RoboCup Rescue Simulation League Agent IranOpen 2017 Competition Rules and Setup

Farshid Faraji, Luis G. Nardin, Ali Modaresi, Dina Helal and Nobuhiro Ito

Abstract

This document describes the rules and the ranking formula for the IranOpen 2017 RoboCup Rescue Agent Simulation League Competitions.

1 Introduction

The Robocup Rescue Simulation Platform is a comprehensive simulation environment for research in disaster response management. During rescue operations after a disaster, cooperation is a must. In general the problem is not solvable by a single agent, and a heterogeneous team that dynamically combines individual capabilities in order to solve the task is needed. This requirement is due to the structural diversity of disaster areas, variety of evidence the sensors can perceive and to the necessity of quickly and reliably examining large regions. Yet, the performance of a joint rescue team depends on assembling the right mixture of capabilities and has to be designed as a whole. The goal of this league is to take this technological and scientific challenge and extend current rescue robot platforms with planning, learning, and information exchange capabilities needed to coordinate their efforts and to accomplish the rescue mission as a team. Robocup Rescue Simulation is an education and research project intended to promote the development of robotic agents for search and rescue.

Efficacious response plans are essential for reducing the negative impacts of natural disasters. These plans are usually elaborated by policy-makers based on previous experiences and with limited evaluation of efficacy. Assess response plans, however, is a challenging task as one cannot reproduce the real conditions of a natural disaster and its effects in a real environment just for assessment purposes. In this case, simulation plays a key role in assisting to assess response plans as it enables the reproduction of the real conditions of a disaster in an artificial environment.

2 RoboCup Rescue Simulation Agent Competition

The competition involves primarily evaluating the performance of agent teams developed using exclusively ADK Framework (https://github.com/RCRS-ADF/RCRS-ADF) on different maps of the RoboCup Rescue Agent Simulation (RCRS) platform (http://sourceforge.net/projects/roborescue).Specifically, it involves evaluating coordination algorithms enabling teams of Ambulances, Police Forces, and Fire Brigades to rescue civilians and extinguish fires in cities where an earthquake has just occurred. Each team is evaluated on a set of scenarios that represent different disaster situations (e.g., civilians and fires, major fire in one corner of a city, blocked roads to refuges, damaged platoon agents, fire maps, civilian maps where only fires need to be extinguished or civilians need be saved respectively).

This competition is divided in three rounds (i.e., preliminary, semi-final, and final rounds). At each round, the teams provide their code that will be executed on a set of scenarios and a score calculated to each scenario.

A scenario is composed of a map, a set of initial rescue agent/civilian positions, and a set of configuration options for each of the simulator components. Each map is constrained to a maximum of 10,000 roads and 10,000 buildings. The buildings and roads entrances are supposed to be fully connected. A validation tool will be used to check the full connectivity of roads and building entrances in each map. However, teams do not have the right to complain in case roads or building entrances are not fully connected if evidenced that this was not detected by the validation tool.

The RoboCup Rescue Agent Simulation Technical Committee is responsible to elaborate a set of scenarios for the competition and all teams code are executed on the same set of scenarios.

2.1 Implementation

All participating teams MUST implement their code using the ADK framework.

The teams should implement their own code to replace or extend the following ADF classes:

adf.component.algorithm.Clustering

adf.component.algorithm.PathPlanning

adf.component.complex.TargetSelector¡rescuecore2.standard.entities.Building¿

adf.component.complex.TargetSelector¡rescuecore2.standard.entities.Human¿

adf.component.complex.TargetSelector¡rescuecore2.standard.entities.Blockade¿

Changes in other classes will not be allowed, especially the Tactics classes.

2.2 Rules

(a) **Remote participation:** Remote participation is allowed to international teams.

(b) **Rounds:** The competition is structured into three rounds: one preliminary round, one semifinal, and one final round. The preliminary round will be executed in two consecutive days (first and second days of competition), while the semifinal and final rounds are executed in one day each (third and fourth days of competition).

(c) **Sessions:** Each round consists of several sessions. A session is comprised of a set of simulations in different scenarios. A member of the Technical Committee will chair each session. The session chair is responsible for executing the simulations, collecting scores and logs, and handling any issues that arise during the session.

