



# A distributed approach for road clearance with multi-robot in urban search and rescue environment

Amar Nath<sup>1</sup> · A. R. Arun<sup>1</sup> · Rajdeep Niyogi<sup>1</sup>

Received: 24 July 2019 / Accepted: 1 November 2019 / Published online: 13 November 2019  
© Springer Nature Singapore Pte Ltd. 2019

## Abstract

In urban search and rescue domains, robots explore the affected terrain to search and assist disaster victims. RoboCup Rescue simulation provides a platform for disaster management where heterogeneous field agents (fire-brigade agent, ambulance agent, and police force agent) collaborate to manage a mimicked calamity situation. The role of police force agents is crucial, as they clear the blocked roads to allow other agents to perform their tasks. In this paper we suggest a distributed multi-robot coordination approach for clearing a road blocked with heavy obstacles. The proposed framework is implemented and simulated in ARGoS, a multi-robot simulator. The experimental results show the validity and satisfactory performance of the proposed approach.

**Keywords** Urban search and rescue · Multi-robot coordination · Task allocation · Distributed algorithm

## 1 Introduction

A set of cooperative and collaborative robots can accomplish a mission in a faster and better way. Physical foraging tasks, i.e., cooperative object transportation, require considerable effort from multiple robots for task execution. A distributed autonomous multi-robot system (MRS) is applicable in many domains such as urban search and rescue, military surveillance, landmine detection, and warfare (Yan et al. 2013). MRS exhibits scalable, fault-tolerant, and accelerated task completion capabilities when compared to a single standalone robot. The application of MRS for urban search and rescue (USAR) is quite challenging because of the uncertain and dynamic nature of the rescue environment.

Rescue operation with intelligent robots has several advantages over rescuing with human being. A robot does not feel stressed or fatigued like their human counterparts. Rescue operation can be speed-up by increasing the number

of robots. Moreover, robots are easily scalable and a damaged robot can be replaced with a new one, but the loss of a human being is everlasting (Burke et al. 2004).

RoboCup Rescue simulation environment is used to promote research in the area of USAR. It provides a platform for disaster management where heterogeneous field agents (fire-brigades, ambulance, and police force agents) collaborate to manage a mimicked calamity situation. The agents are responsible for extinguishing fire, save life of human beings, and clearing road. A fully autonomous USAR robot explores the rescue environment and search victims. In USAR domain, the role of police force agents is crucial, as they clear the blocked roads to allow other agents, i.e., ambulance and fire-brigade, to perform their tasks. After a disaster, a road in the city may get be blocked by any type of obstacle (road-blockers), i.e., light or heavy. The obstacles may be scattered randomly on the road. The road clearance problem, in the RoboCup Rescue simulation environment (Uhrmacher and Weyns 2018; Kitano 2000), requires a framework that guarantees road-clearance, regardless of the possibility of heavy obstacles present on the road.

Robots are useful for search and detection missions for trapped victims after disasters due to their capabilities to enter a region, which may endanger a human life. There are numerous risks involved when working in a USAR task force, and the health and safety of the rescue workers and victims are always of primary concern (Liu and Nejat 2013;

✉ Amar Nath  
anath@cs.iitr.ac.in

A. R. Arun  
arunnadar1994@gmail.com

Rajdeep Niyogi  
rajdpfec@iitr.ac.in

<sup>1</sup> Indian Institute of Technology Roorkee, Roorkee,  
India 247667

Casper and Murphy 2003). In the present era, it is being expected to use an intelligent, autonomous multi-robot system for the rescue operation (Sato et al. 2004). The multi-robot approach can enhance the speed of rescue operations by scaling up the appropriate number of robots involved (Visser et al. 2014). Cooperative work by multiple robots has many benefits for search and rescue task execution.

The work (Sedaghat et al. 2005) suggested a task allocation algorithm for police force agents in the RoboCup Rescue simulation environment. Task allocation in a multi-robot system is the problem of deciding which robots should execute which tasks keeping in mind whether the overall system goal is accomplished or not. However, the problem considered in this paper is how to assign tasks at run-time. The work (Sedaghat et al. 2005) considered the problem of task assignment to a single agent, and thus their approach cannot solve the road clearance problem if heavy obstacles block the road.

To clear a large and heavy obstacle, a joint effort by multiple police force agents is required as a single police force agent cannot handle the heavy obstacle. Hence, whenever a heavy obstacle is detected, a coalition of agents is needed, depending on the current state of the agents, to clear the heavy obstacle. A coalition is a finite group of agents that have agreed to cooperate to execute a task. Coalitions are formed temporarily, and once the task is executed, the coalition is dissembled, and from now onward, the members may be a part of some other coalition. This process of coalition formation/dissembling continues until all the tasks are completed. Coalition formation at run-time is a challenging problem as in a distributed environment, no robot knows the states, locations, and skills of other robots (i.e., the absence of global knowledge). Thus, the robots should communicate among themselves to acquire relevant information for task execution without the intervention of any central authority. This necessitates the design of a distributed algorithm (Lynch 1996) for task execution in such an environment. In this paper, we suggest a distributed algorithm for task execution (see Sect. 3).

As per the taxonomy, given in Gerkey and Mataric (2004), the aforesaid problem falls in the category of ST-MR-IA (ST means a robot can execute at most single task at a time, MR means that some tasks can require multiple robots, IA stands for instantaneous allocation (i.e., at run-time) of robots for the tasks). In this paper, a distributed multi-robot approach to clear the road in the RoboCup Rescue simulation environment, even though the road is blocked with heavy obstacles, is suggested. The implementation of the framework is carried out in ARGoS (Pinciroli et al. 2012), a multi-robot simulator.

The remainder of the paper is structured as follows. In Sect. 2 we discuss the related work. In Sect. 3, we suggest a distributed approach for multi-robot coordination. In Sect. 4,

we discuss the implementation of the proposed approach. The experimental results are given in Sect. 5. Some discussions are made in Sect. 6. We conclude in Sect. 7.

## 2 Related work

The RoboCup Rescue project was motivated by the Hanshin-Awaji Earthquake, which hit Kobe City on January 17, 1995 (Uhrmacher and Weyns 2018). The lessons learned from this earthquake indicate that information systems should be built to support the collection of necessary information and prompt planning for disaster mitigation. The RoboCup Rescue project was proposed in Tadokoro et al. (2000). This rescue project intended to promote research and development in the disaster rescue domain at various levels: multi-agent teamwork coordination, physical robotic agents for search and rescue, information infrastructure, personal digital assistants, standard simulator and decision support systems.

When earthquakes occur in urban areas, various types of causalities and accidents occur. These are related to one another. Collapsed buildings injure civilians and block roads with debris. The rescuers must rush the victims to hospitals, help civilians evacuate to safe areas, and prevent fires, if any, from spreading. The accumulated debris hinders the rescue operations (Uhrmacher and Weyns 2018).

The work (Sedaghat et al. 2005) suggested a task allocation algorithm for police force agents in the RoboCup Rescue simulation environment. Task allocation in a multi-robot system is the problem of deciding which robots should execute which tasks in order to accomplish the overall system goal. The work (Sedaghat et al. 2005) considered the assignment of a single task to single agent and thus their approach cannot solve the road clearance problem where heavy obstacles are present. The road clearance problem considered in this paper requires a coalition to be formed in a distributed manner at run-time. We now discuss some approaches for coalition formation and make a brief comparison with our work.

Coalition or team formation is done in a distributed manner in Abdallah and Lesser (2004), Coviello and Franceschetti (2012), Tošić and Agha (2004). The proposed approach differs from these approaches) Abdallah and Lesser (2004; Coviello and Franceschetti 2012; Tošić and Agha 2004) in the following manner. The proposed algorithm does not use any hierarchical structure among agents as in Abdallah and Lesser (2004). In Abdallah and Lesser (2004), the coalition formation process is delegated to other managers, whereas in our approach either the initiator succeeds or fails in the coalition formation process. In our approach, an initiator can communicate with anyone of agents and is not as restrictive as in Coviello and Franceschetti (2012). In our approach, there are different types of messages to control the

different actions of it, whereas in Coviello and Franceschetti (2012) there are only request and accept/reject messages.

