A Multi-agent, Marine Pollution, Collection and Processing System

December, 2020

George Bradley(20113376)
Wai Kit Jay, Lee (1624263)
Lexuan Shao (20060097)
Idris Sardary (1419164)

King's College London
Strand, London, United Kingdom
WC2R 2LS

1 Purpose of Platform

1.1 Problem

Marine pollution is a current and increasing threat and concern for both the health of the oceans and wildlife but also to the wider global population. Every year billions of tonnes of pollutants enter the ocean [1]. Our multi-agent system of marine pollution collection drones and recycling plants aims to counter and help minimise this problem.

1.2 Aims

- Maximise the collection of marine pollution.
- Coordinate a system of marine pollution collection drones.
- Deliver collected pollution to both land and sea based recycling plants.
- Negotiate situations where a collection drone has ceased operating and or inclement weather is encountered.
- Each agent is assigned a certain amount of credits to be used for negotiations and in order to be penalised or remunerated for either breaking or following rules (Section 5 and 8).
- Allocate the collected pollution to recycling plants effectively, in order to minimise the recycling plant's free capacity.

1.3 Agents and Goals

Our platform consists of four agents, a Central Control Station "CCS", Pollution Collection Drones "PCD" and recycling plants both Land Based Recycling Plans "LBRP" and Sea Based Recycling Plants "SBRP". Each of these agents has a macro goal which it seeks to fulfill.

Agent Goals:

- CCS: Monitor, coordinate and collect data on agents.
- LBRP: Process pollution.
- SBRP: Process pollution.
- PCD: Collect pollution and deliver the pollution to LBRPs or SBRPs for processing.

2 High Level Design

2.1 Environment

The environment which the agents operate within is dynamic, as there are other processes and agents which operate on it, such as travelling vessels and changing weather conditions. Additionally, the environment is non-deterministic and non-accessible, there is uncertainty about the state that will result from performing an action and the agents operating are unable to obtain complete and up-to-date information of the entire environment's state. This is primarily due to the size and complexity of their environment.

2.2 Unified Modeling Language (UML) Diagram

Figure 2.2 outlines what data each of the agents stores along with the actions which an agent can perform.The CCS whose goal is to collect data and coordinate agents stores the state data of all PCD and recycling plant agents. Along with agent state data, the CCS also stores the weather forecast which it uses to help coordinate PCD agents in the scenario of inclement weather. As there is only one CCS within the system the relationship between the CCS and the other agents is one-to-many. The PCD stores simple state data about its location the location of neighbours and the its current capacity, it can carry out actions such as collect pollution and transmit and receive signal which allows it to complete its goal. PCDs can also interact with each other on a 1-to-1 basis which is used for negotiation (section 6). The recycling plant agents store simple state data and are able to carry out actions to help them fulfill their goal of processing pollution along with the ability to transmit and receive signals from other agents within the system.

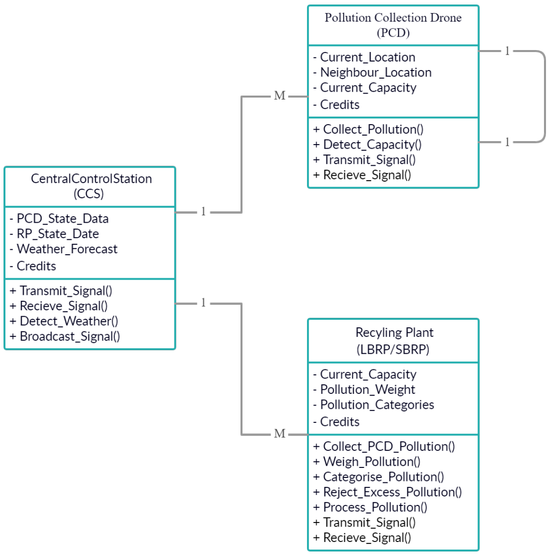

Figure 2.2 UML diagram explaining the data stored and actions available to each agent.

2.3 Entity Relationship Diagram

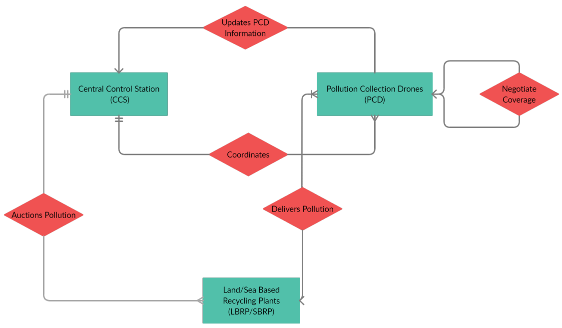

Figure 2.3 Entity Relationship Diagram depicting the relationships between agents in the system.

