Holonic multi-agent system for traffic signals control

Monireh Abdoos a, Nasser Mozayani a,*, Ana L.C. Bazzan b

a School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
b Institute of Informatics, University of Federal Rio Grande do Sul, Porto Alegre, Brazil

Abstract

Agent-based technologies are rapidly growing as a powerful tool for modelling and developing large-scale distributed systems. Recently, multi-agent systems are largely used for intelligent transportation systems modelling. Traffic signals control is a challenging issue in this area, especially in a large-scale urban network. In a large traffic network, where each agent represents a traffic signals controller, there are many entities interacting with each other and hence it is a complex system. An approach to reduce the complexity of such systems is using organisation-based multi-agent system. In this paper, we use an organisation called holonic multi-agent system (HMAS) to model a large traffic network. A traffic network containing fifty intersections is partitioned into a number of regions and holons are assigned to control each region. The holons are hierarchically arranged in two levels, intersection controller holons in the first level and region controller holons in the second level. We introduce holonic Q-learning to control the signals in both levels. The inter-level interactions between the holons in the two levels contribute to the learning process. Experimental results show that the holonic Q-learning prevents the network to be over-saturated while it causes less average delay time and higher flow rate.

1. Introduction

Agent-based technologies are known as powerful tools for development of large-scale distributed systems. The domain of traffic and transportation systems is well suited for an agent-based approach because transportation systems are usually geographically distributed in dynamic changing environments (Chen and Cheng, 2010). The reason for the growing success of agent technology in these areas is that the inherent distribution allows for a natural decomposition of the system into multiple agents that interact with each other to achieve a desired global goal (Chen and Cheng, 2010). The agent technology can significantly enhance the design and analysis of problem domains under the following three conditions (Adler and Blue, 2002):

1. The problem domain is geographically distributed.
2. The subsystems exist in a dynamic environment.
3. The subsystems need to interact with each other more flexibly.

Agent-based transportation systems allow distributed subsystems collaborating with each other to perform traffic control and management based on real-time traffic conditions. Traffic signals control is a principle part of transportation systems management. Applying intelligent techniques to improve the traffic flow of the system is necessary due to increase in the number of vehicles in urban networks and limitation of road infrastructure (Bazzan et al., 2008). It has been shown that traffic signal timing has important effects on traffic delay reduction and consequently reduction in travel time and energy consumption.

Techniques and methods for traffic signals control are usually applied in a single or a small number of intersections (Grégoire et al., 2007; Weiring, 2000). For large networks with multiple intersections, the interdependency between them makes it difficult to set the signals appropriately. An approach to deal with this complicated traffic signals control is using agent-based distributed control techniques which involve multiple intelligent agents, multi-agent systems (MASs).

In this paper, we present a method based on multi-agent systems to control the traffic signals in a large network. Multi-agent systems (MASs) are generally composed of autonomous, reactive, proactive and interacting entities called agents usually engaged in the realisation of a common goal. Although, there are many actors in a traffic network which can be considered as an autonomous agent, in this paper each agent represents an intersection or traffic signal. Hence, a large traffic network contains many agents interacting with each other which made a large-scale MAS.

A common approach for tackling the interactions among a large population of agents is assigning an organisation to agents
for cooperating and working towards a common goal. In the agent organisational paradigm, interactions among the agents are optimised and reduced effectively and this makes it a very popular technique to manage the complexity of large systems. In MAS, an organisation is defined as the collection of roles, relationships and authority structures that govern the behaviour of the multi-agent system.

We present an organisation-based MAS to model large traffic networks which is called holonic multi-agent system. A holonic organisation is a hybrid, recursive and hierarchical structure which is able to generate dynamic linkages to control the structure. The problem of traffic signals control for a large network can be divided into sub-problems each assigned to a holon. This paper also introduce a control strategy based on reinforcement learning that is especially developed for holonic multi-agent systems. In summary, the following main contributions are made in this paper:

• Modelling a large traffic network using holonic multi-agent system which enables us to decompose the system into smaller sub-systems in order to improve the efficiency.
• Development of a hierarchical control strategy based on reinforcement learning which increases reliability and performance of the system.

The rest of the paper is organised as follows: Section 2 reviews the related works and preliminaries about learning model and holonic concepts. The proposed learning method is presented in Section 3. This section also includes basic concepts used in our holonic model. Section 4 gives traffic signals control method based on holonic Q-learning in details. Experimental results are presented in Section 5 and finally the paper is concluded in Section 6.

2. Related works

2.1. Learning models

Classically, Q-learning is modelled using a Markov decision process (MDP) formalism. It means that the agent is in state $s$, performs action $a$ and gets scalar reward $r$ from the environment. The environment changes to state $s'$ according to a given probability transition matrix $T$. The goal of the agent is to choose actions such that maximise discounted cumulative rewards over time. Discounted means that short term rewards are weighted more heavily than distant future rewards. The discount factor, $\gamma = [0..1]$, awards greater weight to future reinforcements if set closer to 1. In Q-learning, $Q(s,a)$ is a state-action value representing the expected total discounted return resulting from taking action $a$ in state $s$ and continuing with the optimal policy thereafter. During the learning process Q-values are updated by:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma V(s') - Q(s,a)]$$

In Eq. (1), $\alpha$ is the learning rate parameter and $V(s')$ is given by Eq. (2). The discount factor, $\gamma$, determines the importance of future rewards. $Q(s,a)$ measures the quality value of the state-action pair and $V(s')$ gives the best $Q$-value for the actions in next state, $s'$. Q-learning helps to decide upon the next action based on the expected reward of the action taken for a particular state. Successive Q-values are built increasing the confidence of the agent as more rewards are acquired for an action. As the agent explores the state space, its estimation of Q-value improves gradually. Watkins and Dayan (1992) have shown that the Q-learning algorithm converges to an optimal decision policy for a finite Markov decision process.

Assume that $x_i$ shows an agent in a MAS. $x_i$ is a tuple $(s_i, P_i, A_i, \phi_i)$ of the set of its possible states, $S_i$ the set of its percepts, $P_i$ and actions, $A_i$, and its agent function, $\phi_i : S_i \times P_i \rightarrow S_i \times A_i$. A classic Q-learning technique for $x_i$, has three main steps:

- The agent estimates the state using its percept, $P$, and determines which state the agent is in at time $t$, $s(t)$.
- The agent selects an action, $a(t)$ according to the learned policy and a mechanism for exploration of the environment.
- The agent updates its policy using the reward value received from environment.

The state of the environment changes and a new learning iteration starts.

