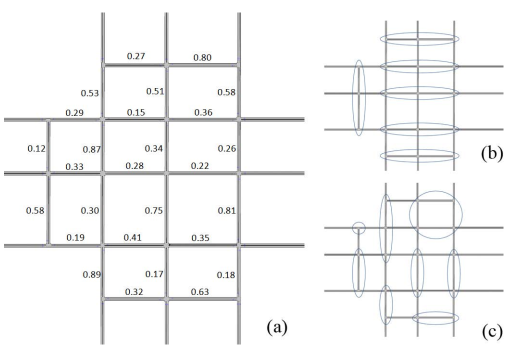


Fig. 3. (a)An example of traffic network modelled by a weighted graph; (b)a random homogeneous holonification; (c) The proposed holonification method

of holons is shown in Fig. 3(b) that the quality measure of it is 1.496 according to Eq. 2. The result of the proposed holonification method is depicted in Fig.3(c) which gives a quality measure of 2.859. As it can be seen, the proposed method has produced a more optimal holonification.

4.2 Intra-holon Timing

In this section, a simple method is utilized to assign green time to different phases of the junctions. Each agent placed at one junction and cooperates with the other connected agents belonging to the same holon by using the dynamic information provided by them. This information is the number of the cars passed through those junction in a specific time interval. The agent, then, uses this information together with the dependencies of its connected nodes to compute the green time of its phases. Formally, let's d_{ij} be the dependency between the agents i and j; T be the cycle time; N_j^t be the number of the cars passed through junction j at t-th time interval. Assigning a separate signal phase to each approaching link of a junction, the green time of the phases of the links outside of the holons are computed by Eq. 5 where i shows the current junction we are assigning the green times and GT_{ij}^t is the green time for the phases corresponding to the link connecting junctions i and j at t-th time interval.

$$GT_{ij}^t = \frac{d_{ij}}{\sum\limits_{k \in neighbor(i)} d_{ik}} \times T \tag{5}$$

. Similarly, for the green times of the phases of the links inside the holon, we have:

$$GT_{ij}^t = \frac{1}{\sum\limits_{k \in neighbor(i)} d_{ik}} \times T \times \frac{d_{ij} \times N_j^{t-1}}{\sum\limits_{l \in neighbor(i); l \in holon(i)} d_{il} \times N_l^{t-1}} \tag{6}$$

where $neighbor(i)$ denotes the set of agents adjacent to agent i and $holon(i)$ denotes the holon which i is a member.

4.3 Experimental Results

The proposed method has been tested on a network that contains 18 intersections and 39 two-way links using the Aimsun traffic simulator [1]. The network configuration parameters are shown in Table 1.

We performed some experiments to determine the average delay time of the network for different holonic structures. To show the impact of holonification, consider the signals timing method mentioned before. It has been tested on the network for two structures, a homogeneous method and the proposed method. The results show the proposed method has reduced the average delay time in comparison with the other one. Figure 4 shows the delay time over the mentioned traffic demand for two networks

[1] `http://www.aimsun.com`

Table 1. Network configuration parameters

Properties	Value	Properties	Value
number of intersections	18	number of lanes per links	3
number of links	78	arrival distribution	Uniform
average length of links	425.38m	simulation duration	10hour
maximum speed	60km/h	traffic demand	5000 veh/hour

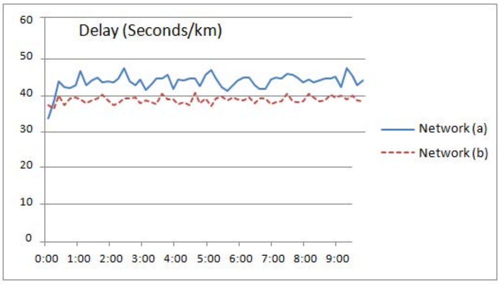


Fig. 4. Delay time tested on both networks

shown in Figs. 3(b) and 3(c). It can be seen that the delay time of the signal timing based on the proposed holonification is less than the one based on homogeneous method.

5 Conclusion and Future Works

In many real world applications we deal with systems that are complex and have a large scale. Holonic multi-agent system can be used as an effective tool for tackling these systems. Holonic systems need to specify the holons (the process of holonification). In this paper we have proposed a holonification method which can be applied to a population of inter-dependant agents in order to recognize and form the holons. In the proposed method, we modelled the agents using an undirected and weighted graph which the weights denote the degrees of the dependencies between the agents and a greedy algorithm was proposed to form the initial holons. The strength of the proposed method was demonstrated in an urban traffic network under two holonic organizations. Furthermore, a general quality measure based on the dependencies between the agents was introduced in this paper. Future works include working on the definition of quality measure based on the properties that are expected from the holons and extends the method for more levels and dynamic holarchies.

