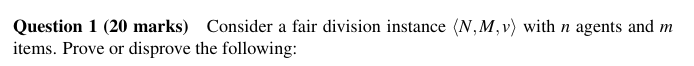
**Fair Allocations COMP4418–Assignment 3**

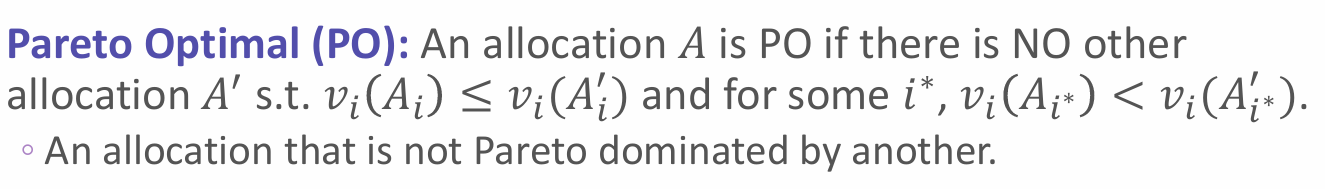
Jiayang Jiang z5319476

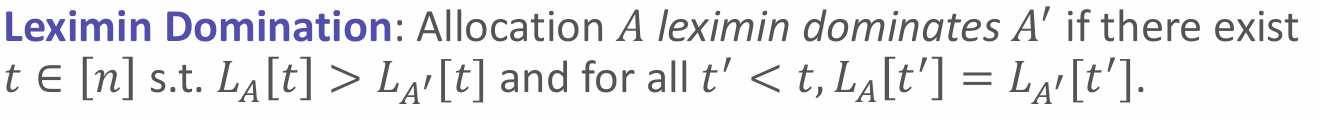


1. **(5 marks) Any Pareto Optimal allocation must also be Leximin Optimal.**

Answer: **Disprove**. Not every Pareto optimal allocation is Leximin optimal.

According to the definition of PO and Leximin Optimal:

****

****

****

Pareto optimality and Leximin optimality are related but distinct concepts. An allocation is Pareto optimal (PO) if there is no other allocation that can make at least one agent better off without making any other agent worse off. An allocation is Leximin optimal if it maximizes the minimal utility among all agents, and subject to that, maximizes the next minimal utility, and so on.

While every Leximin optimal allocation is Pareto optimal, the converse is not necessarily true. A Pareto optimal allocation may not be Leximin optimal because it might not maximize the minimal utility.

**Counterexample:**

Consider an instance with three agents and three items. Their valuations are:



Two allocations:

1. Allocation :
   * ​ gets
   * gets
   * gets
   * Utilities: , sorted:
2. Allocation :
   * ​ gets
   * gets
   * gets
   * Utilities: , sorted:

Both allocations are Pareto optimal, but allocation leximin dominates because it has a higher minimal utility (2 vs. 1). Therefore, not all Pareto optimal allocations are Leximin optimal.

1. **(5 marks) Given any two allocations, one must pareto dominate the other.**

Answer: **Disprove**. There exist allocations where neither Pareto dominates the other.

Pareto dominance is a partial order, meaning that for some pairs of allocations, neither allocation Pareto dominates the other. To illustrate this, let's consider two agents and two allocations with utilities and .

**Allocation A:** utilities

**Allocation B:** utilities

**Analyzing Pareto Dominance:**

* From Allocation to Allocation :
  + 's utility decreases from 2 to 1 (worse off).
  + 's utility increases from 3 to 4 (better off).
  + Since is worse off and is better off, moving from Allocation to Allocation is not a Pareto improvement.
* From Allocation to Allocation :
  + 's utility increases from 1 to 2 (better off).
  + 's utility decreases from 4 to 3 (worse off).
  + Since is worse off and is better off, moving from Allocation to Allocation is not a Pareto improvement.

**Conclusion:**

* Neither allocation Pareto dominates the other:
  + Allocation does not Pareto dominate Allocation .
  + Allocation does not Pareto dominate Allocation .

Therefore, the statement is **false**.

1. **(5 marks) For , any allocation that satisfies PROP is also EF.**

Answer: **Prove**. In the case of two agents, proportionality implies envy-freeness.

For two agents with additive valuations, an allocation is **proportional (PROP)** if each agent receives at least half of the total value of all items according to their own valuation. An allocation is **envy-free (EF)** if no agent prefers the bundle of another agent over their own.

Suppose an allocation is proportional, so each agent gets at least ​ of their total valuation of all items.