The entity relationship diagram figure 2.3 describes the operations which occur between agents.The CCS acts as an auctioneer for the Recycling Plants, the auction is used as a function to best allocate collected pollution for processing (section 8). Based on the outcome of the auction the CCS coordinates the PCDs to travel to the correct recycling plant LBRP or SBRP to deposit their collected pollution for processing. The recycling plants and PCDs send their updated state data to the CCS. The PCDs also negotiate with each other over pollution collection domains outlined in section 6.

3 Reasoning with beliefs, desires and intentions

3.1 Introduction

All of the agents within our system act according to a set of beliefs, desires and intentions which are influenced by their percepts [2]. The BDI process constantly runs except for when agents need to make group decisions (section 7) or negotiations (section 6). The BDI framework joins nicely with our domain as each agent is designed to fulfill a set of desires [3], through the BDI process a series of plans are initialised and constructed through percepts from the environment in order to guide the agents to fulfil their goals.

3.2 Agent's BDI sets

Agent's desires are founded on their core goals outlined in section 1.3.

- The Pollution Collection Drone's (PCD) goal is to collect pollution and deliver this pollution to LBRPs or SBRPs whilst being able to deal with situations where other PCD malfunction or bad weather is expected.
- The Central Control Station's (CCS) goal is to monitor, coordinate and collect data on agents. In order to accomplish this it needs to be able to receive and send information which can be in the form of instruction to other agents within the system.
- Both land and sea based recycling plant's sole aim is to process the most pollution they can, in order to maximise this they need to be able to share their state information with the other agents within the system.

	Set of Beliefs(Bel)	**Set of Desires(Des)**	**Set of Interntions**	**Set of Percepts**
Pollution Collection Drone	{agent's location, agent's health, state of other PCDs, status of the weather}	{reduce pollution}	{collect pollution, move location, deliver pollution, detect tank capacity , communicate status information, detect location, detect other PCD location, detect weather, receive information}	{full capacity reached, detect agent's health, detect weather, detect agent's location, detect surrounding objects, detect information to and from CCS}
Central Control Station	{states of all PCDs, states of all Recycling Plants, status of the weather forecast}	{communicate with agents, coordinate with agents}	{receiving PCD information, receive Recycling Plant information, transmit information}	{weather state, status of all PCDs, status of all Recycling Plants}
Recycling Plants (Land Based & Sea Based)	{current capacity, pollution weight, pollution categories}	{recycling pollution}	{receive pollution, process pollution, communicate status, receive information, check capacity, weigh pollution, categories pollution}	{pollution received}

Table 3.2 The BDI sets for each of the agents within the system.

The pseudocode shown in figure 3.2 describes the logic of when the PCD agent reaches full capacity and transport's the collected pollution to a recycling plant for processing [4].

Initially the sets of beliefs and intentions are empty for an agent until it receives a new percept. The desire of a PCD agent is always to reduce pollution. The PCD, after receiving a new percept and sensing it has reached full capacity plans to travel to the nearest recycling plant which is "in need" in order to deposit its pollution, this then becomes its new intention. The first while loop directs the agent to travel to the recycling plant's location. The second while loop instructs the agent to deposit its pollution, then the while loop ends as the PCD agent has achieved its intention. After which the process begins again with a new set of beliefs, desires and intentions.

```
B0 = {}
I0 = {}

while true do

    new percept = {capacity: 100%, health: 90%, weather: good, location: area A,
    surrounding objects: 0, CSS information: safe}
    new B = {location: Area A, health: 90%, other PCD states: healthy, weather: good}
    new D= {reduce pollution}
    new I = {move location, deliver pollution }
    Pi = { Move location: travel to location, deliver pollution: drop when Move location is
    complete}

    first while loop( Pi is not empty, not succeeded yet nor not impossible to achieve)

        a = move location
        execute(a)
        new Pi = { deliver pollution: drop when Move Location is complete}
        new percept = {capacity: 100%, health: 85%, weather: good, location: Plant A,
        surrounding objects: 0, CSS information: safe}
        new B = {location: Plant A, health: 85%, other PCD states: healthy, weather:
        good}
        reconsider(I, B) = false
        sound(I, B) = true

    second while loop( Pi is not empty, not succeeded yet nor not impossible to achieve)

        a = deliver pollution
        execute(a)
        new Pi = { }
        new percept = {capacity: 0%, health: 84%, weather: good, location: Plant A,
        surrounding objects: 0, CSS information: safe}
        new B = {location: Plant A, health: 84%, other PCD states: healthy, weather:
        good}
        reconsider(I, B) = false
        sound(I, B) = true

     third while loop breaks as Pi is empty which means it has succeeded.
```