Multi-robot coalition formation has been studied by Vig and Adams (2006), where each robot has some capabilities and a coalition has to perform some task for which some capabilities are required. The authors consider the problem of coalition formation such that there is no coalition imbalance, which happens when a large part of resources is responsible for any robot in a coalition. A balance coefficient is thus defined and it is used to form a suitable coalition for a given task. In our algorithm, all the agents in a coalition have required capabilities (skills) for task execution. Thus the notion of imbalance does not arise in our case. Moreover, our algorithm is fully distributed which is not the case in Vig and Adams (2006).

Our strategy of coalition formation differs from the above approaches (Abdallah and Lesser 2004; Coviello and Franceschetti 2012; Tošić and Agha 2004) in the following manner. Our approach does not use any hierarchical structure among agents as in Abdallah and Lesser (2004). In Abdallah and Lesser (2004), the coalition formation process is delegated to other managers, whereas in our approach either the initiator succeeds or fails in the team formation process; delegation is not allowed. In our approach, an initiator can communicate with anyone of agents and is not as restrictive as in Coviello and Franceschetti (2012). In our approach, there are different types of messages whereas in Coviello and Franceschetti (2012) there are only request and accept/reject messages.

Auction-based approaches for team/coalition formation (task allocation) are suggested in Gerkey and Mataric (2002), Kong et al. (2015), Xie et al. (2018). A bidder agent has some resources (e.g., data center, CPU) (Kong et al. 2015), who may bid for multiple auctioneers concurrently. In our work a non-initiator robot (the bidder) will not express its willingness to multiple initiators (auctioneers) concurrently; when more than one request message arrives, the robot stores the requests in its local queue. Having one or more resources specified in the auction is a sufficient condition for an agent to make a bid (Kong et al. 2015). Having the required skills for a task is a necessary but not a sufficient condition for a robot to express its willingness to be part of a team, in our work. The behavior of the robots (modeled using the communicating automation), is determined by its current state, whereas in Gerkey and Mataric (2002), Kong et al. (2015) states need not be taken into consideration. Our algorithm accommodates all these aspects and so it is substantially more complex than auction based protocols used in Gerkey and Mataric (2002), Kong et al. (2015). The work (Xie et al. 2018) considers only free and busy states of the agents and it cannot handle the aforesaid problem (where multiple robots may require to execute a task) directly.

In our work a non-initiator robot (the bidder) will not express its willingness to multiple initiators (auctioneers)

concurrently; when more than one request message arrives, the robot stores the requests in its local queue. Having one or more resources specified in the auction is a sufficient condition for an agent to make a bid (Kong et al. 2015). Having the required skills for a task is a necessary but not a sufficient condition for a robot to express its willingness to be part of a team, in our work. A robot's behavior, in our work, is determined by its current state, whereas in Gerkey and Mataric (2002), Kong et al. (2015) states need not be taken into consideration.

Delays of messages are not of much concern in auction-based algorithms, which however becomes necessary in any real-world environment. Although an auction-based algorithm may be easily implementable for a software agent, the implementation becomes, in general, highly non-trivial for robot systems since several aforementioned aspects are not considered in the design of these algorithms.

The work (Dukeman and Adams 2017) considered a multi-robot task assignment problem using a centralized approach. In Gunn and Anderson (2015), the authors describe a framework for dynamic heterogeneous team formation for robotic urban search and rescue. The task discovery is made by a member of a team and it is sent to the team coordinator for assignment. The team coordinator (central agent) performs the task assignment, ensuring the task is carried out by a robot with the necessary capabilities. A centralized approach for task allocation and human-robot collaboration in urban search is proposed in Saad et al. (2018). The tasks are managed by a centralized manager. In Xie et al. (2018), a decentralized approach for object transport via implicit communication is suggested. Our approach uses a distributed approach for team formation at run-time via explicit communication and thus different from these approaches (Dukeman and Adams 2017; Gunn and Anderson 2015; Saad et al. 2018; Xie et al. 2018).

### 3 Proposed approach

In this section, a distributed multi-robot coordination approach for clearing a road, where heavy obstacles are present, by police force agents, is discussed. To clear a road, coalition formation of police force agents at run-time is required. The police force agents should coordinate among themselves to accomplish the task without any central or human intervention. We consider a wireless network that is lossless, message delay is finite, data is not corrupted during transmission, and messages are delivered in a FIFO manner.

It is assumed that a city has several roads and to rescue the city after a disaster hit, important roads of the city should be cleared first, so that other agents, i.e., fire-brigade and ambulance could get access to the disaster site. The obstacles may be scattered on the road randomly and hence need to be

searched first. The proposed distributed approach is given in Algorithms 1, 2 and 3. The police force agents search and move the obstacles. A distributed control given in Algorithms 1, 2 and 3, is proposed based on the communicating automaton (Figs. 1 and 2). Communicating automaton represents the operational behavior of police force agents based on their states and transitions among them. Each police force agent has a behavioral control system at its core. Although at the higher level, every police force agent is capable of operating in one of 9 states during algorithm run; IDLE, SEARCHING, ANALYSIS, PROMISE, APPROACH, READY, WAITING, COORDINATION and MOVE. Initially, all police force agents are in IDLE state, which significance that agents are not engaged in any activity.

Whenever police force agents receive a road clearance signal, they all change their state from IDLE to SEARCHING and start exploring the road for obstacles. If an agent detects an obstacle during the search, it changes its state from SEARCHING to ANALYSIS and determines whether the obstacle is heavy or light (line no. 3, Algorithm 1). If the obstacle is light then it grabs the light obstacle and shifts it to the nearest location of the road. But, if the obstacle is heavy, agent  $i$ , determines the total number of agents required to move the heavy obstacle (i.e.,  $k$ ) and then starts

coalition formation process by invoking the function `Coalition_Formation` (line number 11, Algorithm 1).

Now, initiator (agent  $i$ ), broadcasts *Request* message (contains initiator's  $id$  and location  $l$ ) (line no. 5, Algorithm 2). The agent, who invokes the coalition formation function (Algorithm 2), termed as the initiator and rest of the agents are termed as non-initiators. The initiator waits for some units of time to get replies from non-initiator. The non-initiators, who are in the SEARCHING state, respond to initiator's *Request* message by sending *Willing* message (contains non-initiator's  $id$  and location  $l$ ). The non-initiator agents who respond to the initiator, changes its state from SEARCHING to PROMISE (line no. 1–3, Algorithm 3). The initiator agent increments its counter ( $c$ ) on the receipt of each *Willing* message (line number 8, Algorithm 2).

Whenever the initiator's waiting time is over, it checks whether it has received the sufficient number of *Willing* messages i.e.,  $c \geq (k - 1)$ , if yes, it selects  $(k - 1)$  nearest non-initiator agents. Initiator then, sends *Confirm* message to the selected  $(k - 1)$  agents and *Not-Required* to the remaining  $c - (k - 1)$  agents. Now, the initiator changes its state from ANALYSIS to WAITING state and waits till all the non-initiators reach its location.

---

#### Algorithm 1: Task execution

---

```

1  clear := false;                                ▷ road is not cleared
2  while (clear ≠ true) do
3      if (state = SEARCHING) AND (agent i finds an obstacles B) then
4          state := ANALYSIS;
5          if (B is light) then
6              | Agent i moves the B to the side of the road;
7          else
8              if (B is heavy) then
9                  | Agent i determines the no. of agents required to move the heavy
10                 | obstacle B (i.e., k);
11                 | Agent i starts the coalition formation process;
12                 | boolean i.b := Coalition_Formation(i, (k - 1), l); ▷ Agent i
13                 | becomes initiator, Algorithm 2,3
14                 | if (i.b = true) then
15                     | obstacle B is moved successfully;
16                 | else
17                     | obstacle B is not moved successfully;
18                 | end
19             end
20         end
21     if (Agent i finds no obstacles on the road) then
22         | clear := true;                                ▷ road is cleared
23         | A message (Road_Is_Cleared) is send to ambulance and fire brigade
24         | agents;
25     end
26 end
27 end
28 Exit;

