FIT5226 - Individual Assignment

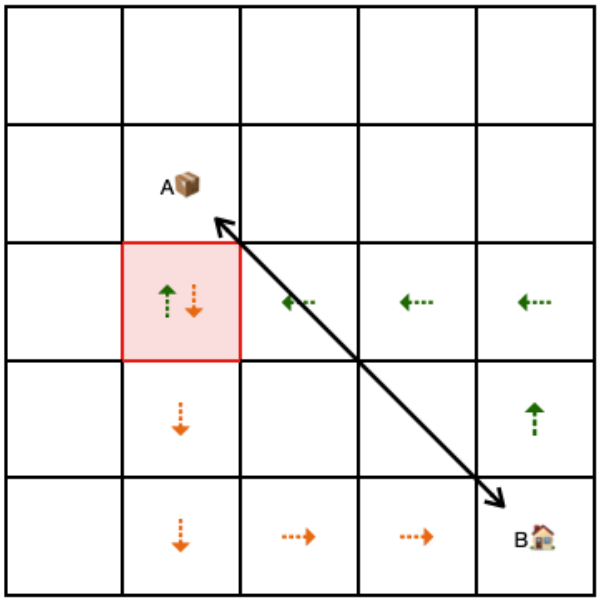
Multi-agent learning of a coordination problem

This document describes the FIT 5226 project which is your individual assignment. It is worth 50% and is **due on May 5, 2025. Note that there is an option to submit part of the assignment early that makes variations of the task accessible (see below, read the instructions carefully and completely now).**

> *"Give me six hours to chop down a tree and I will spend the first four sharpening the axe."*
>
> — Commonly misattributed to Abraham Lincoln

Tasks

You will write code for *multiple* agents in a square grid world to learn a simple transport task that requires coordination between the agents. You are free to use Deep-Q or Tabular-Q as you see fit. You are not allowed to use any learning methods (eg. heuristics) beyond Q.



Your agents still have the same four actions that they can execute: "move north/ south/west/east". A "wait" action in which the agent remains on the same grid square is not allowed. As before, there are two distinct cells A and B. The cells A and B can be in any location and agents are allowed to observe both locations. Each agent starts at one of the two locations A or B (during training, you are free to choose or randomise which one). As before, items need to be transported from A to B. Pickup at A and dropoff at B happen automatically when reaching the respective location without the agent having to take specific action. A and B are the same for all agents (but allocated randomly for each new training and test run).

Instead of performing a single delivery, agents now need to learn to *shuttle indefinitely between A and B, continuing to deliver new items*. (The supply at A and the demand at B are

unlimited). In other words, the task is not episodic anymore, the agents are learning an infinite behaviour.

Most importantly, the agents now have to learn to coordinate their paths in order to avoid collisions! More specifically, they have to avoid head-on collisions between different agents.[1] To simplify matters, we define a *head-on collision* as one that occurs between an agent moving from A to B and another agent moving from B to A (see red cell in the diagram). It is permissible for multiple agents to step on the same cell if and only if all move from A to B (or all move from B to A). Collisions in locations A and B are disregarded.

To keep the learning time within limits we work on a small 5x5 grid as before. We use 4 agents (you must not vary these numbers as otherwise the collision counts in Table 2 will change).

All agents are allowed to share the same DQN or Q-Table or use their individual q-table/network as you see fit.

The agents may observe their own location, the locations of A and B,[2] and, of course, whether they are carrying an item. Note that you can "buy" additional sensors to observe more information if you find this helpful (see below).

All agents normally have to perform their actions *sequentially in a random order*.

Simplifying the task by "purchasing" options

You have the option to balance aiming for higher performing agents with the complexity of the problem structure. The performance that your agents reach influences the marks you will receive for your solution and you can "purchase" options to simplify the task structure so that it becomes easier to train high performing agents. The catch, however, is that you will have to "pay" for each option that you select with points. You can compensate for the "purchase price" by training a better performing agent that is rewarded with "performance points" or you can solve the unmodified task and settle for a lesser performance. The table below details the options available and their impact on marks received.

Your raw mark for the implementation categories "Solution and Functionality" and "Implementation and Coding" of the rubric will be scaled by a factor of

$$\alpha = 1 - \frac{33}{200} max(0,\ C - B),$$

where *C* is the sum of costs for your chosen options and *B* the sum of points for the substantiated performance. In other words, for each 2 points of purchase cost that are not compensated for by outstanding learning performance you will lose ⅓ of the marks in these categories.

---

[1] They are still allowed to run into the wall.