Figure 3.2 BDI Psuedocode example for when a PCD reaches full capacity and transport's its pollution to a recycling plant.

4 Temporal Logic Reasoning (Concurrent MetateM)

4.1 MetateM process

MetateM is a crucial element of our system, it is the method in which the agents within the system communicate with each other in order to fulfill the plans created by the BDI process explained in section 3. A range of communication is carried out using MetateM, from the simplistic agent state updates which are sent to the CCS to more complex communication such as combinatorial auctioning and inter agent negotiations discussed in section 8 and 6 [5].

4.2 MetateM example

Below is an example of communications between the CCS, LBRP/SBRP and PCD agents:

ask1 : Pollution information
ask2 : State of weather
ask3 : Detect locations
ask4 : Detect status of PCDs
ask5 : PCD agent's capacity
give1: Weather forecast update.
give2: Instructions describing which RP the PCD should travel too.

CCS(ask1,ask2,ask3,ask4)[give1,give2]
Start→ $\Box(give1 \wedge give2)$
$\Diamond(ask1 \wedge ask2 \wedge ask3 \wedge ask4) \rightarrow (give1 \wedge give2)$

SBRP/LBRP(give1,give2)[ask1, ask2, ask3, ask5]
Start→ $ask5$
$\Diamond\varnothing(give1 \wedge give2) \rightarrow (ask1 \wedge ask2 \wedge ask3 \wedge ask5)$

PCD(give1,give2)[ask4]
Start→ $ask4$
$\Diamond\varnothing(give1 \wedge give2) \rightarrow ask4$

Table 4.2 demonstrates the execution of the the system's communication. In the defined MetateM the CCS broadcasts the weather forecast to the PCDs and LBRP/SBRPs. The CCS additionally transmits the information of which recycling plant requires pollution to the PCD. The PCD agents sense their current status and broadcast it to the CCS whilst accepting coordination signals from the CCS. The recycling plant agents are able to both receive and transmit information to-and-fro the CCS.

CCS		**LBRP/SBRP**		**PCDs**	
Propositions	Commitments	Propositions	Commitments	Propositions	Commitments
	(give1 give2)	ask5		ask4	
ask4,ask5		ask5		ask4	
ask4,ask5	give1	ask5		ask4	
ask4,ask5,give1	give2	ask5		ask4	
ask4,ask5,give1,give2	give1,give2	ask1, ask2, ask3, ask5		ask4	

Table 4.2. Propositions and commitments for each agent communicating in the system.

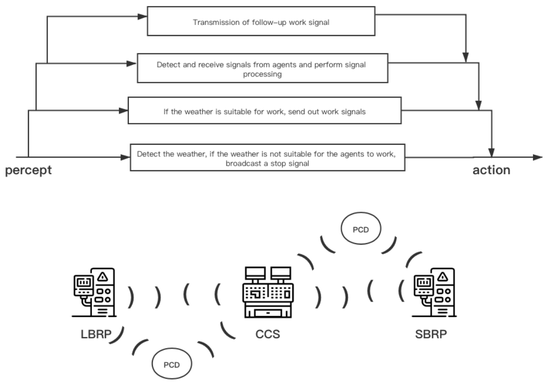

Figure 4.2. Communication between CCS and all other system agents.

5 Prescriptive Norms and Trust Assessment

5.1 Prescriptive Norms

Prescriptive norms form a important part of our system, providing the constraints for the agent's actions whilst simplifying the reasoning process. Referring back to section 1.2 each agent within the system receives 100 credits on initialisation which are deducted from an agent if they execute a prohibited norm, additionally the amount of credits which an agent has is part of the trust assessment described in section 5.2. The credit system is also used as the currency for the combinatorial auctions outlined section 8.