2.2. Holonic models

The term “holonic” is derived from the word “holon”, which introduced by Koestler (1967). The word holon is a combination of the Greek holos=whole, and the suffix –on which means a particle or part. Koestler proposed this word because of two observations. The first observation is from Simon (1969) who concludes that complex systems will evolve from simple systems much more rapidly if there are stable intermediate forms than if there are not. Simon’s analysis reveals why every complex adaptive system is hierarchic. Koestler analysed hierarchies and stable intermediate forms in both living organisms and social organisations and found that “wholes” and “parts” do not exist in an absolute sense. This fact makes his second observation. For example, a human is a composition of organs on one hand, and is a part of a human society on the other hand. Koestler proposed the word holon to describe the hybrid nature of subwholes/parts in real-life systems. Holons simultaneously are self-contained wholes to their subordinated parts, and dependent parts when considered from the inverse direction.

The organisational structure of a holonic system, or holarchy, has some advantages: They are robust against external and internal damages, they are efficient in their use of resources and they can adapt to environment changes. Koestler’s ideas have been applied to manufacturing systems for the first time by Suda (1990). This led to the formation of Holonic Manufacturing Systems (HMS) later. Holons in an HMS are characterised by their holonic attributes, autonomy and cooperativeness which have roughly the same semantics as for multi-agent systems. Autonomy is the ability to create and control the execution of plans and strategies and cooperativeness is joint planning and coordination for a joint plan execution.

Koestler also points out that holons are autonomous self-reliant units, which have a degree of independence and handle contingencies without asking higher authorities for instructions. Simultaneously, holons are subject to control from (multiple) higher authorities. The first property ensures that holons are stable forms, which survive disturbances. The latter property signifies that they are intermediate forms, which provide the proper functionality for the bigger whole.

Finally, Koestler defines a holarchy as a hierarchy of self-regulating holons which function (a) as autonomous wholes in super-ordination to their parts, (b) as dependent parts in sub-ordination to controls on higher levels, and (c) in co-ordination with their local environment.
Koestler (1967) has mentioned a series of grouped rules defining holonic systems, which he calls Open Hierarchical Systems (OHS). we can keep the following principles:

- A holonic system possesses an abstract structure. In fact it can be seen as a set of interwoven hierarchies.
- A holon is considered both as a part of the system and as a whole, a system made up of other holons.
- A holon obeys precise principles, but is able to adopt different strategies according to its need.
- The complex activities and behaviours are situated at the top of the hierarchy, the simple and reactive acts are found at the base of the holarchy.
- The communications must follow the hierarchy and according to the direction, must be filtered or detailed.

We now define a holonic multi-agent system based on the mentioned principles. A holon in an HMAS can be seen as an agent in a multi-agent system. To understand the concept of a holon, Gerber et al. (1999) listed the properties of an agent and extend them to holons:

1. Autonomy
   - **Agents**: Agent autonomy means that agents can control their actions and internal states. They are able to operate in the environment without the direct intervention of humans or others. **Holons**: Several agents form a holon that can act as a single entity. The holon operates in the environment as an autonomous agent. By joining a holon, agents accept some restrictions of their abilities to act and to communicate while they are working towards the goals of a holon.

2. Goal-directed behaviour
   - **Agents**: An agent has represented goals and desires either explicitly or implicitly. **Holons**: Sub-holons of a holon have their private goals while follow at least one common goal of the super-holon. The super-holon's goals emerge from the common goals of sub-holons. It is required that there is no conflict between the goals of sub-holons and the super-holon.

3. Action
   - **Agents**: Actions are defined for reactivity and pro-activeness of the agents. **Holons**: The capability to act is extended to a group level. It means that actions of a super-holon are composed of the actions of sub-holons such that none of the sub-holons could perform actions alone.

4. Belief
   - **Agents**: Agents have implicit and/or explicit representations of their environment. **Holons**: The requirements for an agent's belief remain unchanged: holons have a representation of their environment. This knowledge might be represented explicitly within the super-holon or implicitly distributed to sub-holons by their local knowledge.

5. Bounded rationality
   - **Agents**: A rational agent behaves optimally with respect to its limited resources towards its goal. **Holons**: A holon should be able to control its resources in order to exhibit a bounded rational behaviour. A super-holon distributes guidelines to its sub-holons for local resource management.

6. Communication
   - **Agents**: The ability to communicate with the other agents is an essential part of an agent's autonomy. The agents share a communication language towards their social capability. **Holons**: The right to communicate with other agents is an exclusive resource of the holon (not its sub-holons). Since problem solving inside a holon is cooperative, internal communication is necessary inside a holon.

A holonic multi-agent system must therefore have a flexible organisation in a hierarchical structure. The holons unite to form a whole and each one can be broken down into holonic agents, this is what called the recursive breakdown of the problem.

In a holonic system, a holarchy can be modelled using whole-part relationships and it is managed in a distributed manner by system elements or holons. In HMASs the structure can be 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.

The concept of holon is central to our discussion and therefore a definition of a holon can be helpful. In MAS, the vision of holons is much closer to the one that MAS researchers have of recursive or composed agents. Recursiveness means that holons can contain other inferior levels of holons, that can also be contained in another superior level of holons, being in a recursive architecture.

A holon constitutes a way to gather local and global, individual and collective points of view. A holon is thus a self-similar structure composed of holons as sub-structures and the hierarchical structure composed of holons is called a holarchy. Fig. 1 shows a holonic system arranged in four holarchical levels.

The strength of the holonic paradigm is its recursive definition of holons. Thus it is well adapted for large complex systems where different granularities are required. Holonic systems offer the possibility to model a system from a high-level coarse-grained perspective to a low-level fine-grained one. On the other hand, using holons as modelling approach for a small close system may not always be recommended. If the system does not require multiple levels of granularity, holons will probably introduce an unnecessary overhead. Before applying any approach to model a system, the designer should analyse the adequacy of the underlying concepts. Hence, holonic systems are well suited for analysing and modelling of large systems where multiple levels of abstraction exist. Another type of systems are those that can be decomposed recursively into smaller sub-components (Rodriguez, 2005).

It has been proved that HMAS is an effective solution for several problems associated with hierarchical and self-organising structures (Rodriguez et al., 2005). It has been successfully applied in a wide range of complex systems. For instance, we can mention the works done in transportation (Burckert et al.,

![Fig. 1. A holarchy composed of four holarchical levels.](image-url)
1998a), distributed sensor management (Benaskeur and Irandoust, 2008), adaptive mesh problem (Rodriguez et al., 2003), supply chain management (Marcellino and Sichman, 2010), health organisations (Ulieru and Geras, 2002), biological network simulation (Shafaei and Aghaee, 2008) and complex software systems (Moise, 2008).

In this paper, we develop a holonic multi-agent system to model a large traffic network and use a reinforcement learning method for learning the signals timing in the holarchy. Since a large traffic network is definitely a complex system (Bazazan, 2007), holonic multi-agent system gives a promising tool to model the intersections and manage the traffic jam.

Two overlapping aspects have to be distinguished in holons: the first is directly related to the holonic nature of the entity (a holon, called super-holon, is composed of other holons, called sub-holons or members) and deals with the role of a super-holon. This aspect is common to every holon and thus called the holonic aspect. The second aspect is related to the problem and depends on the application. Hence, we first describe the holonic Q-learning generally and then in the context of traffic signals control.

