Markets spread knowledge to overcome computational complexity

Hassan Andrabi* Peter Bossaerts† Konstantinos Ioannidis†
Robert Woods† Nitin Yadav*

†University of Cambridge, Faculty of Economics
*University of Melbourne, Centre for Brain, Mind and Markets

June 25, 2024

Role of markets

1. Extensively studied
   - Resource allocation
   - Information aggregation
2. Relatively understudied (**??**)
   - Spreading knowledge of solutions to complex problems

?: "knowledge of the kind which by its nature cannot enter into statistics ... the method by which such knowledge can be made as widely available as possible is precisely the problem to which we have to find an answer"

Role of markets

1. Extensively studied
   - Resource allocation
   - Information aggregation
2. Relatively understudied (**??**)
   - Spreading knowledge of solutions to complex problems

?: "knowledge of the kind which by its nature cannot enter into statistics ... the method by which such knowledge can be made as widely available as possible is precisely the problem to which we have to find an answer"

Role of markets

1. Extensively studied
   - Resource allocation
   - Information aggregation
2. Relatively understudied (**??**)
   - Spreading knowledge of solutions to complex problems

?: "knowledge of the kind which by its nature cannot enter into statistics ... the method by which such knowledge can be made as widely available as possible is precisely the problem to which we have to find an answer"

Role of markets

1. Extensively studied
   - Resource allocation
   - Information aggregation
2. Relatively understudied (**??**)
   - Spreading knowledge of solutions to complex problems

?: "knowledge of the kind which by its nature cannot enter into statistics ... the method by which such knowledge can be made as widely available as possible is precisely the problem to which we have to find an answer"

Role of markets

1. Extensively studied
   - Resource allocation
   - Information aggregation
2. Relatively understudied (**??**)
   - Spreading knowledge of solutions to complex problems

> **?**: "knowledge of the kind which by its nature cannot enter into statistics ... the method by which such knowledge can be made as widely available as possible is precisely the problem to which we have to find an answer"

Steam engines in Cornwall (?)

- Improvements to the steam engine occurred at an unprecedented rate during the Industrial Revolution (1760-1840).
- A disproportionately high number of these improvements originated in Cornwall.
  - Most of these improvements were *not patented*.
  - Historians attribute this to a collaborative culture towards innovation, what they call 'collective innovation'.

Steam engines in Cornwall (?)

- Improvements to the steam engine occurred at an unprecedented rate during the Industrial Revolution (1760-1840).
- A disproportionately high number of these improvements originated in Cornwall.
  - Most of these improvements were *not patented*.
  - Historians attribute this to a collaborative culture towards innovation, what they call 'collective innovation'.