Below we define the observation norms, prohibition norms and permission norms for each agent.

5.1.1 Pollution Collection Drone (PCD)

Obligation norm
<all-PCDs, obligation, received-stop-signal, stop, 2-credit-minus>
<all-PCDs, obligation, received-task-signal, process-signal, 2-credit-minus>
Prohibition norm
<all-PCDs, prohibited,on-bad-weather, work, 5-credit-minus>
Permission norm
<all-PCDs, permitted,on-good-weather, work, >

5.1.2 Land/Sea Based Recycling Plants (LBRP/SBRP)

Obligation norm
<both-LBRP-SBRP, obligation, detect-pollution, weigh-pollution and process, 2-credit-minus>
Prohibition norm
<both-LBRP-SBRP, permitted,on-bad-weather, process-pollution, 5-credit-minus>
Permission norm
<both-LBRP-SBRP, permitted,on-bad-weather, provide-shelter, >

5.1.3 Central Control Station (CCS)

Obligation norm
<CCS, obligation, detect-weather, broadcast-signal, 2-credit-minus>
<CCS, obligation, receive-signal, transmit-signal, 2-credit-minus>
Prohibition norm
<CCS, prohibited, on-bad-weather, stop, 5-credit-minus>
Permission norm
<CCS, permitted, receive-emergency-signal, broadcast-high-level-signal, >

5.2 Trust Assessment

Trust assessment is applied to our domain when multiple agents perform a task, the CCS will determine the trust level of the agent based on the agent's past performance. Using the FIRE model, the agent's trust level can be calculated. An agent with a higher trust level will be preferred and assigned to tasks.

5.2.1 Trust Assessment example

The CCS agent, Ag0, determines which PCD agent to assign a task to using the FIRE trust model. The two PCD agents are Ag1 and Ag2.

5 hours ago, Ag0 delegated to Ag1 with rating of 0.9
4 hours ago, Ag0 delegated to Ag2 with rating of 0.8
2 hours ago, Ag0 delegated to Ag1 with rating of 0.6

A interaction which occurred in the present hour is given twice the weighting of an interaction that occurred an hour before. The CCS assigns the task to Ag2, the trust assesment calculation is presented below.

$$\text{Ag1} \quad \frac{0.9 \times 1 + 0.6 \times 6}{7} \approx 0.64$$

$$\text{Ag2} \quad \frac{0.8 \times 1}{1} = 0.8$$

6 Alternating Offers Protocol

The alternating offers protocol is applied in a situation where a PCD stops functioning. Its domain, where it gathers pollution must be taken over by another PCD. The two nearest PCDs enter into a negotiation concerning the allocation of the malfunctioning PCD's domain. The PCD with the greatest amount of free capacity is favoured.

6.1 Alternating Offer Protocol Implementation

(Italic fonts represent the optimal/Nash equilibrium options)

- PCD1 collects some pollution then PCD2 collects some pollution and continues.
- PCD2 collects some pollution then PCD1 collects some pollution and continues.
- *PCD1 collects pollution until it has reached full capacity and then PCD2 collects until full capacity.*
- *PCD2 collects pollution until it has reached full capacity and then PCD1 collects until full capacity.*
- Neither collects

Conflict deal:
Neither PCD collects pollution and the malfunctioning PCD's domain the domain remains unused.

Resource up for division:
Pollution from the malfunctioning PCD's domain

Favoured:
PCD free capacity, smallest is best. If the capacity of both PCDs in negotiation is the same then a PCD will be chosen randomly.

The overall goal is to collect as much pollution as possible as quickly as possible.

6.2 Alternating Offer Protocol Example

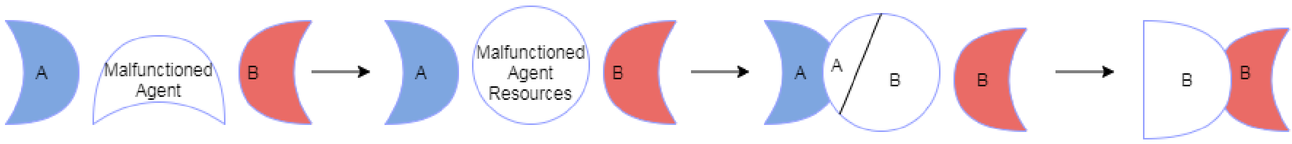

Figure 6.2. Diagram displaying the steps taken in the example outlined in section 6.2

PCD1: Free capacity 80%
PCD2: Free capacity 50%

Estimated pollution within the malfunctioning PCD's domain is equivalent to 70% PCD capacity.