Assume, for contradiction, that agent ​ envies agent :

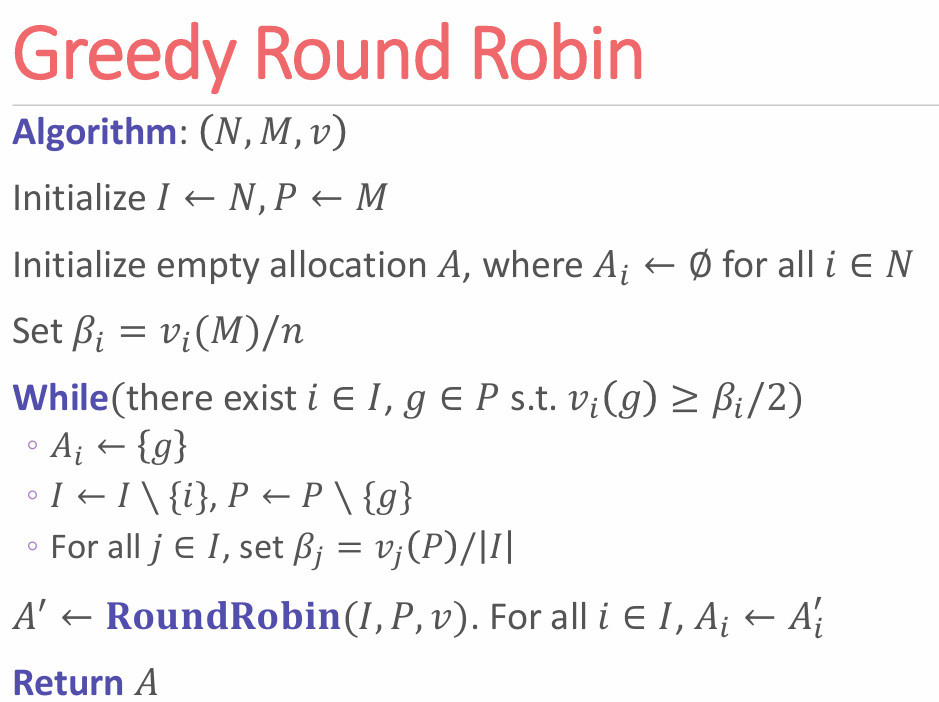
* (by proportionality)
* Total value: + =

Substituting:

This contradicts . Therefore, agent does not envy . Similarly, using the same reasoning process, we can conclude that agent also does not envy agent . Hence, the allocation is envy-free.

1. **(5 marks) Greedy round robin algorithm will return an EF1 allocation.**

Answer: **Disprove**. Greedy round robin algorithm will not always return an EF1 allocation.

****

**Counterexample:**

Consider an instance with two agents and four items. Their valuations are:



**Then, =**  = ,

**Now**  can get since

We removed and , according to the greedy round robin algorithm, will get .

**.**

An allocation is **EF1** if any agent envies another agent, but this envy can be eliminated by removing at most one item from the envied agent's allocation.

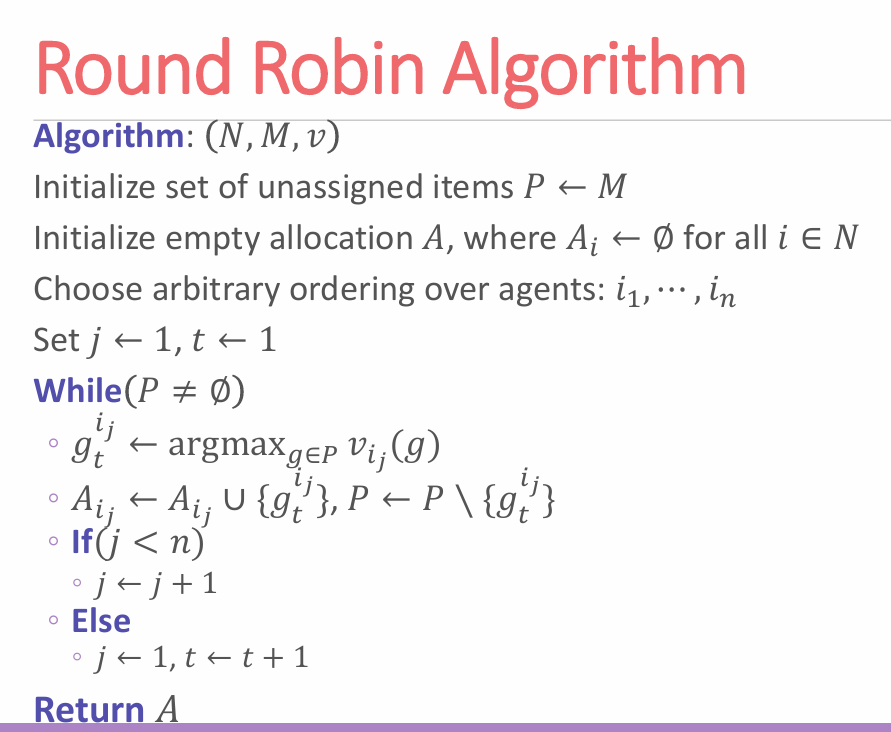
We can see that, after whichever item is removed from , ​ still envies .