(d) **Code submission:** All teams must submit the team' source code (binary code will not be accepted) and adequate compile scripts before the start of each round. The number and time of submissions and specific requirements will be explained in the brief presentation at the competition setup time at the venue. The Technical Committee has the authority to change the time of submissions and to review all submitted teams' source code.

(e) **Scenarios:** The scenarios will be provided by the Technical Committee. Teams shall NOT know the disaster situation (map, random seeds, simulator configuration, parameter values, and phases of execution) of the session before it starts. All the conditions for a particular disaster situation shall be identical for all the teams.

(f) **Agents:** Teams shall implement all types of agents. It is the responsibility of the team to ensure that its code connects the correct number of agents to the server.

(g) **Shared memory:** Agents cannot use any form of shared memory, including static memory accessible to all agents, direct function calls between agents, or writing files for use by other agents during the scenario simulation. The exception is the Pre-Computation phase when agents are allowed to write files (see item (h) for details). The Technical Committee may execute each agent of the team in a different virtual/physical machine if the team is suspected of violating this rule.

(h) **Phases:** The scenario simulation may be performed in two phases of execution of the team's code: the Pre-Computation phase (item (i)) and the Simulation phase (item (j)). The Pre-Computation phase, however, is not mandatory for all scenarios and is assumed a configuration parameter of the scenario. Thus, the execution of the Pre-Computation phase will be defined as a configuration parameter of the scenario (see item (e)).

(i) **Pre-Computation phase:** The Pre-Computation phase allows an agent of each type to pre-process map-specific and scenario-specific data and store them into a file for using during the Simulation phase. Only one agent of each type can connect to the server and execute the Pre-Computation algorithm. This phase is limited to 2 minutes and after the time is elapsed the server will be terminated. Pre-Computation is allowed under the following conditions:

  I. The data must be generated by a computer program with no human interaction or intervention.

  II. Data for all maps must be generated by a single computer program.

  III. The computer program should work for any new map.

  IV. Agent must choose the file to store the pre-computing data.

  V. Agents must be able to work if no Pre-Computation data is present for the map.

  VI. The source of the Pre-Computation program must be released after the competition.

(j) **Simulation phase:** The Simulation phase corresponds to the actual team's simulation in the competition scenario. The team must connect in at most 3 minutes all agents to the kernel in order to perform the scenario simulation. The scenario simulation will begin no later than 3 minutes after the first agent begins its handshake with the kernel. All file permissions, except read permission for previously written files, will be removed.

(k) **Valid map:** The Technical Committee is entitled to define whether a map results is valid or invalid in a session. The decision is based on the results of the map and it may be decided that a map is invalid when all the teams score very close in the map.

(l) **Valid games:** Teams will NOT be entitled to rerun their agent team in most circumstances. It is expected that teams write their agents so that they work correctly. In extreme circumstances teams may have the right of a single rerun. Circumstances that may result in a rerun are:

I. A power failure.

II. Accidental or deliberate termination of a kernel, simulator or agent process.

III. Java Virtual machine crash.

In the case of rerunning, the latest score is used as the official score of the team on that map.

Examples of events that will NOT result in a rerun are:

I. A simulator crash.

II. Agents failing to fully connect before the simulation starts.

III. Agents crashing or failing to act during the run.

IV. Observing apparently incorrect behavior by a simulator or the viewer.

Teams that wish to request a rerun must do so in writing. The request must include the team name, the scenario name, a description of the problem, and the reasons why the team feels a rerun is appropriate. The request must also state whether the request is for a rerun of just that team or for a full session rerun. Only one Java Virtual Machine crash rerun request is accepted in each session.

(m) **Bugs:** It is the responsibility of each team to ensure that its code works correctly with the provided simulators. Although the Technical Committee makes every effort to provide a reliable simulation environment, the Technical Committee has no responsibility for any kind of software failure at competition time. Simulator bugs are not sufficient grounds to request a rerun.

(n) **Committee decisions:** If a problem arises during a session then teams may ask for the session chair to resolve the problem. The session chair may make a decision on the spot or may refer it to the committee. Chair decisions are final, but if a team strongly disagrees then they may submit a written appeal to the committee. In order to allow the competition to continue, appeals will not be heard during a round, but will be discussed by the committee at the end of each day. The Technical Committee will make final decision at any condition.

