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

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

1. Problem definition
2. Objectives
3. Research Questions and Hypotheses
4. Multi-Agent System Design
5. Variables
6. Experiments
7. Conclusions

KU LEUVEN

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_0: The waiting time does not increase with the amount of requests.
  - H_1: The waiting time increases with the amount of requests.
- Do robots drive more (non-idle time) when there are more requests in the system?
  - H_0: Robots do not drive more when there are more requests in the system.
  - H_1: Robots drive 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_0: Increasing the amount of robots does not 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.
- How do waiting times change as the amount of road works changes (dynamism)?
  - H_0: Waiting times do not increase as the amount of road works increase.
  - H_1: Waiting times increase as the amount of road works increase.

4 Outline

4 Multi-Agent System Design: Overview

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

4 Multi-Agent System Design: Ants

- Ants travel to destinations through resource agents
- Follow paths generated by robots (using their static maps)
- Are sent to next resource agents at every simulation tick

- **Desire ants**: sent to all delivery tasks in the city and gather information about each task (amount of pizzas left, waiting time, estimated travel time).
- **Exploration ants**: estimate travel time for paths (accounting for road works) and get amount of pizzas left in case it changed since desire ant
- **Intention ants**: create reservations (with expiration times) for deliveries with possibly multiple destinations and confirm the intentions of robots

4 Multi-Agent System Design: Robot Strategy I

- Simplified version of strategy to fit presentation in time slot
- Robot strategy loop:
  1. Read all new messages (ants)
  2. Process ants if all ants of a type have returned
  3. Reconsider intentions if not waiting for ants
  4. Resend exploration or intention ants if respective timer has passed
  5. Do action if intention is present, not charging, and not waiting for ants
- Will explain 2, 3, and 5 in more detail

4 Multi-Agent System Design: Robot Strategy II

- 2.A) Process desire ants if all have returned:
  - (Desire ants contain information about waiting delivery tasks)
  - Select as many tasks as the robot can handle at once
  - Select tasks using fitness proportionate selection where fitness is waiting time
  - Send exploration ants along alternative paths that visit all selected tasks
  - Alternative paths are found using static map and A* with penalty for common nodes on paths [1]

4 Multi-Agent System Design: Robot Strategy III

- 2.B) Process exploration ants if all have returned:
  - (Exploration ants contain travel time for the path they traveled)
  - Select the path with the lowest estimated travel time
  - If there is a current intention and current estimated travel time left is lower than new estimated time, then stop here
  - If intended destination is pizzeria, set intention to selected path
  - Else send intention ant along this path

- 2.C) Process intention ant if returned:
  - If want to go to charging station, set intention to path of ant
  - Else (want to go to delivery destinations):
    - If not carrying pizzas, get pizzas from pizzeria
    - Set intention to path of ant

4 Multi-Agent System Design: Robot Strategy IV

- 5) Do action if intention is present, not charging, and not waiting for ants:
  - Move along the current intention
  - If currently at a delivery task destination, deliver pizzas
  - If was going charging station and arrived, then set up for charging
  - If was going to pizzeria and arrived, then set up for new delivery tasks
  - If last node on intention was reached, remove it so that intention reconsideration can find a new one

4 Multi-Agent System Design: Robot Strategy V

- 3) Reconsider intentions if not waiting for ants:
  - If intention is present, but no connection exists to next node on path, delete the intention
  - If no intention is present
    - If going to charge, explore paths towards charging station
    - If carrying pizzas for delivery tasks, explore paths for the tasks
    - If at pizzeria, get waiting delivery tasks from pizzeria and send desire ants
    - If not at pizzeria, not charging, and not going to charge, then explore paths towards pizzeria (to get new tasks)
  - Else (intention is present, but check if it should be replaced by a more important one)
    - If battery is low, delete current intention (keep any pizzas being carried) and explore paths towards charging station

5 Outline

5 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 $(0.0025 * city\_size)$

5 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

5 Other Variables I

- l_{tick} = length of a single simulator tick (1000)
- $l_{simulation}$ = simulation length in experiments $(10 hours)$
- $experiment\_repeats$ = 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)