Therefore, greedy round robin algorithm will not always return an **EF1 allocation**.

A table with numbers and text

Description automatically generated





According to the Round Robin Algorithm, the agent will always choose the item which is available with largest value.

For , has the largest value (5) but , so is not the first agent.

For , has the largest value (9) but , so is not the first agent.

For , has the largest value (10) but , so is not the first agent.

For , and has the largest value (10) and , **so is the first agent.**

**Since is the first agent, item** **is removed:**

For , and has the largest value (4) but , so is not the second agent.

For , has the largest value (10) but , so is not the second agent.

So,  **is the second agent.**

**Now, and are removed:**

For , has the largest value (4) but , so is not the third agent.

So,  **is the third agent.**

Therefore, the ordering is **.**

****

**We use the same ordering:**

For , we can choose instead of , then will choose (largest value).

will choose and will choose .

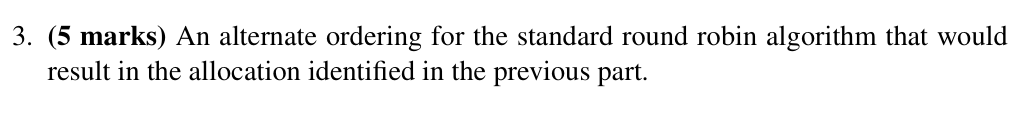
**Second round:**

will choose , then will choose will choose and will choose .

For allocation

For allocation

**Therefore, every value in**  is equal or greater than that in allocation , pareto dominates .

****

To result in the allocation identified in the previous part, we can use this ordering:

**.**

**Process:**

For , we can choose , then will choose (largest value).

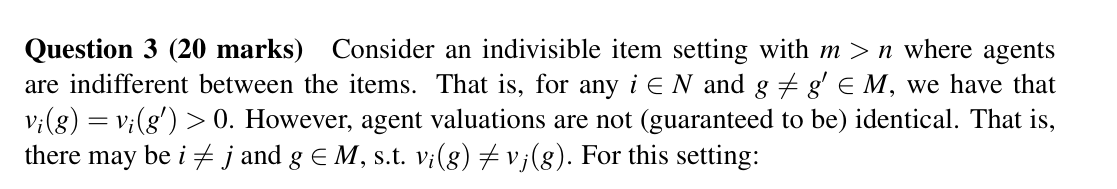
will choose and will choose .

**Second round:**

will choose , then will choose will choose and will choose .

**Compare the value:**

Therefore, the allocation generated by ordering pareto dominates .



**1. (5 marks) Show that an MMS allocation always exists.**

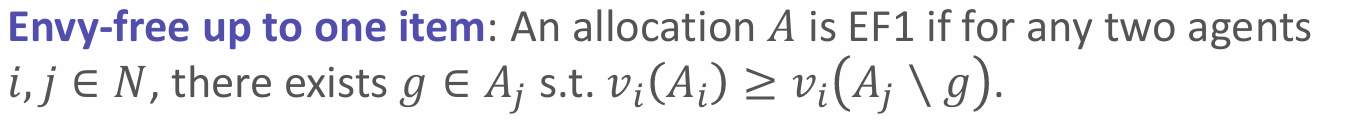


Yes, an MMS allocation always exists in this setting. Since agents value all items equally, each agent maximin share is . To construct an MMS allocation, we can proceed as follows:

* Divide the items into bundles, each containing at least items.
* Assign one bundle to each agent arbitrarily.

Since , each agent receives at least one item, ensuring that their utility is at least . Any remaining items can be distributed arbitrarily, possibly increasing agents' utilities above their MMS values . Thus, an MMS allocation exists.

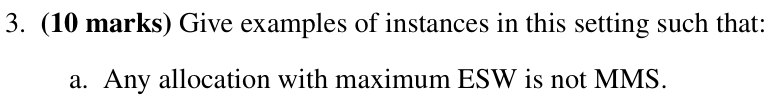
**2. (5 marks) Show that an EF1 allocation will always be MMS.**



Yes, in this setting, **an EF1 allocation will always be MMS**. In an EF1 allocation, for any pair of agents and , agent does not envy agent after removing any single item from bundle. Since agents value all items equally, their utilities are proportional to the number of items they receive.