```

---

**Algorithm 2:** Coalition\_Formation: Initiator

---

```

1 if (state = ANALYSIS) then
2   boolean b := false;
3   int c := 0;           ▷# Willing messages
4   int a := 0;           ▷# robots, have reached to location l
5   broadcast Request(init, loc(init));
6   initiator wait for few units of time;
7   if ((Waiting time of initiator is not over) AND (Agentinit receives Willing
      message)) then
8     | c := c + 1;
9   end
10  if (Waiting time of initiator is over) AND (c ≥ (k − 1)) then
11    | initiator selects (k − 1) nearest agents for the coalition;
12    | send Confirm to these (k − 1) non-initiators;
13    | send Not-Required to (c − (k − 1)) non-initiators if any;
14    | state := WAITING;           ▷ initiator waits till all non-initiators
      | reach to initiator's location
15  else
16    | if (Waiting time of initiator is over) AND (c < (k − 1)) then
17      | sends Not-Required to c non-initiators;
18      | state := SEARCHING;
19      | return b; ▷ coalition not formed successfully
20      | initiator leaves the task; state = SEARCHING;
21    | end
22  end
23 end
24 while (state = WAITING) do
25   if (receives Arrived) then
26     | a := a + 1;
27     | if (a = (k − 1)) then
28       | initiator decides target location where to shift the obstacle;
29       | state = COORDINATION;
30       | initiator sends Sync message to (k − 1) non-initiators;
31     | end
32   end
33 end
34 if ((state = COORDINATION) AND (initiator receives Beacon signal from all
      (k − 1) non-initiators)) then
35   | initiator sends go_Beacon signal to (k − 1) non-initiator
36   | state := MOVE;
37   | Agent start moving towards goal location in MOVE state;
38   | b := true;
39   | return b;
40 end
41 if ((state = MOVE) AND (initiator have reached to the side of the road)) then
42   | initiator leaves the obstacle;
43   | state = SEARCHING;
44 end

```

---

**Algorithm 3:** Coalition\_Formation: Non-initiator

---

```

1 if (state = (SEARCHING)) AND (receives Request) then
2   | state := PROMISE;
3   | sends Willing;
4 else
5   | Ignore the Request message;
6 end
7 if ((state = PROMISE) AND (receives Confirm)) then
8   | state := APPROACH;
9 end
10 if ((state = PROMISE) AND (receives Not – required)) then
11   | state := SEARCHING;
12 end
13 if ((state := APPROACH) AND ( $|initiator.loc() - non-initiator.loc()| \leq \delta$ )) then
14   | non-initiator grab the obstacle;
15   | non-initiator sends Arrived;
16   | state := READY;
17 end
18 if ((state := READY) AND (non-initiator receive Sync message)) then
19   | non-initiator synchronize itself with initiator;
20   | non-initiator send Beacon signal;
21 end
22 if ((state := READY) AND (non-initiator receives go_Beacon signal)) then
23   | state := MOVE;
24   | non-initiator starts moving towards the location, decided by initiator;
25 end
26 if ((state = MOVE) AND (non-initiator have reached at the side of the road)) then
27   | non-initiator leaves the obstacle;
28   | state := SEARCHING;
29 end

```

---

If a non-initiator receives a *Not-Required* message, it changes its state from *PROMISE* to *SEARCHING*. But, if it receives a *Confirm* message, it changes its state from *PROMISE* to *APPROACH*. The non-initiator agent in the *APPROACH* state moves toward the initiator's location. Whenever a non-initiator agent reaches very close to the initiator, it changes its state from *APPROACH* to state *READY*, grabs the obstacle and sends an *Arrived* beacon signal to the initiator (line number 13–16, Algorithm 3). On receipt of *Arrived* beacon signal, initiator increments its counter  $a$  by one and when  $a$  becomes equals to  $(k - 1)$ , it means that all the non-initiator agents have reached to the location of the initiator. Now, the initiator changes its state from *WAITING* to *COORDINATION* and sends *Sync* message for synchronization (line number 24–32, Algorithm 2).

The non-initiator synchronizes and aligns itself according to the instruction provided by the initiator via *Sync* message. The *Sync* message contains the target location where to move the obstacle. After synchronization, the non-initiator sends a *Beacon* signal to indicate that it has synchronized. After sensing the *Beacon* signal from all  $(k - 1)$  non-initiators, the initiator sends a *go\_Beacon* signal to indicate that all the team members are ready to move. After sensing

the *go\_Beacon* signal, a non-initiator changes its state from *READY* to *MOVE* and starts moving by grabbing the obstacle.

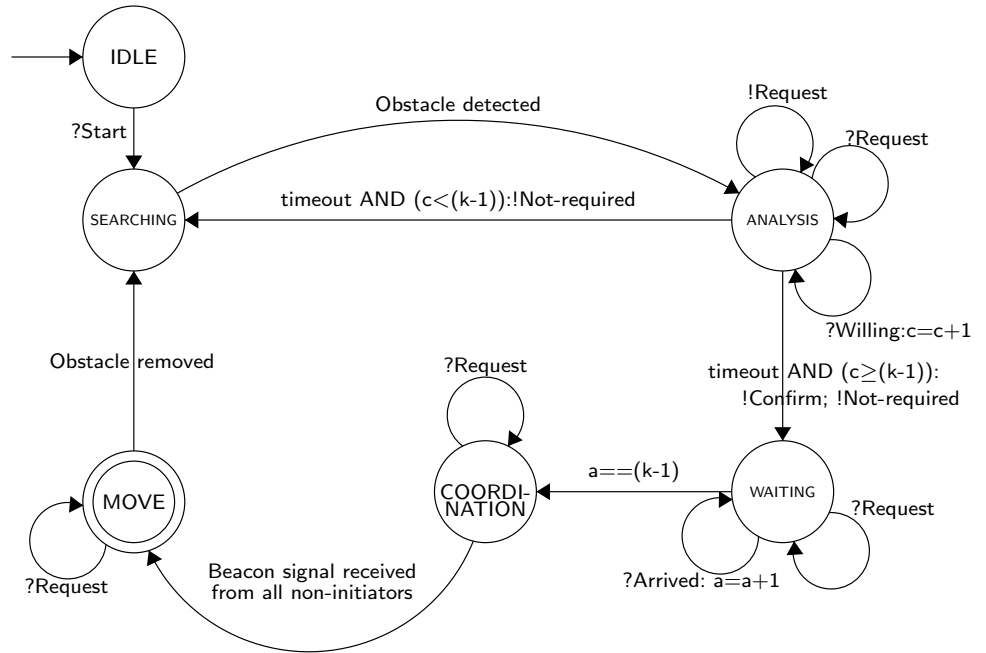
As the agents of the coalition carrying heavy obstacle reach the side of the road, they leave the heavy obstacle there and again start searching for more obstacles if any. The initiator as well as the non-initiator changes their states from *MOVE* to *SEARCHING* (line number 40–42, Algorithm 2, and line number 26–29, Algorithm 3). Control passes back to Algorithm 1 in line number 11, in both the case; after shifting the obstacle successfully (line number 36, Algorithm 2); coalition is not formed successfully (line number 19, Algorithm 2). In this way, if no obstacles are found on the road, the algorithm terminates (line number 19–21, Algorithm 1). Finally, a message (*Road\_Is\_Cleared*) is sent to the ambulance and fire-brigade agents to let them know about the road clearance and so that they can proceed now. The proposed approach is robust, as, if a robot fails while executing a task, the initiator can replace it with another robot present in the environment by using Algorithms 2 and 3.

The communicating automaton (CA) (Bérard et al. 2013) based model is used to capture the behavior of robots, shown

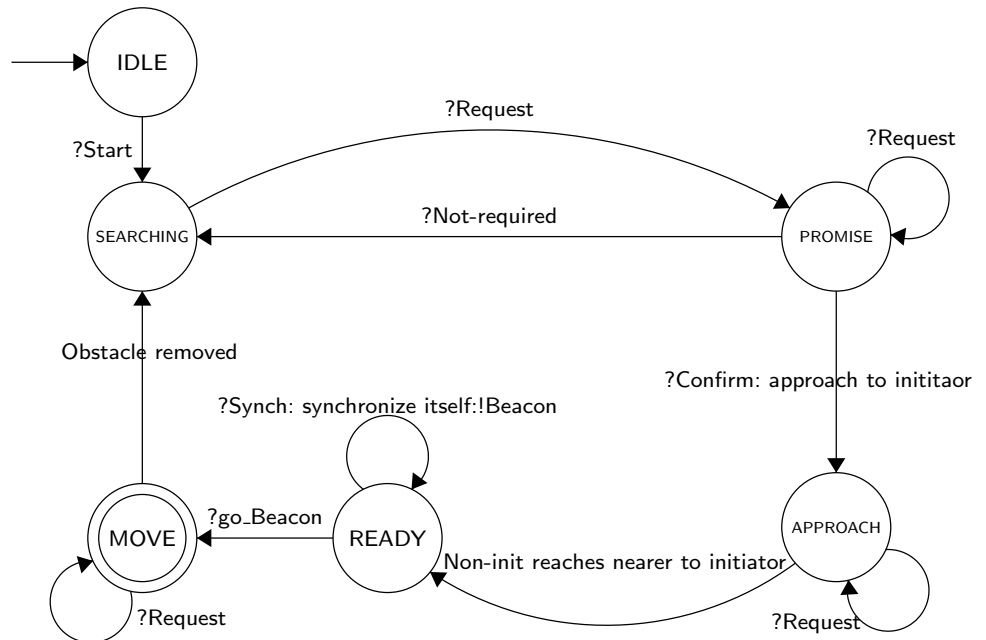
in Figs. 1 and 2. This model is very helpful for designing, understanding and implementing the proposed algorithm. The state of robots changes on receipt of messages (e.g., ?Start, ?Request, ?Arrived, etc.). Initially, all the robots are in IDLE state and on task detection, a robot goes to ANALYSIS state. The initiator and non-initiator behavior is captured by communicating automata given in Figs. 1 and 2 respectively. The overall description of the communicating automata is discussed in Sect. 3 (description of the Algorithms 1, 2, and 3).