Acknowledgements. The first authors would like to thank Research Institute for Information and Communication Technology - ITRC (Tehran, Iran) for their supports.

References

1. Weiss, G.: Multiagent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intelligence. MIT Press (1999)
2. Theocharopoulou, C., Partsakoulakis, I., Vouros, G.A., Stergiou, K.: Overlay networks for task allocation and coordination in dynamic large-scale networks of cooperative agents. In: Proc. of 6th International Joint Conference on Autonomous Agents and Multi Agent Systems, Hawaii, pp. 295–302 (2007)
3. Irandoust, H., Benaskeur, A.R.: Multi-organizational structures. In: Proc. of Association for the Advancement of Artificial Intelligence, Chicago, pp. 25–33 (2008)
4. Tianfield, H.: A new framework of holonic self-organization for multi-agent systems. In: Proc. of IEEE International Conference on Systems, Man, and Cybernetics, Montreal, pp. 753–758 (2007)
5. Koestler, A.: The Ghost in the Machine. Hutchinson (1967)
6. Rodriguez, S., Hilaire, V., Koukam, A.: Towards a holonic multiple aspect analysis and modeling approach for complex systems: Application to the simulation of industrial plants. Simulation Modelling Practice and Theory 15, 521–543 (2007)
7. Rodriguez, S., Hilaire, V., Koukam, A.: Formal specification of holonic multi-agent. In: Proc. of International Conference on Computational Science, pp. 719–726 (2005)
8. Deen, S.M.: A cooperation framework for holonic interactions in manufacturing. In: Proc. of the Second International Working Conference on CKBS, pp. 103–124 (1994)
9. Burckert, H.J., Fischer, K., Vierke, G.: Transportation scheduling with holonic mas - the teletruck approach. In: 3d International Conference on Practical Applications of Intelligent Agents and Multiagents, pp. 577–590 (1998)
10. Rodriguez, S., Hilaire, V., Koukam, A.: Towards a methodological framework for holonic multi-agent systems. In: Proc. of Fourth International Workshop of Engineering Societies in the Agents World, London, pp. 179–185 (2003)
11. Ulieru, M., Geras, A.: Emergent holarchies for e-health applications: a case in glaucoma diagnosis. In: Proc. of 28th Annual Conference of the Industrial Electronics Society, vol. 4, pp. 2957–2961 (2002)
12. Brandes, U., Delling, D., Gaertler, M., Gorke, R.: On finding graph clusterings with maximum modularity. In: Proc. of 33rd International Workshop on Graph-Theoric Concepts in Computer Science, pp. 121–132 (2007)
13. Schaeffer, S.E.: Survey graph clustering. Computer Science Review, 27–64 (2007)
14. Kok, J.R., Hoen, P.J., Bakker, B., Vlassis, N.: Utile coordination: Learning interdependencies among cooperative agents. In: Proc. of IEEE Symposium on Computational Intelligence and Games, Colchester, pp. 29–36 (2005)
15. Melo, F.S., Veloso, M.: Learning of coordination: Exploiting sparse interactions in multiagent systems. In: Proc. of 8th International Conference on Autonomous Agents and Multi Agent Systems, Budapest, pp. 773–780 (2009)
16. Doniec, A., Mandiau, R., Piechowiak, S., Espie, S.: A behavioral multi agent for road traffic simulation. Engineering Applications of Artificial Intelligence 21, 1443–1454 (2008)
17. Hirankitti, V., Krohkaew, J., Hogger, C.: A multi-agent approach for intelligent traffic-light control. In: World Congress on Engineering, London (2007)
18. Zhang, H.T., Yu, F., Li, W.: Step-coordination algorithm of traffic control based on multi-agent system. International Journal of Automation and Computing 6, 308–313 (2009)
19. Hu, M.B., Jiang, R., Wu, Y.H., Wang, W.X., Wu, Q.S.: Urban traffic from the perspective of dual graph. The European Physical Journal B(63), 127–133 (2008)
20. Chen, S.W., Yang, C.B., Peng, Y.H.: Algorithms for the traffic light setting problem on the graph model. In: Taiwanese Association for Artificial Intelligence, pp. 1–8 (2007)