For agent not to envy agent up to one item, it must be that:

This condition ensures that no agent has significantly more items than another **(differing by at most one item)**. As a result, each agent receives at least items, ensuring they get at least their MMS value . Therefore, an EF1 allocation will always be MMS allocation in this setting.

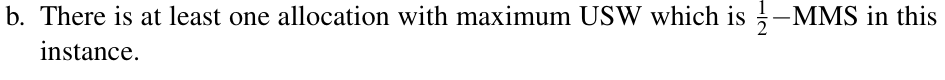


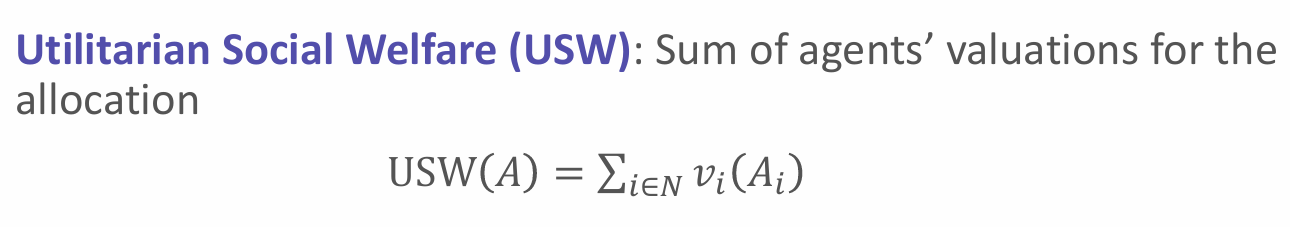
A math equations and symbols

Description automatically generated with medium confidenceYes, here is an example:

* **Instance:**
  + items, agents.
  + Agent 1 values each item at .
  + Agent 2 values each item at .
* **Agents' MMS values:**
  + .
  + .
* **Allocation maximizing ESW:**
  + Allocate 3 items to Agent 1: .
  + Allocate 1 item to Agent 2: .

In this allocation, Agent 2 receives less than her MMS value . Therefore, this allocation that maximizes ESW results in Agent 2 not receiving her MMS, as giving more items to Agent 1 increases the minimum utility but deprives Agent 2 of her MMS.





We assume that we have 2 agents , and 3 items.

**Valuations:**

* assume both Agent 1 and Agent 2 value each item equally at . That is, all agents value all items the same.

Since all items have the same value, we can allocate 2 items to and 1 item to to achieve **maximum USW**, which is .

**Utility of Agent 1:**

Agent 1's utility is greater than half of their MMS value.

**Utility of Agent 2:**

Agent 2's utility is equal to half of their MMS value.

**Conclusion:** In this allocation that maximizes USW, agent 1's utility is greater than half of their MMS value and agent 2's utility is exactly equal to half of their MMS value. Thus, it satisfies the requirement of the problem: there exists an allocation that maximizes USW, in which at least one agent's utility is equal to half of their MMS value.

A white paper with black text

Description automatically generated

**1. (10 marks) probabilistic serial (PS)**

**PS (Probabilistic Serial):**

|  |  |  |  |
| --- | --- | --- | --- |
| Agent 1 |  |  |  |
| Agent 2 |  |  |  |
| Agent 3 |  |  |  |

**PS () =**

**2. (10 marks) random serial dictator (RSD)**

A white background with black text

Description automatically generated

**Possible Orders (Permutations):**

There are possible orders:

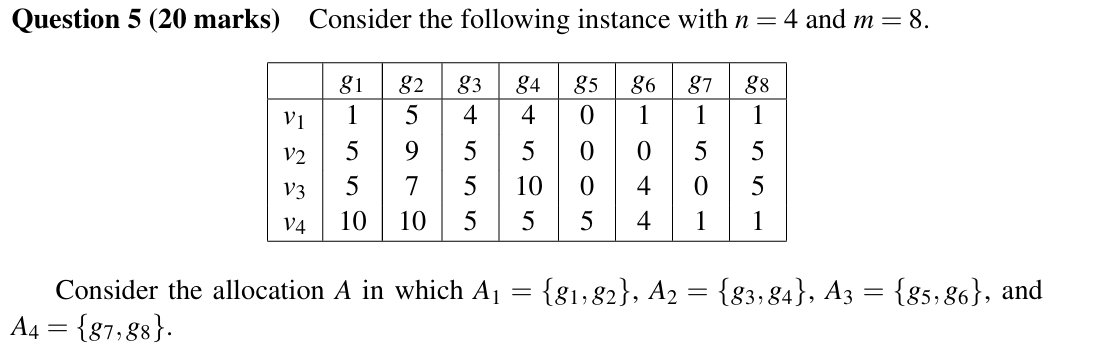
1. (1,2,3)
2. (1,3,2)
3. (2,1,3)
4. (2,3,1)
5. (3,1,2)
6. (3,2,1)