A CA is like a finite automaton where the transitions may involve sending/receiving of messages. A label of a transition, in the CA that we use, has a more general form  $\chi : \gamma$ , where  $\chi$  can either be an input  $a$  (send message  $!m$ , receive message  $?m$ ), or a state condition  $g$ , and  $\gamma$  can either be a sequence of actions  $seq$ , or empty. The semantics of the transitions are:  $s_1 \xrightarrow{a} s_2$  means switch from  $s_1$  to  $s_2$  on input  $a$ ;  $s_1 \xrightarrow{g} s_2$  means switch from  $s_1$  to  $s_2$  if a condition  $g$  holds at  $s_1$ .

**Fig. 1** Communicating automata model of initiator agent



**Fig. 2** Communicating automata model of non-initiator agent



### 3.1 Communication complexity

As per the proposed distributed framework (Algorithm 1, 2 and 3), the messages are not exchanged while agents accomplish simple tasks (light obstacles). The messages are exchanged only when dealing with complex tasks. Communication is needed to form the coalition and to coordinate the activities of agents. Let, the total number of tasks (obstacles on the road) and police force agents are  $n$  and  $(m)$  respectively. In the worst scenario, assume, all of the tasks are complex (heavy obstacles). At some moment in time, each robot finds the heavy obstacle, where  $(m \geq n)$ . Each robot start its coalition formation process by broadcasting  $(m - 1)$  Request message. In this case, there is  $n$  number of initiators because the total number of heavy tasks found is  $n$ , hence a total of  $n * (m - 1)$  messages are sent. The total number of replies (Willing) messages send by non-initiator would be at most  $(m - n)$ , because at this time at most  $(m - n)$  agents may be in the SEARCHING state. Now, each initiator agent determines whether the coalition can be formed or not and then, it sends  $(m - n)$  replies (Confirm or Not-required).

After receiving the Confirm message, the non-initiator robots reach the initiator's location and send a maximum of  $(m - n)$  Arrived beacon signal that do not effect the communication overhead as its are signal not messages. When all robots of the coalition have reached the location of initiator, initiator sends  $m - n$ , Sync beacon signal to the non-initiators and similarly do not have any effect. In this way, total number of messages would be  $(n * (m - 1))$  broadcast +  $(m - n)$  Willing +  $(m - n)$  Confirm or Not-required + 0Arrived + 0Sync and this becomes  $(mn + 2 * (m - n) - n = (mn + 2m - 3n))$ . Hence, the message complexity is  $O(mn)$ .

## 4 Implementation using ARGoS

The proposed framework is implemented using ARGoS (Autonomous Robots Go Swarming), a multi-robot simulator (Pinciroli et al. 2012). The main devices of the foot-bot robot supported by ARGoS are given in Fig. 3. One of the most interesting features of the ARGoS simulator is that, it allows a user to modify every aspect of a simulation.

In this paper, we have used foot-bots; a type of robot supported by ARGoS. The police force agents (foot-bots) search for road-blockers (obstacles) inside the road (arena created as shown in Fig. 5). If a robot detects a road-blocker, it approaches to the road-blocker to grab it. A robot checks whether road-blocker is a light or heavy. If it is a light road-blocker, the robot moves it on its own to the nearest place of the road. On the other hand, if the road-blocker is heavy it calls for help (as per algorithms discussed in Sect. 3). After, the non-initiators have arrived to the location of

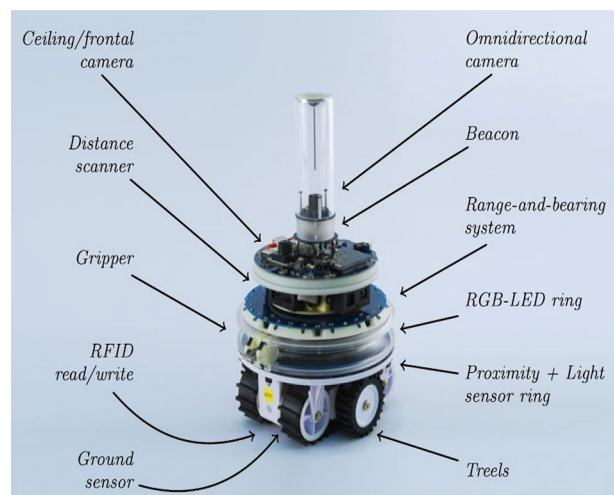


Fig. 3 The foot-bot robot supported by ARGoS (Pinciroli et al. 2012)

Table 1 Sensors and actuators used in foot-bot robot

Sensors	Actuators
Positioning	Footbot_turret
Colored_blob_omnidirectional_camera	Footbot_gripper
Footbot_motor_ground	Leds
Differential steering	Range_and_bearing
Footbot_proximity	Differential_steering
Range_and_bearing	

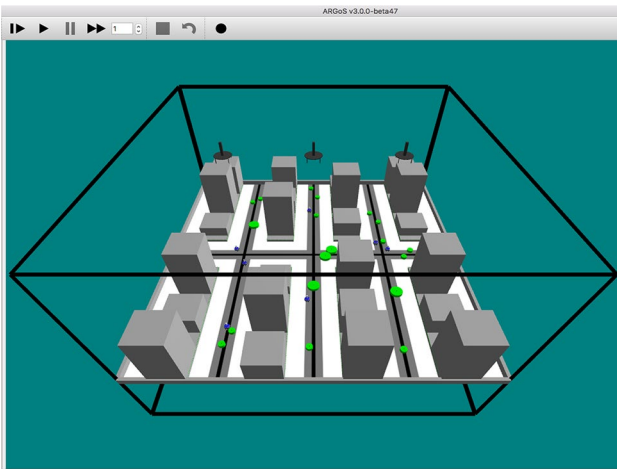
road-blocker, they jointly move it. After successfully moving the road-blockers, the robots go back to the search state and repeats the same activities until all the road-blockers are moved. To perform the search and other activities, the foot-bots use sensors and actuators given in Table 1.

### 4.1 Experimental setup

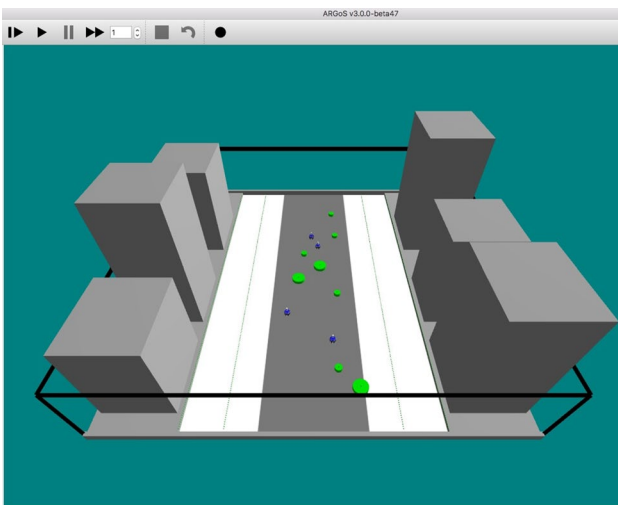
A prototype is constructed in ARGoS to mimic the road clearance scenarios of USAR. The foot-bot robot in ARGoS plays the role of a police force agent of USAR. The initial setting of a city having several roads is shown in Fig. 4. Buildings are immovable boxes and the borders of a road are denoted by green light.

