Obligatory assignment 1 in TEK5010 Multiagent systems 2023

Report delivery date October 11 by e-mail to hjmoen@its.uio.no.

The report should contain answers to all the questions, include discussions on simulation results in the form of graphs, tables or figures, and an appendix with the simulation program code. The report should be delivered as a pdf file. You can use any programming language of your choosing for this oblig.

Search and task allocation in multiagent systems: Swarm intelligence

Background:

In the obligatory assignments this year we are going to explore Search, Task Allocation and Completion (STAC) problems in multiagent systems (MAS). STAC problems are considered a general class of problems in MAS and many real-world problems could be formulated as a STAC problem. See Ijspeert et al. for an example of STAC experimenting in swarm robotics [1] and Minos-Stensrud et al. for a generalized analysis of STAC problems [2].

In the first oblig we are going to study STAC in relation to reactive agents, i.e. swarm intelligence, and in the second oblig we are going to employ interactive agents, i.e. game theory, for solving the same STAC problem. The goal is to analyze how these two MAS concepts differ and to understand when it is appropriate to employ the different algorithms under varying STAC conditions. The main focus will be to analyze how the agents' ability to share information affects system performance in STAC problems.

Definition of the search and task allocation problem:

The *search area A* is a bounded square spanned by the two points (0, 0) and (1000, 1000).

The *tasks T* are randomly distributed over the search area. As soon as a task is completed a new task is spawned at a random position in the search area. The tasks have a task capacity *Tc* indicating how many agents that are required to solve a task, e.g. *Tc*=3 means that 3 agents are required to solve the task. The task is automatically completed if *Tc* agents are within the task radius *Tr*.

The *agents R* move randomly around the search area at a speed *Rv*. When an agent is inside the task radius *Tr* of a task, the agent will stop and wait for other agents to complete the task. The agent could also call for help by engaging in communication with nearby agents. The communication distance *Rd* determines the information sharing process between agents, e.g. *Rd*=250 means that any agent that is within distance of less than 250 from the agent will hear the communication signal. The information communicated and corresponding response will depend on the STAC condition and choice of MAS algorithm employed.

Questions:

a) Could you simulate one agent *R*=1 moving around at speed *Rv*=25 per iteration solving one task with a capacity of *Tc*=1 and task radius *Tr*=50? What would be a good model for moving the agents randomly around the search area? Could you plot how many tasks the agent solve per time when a new task is spawned every time a task is completed? Is this a good measure for assessing performance in the STAC problem?

b) Could you plot the performance as a function of added agents, say for *R*=3, 5, 10, 20 and 30?

c) Now, repeat the simulations using task capacity *Tc*=3 and plot the performance as a function of number of agents.

d) What happens if we increase the number of tasks to be completed, say *T*=2, 10 and 20?
Discuss how many iterations you have to simulate to get into steady-state. And also, how many simulations do you need to repeat to get a good statistical estimation of the average performance? This case of no communication among the agents could be considered the 'random' benchmark for the STAC problem.

e) 'Call-out' is a basic swarm intelligence method for signaling nearby agents of need for assistance [2]. An agent that detects a task emits a simple unmodulated signal that is picked up by other agents in close proximity. The nearby agents would then reactively respond to the signal by moving in the signal direction until task is reached (or helping agent have reached the edge of the search area).
Could you simulate and plot the system performance as a function of communication distance *Rd*=0, 100, 200, 300, 400, 600, 1000 and 1400 when using 'call-out'? Assume *T*=2, *Tr*=50, *Tc*=3 and *R*=30. Comment on your findings.

f) 'Call-off' is an altered 'call-out' by including a call-off signal to all nearby agents when task is complete, whereby releasing all committed agents into search modus again. Simulate and plot the same setting as in e) but with 'call-off' implemented. Comment on your findings again.

References

[1] Auke Jan Ijspeert, Alcherio Martinoli, Aude Billard and Luca Maria Gambardella, "Collaboration through the exploitation of local interactions in autonomous collective robotics: the stick pulling experiment", Autonomous Robots 11, 149–171, 2001. https://doi.org/10.1023/A:1011227210047 or http://people.idsia.ch/~luca/AR2001.pdf

[2] M. Minos-Stensrud, H. Jonas Fossum Moen and J. Dyre Bjerknes, "Information sharing in multi-agent search and task allocation problems," 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 2021, pp. 1-7, doi: 10.1109/SSCI50451.2021.9660121.