5 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** $(l_{city} * l_{city} * d_{node} * 1000)$

KU LEUVEN

5 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)

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
- Labels in plots show the independent variables that were varied

6 Experiments: Question 1 (1)

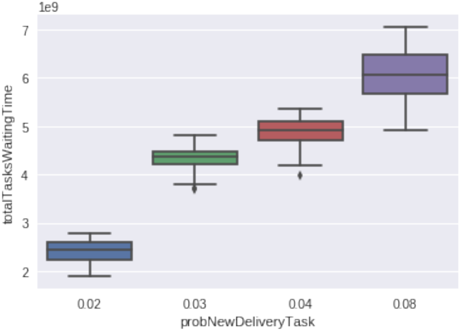
- What is the relation between the amount of requests that RoboPizza receives and the waiting time for customers?

- 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 H_0
- $\mu_2 \geq \mu_3$
  - t-value $= 14.425$
  - p-value $= 0.000$
  - $\rightarrow$ Reject H_0
- $\mu_3 \geq \mu_4$
  - t-value $= 21.355$
  - p-value $= 0.000$
  - $\rightarrow$ Reject H_0

6 Experiments: Question 2 (1)

- Do robots drive more (non-idle time) when there are more requests in the system?

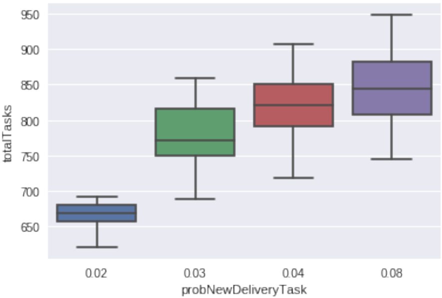
- μ_1: mean travel time for lower amount of requests
- μ_2: mean travel time for higher amount of requests

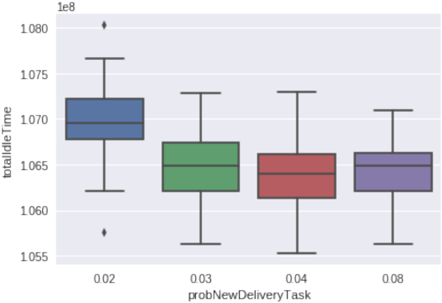
- H_0: Robots do not drive more when there are more requests in the system ($\mu_1 \geq \mu_2$)
- H_1: Robots drive more when there are more requests in the system ($\mu_1 < \mu_2$)

- Interesting to check if communication overhead for task distribution increases under high load

6 Experiments: Question 2 (2)

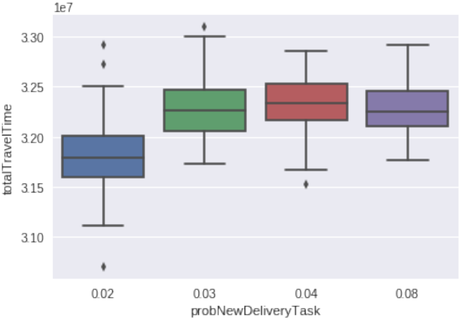
- Total tasks and robot idle time





6 Experiments: Question 2 (3)

- Total travel time



- $\mu_1 \geq \mu_2$
  - t-value = 10.583
  - p-value = 0.000
  - $\rightarrow$ Reject H_0
- $\mu_2 \geq \mu_3$
  - t-value = 0.590
  - p-value = 0.556
  - $\rightarrow$ Cannot reject H_0
- $\mu_3 \geq \mu_4$
  - t-value = -0.582
  - p-value = 0.561
  - $\rightarrow$ Cannot reject H_0

6 Experiments: Question 3 (1)

- Does increasing the amount of robots decrease customer waiting time when there are many requests ($p_{task} = 0.04$)?