2.3. Learning and holonic models for traffic simulation

Intersections are bottlenecks in urban transportation systems. Multi-agent systems have been used for intersection control in many works. It is beyond the scope of this paper to review the literatures on agent-based technologies in traffic and transportation systems, the readers are referred to Chen et al. (2009) for a survey in this area. We mention some related works which use agent-based hierarchical control of traffic signals control or apply reinforcement learning in order to find the traffic control policy. However, for the survey of existing methods on learning and multi-agent systems in traffic control the readers are also referred to Bazazan (2009).

Roozemond (2001) proposed an agent-based urban intersection control system that reacts to changes in the traffic environment and adapts itself to changing environments based on internal rules. The system consists of several intersection traffic signalling agents (ITSA), road segment agents (RSAs), and some authority agents. The ITSA manages the intersection controls helped by RSAs. The authority agents control and coordinate ITSA to achieve a globally optimal system performance. Although the interesting issues tackled in this research are cooperation between multiple agents, and estimation of prediction model and adaptation to dynamic environments, the drawback is necessity for some knowledge in the form of rules. Hence, the performance highly depends on the rules.

In Cai and Yang (2007) a system composed of four types of agents, segment agent, crossing agent, section agent and central decision agent are presented. The system can achieve the global optimisation of the region, alleviate traffic jam, reduce the emission of vehicles and then increase the efficiency of traffic management by information sharing in a region. The central decision agent which stands on the top of the hierarchy, analyses the decisions from section agents. Since central decision agent controls all management processes, their system has the disadvantages regarding centralised control such as design complexity and robustness.

Hierarchical decentralised agent-based control methods decompose the system into smaller sub-systems that have interactions with each other. They generally are used in large networks that decomposition causes a reduction in dimensionality.

Choy et al. (2003) and Srinivasan et al. (2006) presented a hierarchical multi-agent system that consists of three layers of agents with the lowest layer of intersection controller agents (ICAs), the middle layer of zone controller agents (ZCAs), and the highest layer of regional controller agents (RCAs). A ZCA controls several preassigned ICAs, and one RCA controls all of the ZCAs. The implementation of agent is based on feed-forward neural network and fuzzy logic theories. The neural network consists of five layers such that different parts of fuzzy reasoning mechanism stand between the layers. The system also provides online learning facilities, including reinforcement learning, learning rate and weight adjustment, and dynamic update of fuzzy relations using evolutionary algorithm, to allow agents dynamically adapting to changing environment. However, the hybrid method controls the traffic signals effectively in a dynamically changing environment, the design and implementation of different parts of the system is complicated and computationally expensive.

Learning techniques which were developed for multi-agent systems, can potentially give decisive contributions to control and management of traffic systems, as they meet the demands of dynamic, changing systems of many actors with different goals, distinct cognitive capabilities and learning pace. Reinforcement learning (RL) offers some potentially significant advantages in dealing with dynamic environments. Hence, RL has become a good solution to control single intersection and also network of intersections specially when there is no previous knowledge about the environment.

Some researchers have tried to use a model-based reinforcement learning method for traffic signals control. Grégoire et al. (2007) designed a single adaptive traffic signals controller using learning agents. The agents learn a control policy trying to optimise the traffic flow by interacting directly in the system. However, they showed that the results are better than optimal pre-timed controllers, they focused on a single intersection. There are some problems which arise from working a large number of controller together specially when the intersections are controlled by learning.

Weir (2000) describes the use of reinforcement learning by traffic signal agents in order to minimise the overall waiting time of vehicles in a small grid. Those agents learn a value function that estimates expected waiting times of vehicles given different settings of traffic signals. One interesting issue tackled in this research is that a kind of co-learning is considered: value functions are learned not only by traffic signals, but also by the vehicles that can thus compute policies to select optimal routes to their destinations. One drawback of his method is that the kind of communication and knowledge may have a high cost specially when we extend the method to a larger network.

A similar model-based RL method for controlling traffic signals is presented in Steingröver et al. (2005) to minimise the total travel time of all vehicles in the network. Thus, the control perspective is a global one, although actions are local to the agents. Agents here are the traffic signals but the learning task is formulated in a way that the state representation is vehicle-based (waiting times for individual vehicles), aggregated over all vehicles around the intersection. The network used in their simulation includes fifteen intersections. An issue is that the more information about the individual vehicle, the bigger the state space, specially when larger network is used. Hierarchical methods that use a divide-and-conquer strategy can be used to deal with large problems.

Since model-based RL can work in stationary environments, when dealing with non-stationary environments, they cannot adapt to the changing dynamics of the environment. In Silva et al. (2006) described a method for complementing RL algorithms so that they perform well in a specific class of non-stationary environments. They have presented a reinforcement learning with context detection that creates a partial model of the environment on demand. They validated their proposed method on traffic signals control containing nine intersections. The results
showed that the performance of their method is better than both model-based and model-free RL assuming some properties that cause the method cannot scale up to larger network.

A model-free Q-learning has also been presented in Abdoos et al. (2011) to control traffic signals in the non-stationary environment of a large network. They used a parametric representation of action space which can be used in different forms. Although the results show improvements on delay time for a large network comparing with fixed timing method, the agents work independent of each other without any interaction.

Adaptive reinforcement learning has been presented to control a model free traffic environment in Wen et al. (2008) and Dai et al. (2010). Adaptive control of traffic signals has also been studied in Prashanth and Bhatnagar (2010), which uses a function approximation method as a mapping between the states and signal timing. However, the mentioned model free RL methods have some advantages regarding non-stationary environments, they tested on isolated intersection (Wen et al., 2008) or regular networks either grid (Dai et al., 2010; Prashanth and Bhatnagar, 2010) or corridor (Prashanth and Bhatnagar, 2010) networks. These algorithms could not be implemented on larger road networks due to the exponential increase in the computational complexity.

A number of works have used reinforcement learning for traffic signals control in a large network. For example a collaborative reinforcement learning introduced in Salkham et al. (2008) which tried to exploit local knowledge of neighboring agents in order to learn signal timings. Balaji et al. (2010) have also presented a method in which agents try to compute a new phase length based on both local and communicated information that received from neighbours.

As mentioned before, hierarchical agent-based control methods decompose the system into smaller sub-systems. The idea of using reinforcement learning method to hierarchical control of the traffic signals in a large network explored in Bazzan et al. (2010). They organised agents in groups of limited size so that the number of joint actions is reduced. These groups are then coordinated by another agent, a tutor or supervisor that recommend a joint action. The paper investigates and compares the task of multi-agent reinforcement learning for control of traffic signals under the following two situations: agents act individually and agents can be tutored by another agent with a broader view of the system.

