KU LEUVEN

Multi-Agent Systems Based Solution for Pickup-And-Delivery

Thierry Deruyttere (r0660485)
Armin Halilovic (r0679689)
KU Leuven - Multi-Agent Systems

0 Outline

1 Outline

KU LEUVEN

1 Problem definition - Setting

- Pizzeria chain RoboPizza
- Pizza delivery using robots (AGVs)
- RoboPizza receives pizza delivery requests (tasks), robots deliver the pizzas

1 Problem definition - Environment

- Grid city system
- AGVs can go in both directions on streets
- AGVs can pass each other
- Dynamism:
  - Road works cause streets to be closed off
- Efficiency measure:
  - Total waiting time for tasks

1 Problem definition - Robots

- Can move from and to any position in the city
- Have static maps and can compute paths between locations
- Can carry up to 5 pizzas at once
- Can only communicate with entities that are close to them (i.e. city node they are on)
- Run on batteries which need to be recharged
  - Can charge at charging station
  - Station supports limited amount of robots at once
- Potential crash:
  - Can run out of battery $\rightarrow$ reset battery after time delay as simulation of manual replacement

1 Problem definition - Tasks

- Consist of picking up (multiple) pizzas and delivering them to a destination in the city
- If there are more than 5 pizzas in a task, it has to be split up
- Pizzas have no preparation time and can be picked up instantly
- Created each simulation tick with low probability that scales with city size:
  - Amount of pizzas by Gaussian distribution
  - Delivery position uniformly random in city

2 Outline

2 Objectives

- Develop a BDI & Delegate MAS algorithm for the described setting
- Analyze the performance for certain parameter settings

3 Outline

3 Research Questions and Hypotheses I

- What is the relation between the amount of requests that RoboPizza receives and the waiting time for customers?
  - H_1: The waiting time increases with the amount of requests.
  - H_0: The waiting time does **not** increase with the amount of requests.
- Do robots drive more (non-idle time) when there are more requests in the system?
  - H_1: Robots drive more when there are more requests in the system.
  - H_0: Robots do **not** driving more when there are more requests in the system.

3 Research Questions and Hypotheses II

- Does increasing the amount of robots decrease customer waiting time when there are many requests?
  - H_1: Increasing the amount of robots decreases customer waiting time when there are many requests.
  - H_0: Increasing the amount of robots does **not** decrease customer waiting time when there are many requests.
- How do waiting times change as the amount of road works changes (dynamism)?
  - H_1: Waiting times increase as the amount of road works increase.
  - H_0: Waiting times do **not** increase as the amount of road works increase.

4 Outline

4 Independent Variables

- n_{robots} = amount of delivery robots $_{(4)}$
- p_{task} = probability for a new task $_{(0.001 * city\_size)}$
- p_{road_works} = probability for road works to start $_{(0.0025 * city\_size)}$

4 Dependent Variables

- n_{task} = amount of tasks
- $n_{finished}$ = amount of finished tasks
- t_{wait} = waiting time for customers
- n_{pizzas} = amount of pizzas carried by robots
- $n_{distance}$ = distance all robots have traveled
- $t_{driving}$ = time robots spent driving
- t_{idle} = time robots were idle
- $t_{charging}$ = time robots spent charging
- n_{road_works} = amount of road works
- $n_{dropped}$ = amount of parcels that were dropped

4 Other Variables I

- l_{tick} = length of a single simulator tick (1000)
- $l_{simulation}$ = simulation length in experiments $(10hours)$
- $experiment\_repeats$ = amount of times experiments are ran (100)
- l_{city} = **city grid size** (20)
- l_{robots} = robot length (1)
- d_{node} = distance between two nodes on city grid $(2 * l_{robot})$
- v_{robot} = robot speed (1)
- $distance\_unit$ = distance unit (m)
- $speed\_unit$ = speed unit (m/s)

