SET10111 Multi-Agent Systems
Coursework

Learning Outcomes Covered:	LO3: Evaluate the effectiveness of multi-agent technologies when applied to a specific problem. LO4: Specify, implement and evaluate a multi-agent system in response to a particular problem.
Assessment Type:	Report
Overall module assessment	60% Coursework, 40% Exam
For this assessment:	60%
Assessment Limits:	Four pages using the IEEE conference template on the module Moodle page
Submission Date:	Friday, 01 December 2023
Submission Time:	15:00
Submission Method:	Via Moodle
Turnitin:	Unlimited attempts
Module leader:	Simon Powers
Tutor with Direct Responsibility:	Simon Powers

- You are advised to keep a copy of your assessment solutions.
- Please note regulation Section B5.3.b regards component weighting.
- Late submissions will be penalised following the University guidelines as follows: Up to 5 working days the mark will be capped at 40%, and 0% if 5 working days.
- Extensions to the submission date may only be given by the Module Leader for exceptional circumstances. – by submitting appropriate request form from Extenuating circumstances.
- Feedback on submissions will normally be provided within three working weeks from the submission date.

The University rules on Academic Integrity will apply to all submissions. The student academic integrity regulations contain a detailed definition of academic integrity breaches which includes use of commissioned material; knowingly permitting another student to copy all or part of his/her own work

You must not share your work with other students - this includes posting any of your work in any repository that is accessible to others (such as GitHub) and applies also after you have completed the course. You must not ask coursework-related questions in online for a (such as Stackoverflow) and you must not use ChatGPT or other generative AI tools – this would constitute academic misconduct as it would be commissioning material.

By submitting the report, you are confirming that:

- It is your own work except where explicit reference is made to the contribution of others.
- It has not been submitted for any module or programme degree at Edinburgh Napier University or any other institution.
- It has not been made with the assistance of Artificial Intelligence (AI) tools.

SET10111 Multi-Agent Systems Coursework: Design, implement and evaluate a multi-agent auction system for resource allocation in edge computing

Resource allocation poses a frequent challenge in multi-agent systems. Imagine assigning charging spots to self-driving cars, distributing web search engine ad spaces to advertisers, or allocating computing power to agents tackling intensive computational tasks. The core issue revolves around deciding which agents should receive access to which resources. Auctions commonly provide a solution, employing bidding processes to disperse resources among various agents. This method ensures resources are distributed both efficiently and in a manner that considers the diverse preferences and capabilities of each agent.

In this coursework, you will create and evaluate an auction system to distribute edge computing resources to mobile devices. This will give you experience of designing and developing a multi-agent system for a cutting-edge commercial application area. You will build your auction system using ActressMAS and present your design and evaluation in a paper of **no more than four pages** (including references, figures and tables) **using the IEEE conference template** supplied on Moodle.

Background to the problem

Edge Computing refers to computational processing that happens close to the data source or "edge" of the network, rather than in a centralised data centre location. In simpler terms, it involves processing data near the location where it is needed or generated, rather than sending it to a central cloud. Edge computing is especially significant in environments where low latency is crucial, and data needs to be processed in real-time or near real-time, such as in autonomous vehicles, image recognition from cameras, wearable health devices, and real-time voice translation [1]. These devices lack sufficient processing power themselves, and often need to offload computation to more powerful machines. However, cloud computing services, such as AWS and Azure, may not be suitable because of the latency involved in sending vast amounts of a data to a server that is far away. Edge computing solves this problem by locating computing resources on edge servers close to the devices that need them, reducing latency, bandwidth use and data transmission costs. The market for edge computing infrastructure was valued at 11.24 billion US dollars in 2022, and is expected to grow at 38% per year for the rest of this decade [2].

But despite its advantages, edge computing raises significant resource allocation problems. This is because edge servers typically have limited resources (CPU cycles, storage, network resources), which many end user devices are in competition for. Current research is using multi-agent systems to address this. The basic idea is that an edge server is represented by an agent that offers its services for auction, while end user devices are each represented by an agent that bids on these resources. Your task involves developing an auction system to handle the allocation of edge server resources to end user devices.

Practical 5 and Lecture 6 introduce multi-agent system auctions. For higher marks, your design should be influenced by the literature on auctions in multi-agent systems, especially in auctions for allocating edge computing resources. Reference [3] provides a good starting point.

Detailed specification

Your auction system needs to meet the following requirements.

Agent types

Your system will need to have at least the following types of agents:

- **Device agents**: Your system should implement a set of agents representing devices with tasks that they wish to perform on edge servers. Initially, you can assume that each device agent has a single task that it wishes to perform, and that this task is represented by the amount of data (in Mb) that it needs an edge server to process. A device agent also has a valuation for the task – this is how much it is worth to the agent, in terms of money, to have the task completed by an edge server. You may initially assume that these valuations are randomly generated, but you may choose to change this as part of your evaluation.
- **Edge server agents**: Your system should also have agents that represent edge servers. Each edge server has capacity to process a given amount of data (in Mb). It should sell this capacity to the device agents using an auction protocol. For simplicity, you may wish to assume that the capacity is sold in lots of 10Mb. Each edge server agent also has a cost (measured in pence) that it needs to pay per 10Mb of data that it processes (this represents its energy and network costs). You can initially assume that this cost is randomly generated for each agent.

You may also include additional types of agents if needed for your design (e.g. an Auctioneer agent). Finally, you can assume that if a device agent does not manage to complete its task (i.e. it does not win any auction) then it has to send the task to a centralised cloud to complete, and pays a penalty for doing this (e.g. a cost greater than its valuation).