Although hierarchies are suitable for modelling large-scale complex systems, there are some important issues that made them inadequate: hierarchical control usually creates a rigid communication hierarchy and an authoritarian top-down flow of commands; the systems will be more vulnerable upon failures in one or more of the master controllers. Holonic organisation tries to overcome these defects with the help of autonomy, cooperation, and self-organisation (see more description in the next section).

Holonic multi-agent system has been applied in transportation by Burckert et al. (1998a) for the first time in 1998. They introduced a system called TELETRUCK for dynamic planning and optimisation of transport orders based on holonic agents (Burckert et al., 1998a,b). Holonic models have also been used for railway traffic control system (Ciufudean and Filote, 2009) and for communication of knowledge on traffic network in Adam et al. (2012). Adam et al. (2012) presented a multi-level architecture, inspired from holonic principals which propagate knowledge between mobile agents. They have used flexibility of holonic organisation for knowledge transfer on traffic network.

Holonic MAS has been used for transportation modelling and optimisation using negotiation in Kožjak and Pisarski (2011). They introduced theoretical representation of the abstract holon and a model for building and reorganising holons which was used for transportation management and a delivery strategy development in a company.

In this paper, we develop a holonic multi-agent system to model a large traffic network and use a reinforcement learning method for learning the signals timing.

Although holonic multi-agent systems have already used in traffic networks, the main contributions of this paper are application of HMAS to model traffic signals and using reinforcement learning in holonic organisation to control the signals. Since holonic organisation is composed of self-similar holons which stand in a hierarchy, reinforcement learning method is used for all holons in different levels in the same way. Therefore, all holons in the hierarchy try to learn the control policy simultaneously.

3. Holonic multi-agent Q-learning

Reinforcement learning is best suited learning method for domains in which we have limited or no previous knowledge. RL offers a promising way to model learning in problem solving tasks because it defines problems in a compatible way using states, actions and goals. Q-learning is a particular reinforcement learning algorithm which is model-free. In this section, first we give a brief definition of Q-learning and then introduce the holonic Q-learning method which is used for traffic signals control.

3.1. Basic concepts

According to the model presented by Hilaire et al. (2000), there are three interrelated concepts in organisational models: role, interactions and organisation. This concepts are defined as follows:

- Role: A role is defined as an abstraction of behaviour in a given context.
- Interaction: An interaction determines the connection between two roles such that an action in each of them produces a reaction in the other role.
- Organisation: A set of roles and their interactions pattern define an organisation in a specific domain. The concept of organisation combines roles and their interactions.

Holonic organisations present a powerful abstraction tool to model large-scale systems. Different roles and interaction model should be defined when holonic organisation is used for modelling.

3.1.1. Roles

Holons are composed of other holons which referred as super-holon and sub-holon, respectively. A super-holon can be seen as a set of interacting sub-holons. The interactions are required to achieve the goal of the super-holon. Three roles are defined in holonic organisation (Rodriguez, 2005): head, part and multi-part. The head is representative of the holon which composed of a set of holons and represents the shared intentions of the holon. It negotiates these intention to other holons either its members or other holons. For the members two different roles are defined (Rodriguez, 2005): The part role represents the members which belong to one super-holon; the multi-part role represents the member which belongs to more than one super-holons. Multi-part roles lead to overlapping holons which is beyond the scope of this paper. Hence, we consider two roles, head and part which are equivalent to super-holon and sub-holon, respectively.

The members of a super-holon interact and cooperate to achieve the goal of super-holons. As mentioned before, the
be behaviour of holons in higher levels is more complex and reactive behaviour is done at the bottom of the hierarchy. A super-holon is created to fulfill a task for specific goal. To achieve the goal, the members should interact with each other and coordinate their actions according to the policy of the super-holon. Super-holons make abstraction of interactions to concentrate on the behaviour of the members. Super-holons on the higher levels make more abstraction comparison with lower level super-holons. The abstraction level of a holon is generally defined by

$$\text{level}(h) = \begin{cases} 0 & \text{if } h \text{ is an atomic agent} \\ \max(\text{level}(\text{subholon})) + 1 & \text{if } h \text{ is an abstract agent} \end{cases}$$

According to Eq. (3) the abstract level of the first level agents is 0. A holon with recursion level $n$ such that $n > 1$ is made up of holons with abstract level $k$ such that $k < n$. Depending on the complexity of the system, designers can use different abstract levels with more or less abstraction.

3.1.2. Interaction model

Holonic organisation includes a set of holons arranged in different levels. Distributed holons try to achieve a common goal through cooperation and coordination. Two types of interactions can be seen in a holonic organisation as shown in Fig. 2 (Rodriguez, 2005):

- Intra-level interaction: Agents interact with each other agents at the same level. This type of interaction sometimes referred to as horizontal interaction.
- Inter-level interaction: Agents of two different levels in the hierarchy interact with each other. It sometimes referred to as vertical interaction.

Intra-level interaction has been investigated in multi-agent system research. A holonic or any other intelligent agent-based framework which is entirely based on intra-level negotiation or interaction is likely to be inadequate. In practice, in addition to the intra-level interaction among the agents, there are vital interactions that take place between agents in different levels. For example, in any multi-level hierarchical multi-agent system, inter-level interactions play an important role.

Although a number of mechanisms are introduced in order to handle inter-level interaction, there is no general mechanism to transfer information between different levels of abstraction (Rodriguez, 2005). The approaches in this area usually propose their own mechanism for information transfer.

In this context, interactions between agents are implemented through asynchronous message exchange in two opposite directions: upward to transfer data towards higher level holons and downward in the opposite direction. A message is composed of four components $\langle \text{from}, \text{to}, \text{type}, \text{value} \rangle$:

- from: holonID (sender identification),
- to: holonID (receiver identification),
- type: $i$, $i = 0, 1, 2$ (0: abstract state, 1: abstract action, 2: reward),
- content: value (state index, action index, reward value).

Three types of messages are used in the proposed system. Depending on the direction, the messages contain abstract state, abstract action or reward.

3.2. The proposed learning method

Developing a distributed cooperation and coordination mechanisms for agents with a partial view of the whole system is challenging for complex systems. Agents need to estimate the unobserved states of the system and adapt their actions to the dynamics of the environment. These problems have caused that using classical learning methods is hard or even impossible for large scale domains. In order to handle these problems, some heuristics have been used in the literatures. Some of them tried to reduce the policy search space (Makar et al., 2001) while the others used some heuristics to guide the policy search (Zhang et al., 2009).

Holonic organisation provides a mechanism for abstract data transfer in multi-agent systems that makes it possible to use learning methods in complex systems. Abstract data usually contains useful information that the entities need them for their operation in the environment.

Classic Q-learning usually has three main steps as mentioned before: state estimation, action selection and reward calculation. However, the holonic Q-learning contains the same steps, the transferred abstract data contribute to different steps. We define three abstract data types that transfer between holons in different levels.