4 Other Variables II

- cap_{robot} = amount of pizzas a robot can carry (5)
- $cap_{charging\_station}$ = amount of robots that can charge simultaneously (2)
- $cap_{battery}$ = robot battery capacity $(l_{city} * l_{city} * d_{node})$
- $cap_{recharge}$ = battery capacity recharged per tick $(0.01 * cap_{battery})$
- $paths\_to\_explore$ = **amount of paths to explore towards destinations** (3)
- t_{road_works} = **time duration of road works** $(l_{city} * l_{city} * d_{node} * 1000)$

4 Other Variables III

- $t_{battery\_rescue}$ = time delay for replacement of empty batteries $(5 * (cap_{battery}/cap_{recharge}) * l_{tick})$
- $t_{intention\_reservation}$ = **time duration of intention reservations** $(4 * l_{city} * d_{node}/v_{robot}) * 1000)$
- $t_{refresh\_intention}$ = the amount of time it takes before a robot resends intention ants $(0.5 * t_{intention\_reservation})$
- $t_{refresh\_exploration}$ = the amount of time it takes before a robot resends intention ants $(0.4 * t_{intention\_reservation})$
- $\mu_{pizza}, \sigma_{pizza}$ = **pizza amount parameters** $(4, 0.75)$
- t_{pizza} = preparation and pickup time of a pizza (0)

5 Outline

5 Multi-Agent System Design: Overview

- Agents
  - Robot Agent for each robot
  - Resource Agent for each node on city graph
- Ant-based Delegate MAS
  - Desire Ants
    - Find delivery tasks
  - Exploration Ants
    - Find paths towards destinations
  - Intention Ants
    - Choose path and create reservation
- Buildings
  - 1 Pizzeria
  - 1 Charging Station
  - Random Road Works

5 Multi-Agent System Design: Ants

Ants are send on a hop by hop basis through resource agents. A resource agent can only send an ant to its neighbors. The ant knows to which neighbor it wants to go and notifies the resource agent of this.

1. Desire ant: are sent to all the available delivery tasks in the city and will gather information about each task. (amount of pizzas required, amount of time waiting, ...).
2. Exploration ant: Estimates travel duration for paths and also updates needed amount of pizzas if necessary
3. Intention ant: Creates reservations for best path for selected tasks

5 Multi-Agent System Design: Robot Strategy

1. send desire ants towards tasks
2. chooseBestTask()
3. sendExplorationAnts() (send explor. ants to chosen desires)
4. Send intention ants to confirmed tasks
5. moveToDestinations()

5 Multi-Agent System Design: chooseBestTask

1. collect sent desire ants
2. while agent has capacity left
    1. Choose a task t with fitness proportionate selection where fitness is waiting time.

5 Multi-Agent System Design: sendExplorationAnts

1. for i in range(Alternative paths):
   1. Calculate a path between chosen tasks with A* algorithm on static city graph.
2. send exploration ants according to calculated paths.

5 Multi-Agent System Design: moveToDestinations

1. follow path for confirmed intentions
2. check if need to recharge
   1. if True: chargeLogic()
   2. if False, continue
3. refresh exploration and intention ants if refresh timer is fired
4. if at intention:
   1. if all intentions finished send out exploration ant to find pizzeria and set intention to pizzeria.

5 Multi-Agent System Design: chargeLogic

1. save current intentions
2. set intention to charging station
3. move to charging station
4. charge
5. while the agent is charging update the path the agent needs to follow for its old intentions
6. if charging is finished:
    1. set intentions to old intentions

6 Outline

6 Experiments

- Different parameter settings for each hypothesis
- Each experiment is run 100 times
- Experiment end statistics are written to files
- Test method: two-sample t-test
- Significance level: 0.05
- Hypotheses entail one-tailed tests. Null hypothesis will be rejected if the mean difference between sample means is too small.

- Could not experiment with varying dynamism because of an error with RinSim Experiment repeats we couldn't fix

6 Experiments: Question 1 (1)

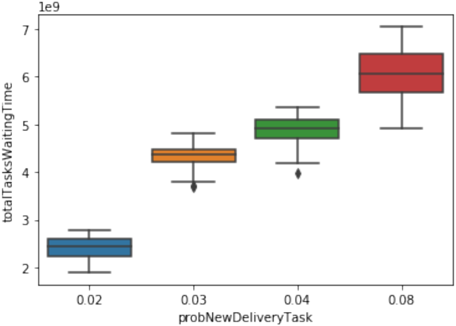
- What is the relation between the amount of requests that RoboPizza receives and the waiting time for customers?