- PCD1 and PCD2 begin negotiations with the respective states "80%" and "50%".
- PCD2 has all the bargaining power since it has smallest free capacity and therefore the needs the least pollution to reach full capacity.
- PCD2 proposes (50,20), PCD1 accepts.

Accepted deal after negotiations:
PCD2 operates within the malfunctioning PCD's domain until it reaches full capacity and then PCD1 operates with the domain collecting the remaining pollution.

7 Voting Accounting for Preferences

Group decision voting is applied when the PCDs decide among themselves where they will travel in order to collect pollution. The environment which the PCDs operate in can be visualised as a grid with each element of the grid being a fixed distance i.e: 1x1miles. The PCDs come to a group decision about which area they will operate within, this area is referred to as their domain.

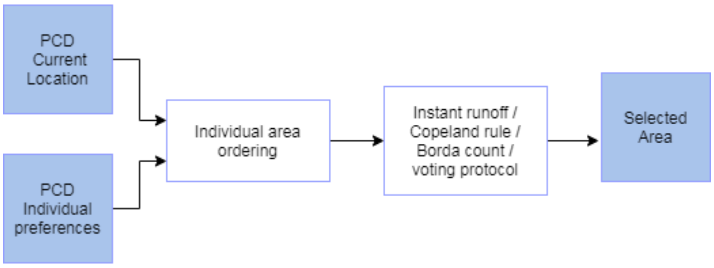

Figure 7. Depiction of the voting process for PCD group decisions when deciding on operating domains

7.1 Voting protocol example

In this example we assume that there are three areas which the PCDs can operate within, a total of 60 PCDs have voted and listed their preferences. Multiple voting protocols can be used in order to reach a group decision which are described below.

Environment areas: A,B,C

10 voted for A >C >B	30 voted for C >A >B	20 voted for B >C >A

Table 7.1. The number of votes for each preference.

7.1.1 Voting Methods

Instant runoff voting protocol:

Round	1	2
Procedure	Area A : 10 top choice votes Area B : 20 top choice votes Area C : 30 top choice votes	Area B : 20 top choice votes Area C : 40 top choice votes
Eliminated	Area A	Area B

Table 7.1.1. Depiction of the voting process for PCD group decisions when deciding on operating. domains

Result: Area C wins

Copeland rule voting protocol:

Pairs:
Area A vs Area C = 10 vs 50 = C wins: +1, A loses: -1
Area A vs Area B = 40 vs 20 = A wins: +1, B loses: -1
Area C vs Area B = 40 vs 20 = C wins: +1, B loses: -1

Areas	A	B	C
Score	0	-2	2

Copeland rule voting calculations

Result: Area C wins

Borda count voting protocol:

Area A : 2x10 + 1x30 = 50
Area B : 2x20 = 40
Area C : 2x30 + 1x30 = 90

Result: Area C wins

8 Combinatorial Auctions

Combinatorial auctions are applied by having recycling plants act as agents bidding for the pollution carried by the PCDs whilst the CCS assumes the role of the auctioneer. The credit system outlined in section 1.2 and 5.1 is the bidding currency of the recycling plants.

8.1 Combinatorial Auction example

A SBRP is bidding against a LBRP with the CCS agent acting as the auctioneer. The goods that are being bid on are two PCD agents which are carrying different weights of pollution.

Z = PCDa, PCDb
Ag = SBRP, LBRP

Scale: 100% PCD capacity == 10% Recycling plant capacity.

Recycling plant and PCD capacity:
Plant SBRP: 90% capacity - 10% free capacity
Plant LBRP: 40% capacity - 60% free capacity
PCDa: 10% capacity
PCDb: 30% capacity

Valuation Logic:
Plants that are close to full capacity (high capacity, low free capacity) prefer PCDs with less pollution to avoid overloading and having to reject excess pollution.

Valuation function: (100 - | PCD CAPACITY - RECYCLING PLANT FREE CAPACITY | *100%)

V1 from plant SBRP.
V2 from plant LBRP.