Consider a city with multiple roads as shown in Fig. 4. A road, blocked with obstacles is taken as a case (see Fig. 5). Now, this road needs to be cleared. The police force agents will clear the road, so that other agents, i.e., fire-brigade and ambulance could get access to the disaster site. A snapshot of the blocked road is shown in Fig. 5. The small obstacles are represented by movable cylinders of radius 0.1 m with a red light in the top-center of the cylinder. The large obstacles are represented by a cylinder of radius 0.2 m with a blue light in the top-center of the cylinder (Fig. 6).

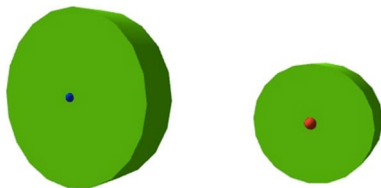




**Fig. 4** A city scenario after disaster where road-blockers are shown in green and police force agents are shown in blue



**Fig. 5** A snapshot of a blocked road



**Fig. 6** Heavy and light obstacles

## 5 Experimental results

The experiments are carried out by running ARGoS simulator (3.0.0-beta50 version) on Intel<sup>®</sup> Core<sup>™</sup> i7 2600 CPU @ 3.40GHz×8 Processor, 8-GB of RAM and Ubuntu operating

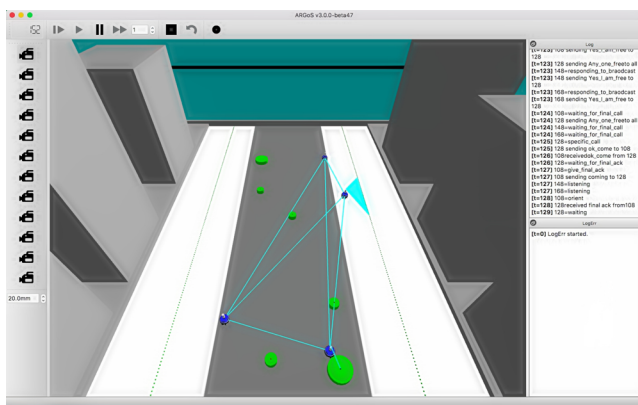
system.. Several experiments are carried out by varying the number of robots and number of tasks (obstacles). Experiments with two scenarios are carried out; (1) Road blocked with mixed type of obstacles, i.e., light and heavy, and (2) Road blocked with heavy obstacles only.

### 5.1 Scenario 1: road blocked with light and heavy obstacles

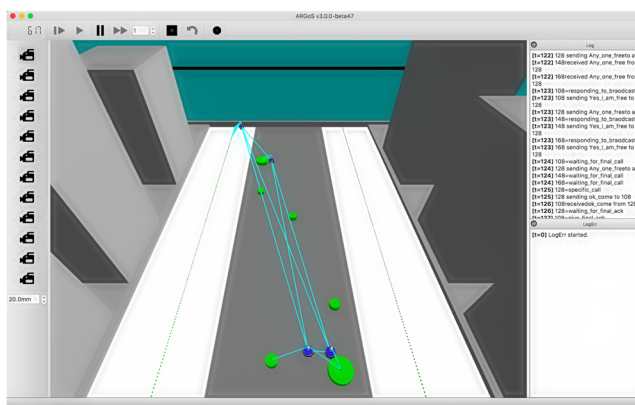
In scenario 1, it is assumed that the road is blocked with a mixed type of obstacles, i.e., light and heavy. For this scenario, several experiments are carried out by varying the number of police force agents (robots) (i.e., 4, 6, 8, and 10) and set of obstacles (8 light and 2 heavy, 8 light and 4 heavy, and 8 light and 6 heavy). The number of light and heavy obstacles for each robot size is kept the same for each robot size. Further, for each combination of robot size and obstacles, the algorithm is executed for 10 independent runs. The overall process of obstacle detection and clearance is shown in Fig. 7. Whenever a police force agent finds a heavy obstacles (see Fig. 7a), it start the coalition formation process (as per the Algorithms 2 and 3). The agents, selected in the coalition, approach to the initiator (see Fig. 7b). When a non-initiator reaches the location of the initiator, they synchronize themselves (see Fig. 7c). After synchronization, they jointly pick the heavy obstacle and move it to the side of the road (see Fig. 7d). They drop the obstacle inside the road and come back to the road and start searching for more obstacles if any present in the road (see Fig. 7e). This process continues until the road is cleared. After clearing all the obstacles, the algorithm terminates (see line no. 20–23, Algorithm 1).

The experimental results are obtained and recorded for scenario 1 (given in Tables 2, 3, and 4) by carrying out several iterations of the experiment, by keeping tasks (light and heavy obstacles) fixed and varying the number of robots. Each experiment has been repeated several times to get an average value of time taken for the better accuracy of the results. The number of robots has been varied from 4 to 10 by keeping the number of tasks fixed to 10 (light obstacles = 8 and heavy obstacles = 2), 12 (light obstacles = 8 and heavy obstacles = 4), and 14 (light obstacles = 8 and heavy obstacles = 6). The findings illustrate that the average time taken to complete the tasks decreases with the increased in the number of robots (see Fig. 7). The results reflect the natural phenomenon of the real-world scenario. When the number of robots to clear the road is kept low, the time taken to clear the road is large, but with more robots, the time taken to clear the road decreases. However, after a certain increase in the number of robots, the time reduction rate decreases.

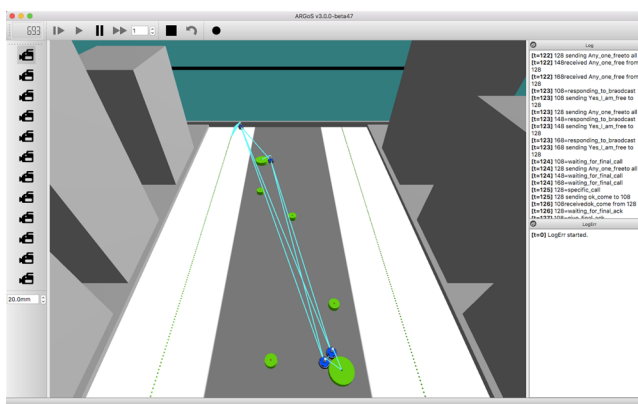
This phenomenon exhibit the mimicry of the real-life disaster scenario where, the rescue operation is slow with



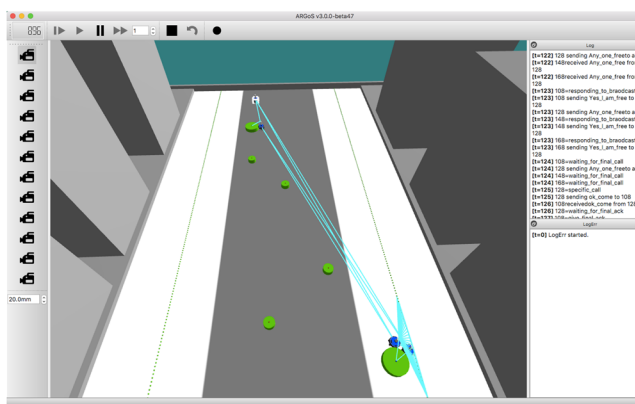
(a) A police force agent (robot) finds a heavy obstacle



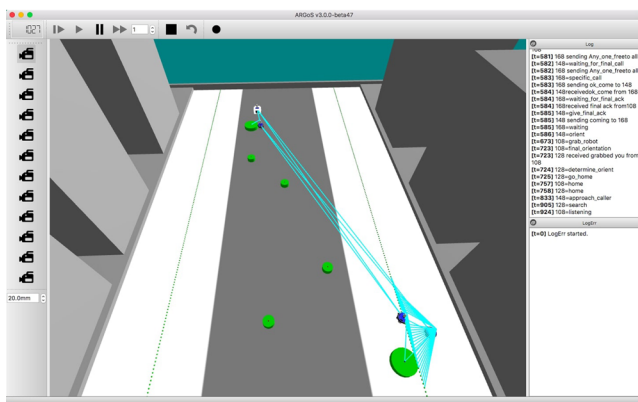
(b) Robot starts coalition formation process