[2] But see Table 1 regarding the option to choose a fixed location B

Table 1: Options available

Option Type	Item	Description	Cost
Sensors	State of neighbouring cells	You may augment the observation space for all agents with the occupancy state of all cells or chosen cells in the agent's immediate 8-neighbourhood (unoccupied, occupied)	2
	State of neighbouring cells checked for agents of opposite type	As per previous entry, only cells that contain an agent going in the opposite direction (as defined above) will be marked as occupied.	3
Coordination	Central clock	This allows you to coordinate the update schedule of your agents. Instead of having to update all agents in random order (as described above) you can update them in round-robin fashion or any other order that you determine.	1
Training conditions	Off-the-job training	Instead of having to learn with each episode starting from random locations for all agents and A, B you can define a schedule for the configuration at the start of training episodes.	2
	Staged Training	Instead of letting the agents learn everything in a single type of training run you may break the training into different phases (eg. with different grid sizes, different numbers of agents, different penalty schemata, etc.) Q-tables or q-networks may be passed between the stages.	3
Setup	Fixed delivery location B	Instead of having to service an (observable) random delivery location, the target location B is always in the bottom right corner of the grid. However, if you choose this option, the agents can no longer observe the location of B. Instead,	4

		they have to discover it.	

Table 2: Performance Points

You can receive up to 2 performance points for two categories, so that a maximum total of 4 is possible. To receive these you must provide code that measures the claimed performance!

Performance level *after training* (percentage of scenarios solved in less than 20 steps[3] and without any collisions)	Total number of collisions occurring *during training* to the claimed performance level	Performance Points
>95%	<500	2
>85%	<1,000	1

Required minimum performance

Any state-of-the art LLM will willingly solve the above task for you out-of-the box. However, it will solve it in a pretty naive, mediocre way. We act in the spirit of some major retail chains: "*If you see the same item somewhere else we'll beat it by 20%*". So your job is to beat the performance of these naive LLM products. Thus, your training must be completed within the following limits to achieve full marks:

Step budget: 1,500,000 (Maximum number of agent steps allowed during training)
Collision budget: 4,000 (Maximum number of head-on collisions allowed during training)
Walltime budget: 10 minutes (Maximum runtime allowed for training)
Final performance: after training, your agents must be able to perform a single delivery (starting at B) in at least 75% of all scenarios within at most 25 steps collision-free.

Task Details

You are free to use any work that was produced by you in the tutorials. You are even free to use the example solutions!

1. **Contract:** Determine your "shopping list", ie. the list of options you want to use. **During the first week of the assignment, the options will be on sale**. Each will only "cost" half the points listed in Table 1. You must submit your list of chosen options on Moodle by **April 17, 2025 if you want to obtain them at the early bird cost.** You can still pick other additional options after this date, but they will be

[3] Steps are counted by starting at A moving to B and returning to A.

"charged" at the full price. If you submit a list of early bird option purchases, this is deemed final and binding, ie. you will be "charged" for these options even if you later decide not to use them. Late submissions of the early bird purchase list will not be accepted unless formal special consideration for Assignment 1 applies and **an unsubmitted list will be taken to mean "no options selected."** You are not allowed to confer with other students in the class about your choices before the submission. This constitutes collusion. Please fill in the contract form available in the Moodle Assignment section and submit it through the Moodle portal.

2. **Implementation:** Implement the Q-Learning scenario in Python as a Jupyter notebook using any chosen options. If you want to use any other additional libraries apart from the provided Skeleton, Numpy and Matplotlib check with the unit coordinator beforehand whether these are admissible. Train and test your Q-Learning environment.To obtain performance points you must provide code so that the claimed performance can directly be tested. Observe the minimum requirements listed above.
3. **Documentation:** Explain your training approach in full detail. Devise a test procedure and use appropriate metrics and visualisations to show convincingly that your agents learn the task successfully and that they learn to coordinate effectively. Document this in full detail in writing.

Please be aware that you will have to participate in an interview where you have to demonstrate and explain your submission. Even if you use AI to solve the assignment (see below) you have to be able to fully explain every aspect of your submission to receive full marks, including the Python code.

Submission Instructions

Submission is due on MAY 5.
You must submit a single *.ipynb file via the Moodle Assignment section.

Use of Generative AI

You are allowed to use Generative AI to solve your assignment. If you decide to do so, you must treat the AI like another external author (as a non-authoritative author whom you mistrust, given how much content is made up by Chat GPT and similar AIs). It is entirely your own responsibility that the content is correct and you can only use generated content to the extent that you could use materials provided by an external author. The AI is not part of your project team.

Any use of generative AI must be appropriately acknowledged (see Learn HQ).

AI use that has not been declared will lead to the submission being disqualified.