- μ_1: mean waiting time for lower amount of requests
- μ_2: mean waiting time for higher amount of requests

- H_0: The waiting time does not increase with the amount of requests. $(\mu_1 \geq \mu_2)$
- H_1: The waiting time increases with the amount of requests. $(\mu_1 < \mu_2)$

6 Experiments: Question 1 (2)

- Total task waiting time



- $\mu_1 \geq \mu_2$
  - t-value $= 56.908$
  - p-value $= 0.000$
  - $\rightarrow$ Reject null hypothesis
- $\mu_2 \geq \mu_3$
  - t-value $= 14.425$
  - p-value $= 0.000$
  - $\rightarrow$ Reject null hypothesis
- $\mu_3 \geq \mu_4$
  - t-value $= 21.355$
  - p-value $= 0.000$
  - $\rightarrow$ Reject null hypothesis

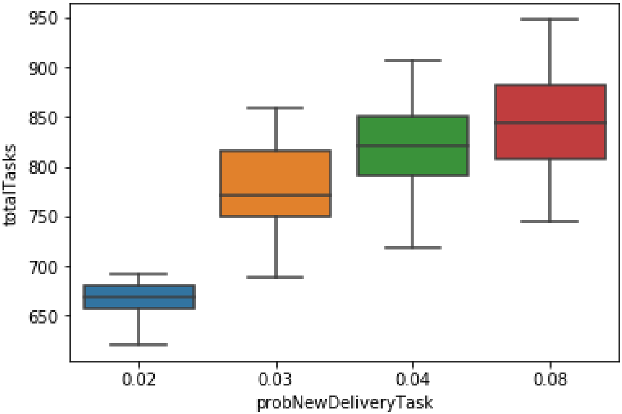
6 Experiments: Question 2 (1)

- Do robots drive more (non-idle time) when there are more requests in the system?
  - H_1: Robots drive more when there are more requests in the system.
  - H_0: Robots do **not** driving more when there are more requests in the system.

- TODO: Hypotheses in termen van $H_0 : \mu_1 \geq \mu_2$ en $H_1 : \mu_1 < \mu_2$
- http://stattrek.com/hypothesis-test/difference-in-means.aspx

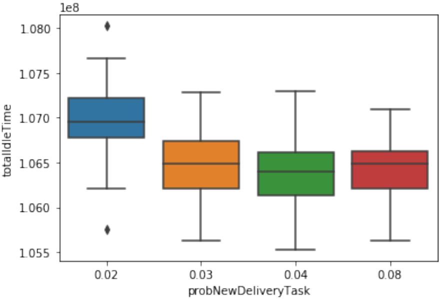
6 Experiments: Question 2 (3)

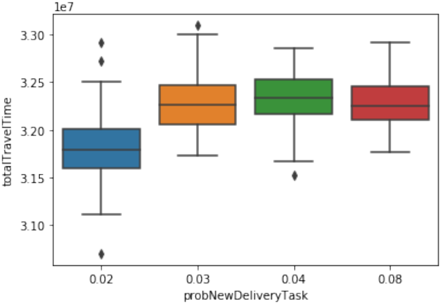
- Total tasks



6 Experiments: Question 2 (2)

- Total idle and travel time





6 Experiments: Question 3 (1)

- Does increasing the amount of robots decrease customer waiting time when there are many requests?
  - H_1: Increasing the amount of robots decreases customer waiting time when there are many requests.
  - H_0: Increasing the amount of robots does **not** decrease customer waiting time when there are many requests.

- TODO: Hypotheses in termen van $H_0 : \mu_1 \geq \mu_2$ en $H_1 : \mu_1 < \mu_2$
- http://stattrek.com/hypothesis-test/difference-in-means.aspx

6 Experiments: Question 3 (2)

- grafieken ofzo

7 Outline

7 Conclusions

- conclude our Conclusions

Questions?