According to characteristics of HMAS defined by Gerber et al. (1999), the agents of level $n$ are composed of agents of level $n-1$. The holons in lower level are reactive holons while the higher level holons are proactive according to the rules defined by Koestler (as mentioned before). The complex activities and behaviour are related to the top of the holarchy and the lower level holons act directly in the environment. According to the

---

**Fig. 2.** Two types of interactions between holons in a holarchy.
directions shown in Fig. 1, the data stream for the main steps of the learning is as follows:

- Abstract state stream from level \( i \) to level \( i+1 \): Each holon uses the abstract state data received from sub-holon (if any exists), estimate the state and send the abstract state information to its super-holon. This data stream allows super-holons to estimate their state using data received from their members.
- Abstract action stream from level \( i \) to level \( i-1 \): The holon selects an action according to the abstract action received from super-holon (if any exists), send selected action to its sub-holons (if any exists). The concept behind this stream is that the members of a holon select an action correspond to the action selected by the super-holon.
- Reward stream from level \( i \) to \( i-1 \): A part of the reward for the holons depend on their super-holon. Since the super-holon has a broader view of the system and act as a supervisor, they assign rewards to their members. It means that, the holon calculates the reward according to the reward received directly from the environment and the reward received from its super-holon. This stream determines the efficiency of coherence of the members in the holon. It can be considered as a social reward or measure of the collaboration ability of the members.

Fig. 3 represents an example of HMAS containing four levels. Transferred messages are graphically shown by numbered arrow. The numbers show the order of messages considering one holon in each level. Let \( x_1, \beta_1, \gamma_1 \) and \( \alpha_1 \) are the marked holons such that: \( \text{level}(x_1) = 0, \text{level}(\beta_1) = 1, \text{level}(\gamma_1) = 2 \) and \( \text{level}(\alpha_1) = 3 \). Arrows 1–3 shows the abstract state stream which starts from \( x_1 \) up to \( \alpha_1 \). Abstract action stream is shown by arrows 4–6 and the reward stream corresponds to arrows 7–9 start from \( \alpha_1 \) down to \( x_1 \).

According to our interaction model, a message is composed of four components \(\langle \text{from}, \text{to}, \text{type}, \text{value} \rangle\), which are given in Table 1.

![Fig. 3. Data streams for a four-level holarchy. Arrows 1–3 show the abstract state stream; arrows 4–6 show the abstract action stream; and arrows 7–9 show the reward stream.](image)

### Table 1

<table>
<thead>
<tr>
<th>Abstract state stream</th>
<th>Abstract action stream</th>
<th>Reward stream</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\langle x_1, \beta_1, 0, p_1 \rangle)</td>
<td>(\langle \alpha_1, \gamma_1, 1, p_4 \rangle)</td>
<td>(\langle \alpha_1, \gamma_1, 2, p_5 \rangle)</td>
</tr>
<tr>
<td>(\langle \beta_1, \gamma_1, 0, p_2 \rangle)</td>
<td>(\langle \gamma_1, \beta_1, 1, p_5 \rangle)</td>
<td>(\langle \gamma_1, \beta_1, 2, p_6 \rangle)</td>
</tr>
<tr>
<td>(\langle \gamma_1, \alpha_1, 0, p_3 \rangle)</td>
<td>(\langle \gamma_1, \alpha_1, 1, p_7 \rangle)</td>
<td>(\langle \beta_1, x_1, 2, p_8 \rangle)</td>
</tr>
</tbody>
</table>

4. Traffic signals control

The traffic network is a complex system (Bazzan, 2007) composed of many different entities interacting with each other, whose actions are highly coupled. The traffic network is considered as a system composed of intelligent agents that each agent controls an intersection. In particular, traffic signals controllers located at intersections can be seen as autonomous agents. Since the number of agents is high in real urban network, holonic organisation is useful for modelling the traffic signals controllers. The network used for simulation is shown in Fig. 4. It contains 50 four-way intersections and more than 100 links. We used four-way intersections with three lanes in each link.

Organisation of a HMAS is formation of different holons and makes different levels in the holarchy. Ideally, this process carried out dynamically during the time. Dynamic organisation in HMAS includes creation and integration of new members, and self-organisation (Rodriguez et al., 2006). Since dynamic holons lead to challenges regarding the curse of instability which depends on the application, a fixed formation and grouping of the sub-holons is assumed according to the problem in most applications. In this paper, we use a holon formation method that is described in Abdoos et al. (2012). Since the formation of holons in the holarchy...
is out of the scope of this study, readers are referred to Abdoos et al. (2012) for more details.

4.1. Network configuration

A holonic organisation composed of two layers is used to model the network. In this model, each holon controls an intersection in the first level while a holon in the second level controls a region of network which contains a number of intersections. The proposed method can be easily extended to three levels if we have a policy to control a number of regions. Developing a control strategy is necessary to extend the number of levels. The model is extendible even for more levels, if we can develop a proper control strategy for each level. Since interactions between neighbouring intersections are only considered in this model, our model consists of two levels. Fifty agents are used for the first level (one for each intersection) which are arranged in 13 holons as it can be seen in Fig. 4. The holons are determined according to a graph-based algorithm that is described in Abdoos et al. (2012).

The intersections and links are considered as nodes and edges of the graph, respectively. Hence, the network is modelled using a weighted graph which the weights show the dependency between every two intersections. The dependency is computed according to the distance between them and the traffic rate between every two intersections. The dependency is computed according to the permutation of the four lanes, $l_1, l_2, l_3$ and $l_4$ (as shown in Table 3).

We discuss the learning process of holons in two levels separately. The holons in the first level and the second level are called sub-holons and super-holons, respectively.

4.2. First level

The traffic network is considered as a system composed of 50 agents that each agent controls an intersection in the first level.

For these agents we use the parametric Q-learning method presented in Abdoos et al. (2011). In each learning iteration, the agents perceive the environment and receive statistical data in a fixed cycle. Since all the intersections are 4-way crossroad, the system contains homogeneous agents. The signals groups are also shown in Fig. 5. A signals group is assigned to each approaching link. Therefore, each agent controls a signalised intersection which has four phases. Each agent controls a signalised intersection which has four phases, each assigned to an approaching link. A simple way to determine the states is to consider the average queue length in the approaching links. The state space is discretised by ranking the approaches according to the statistic gathered in the last cycle of time. The agents use the average queue length of approaching links as statistical data in a fixed cycle for state estimation. Let $l_i$ is the $i$th approaching link for an intersection, then $l_1 \geq l_2 \geq l_4 \geq l_3$ shows a state in which the average queue length of approaching link $l_1$ is the longest and $l_3$ is the shortest. If two approaching links have the same average queue length, they are ranked according to their order in the signal plan. The number of states is 24 which is the number of the permutations of the four lanes, $l_1$, $l_2$, $l_3$ and $l_4$ (as shown in Table 3).