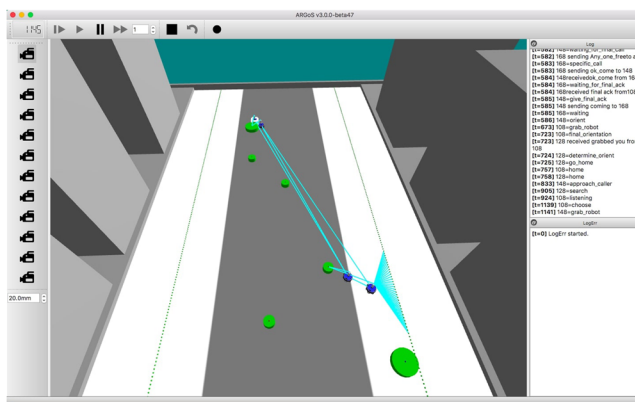
(c) Non-initiator reaches to initiator's location



(d) Robots carry the heavy obstacles to one side of the road



(e) Robots have moved the obstacle



(f) Robots are going back to the road

Fig. 7 Scenario 1: road is blocked with light and heavy obstacles

few rescuers, but it becomes faster with more rescuers. But, after a certain increase in the number of the rescuer, instead of being faster the rescue operation gets slow as the crowd of rescuers, itself creates hindrance in the rescue operation. The obtained results also follow the phenomenon mentioned above, as if the number of robots in any specific area

is increased, then robots themselves become obstacles for other robots. In such a crowded scenario maximum time is spent in obstacle avoidance and robots take more time in traveling from one place to another. The findings illustrate that as the number of robots is increased, the average time is taken to complete the tasks decreases as shown in Fig. 8.

**Table 2** Results: Road blocked with 8 light and 2 heavy obstacles

S. no	#Robots	Time taken (s)			
		Minimum	Maximum	Average	Std.
1	4	353.6	404.7	378.86	24.47
2	6	312.5	344.5	324.04	12.12
3	8	210.2	268.7	238.24	22.25
4	10	183.7	205.1	193.72	09.11

**Table 3** Results: Road blocked with 8 light and 4 heavy obstacles

S. no	#Robots	Time taken (s)			
		Minimum	Maximum	Average	Std.
1	4	412.3	650.0	540.20	84.53
2	6	410.5	567.5	497.08	70.88
3	8	309.7	368.2	340.84	23.03
4	10	277.4	367.2	317.92	32.35

**Table 4** Results: Road blocked with 8 light and 6 heavy obstacles

S. no	#Robots	Time taken (s)			
		Minimum	Maximum	Average	Std.
1	4	681.4	710.0	694.44	11.14
2	6	521.2	642.0	585.67	43.33
3	8	421.0	496.0	446.30	28.63
4	10	354.7	407.1	380.96	18.58

We can conclude from the obtained results, given in Tables 2, 3 and 4, that if the number of robots is kept fixed, then the average time (taken in cleaning the road) increases

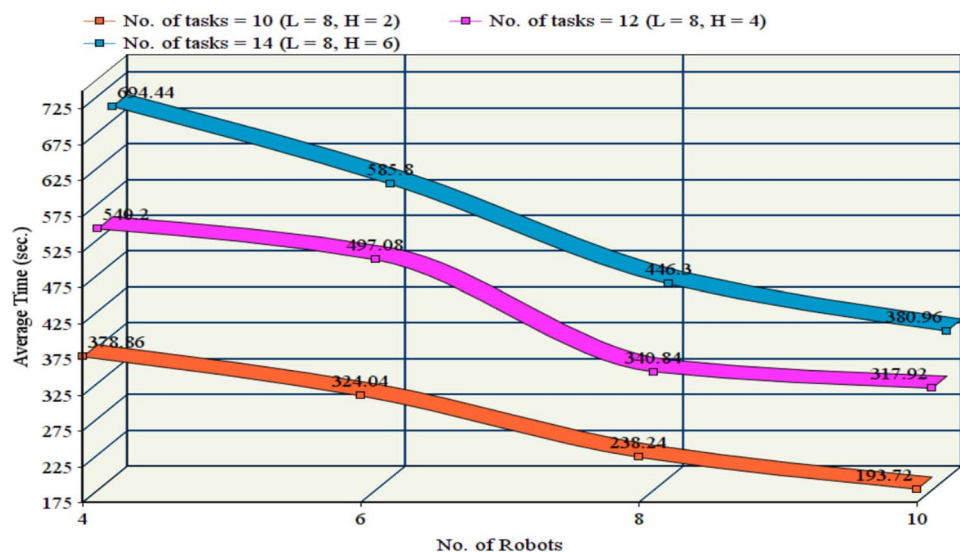
linearly with the increase in the number of the obstacles present on the road.

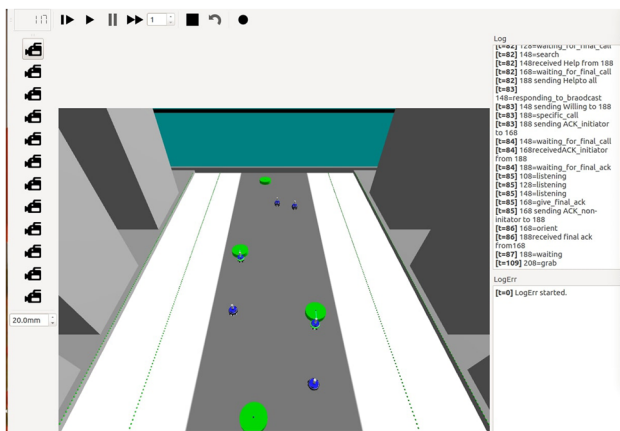
## 5.2 Scenario 2: road blocked with heavy obstacles

In this scenario, it is assumed that the road is blocked only with heavy obstacles that can be moved/shifted by a coalition of robots. The overall process for scenario 2 as shown in Fig. 9. Whenever the robots, assigned for road clearance in a particular region of the city is triggered, they start searching for the obstacles. As soon as a robot finds an obstacle it determines the number of robots required to shift the obstacle (see in Fig. 9a). The initiator starts the coalition formation process and if the coalition is formed successfully, it waits till all the coalition robots reach to obstacle's location (see Fig. 9b). The initiator also determines the closest location to the side of the road. When all robots of the coalition reach to initiator's location, they synchronize themselves and move with heavy obstacles towards the target location decided by the initiator (see Fig. 9c). After reach to the side of the road, coalition robots drop the obstacle and again come back to the road and again start searching for obstacles (see Fig. 9d). In all experiments, the most important aspects of the multi-robot system, i.e., collision and obstacle avoidance are also taken into consideration.

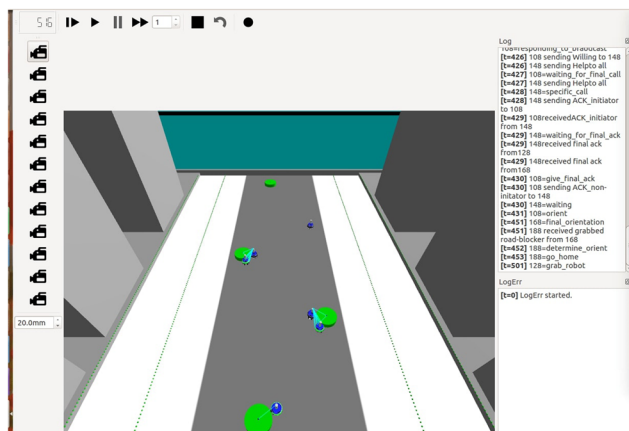
Figure 9 also exhibits parallel multiple tasks execution. If multiple obstacles are detected by different robots then, they all start the coalition formation process simultaneously and if they succeeded in their coalition formation process, the individual coalition's robots move the corresponding obstacles simultaneously. In this way, the proposed approach also handles the parallel task execution.