True Valuation function from both agents (Recycling plants):

V1 (PCDa) = 100- $\left|10\% - 10\%\right|$ *100% = 100
V1 (PCDb) = 100- $\left|30\% - 10\%\right|$ *100% = 80
V2 (PCDa) = 100- $\left|10\% - 60\%\right|$ *100% = 50
V2 (PCDb) = 100- $\left|30\% - 60\%\right|$ *100% = 70

Possible allocations: $(\varnothing, \varnothing), (\varnothing, a), (\varnothing, b), (a, \varnothing, (b, \varnothing, (a, b), (b, a), (a, b, \varnothing) and (\varnothing, a, b)$

Allocation	$(\varnothing, \varnothing)$	$(\varnothing, \{a\})$	$(\varnothing, \{b\})$	({a},∅})	({b},∅}	({a},{b})	({b},{a})	({a,b},∅)	$(\varnothing, \{a, b\})$
Social welfare	0	50	70	100	80	170	130	180	120

Table 8.1. The social welfare score for each possible allocation.

Optimal allocation (maximised social welfare): (a,b,∅)

Amount of credits each agent must pay:
Plant SBRP: 120 - 0 = 120, (if SBRP reported 0 for both, optimal allocation would be $(\varnothing, a, b)$)
Plant LBRP: 180 - 180 = 0, (if LBRP reported 0 for both, optimal allocation would be (a,b,∅))

Utility each agent get:
Plant SBRP: 180 - 120 = 60
Plant LBRP: 0 - 0 = 0

9 Reflection

The function of our agents are predefined, the PCDs and the recycling plants all hold goals which they wish to achieve, ie: collect and process pollution. The goal orientated definition of our agents and allows for an effective integration with the BDI framework.

Prescriptive norms are a useful concept for our system as they create constraints which define the actions of the agents. Trust Assessment can be successfully implemented to a certain extent as it allows for the identification of reliable agents, however performs best when a system's agents have incentive to compete with one another which does not hold true within our system.

The alternating offer protocol translates well to our domain, the protocol acts as a resolution function in the event of an agent malfunctioning. Currently the protocol is driven by one metric from each agent the agents current capacity, perhaps for practical use a protocol driven by multiple metrics may lead to a more nuanced and optimal resolution.

Voting is successfully applied to our domain in order to decide where each PCD will travel to collect pollution this allows the PCDs to operate in an autonomous, decentralised cohesive manner in order to effectively divide the operating environment eliminating the possibility of PCDs trending to a certain positions whilst meeting the preferences of individual PCDs.

Combinatorial auctions are an effective method for our central control station to allocate resources efficiently whilst maximising the systems goal of recycling pollution.

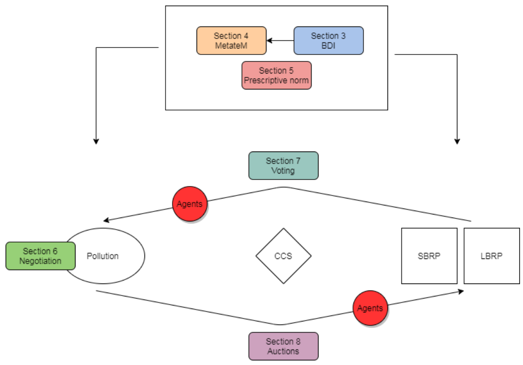

Figure 9. A graphic representation of the interaction between all the agents and concepts used within the system.

10 References

References

[1] ["Ocean pollution — National Oceanic and Atmospheric Administration," Noaa.gov, 2015. https://www.noaa.gov/education/resource-collections/ocean-coasts/ocean-pollution (accessed Dec. 02, 2020).]

[2] [Wooldridge, M. (2009). An Introduction to MultiAgent Systems, 2nd edition. Wiley. Pages 66 - 69].

[3] [Dastani, M. (2008). 2APL: a practical agent programming language. Autonomous Agents and Multi-Agent Systems. 16:214–248.].

[4] Black, Dr Elizabeth. "Overview of Topic 4: Practical Reasoning Agents." BDI Agents - Part 4, keats.kcl.ac.uk/mod/lesson/view.php?id=4567057.

[5] [Wooldridge, M. (1996). In Proceedings of the International Workshop on Agent Theories, Architectures, and Languages. Springer. Pages 357 - 374]

[6] [E. Tardos, J. Chen. (2017). Lecture material: Algorithmic Game Theory Lecture 29. Cornell University.]