The action space determines a plan for agents in such a way that each approaching link has an individual portion of green time in a fixed order. In the experiments, 2 s considered for yellow time after the green time assigned to four signal groups. The cycle adjusted to 100 s, and four 2 s have been assigned to yellow time. The total green time splits to four time interval, each assigned for a signal group as shown in Table 4. For example, action 12 is graphically shown in Fig. 6. Readers are referred to Abdoos et al. (2011) for more details.

Moreover, we use $\epsilon$-greedy as a mechanism to select an action in Q-learning. This means that the best action is selected for a proportion $1-\epsilon$ of the trials, and a random action is selected (with uniform probability) for a proportion $\epsilon$.

In this configuration, the number of signal groups is four as it can be seen in Fig. 5. The actions are defined by assigning different time split to the available signal groups in a fixed order.

### Table 3

<table>
<thead>
<tr>
<th>Learning steps</th>
<th>Sub-holons</th>
<th>Super-holons</th>
</tr>
</thead>
<tbody>
<tr>
<td>State estimation</td>
<td>Estimate the state by local information received from the environment</td>
<td>Estimate the state by local information and abstract state information received from sub-holons</td>
</tr>
<tr>
<td>Action selection</td>
<td>Select an action according to the learned policy and abstract action received from super-holon</td>
<td>Select an action according to the learned policy like classic Q-learning</td>
</tr>
<tr>
<td>Update policy</td>
<td>Update the policy using the reward received directly from the environment and reward received from super-holon</td>
<td>Update the policy using the reward received directly from the environment</td>
</tr>
</tbody>
</table>

**Fig. 4.** The network used for the simulation. The network is divided to 13 regions, each represented by a holon.

**Fig. 5.** An example of a simple intersection with four signal groups.
4.3. Second level

The holons are extracted according to the algorithm described in Abdoos et al. (2012). A number of neighboring intersections forms a region which is controlled by a holon in the second level. Since all holons in the second level contains a set of first level holons, let us call them as super-holon. Since super-holon has a broader view of the environment in comparison to its members, it can estimate the state of the corresponding holon using more global information.

Assume that \( x_i \) is a super-holon of \( m \) holons (\( x_{i1}, x_{i2}, \ldots, x_{im} \)) and each member has four approaching links: \( l_{ij}, x_{ij}, \) and \( x_{ip} \) as shown in Fig. 5.

\( x_i \) estimates the state of the holon based on the density of vehicles in all links inside the holon. Consider \( \eta \) is the maximum density value of approaching links. The state of \( x_i \) is estimated using \( \eta \), as it shown in Table 5.

\[
\eta = \max_{j=1,...,m; k=1,...,4} \text{density}(l_{ik})
\]  

Fig. 6. Graphical representation of an action (ActionID=12).

Fig. 7. Two members of a holon which connected to each other through the link with maximum density value.

A super-holon plays its role in the corresponding holon by restricting the action space of the members. The super-holon tries to decrease the traffic jam in the approaching link which has the maximum density of vehicles. To this end, it affects members that are directly connected to that link.

Assume that \( l_{ij} \) is the link which has the maximum density among all links inside the holon \( x_i \). Furthermore, assume that \( x_{ij} \) is another member of \( x_i \) which is directly connected to \( x_{ij} \) through \( l_{ij} \) (see Fig. 7). The action space of super-holon is defined such that it increases the output rate of \( l_{ij} \) on one hand and decreases the input rate of \( l_{ij} \) through \( x_{ip} \) on the other hand. Hence, \( x_{ip} \) prevents vehicles to enter to \( l_{ij} \) by assigning specific time splits to signal groups. There are three possible directions for each signal groups: straight, turn left and turn right. Therefore, \( x_{ip} \) assigns green time to \( l_{ij} \), which connects \( x_{ij} \) to \( x_{ip} \) more than the other approaching links.

According to this description, the action space for a super-holon, \( x_i \), is defined as follows:

- **Action 1**: No command.
- **Action 2**: Change priority of \( x_{ij} \) such that the green time assigned to \( l_{ij} \) is more or equal than the other links.
- **Action 3**: Preemption of green time for \( l_{ij} \) such that the shortest green time is assigned to the other phases and the remaining green time is assigned to \( l_{ij} \).
- **Action 4**: Joint action for \( x_{ij} \) and \( x_{ip} \), i.e. \( x_{ij} \) assigns a high priority to \( l_{ij} \) and \( x_{ip} \) assigns a high priority to \( l_{ip} \).
- **Action 5**: Joint action for \( x_{ij} \) and \( x_{ip} \), such that \( x_{ij} \) preempts the green time for \( l_{ij} \) and \( x_{ip} \) changes the priority of \( l_{ip} \) to higher priority.

For example, for \( x=1 \) and \( y=3 \), Table 6 shows the relation between the action selected by \( x_i \) and the two involved members, i.e. \( x_{ij} \) and \( x_{ip} \) in this case. The subset of actions is determined as they satisfy the conditions for green time assignment. Actions 4 and 5 are joint actions that involve two members. Obviously, the subset of actions are determined according to \( x \) and \( y \) and hence it definitely depends on \( x \) and \( y \).
4.4. Algorithm

The proposed Q-learning process contains the following steps:

1. Sub-holons in the first level receive information from the environment and estimate their state at time \( t \). The so-called information is the average queue length of approaching links in a fixed cycle length.
2. The sub-holons send the density of vehicles in all approaching links to their super-holon as abstract state information. This data stream forms the abstract state stream as mentioned before.
3. The super-holons (the holons in the second level) estimate the state of the corresponding holon using the information received from all members according to Eq. (5).
4. Each super-holon selects an action and sends it to its members as an abstract action. This data stream forms the abstract action stream.
5. The sub-holons receive the abstract action and find the corresponding subset of their action (for example Table 6).
6. The sub-holons select an action from the sub-set of actions and assign different green time to the signals.
7. Each super-holon perceives the environment and calculates the reward. The super-holon receives the reward value of its own selected action. It also determines the reward value assigned to each member.
8. The super-holons send the assigned reward value to their members. This data stream forms reward stream.
9. The sub-holons calculate the overall reward by local reward and the reward received from their super-holon using a weighted average method.
10. All holons in both levels update their own policy.

The holonic Q-learning process for two levels is described in Algorithm 1. The algorithm includes three main parts (that is determined in the body of the algorithm): state estimation, action selection and reward calculation.