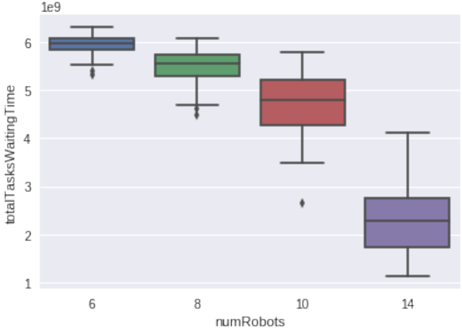
- μ_1: mean waiting time for lower amount of robots
- μ_2: mean waiting time for higher amount of robots

- H_0: Increasing the amount of robots does not decrease customer waiting time when there are many requests ($\mu_1 \geq \mu_2$)
- H_1: Increasing the amount of robots decreases customer waiting time when there are many requests ($\mu_1 < \mu_2$)

- Interesting to see the effect of more robots under high load. Higher load and more robots each cause more communication.

6 Experiments: Question 3 (2)

- Total task waiting time



- $\mu_1 \geq \mu_2$
  - t-value = -12.150
  - p-value = 0.000
  - $\rightarrow$ Cannot reject H_0
- $\mu_2 \geq \mu_3$
  - t-value = -11.090
  - p-value = 0.000
  - $\rightarrow$ Cannot reject H_0
- $\mu_3 \geq \mu_4$
  - t-value = -24.925
  - p-value = 0.000
  - $\rightarrow$ Cannot reject H_0

6 Experiments: Question 4 (1)

- How do waiting times change as the amount of road works changes (dynamism)?

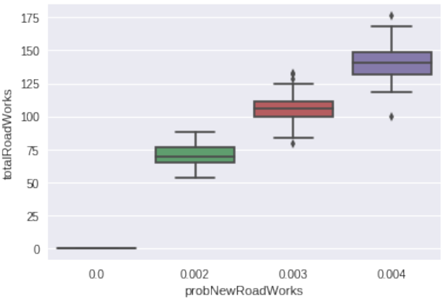
- μ_1: mean waiting time for lower amount of road works
- μ_2: mean waiting time for higher amount of road works

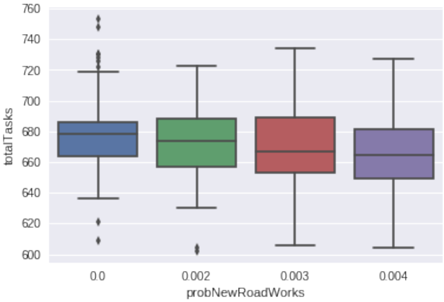
- H_0: Waiting times do not increase as the amount of road works increases ($\mu_1 \geq \mu_2$)
- H_1: Waiting times increase as the amount of road works increases ($\mu_1 < \mu_2$)

- Interesting to see how the system copes with dynamism

6 Experiments: Question 4 (2)

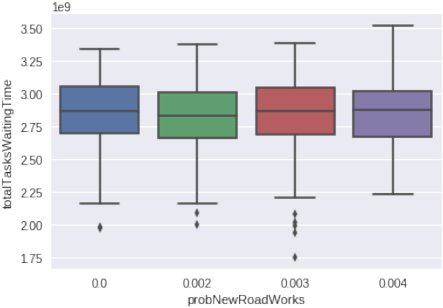
- Total road works and delivery tasks





6 Experiments: Question 4 (3)

- Total task waiting time



- $\mu_1 \geq \mu_2$
  - t-value = -0.636
  - p-value = 0.525
  - $\rightarrow$ Cannot reject H_0
- $\mu_2 \geq \mu_3$
  - t-value = 0.391
  - p-value = 0.696
  - $\rightarrow$ Cannot reject H_0
- $\mu_3 \geq \mu_4$
  - t-value = 0.732
  - p-value = 0.465
  - $\rightarrow$ Cannot reject H_0

7 Outline

KU LEUVEN

7 Conclusions

- We have:
  - Developed a Multi-Agent Systems based solution for a pickup-and-delivery problem
  - Using the BDI model in combination with Delegate MAS
  - And have studied the effect of certain parameters on its performance

7 Bibliography

Hoang Tung Dinh, Rinde van Lon, and Tom Holvoet.
Multi-agent route planning using delegate mas.
In *Workshop on Distributed and Multi-Agent Planning*, pages 24–32, 2016.

Questions?