**Fig. 8** Performance based on varying the number of robots and tasks for scenario 1

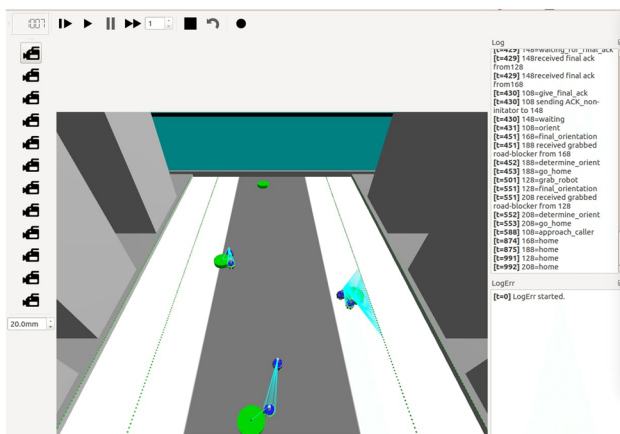




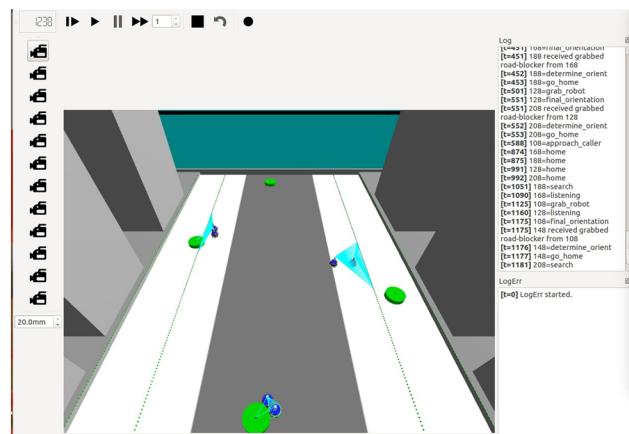
(a) Police force agent (robot) finds obstacle



(b) Robot decides shortest path



(c) Robots drop the obstacle in safe zone



(d) Robots return back to search other obstacles

Fig. 9 Scenario 2: road is blocked with heavy obstacles only

Performance based on varying the number of robots and tasks for scenario 2 is given in Fig. 10.

## 6 Discussion

The experimental results were performed on different scenarios. Scenario 1: road blocked with mixed types of obstacles, i.e., light and heavy. Scenario 2: road blocked with heavy obstacles. The time taken to clear a road blocked with mixed type of obstacles is lesser than the time taken to clear the road, blocked only with heavy of obstacles. This is because, when tasks are of mixed types, some of the obstacles are light and hence no coalition formation is required, but to move heavy obstacles, coalition formation is required. To clear a heavy obstacle some amount of time is required for coalition formation, and robots take some time to reach the initiator's location.

The results obtained for scenarios 1 and 2 are compared for the case where the road is blocked with 8 light and 6

heavy obstacles (Table 4) to the case where the road is blocked with 6 heavy obstacles (Table 6). The minimum, maximum, average, and standard deviation of the execution time of our approach for different number of police force agents (robots), i.e., 4, 6, 8, and 10, are given in Fig. 11.

The average time taken to clear the road in both the scenarios are almost similar although in scenario 1, more obstacles are present on the road. The reason is that when there are only heavy obstacles on the road, there is a high chance of simultaneous detection and start of the coalition formation process by multiple robots. Because of the simultaneous start of the coalition formation process, only a few of them can succeed in the coalition formation process due to the unavailability of free robots in the environment. Hence, for scenario 2 almost the same time is taken as for scenario 1.

In this paper, we develop a prototype model for road-clearance scenario in a realistic multi-robot simulation environment (ARGoS) and apply the proposed algorithm. The results given in Tables 2, 3, 4, 5, 6 and 7 are satisfactory. The proposed framework is applicable to other domains

**Table 5** Results: Road blocked with 4 heavy obstacles

S. no.	#Robots	Time taken (s)			
		Minimum	Maximum	Average	Std.
1	4	465.0	642.2	571.88	66.34
2	6	432.0	580.0	502.44	62.01
3	8	382.2	512.6	436.42	60.12
4	10	197.8	308.0	249.26	48.46

**Table 6** Results: Road blocked with 6 heavy obstacles

S. no.	#Robots	Time taken (s)			
		Minimum	Maximum	Average	Std.
1	4	643.5	843.4	772.26	75.88
2	6	425.3	601.1	509.90	68.06
3	8	386.3	526.3	448.62	64.41
4	10	302.5	460.9	354.78	58.90

**Table 7** Results: Road blocked with 8 heavy obstacles

S. no.	#Robots	Time taken (s)			
		Minimum	Maximum	Average	Std.
1	4	786.0	962.2	863.62	77.72
2	6	498.0	662.3	570.92	76.49
3	8	456.0	432.5	546.20	72.73
4	10	383.7	555.6	460.20	66.98

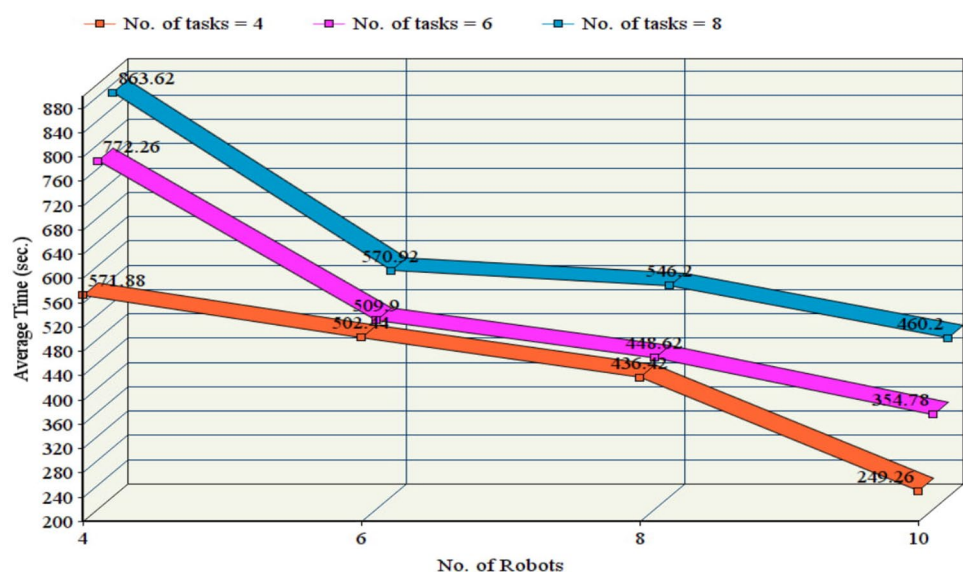
(e.g., landmine detection and detonation, cooperative object transportation) where, coalition formation has to be done at run-time and multi-robot coordination is required for task execution.