**Algorithm 1.** The proposed holonic Q-learning algorithm.

```plaintext
for all \( h_i \in H \) do
    for all \( x_{ij} \in h_i \) do
        // state estimation
        estimate the state, \( s_{ij}(t) \), using queue length values
        estimate the density, \( d_{ij}(t) \)
        sendMessage(two tuple \( j, i, 0, d_{ij}(t) \));
        end
        calculate \( \eta \) (according to Eq. (5))
        estimate the state, \( s_{ij}(t) \), using \( \eta \)
        select an action, \( a_{ij}(t) \)
        for all \( x_{ij} \in h_i \) do
            // action selection
            sendMessage(two tuple \( j, i, 1, a_{ij}(t) \));
            find the subset of \( A_{ij} \) using \( a_{ij}(t) \)
            select an action, \( a_{ij}(t) \)
            control traffic signal \( j \) according to \( a_{ij}(t) \)
            end
            calculate the rewards, \( r_{ij}(t−1) \) and \( r_{ij}(t−1) \)
            for all \( x_{ij} \in h_i \) do
                // reward calculation
                sendMessage(two tuple \( j, i, 2, r_{ij}(t−1) \));
                update Q-table, \( Q_{ij} \)
                end
                update Q-table, \( Q_i \)
            end
```

### Table 6

The relation between the actions of the first level holon and its super-holon (it is assumed that the density of \( h_{ij} \) is the maximum value and \( x_{ij} \) is corresponding neighbors of \( x_{ij} \)).

<table>
<thead>
<tr>
<th>Action ID</th>
<th>( A_{ij} )</th>
<th>( A_{ip} )</th>
<th>( A_{aq} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( A_{m} )</td>
<td>( A_{p} )</td>
<td>( A_{q} )</td>
</tr>
<tr>
<td>2</td>
<td>{1,2,3,7,8,9}</td>
<td>( A_{p} )</td>
<td>( A_{q} )</td>
</tr>
<tr>
<td>3</td>
<td>{53,13,13,13}</td>
<td>( A_{p} )</td>
<td>( A_{q} )</td>
</tr>
<tr>
<td>4</td>
<td>{1,2,3,7,8,9}</td>
<td>{2,4,6,13,14,15}</td>
<td>( A_{q} )</td>
</tr>
<tr>
<td>5</td>
<td>{53,13,13,13}</td>
<td>{2,4,6,13,14,15}</td>
<td>( A_{q} )</td>
</tr>
</tbody>
</table>

4.5. Cost of interactions

According to the framework proposed by Gerber et al. (1999), the right to interaction with other agents is an exclusive resource of the holon (not of its sub-holons). Hence, there is a data channel between two equally ranked holons which is managed only by the super-holon.

The constraints in holonic models lead to increase in the cost of interaction, especially between sub-holons of different holons. It is not possible to compute the precise order of cost of interaction generally, because it highly depends on the structure characteristics such as the number of levels and holons, the number of members in each holon and allowed interactions between different entities in the holonic organization. Let focus on a two-level holonic organization. Assume that \( a \) and \( b \) are two sub-holons that belong to super-holons \( h_1 \) and \( h_2 \), respectively. A message passing between \( a \) and \( b \) includes: (i) \( a \) sends a message to \( h_1 \); (ii) \( h_1 \) sends the message to \( h_2 \) (if it is allowed); and (iii) \( h_2 \) sends the message to \( b \). Hence, the cost will be three times higher than a direct message passing.

For a fully connected network composed of \( n \) agents the order of cost of interaction increase from \( O(n^2) \) to \( O(n^3) \) in a two-level holarchy. Since we only consider the interactions between sub-holons and their super-holons, the cost order of interactions is \( O(n) \) in the proposed model.

4.6. An example

Consider a part of the network managed by holon \( h_1 \) which has four members: \( h_{11}, h_{12}, h_{13} \) and \( h_{14} \) as shown in Fig. 8. The average queue length and density value of the links in a fixed cycle are reported in Table 7.

The state of the members is estimated as follows:

\[
\begin{align*}
&h_{11}: \overline{t}_1 \prec \overline{t}_2 \prec \overline{t}_3 \prec \overline{t}_4 \quad \Rightarrow \quad S_1(t) = \text{State 10} \\
&h_{12}: \overline{t}_2 \prec \overline{t}_3 \prec \overline{t}_4 \prec \overline{t}_1 \quad \Rightarrow \quad S_2(t) = \text{State 5} \\
&h_{13}: \overline{t}_3 \prec \overline{t}_4 \prec \overline{t}_1 \prec \overline{t}_2 \quad \Rightarrow \quad S_3(t) = \text{State 14} \\
&h_{14}: \overline{t}_4 \prec \overline{t}_1 \prec \overline{t}_2 \prec \overline{t}_3 \quad \Rightarrow \quad S_4(t) = \text{State 12} 
\end{align*}
\]

Each \( h_{1j}, j = 1, \ldots, 4 \) sends a message containing density value to its super-holon, \( h_1 \), which is used to estimate state of the holon. Since the maximum density is 0.226, the state of the holon is estimated as State 2.

Assume that action 4 is selected by \( h_1 \). Action 4 is mapped to \( A_{1j}, j = 1, \ldots, 4 \) for the members as follows:

\[
\begin{align*}
&h_{11} = \{2,4,6,13,14,15\} \\
&h_{12} = \{1,2,3,7,8,9\} \\
&h_{13} = \{1,2, \ldots, 19\} \\
&h_{14} = \{1,2, \ldots, 19\} 
\end{align*}
\]
5. Experimental results

The proposed method has been tested on a large network that contains 50 intersections and 112 two-way links as depicted in Fig. 4. Aimsun traffic simulator\(^1\) has been used to create the topology of the network. The number of lanes per link is three for all links and the maximum speed allowed to vehicles is 50 km/h. Furthermore, the vehicles are inserted to the network through 20 sources which lie on the border of the network. The number of lanes per link is three for all links and the maximum speed allowed to vehicles is 50 km/h. The average length of links is 847.44 m.

Although the performance of the methods depends on the parameters value, most of them have a common effect on the performance for different methods. For example, an increase in the maximum allowed speed decreases the delay value in simulation when different methods are used. Among different simulation parameters, the most effective one on delay time is traffic demand value. On the other hand, traffic demand usually varies in different hours of a day. A practical approach for traffic signals control, should be enabled to work in different traffic demands. Therefore, we only consider the impact of traffic demand values on average delay time during simulations.

Some experiments were performed in order to compare the proposed method in two modes: the individual Q-learning in the first level and the holonic Q-learning designed for two holarchical levels. Learning parameters values which are used in the simulation are given in Table 9. A series of experiments was performed to evaluate delay time and flow of the network for the proposed approach. The experiments determine the simulated average delay time for different traffic demands (Table 9).

Among different traffic demands we choose three values as low, medium and high traffic congestion (traffic demands 1–3 in Table 9). In these experiments, the average delay time and flow are used for performance evaluation.

In the first experiments, there is no holonic control and first level agents run the classic Q-learning algorithm in an individual way. In the second experiments, the proposed holonic Q-learning is used to control the traffic signals. The average delay time and flow of the individual Q-learning and the holonic Q-learning for different traffic demands are shown in Table 9.

\(^1\) http://www.aimsun.com.