Auction protocol

You should design an auction protocol sell the 10Mb lots of edge server agents to device agents. You should begin by studying the protocols in Lecture 6, and their implementations provided with ActressMAS and in Practical 5, and think about how you might adapt those to this problem. You can design and implement a system with varying degrees of difficulty:

- **Bare pass** (40-49%): You will use one of the auction protocols from Lecture 6 (English, Dutch, first-price sealed bid or Vickrey second-price sealed bid). You will have a single edge server agent that auctions its lots to multiple device agents. Each device agent could have a single task of 10Mb that it wishes to perform.
- **Good** (50-59%): As above, but you will additionally have multiple edge server agents offering lots for sale.
- **Very good** (60-69%): As for good plus you will consider extensions to the basic scenario to increase realism. For example, you might consider device agents and edge server agents requesting and offering different types of resources, such as CPU, RAM and network bandwidth, which need to be bid on simultaneously. Or you might consider that device agents and edge server agents each have a location, and device agents would be willing to pay less for edge server agents that are further away from them (due to increased latency). But these are just suggestions – you are encouraged

to take the scenario in your choice of direction, provided this makes sense for the problem. You can check with Simon if your ideas are suitable.

- **Excellent** (70%+): As for very good, plus you will research and use an auction protocol from the literature that has not been covered in Lecture 6, and that is suitable for the problem. For example, you may consider a double auction protocol that simultaneously matches offers from edge server agents to bids from device agents [3].

Implementation of your system

Your system must be implemented as a console application using ActressMAS. It should allow the number of device agents and edge server agents to be varied on different runs. For demonstration purposes, it should be able to print to the console the messages sent between agents. It should also be able to output a list of all the sales that take place during the auction, including the agent names of the buyer and seller, and the price paid.

Evaluating your system

You should run **at least two** experiments to demonstrate how effectively your system allows device agents and edge server agents to achieve their goals under different conditions. For example, you may wish to vary the strategy of the agents when buying or selling, or vary the ratio of device agents to edge server agents to affect supply and demand, or vary the distribution of valuations that device agents have…

You should decide and justify what you will measure, e.g. average profit of buyers, average profit of sellers, social welfare…

Because the device agent valuations and edge server agent costs are randomly generated, you should run each experiment at least 10 times and look at statistical performance (e.g. average and standard deviation across the 10 runs). For each experiment, you should explain why you chose that experiment, and give a hypothesis for what you think will happen before you run it. You should then **present the results in figures or tables**, and comment on their implications.

To complete your evaluation, you should also investigate how the number of messages passed between agents in your system scales with the total number of agents (you may analyse this theoretically or experimentally by recording the number of messages), and comment on the implications of this for implementing your system in the real world.

Mark scheme and paper requirements

The maximum mark available is 60. **Your implementation is worth up to 18 marks**, and is marked as stated in the "Auction protocol" section above. The evidence for this will come from your demonstration, and the code that you submit to Moodle. The remainder of the marks come from the following sections in your paper (you may break these down into subsections as appropriate):

Paper section	**Requirements**
Abstract Introduction (5 marks)	The abstract should summarise your design, your experimental results, and their implications. The introduction should introduce the edge computing resource allocation problem in your own words using appropriate references to support your description, focussing on why multi-agent systems are a suitable technology for addressing it.
Design (21 marks) I used double auction, which allows you to have multiple buyers and sellers at the same time.	Introduce your auction protocol**, briefly** explaining how it works, including a description of the agent types and the types of messages they exchange. (3 marks) **Justify** why you have chosen that protocol by considering other possibilities and drawing on academic literature. (6 marks) State precisely how device agents and edge server agents choose how much to bid, or the minimum price to sell for. **Justify** your choice for this. (3 marks) Explain any extensions you have implemented to the base scenario, as described above for "very good" (3 marks) Where you have used a protocol not from the taught material you can receive up to an additional 6 marks, as described above for "excellent".
Evaluation (13 marks)	Present at least two experiments following the instructions in the Evaluation section above. If you are not able to run the experiment, then you can still pick up marks for showing your experimental plan. (6 marks + up to an additional 5 marks if your experiments use the features described in "very good" and "excellent" above).

	Evaluate how the performance of your system scales as the number of agents increases. (2 marks)
Conclusion (3 marks)	Briefly summarise your results and make suggestions for future work.
References	Include a reference list at the end of the document, using the IEEE reference format in the template.

Submission

1. Your paper should be submitted to Moodle as a PDF.
2. Your source code must be submitted to Moodle. If you have used the Visual Studio IDE then your code should be submitted as a zip file of your Visual Studio solution. If you have used another toolchain then please submit a make or cmake build script with your source code.
3. You must demonstrate your system to the teaching team during Week 14 (week beginning 4th December). Arrangements for this will be announced on Moodle. Your work cannot be marked without a demonstration.

References

[1] A. Galanopoulos, T. Salonidis, and G. Iosifidis, ‘Cooperative Edge Computing of Data Analytics for the Internet of Things’, *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 4, pp. 1166–1179, Dec. 2020, doi: 10.1109/TCCN.2020.3019610.

[2] ‘Edge Computing Market Size, Share & Growth Report, 2030’. Accessed: Oct. 10, 2023. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/edge-computing-market

[3] H. Qiu, K. Zhu, N. C. Luong, C. Yi, D. Niyato, and D. I. Kim, ‘Applications of Auction and Mechanism Design in Edge Computing: A Survey’, *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 1034–1058, Jun. 2022, doi: 10.1109/TCCN.2022.3147196.