**Allocations for Each Order:**

1. **Order (1,2,3):**
   * Agent 1 picks ​.
   * Agent 2 picks ​.
   * Agent 3 picks .
2. **Order (1,3,2):**
   * Agent 1 picks ​.
   * Agent 3 picks ​.
   * Agent 2 picks ​.
3. **Order (2,1,3):**
   * Agent 2 picks ​.
   * Agent 1 picks ​.
   * Agent 3 picks ​.
4. **Order (2,3,1):**
   * Agent 2 picks ​.
   * Agent 3 picks ​.
   * Agent 1 picks ​.
5. **Order (3,1,2):**
   * Agent 3 picks ​.
   * Agent 1 picks ​.
   * Agent 2 picks ​.
6. **Order (3,2,1):**
   * Agent 3 picks g2​.
   * Agent 2 picks g1​.
   * Agent 1 picks g3​.

**Counting Allocations:**

* **Agent 1:**
  + Gets ​ in orders 1, 2, 5 (3 out of 6 times).
  + Gets ​ in order 3 (1 out of 6 times).
  + Gets in orders 4, 6 (2 out of 6 times).
* **Agent 2:**
  + Gets in orders 3, 4, 6 (3 out of 6 times).
  + Gets in order 1 (1 out of 6 times).
  + Gets in orders 2, 5 (2 out of 6 times).
* **Agent 3:**
  + Gets ​ in orders 2, 4, 5, 6 (4 out of 6 times).
  + Gets ​ in orders 1, 3 (2 out of 6 times).



**1. (5 marks) Prove or disprove that** **the allocation is envy-free.**

From the graph and the Allocation, we know that

and , so 4 is envious of 1.

To make 4 be made non-envious, we need give .

However, .

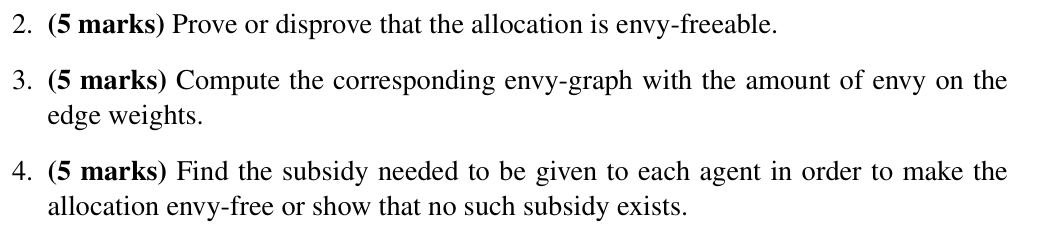
And ,

If we give , then 1 will be envious of 4.

A close-up of a white background

Description automatically generated

Additionally, this allocation doesn’t follow the theorem, **so** **the allocation is not envy-free.**



**The Envy-graph:**

**A drawing of a diagram on a lined paper

Description automatically generated**

A white paper with black text

Description automatically generated

According to this theorem, since there are many positive weight cycles in the envy-graph,

**The allocation is not envy-freeable.**

To eliminate envy using subsidies, we need to adjust each agent's utility so that no agent envies another after subsidies are applied.

Let:

* ​ be the subsidy given to agent.
* Adjusted utility for agent : .

**Constraints:**

For every pair of agents and, the following must hold:

This ensures agent does not envy agent.

**Set Up the Inequalities:**

**From Agent v₁:**

1. (Does not envy v₂)
2. (Does not envy v₃)
3. ​ (Does not envy v₄)

From Agent v₂:

1. (Does not envy v₁)
2. ​ (Does not envy v₃)
3. ​ (Does not envy v₄)

From Agent v₃:

1. ​ (Does not envy v₁)
2. (Does not envy v₂)
3. (Does not envy v₄)

From Agent v₄:

1. (Does not envy v₁)
2. (Does not envy v₂)
3. (Does not envy v₃)

**Simplify the Inequalities**:

**Analyze the Inequalities:**

* From inequalities 1 and 4:
  + Adding both: , which is impossible.

**Conclusion:**

The system of inequalities is inconsistent. Therefore, **no such subsidy** exists that can make the allocation envy-free.