### Table 7

<table>
<thead>
<tr>
<th>Link number</th>
<th>Id</th>
<th>Avg. queue length</th>
<th>Density</th>
<th>Link number</th>
<th>Id</th>
<th>Avg. queue length</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>11.6</td>
<td>0.154</td>
<td>9</td>
<td>1</td>
<td>6.0</td>
<td>0.157</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>10.6</td>
<td>0.149</td>
<td>8</td>
<td>2</td>
<td>13.0</td>
<td>0.196</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>8.0</td>
<td>0.132</td>
<td>11</td>
<td>3</td>
<td>6.0</td>
<td>0.146</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>17.6</td>
<td>0.197</td>
<td>12</td>
<td>4</td>
<td>9.3</td>
<td>0.170</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>18.6</td>
<td>0.226</td>
<td>13</td>
<td>5</td>
<td>10.0</td>
<td>0.113</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>12.0</td>
<td>0.187</td>
<td>14</td>
<td>6</td>
<td>4.3</td>
<td>0.087</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>14.0</td>
<td>0.201</td>
<td>15</td>
<td>7</td>
<td>7.6</td>
<td>0.098</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>13.3</td>
<td>0.204</td>
<td>16</td>
<td>8</td>
<td>10.6</td>
<td>0.121</td>
</tr>
</tbody>
</table>

\(\alpha_{1,1}\) and \(\alpha_{1,2}\) select an action from limited action space, i.e. \(A_{1,1}\) and \(A_{1,2}\), respectively, while \(\alpha_{1,3}\) and \(\alpha_{1,4}\) select an action from the whole set, i.e. \(A_{1,3}\) and \(A_{1,4}\).

### Table 8

<table>
<thead>
<tr>
<th>Properties</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of intersections</td>
<td>50</td>
</tr>
<tr>
<td>Number of links</td>
<td>224</td>
</tr>
<tr>
<td>Average length of links</td>
<td>847.44 m</td>
</tr>
<tr>
<td>Number of lanes per links</td>
<td>3</td>
</tr>
<tr>
<td>Maximum speed</td>
<td>50 km/h</td>
</tr>
<tr>
<td>Number of input/output centroid</td>
<td>20</td>
</tr>
<tr>
<td>Arrival distribution</td>
<td>Normal</td>
</tr>
<tr>
<td>Simulation duration</td>
<td>10 h</td>
</tr>
<tr>
<td>Traffic demand 1</td>
<td>10,000 veh/h</td>
</tr>
<tr>
<td>Traffic demand 2</td>
<td>20,000 veh/h</td>
</tr>
<tr>
<td>Traffic demand 3</td>
<td>30,000 veh/h</td>
</tr>
</tbody>
</table>

### Table 9

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\beta)</td>
<td>Learning rate</td>
<td>0.9</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>Discount factor</td>
<td>0.4</td>
</tr>
<tr>
<td>(\epsilon)</td>
<td>Exploration rate</td>
<td>0.9</td>
</tr>
</tbody>
</table>

### Table 10

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Traffic demand number</th>
<th>Individual Q-learning</th>
<th>Holonic Q-learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average delay time</td>
<td>1</td>
<td>49.6077</td>
<td>48.9557</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>139.2482</td>
<td>62.07374</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Standard deviation of delay</td>
<td>1</td>
<td>25.653</td>
<td>21.0913</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>202.5363</td>
<td>26.6049</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Average harmonic speed</td>
<td>1</td>
<td>30.9975</td>
<td>31.1663</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>17.67019</td>
<td>27.99178</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Standard deviation of</td>
<td>1</td>
<td>6.61357</td>
<td>6.06995</td>
</tr>
<tr>
<td>harmonic speed</td>
<td>2</td>
<td>11.85251</td>
<td>6.138003</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>
traffic demand 1 are shown in Figs. 9 and 10. The reported delay values are the average computed over 10 runs. Table 10 shows the average delay time and the average standard deviation of delay time computed over 10 simulations during 10 h. Furthermore, the average harmonic speed and standard deviation of the speed are also given.

As it can be seen from the figures, the performance is almost the same for the holonic Q-learning and individual Q-learning. According to the data reported in Table 10, the average standard deviation of delay time of the holonic Q-learning method is less than the individual Q-learning method. A high standard deviation shows that the data is widely spread (less reliable) and a low standard deviation shows that the data are clustered closely around the mean (more reliable).

Figs. 11 and 12 depict the average delay time and flow of the network for traffic demand 2. The performance is good at the beginning because of the time necessary for the network to saturate. However, this changes during the time, the figures show a significant increase in performance of the holonic Q-learning. The network begins to saturate after some hours when individual Q-learning is used for signals control while the holonic Q-learning keeps the average delay time value around a fixed value. The figures also show that the holonic Q-learning controls the signals more reliable than the individual one. The average delay time values reported in Table 9 indicate that the holonic control of the network leads to less delay time and more flow rate while the standard deviation indicates the reliability of the holonic Q-learning.

The average delay time and flow of the network for traffic demand 3 are shown in Figs. 13 and 14. Although the network begin to saturate after a few hours using the both methods, but the performance of the methods are different. When the individual Q-learning is used, the network starts to saturate before four hours and when the holonic Q-learning is used the network starts to saturate after six hours. Since the statistical data are not valid after saturation, it is not possible to calculate the average value and standard deviation for this traffic demand. It is worth noting that there are some situations where preventing from saturation is unavoidable. The holonic Q-learning is highly advantageous due to postponing the network saturation.

6. Conclusion and future works

Multi-agent system is usually used to model distributed complex real-world problems. An approach to cope with large-scale complex system is to organise agents and decompose the problem in some smaller sub-problems. Holonic multi-agent system provides such an organisation in which self-similar holons arranged in a multi-level holarchy. This paper uses a holonic multi-agent system to model a large number of signals controller agents. A holonic Q-learning method is applied to learn signals timing in two holarchical levels. A network containing fifty intersections decomposes to thirteen holons each assigned to a holon which stands on the second level.
The holons in the first level use a large state and action space while the holons in the second level use a smaller space. The holons in both levels try to learn the optimal policy simultaneously. Lower level holons send message to higher-level holons which is used to estimate the state of a part of the network. Higher-level holons learns how restrict the action space for the members and lower-level holons learns which actin leads to the best future value among the set/subset of actions at each state. During all the learning process each holon updates its own policy using the data that transfer between the holons in two levels. The results show that the holonic Q-learning stands out the individual Q-learning.

We are currently working on the holonic aspect of the proposed method and the analysis of the holonic Q-learning that will be published in future. How to organise the agents to form the holons in different levels is another issue that we are working on. We also wants to implement a learning method that deal with dynamic organisation of holons in different levels.

Future works include extending our method to more holarchical levels in order to contribute more abstracted data that contain broader view of the system; to apply the proposed holonic Q-learning method in other real-world applications. Furthermore, analysis of the impact of higher level data to speed up the convergence of the Q-learning is another open issue in this area.

Acknowledgement

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

References


Balaji, P., German, X., Srinivasan, D., 2010. Urban traffic signal control using the data that transfer between the holons in two levels. The first authors would like to thank Research Institute for Information and Communication Technology—ITRC (Tehran, Iran) for their supports.