(o) **Comments from teams: Only the team leader** of participating teams can comment and make suggestions to the Technical Committee about the running of the competition. If these comments or suggestions are deemed derogatory or abusive then the matter will be referred to the RoboCup Trustees and may result in penalties for the team concerned. Penalties may include points reduction or, in the worst case, disqualification. Expect team leaders, other team members are not entitled to comment or make suggestions to the Technical Committee.

(p) **Bugs exploitation:** A team that knowingly uses bugs in the simulation package to gain advantage will be disqualified from the competition. Disqualifications will be made only after consultation with the RoboCup Trustees.

(q) **Dispute resolution:** If there is an ambiguity in the rule and any unexpected situation happens, a temporary committee composed of the Technical, Organization and Executive committees members and the local chair have the power to take a decision regarding the issue. The temporary committee decision has the same effect as a rule.

(r) **Open source policy:** Source code files must be released open-source immediately after the end of the competition to guarantee fair play and to encourage community activity after competition. Log files and related parameter files will be open access.

2.3 Ranking

Each round is composed of several sessions (S), and at each session the participating teams receive an identification ranging from t_1 to t_n, where n represents the number of teams participating on that session. Each session is comprised of a set of scenarios (M), and each scenario also receives an identification ranging from m_1 to m_p, where p represents the number of scenarios in that session. A score SC_{ji}^{k} is assigned to each team $i \in T$

($T = \{t_1, \ldots, t_n$) at each session $k \in S$ ($S = \{s_1, \ldots, s_n$) for each scenario $j \in M$ ($M = \{m_1, \ldots, m_p\}$).

For each session k and scenario j, the Selective Minimum (SM_j^k) is calculated as

$$SM_j^k = max\left(SC_{ji}^k\right) - \left(\left(max\left(SC_{ji}^k\right) - mean\left(SC_{ji}^k\right)\right) \times 2\right), \tag{1}$$

and the Maximum Score (MS_j^k) is calculated as

$$MS_j^k = n \times SDC, \tag{2}$$

where n is the number of teams on session k, and SDC is the coefficient indicating the step between points among teams (we will use $SDC = 2$ in RoboCup Rescue 2016 competition).

The maximum value of each step is calculated as

$$MSS_{j/step \in \{1,\ldots,MS_j^k\}}^k = \frac{\left(\left(max\left(SC_j^k\right) - SM_j^k\right)\right)}{\left(MS_j^k \times \left(MS_j^k - step\right)\right)} \tag{3}$$

To each team is assigned the step value, whose $MSS_{j/step}^k$ value is lower than the team' score, but the $MSS_{j/step+1}^k$ value is greater than the team' score.

$$TP_{ji}^k = step \therefore MSS_{j/step}^k < SC_{ji}^k < MSS_{j/step+1}^k \tag{4}$$

The final team points for each scenario and team are calculated as

$$FTP_i^k = \sum_{j=m_1}^{m_p} TP_{ji^k} \tag{5}$$

The final team points is then used to generate a ranking of all the teams for that session. The team with the highest mean points is ranked as first, the second highest as second, and so on.

2.4 Hardware & Software Configuration

In the competition venue, there will be several clusters with 4 computers each. One PC per cluster will be reserved for the simulator components; the remaining three will be available to run the agent teams. The hardware and software configurations are

> *Hardware:* Intel i7 CPU processor 3.0GHz or higher, and 8.0 GB or higher RAM

Operating Systems: Linux Ubuntu 12.04 LTS (64bit) or higher with Oracle Java 1.8 run-time or higher.

Simulation Packages: The simulation packages used in the 2016 RoboCup Rescue Simulation League competition is available at `https://sourceforge.net/projects/roborescue/files/roborescue/v1.2/`.

Parameters: The parameters and their possible values are the same as those in the 2014 Rules, which are listed in Section 8 of the document RoboCup 2014—RoboCup Rescue Simulation League Agent Competition (`http://roborescue.sourceforge.net/web/2014/rules2014.pdf`).

ADF: The ADK framework is available at `https://github.com/RCRS-ADF/RCRS-ADF`.

ADF documentation: The documentation of the ADK Framework is available at `https://github.com/RCRS-ADF/RCRS-ADF`.

3 Presentation

The presentation aims to share the knowledge of the teams and improve the academic research aspects of the league. Each team will have 20 minutes to present their implementation and another 10 minutes for questions and answers. The presentation will be evaluated by committee members and teams. Best presentation award will be awarded.