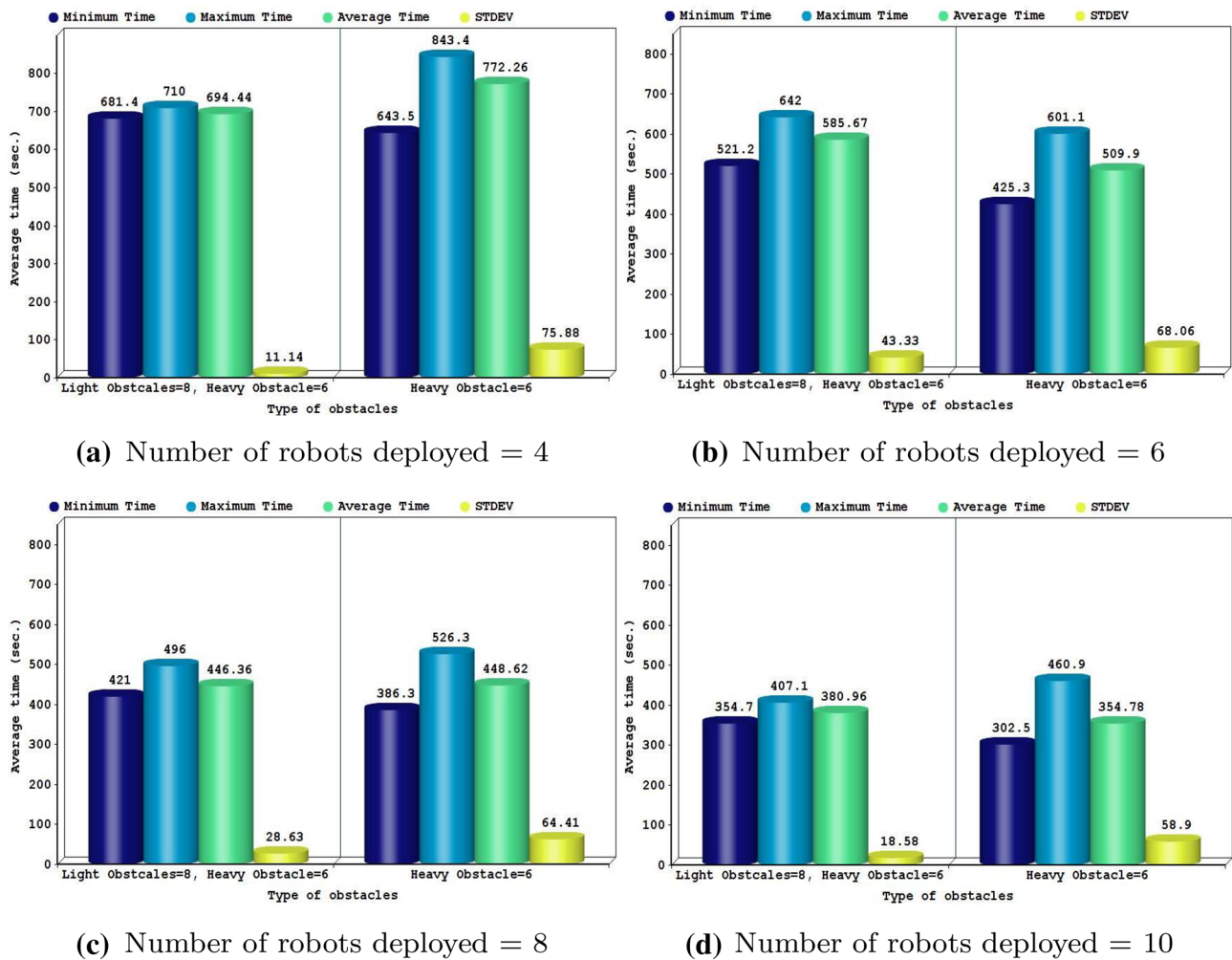
## 7 Conclusion

In this paper, a distributed coordination approach is proposed that ensures road clearance when the road is blocked with heavy obstacles. The salient features of the approach are: all the police force agents perform search and road clearance activities collaboratively; the actions of the agents are self-controlled; agents perform obstacle avoidance and self-alignment. The proposed approach also takes care of multiple task execution simultaneously, i.e., if multiple agents detect different obstacles at the same time, the coalition formation process for each obstacle can be started.

The role of the agents is not fixed a priori, an agent who was an initiator for a task may be a non-initiator for another task. The communicating automaton for the agent's behavior is presented which helps in designing and implementing the proposed approach. The approach is robust, since in the event of a failure of an agent during execution, the initiator can replace it with other agents by using Algorithms 2 and 3. The experimental results performed on different scenarios exhibit satisfactory and desired performance of the proposed approach. As part of our future work, we would like to extend our framework for other domains like landmine detection and detonation.

**Fig. 10** Performance based on varying the number of robots and tasks for scenario 2





**Fig. 11** Comparison of min., max., avg., and standard deviation of execution time for scenario 1 (8 light and 6 heavy obstacles) and scenario 2 (6 heavy obstacles): with varying the number of police force agents (robots)

**Acknowledgements** The third author was in part supported by a research grant from Google.

**Compliance with ethical standards**

**Conflict of interests** The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**References**

Abdallah, S., Lesser, V.: Organization-based cooperative coalition formation. In: International Conference on Intelligent Agent Technology, pp. 162–168, (2004)

Bérard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., Schnoebelen, P.: Systems and Software Verification: Model-Checking Techniques and Tools. Springer Science & Business Media, Berlin (2013)

Burke, J.L., Murphy, R.R., Covert, M.D., Riddle, D.L.: Moonlight in Miami: field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise. *Hum. Comput. Interac.* **19**(1–2), 85–116 (2004)

Casper, J., Murphy, R.R.: Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **33**(3), 367–385 (2003)

Coviello, L., Franceschetti, M.: Distributed team formation in multi-agent systems: stability and approximation. In 51s. In: Annual Conference on Decision and Control (CDC), pp. 2755–2760, (2012)

Dukeman, A., Adams, J.A.: Hybrid mission planning with coalition formation. *Auton. Agents Multi-Agent Syst.* **31**(6), 1424–1466 (2017)

Gerkey, B.P., Mataric, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robot. Res.* **23**(9), 939–954 (2004)

Gerkey, B.P., Mataric, M.J.: Sold!: auction methods for multi-robot coordination. *IEEE Trans. Robot. Autom.* **18**(5), 758–768 (2002)

- Gunn, T., Anderson, J.: Dynamic heterogeneous team formation for robotic urban search and rescue. *J. Comput. Syst. Sci.* **81**(3), 553–567 (2015)
- Kong, Y., Zhang, M., Ye, D.: An auction-based approach for group task allocation in an open network environment. *Comput. J.* **59**(3), 403–422 (2015)
- Liu, Y., Nejat, G.: Robotic urban search and rescue: a survey from the control perspective. *J. Intell. Robot. Syst.* **72**(2), 147–165 (2013)
- Lynch, N.A.: *Distributed Algorithms*. Elsevier, Amsterdam (1996)
- Pinciroli, C., Trianni, V., O’Grady, R., Pini, G., Brutschy, A., Brambilla, M., Birattari, M.: ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intell.* **6**(4), 271–295 (2012)
- Kitano, H.: RoboCup rescue: a grand challenge for multi-agent systems. In: *ICCCN*, pp. 1–8, (2000)
- Saad, E., Hindriks, K.V., Neerinx, M.A.: Ontology design for task allocation and management in urban search and rescue missions. *ICAART* **2**, 622–629 (2018)
- Sato, N., Matsuno, F., Yamasaki, T., Kamegawa, T., Shiroma, N., Igarashi, H.: Cooperative task execution by a multiple robot team and its operators in search and rescue operations. *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, **2**, 1083–1088 (2004)
- Sedaghat, M.N., Nejad, L.P., Iravanian, S., Rafiee, E.: Task allocation for the police force agents in robocuprescue simulation. In: *Robot Soccer World Cup*, pp. 656–664, (2005)
- Tadokoro, S., Kitano, H., Takahashi, T., Noda, I., Matsubara, H., Shinjoh, A., Hatayama, M.: The robocup-rescue project: a robotic approach to the disaster mitigation problem. *IEEE Int. Conf. Robot. Autom. (ICRA)*, **4**, 4089–4094 (2000)
- Tošić, P.T., Agha, G.A.: Maximal clique based distributed coalition formation for task allocation in large-scale multi-agent systems. In *Int. Workshop on Massively Multiagent Systems*, pp. 104–120, (2004)
- Uhrmacher, A. M., Weyns, D.: RoboCup Rescue: Challenges and Lessons Learned Tomoichi Takahashi. In: *Multi-Agent Systems: Simulation and Applications*, pp. 437–464, (2018)
- Vig, L., Adams, J.A.: Multi-robot coalition formation. *IEEE Trans. Robot.* **22**(4), 637–649 (2006)
- Visser, A., Ito, N., Kleiner, A.: Robocup rescue simulation innovation strategy. In: *Robot Soccer World Cup*, pp. 661–672, (2014)
- Xie, B., Chen, S., Chen, J., Shen, L.: A mutual-selecting market-based mechanism for dynamic coalition formation. *Int. J. Adv. Robot. Syst.* **15**(1), 1–10 (2018)
- Yan, Z., Jouandeau, N., Cherif, A.A.: A survey and analysis of multi-robot coordination. *Int. J. Adv. Robot. Syst.* **10**(12), 399–416 (2013)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Formal verification.



**Amar Nath** received his B. Tech degree from Bundelkhand Institute of Technology, Jhansi, UPTU, Lucknow in CSE, 2006. He received his Master degree from National Institute of Technology Rourkela, Odisha, India in CSE, 2014. Currently, he is doing his Ph.D. From Indian Institute of Technology Roorkee, India in the Department of CSE. Authors have won best paper awards in a couple of international conferences. His area of interest includes SOA, AI, MAS, MRS, Distributed system and

**A. R. Arun** received his IDD (Integrated dual degree, i.e., B. Tech. and M. Tech.) degree from Indian Institute of Technology Roorkee, India in CSE, 2017.



**Rajdeep Niyogi** is an Associate Professor in the Department of Computer Science and Engineering, Indian Institute of Technology (IIT) Roorkee, India. His research interests include automated planning, multi-agent systems, distributed systems, and applications of logic and automata theory. He did Ph.D. in Computer Science and Engineering from Indian Institute of Technology (IIT) Kharagpur, India, in 2004. He is a member of ACM and ERCIM.