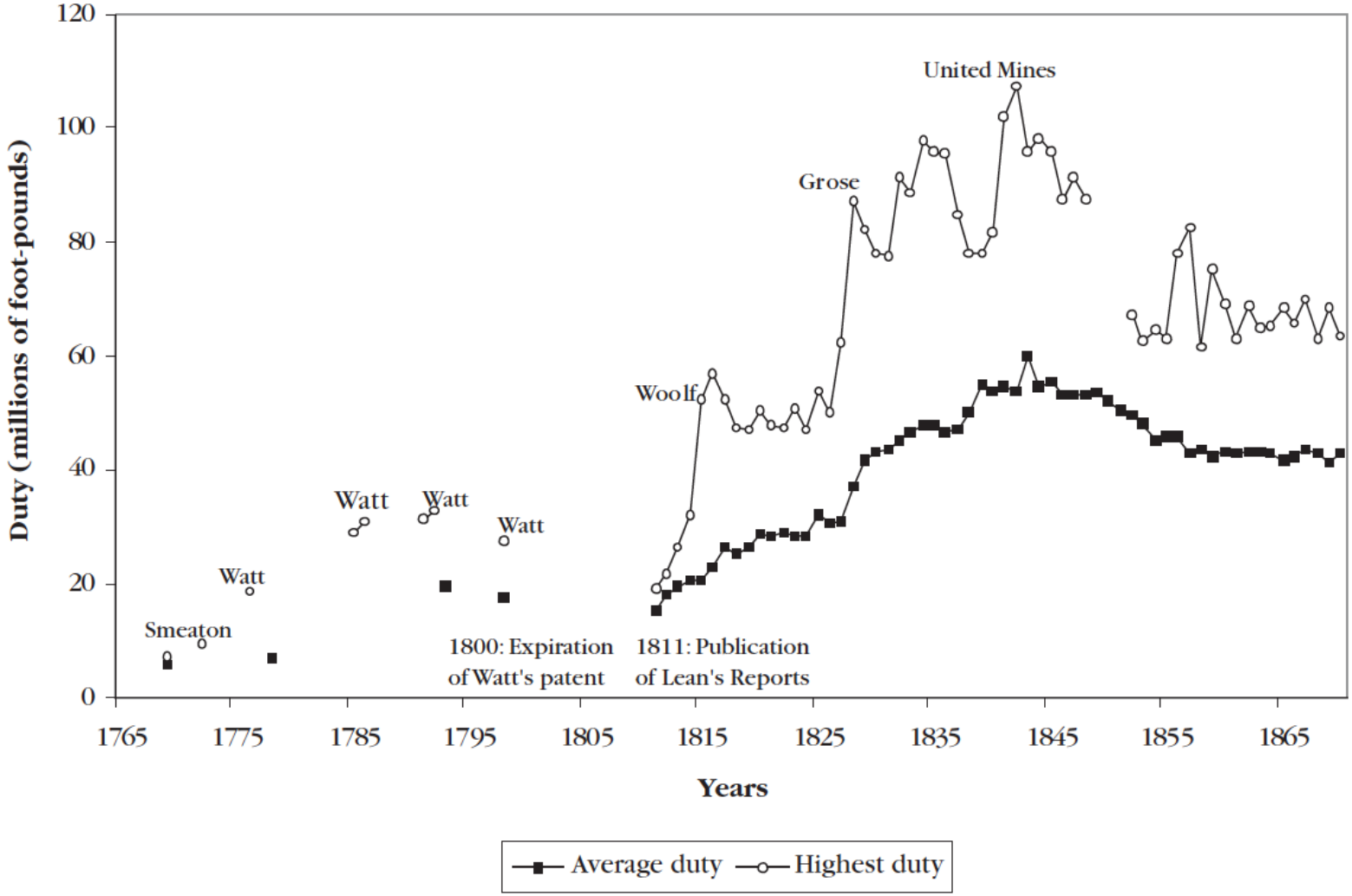
Steam engines in Cornwall (?)

- Improvements to the steam engine occurred at an unprecedented rate during the Industrial Revolution (1760-1840).
- A disproportionately high number of these improvements originated in Cornwall.
  - Most of these improvements were *not patented*.
  - Historians attribute this to a collaborative culture towards innovation, what they call 'collective innovation'.

Steam engines in Cornwall (?)

- Improvements to the steam engine occurred at an unprecedented rate during the Industrial Revolution (1760-1840).
- A disproportionately high number of these improvements originated in Cornwall.
  - Most of these improvements were *not patented*.
  - Historians attribute this to a collaborative culture towards innovation, what they call 'collective innovation'.

Efficiency of steam engines in Cornwall



How did this collaborative culture develop?

- Cornwall system
  - Lean's Reporter: mechanism to share information.
  - Cost-book incentive system: incentive to share information.
- Miners often held shares of many different mine ventures, as opposed to a single mine.
- Overall profitability of the district was more important than that of individual mines.
- $\uparrow$ average aggregate performance of the steam engine $\rightarrow$ $\uparrow$ overall profitability of the district.

How did this collaborative culture develop?

- Cornwall system
  - Lean's Reporter: mechanism to share information.
  - Cost-book incentive system: incentive to share information.
- Miners often held shares of many different mine ventures, as opposed to a single mine.
- Overall profitability of the district was more important than that of individual mines.
- $\uparrow$ average aggregate performance of the steam engine $\rightarrow$ $\uparrow$ overall profitability of the district.

How did this collaborative culture develop?

- Cornwall system
  - Lean's Reporter: mechanism to share information.
  - Cost-book incentive system: incentive to share information.
- Miners often held shares of many different mine ventures, as opposed to a single mine.
- Overall profitability of the district was more important than that of individual mines.
- $\uparrow$ average aggregate performance of the steam engine $\rightarrow$ $\uparrow$ overall profitability of the district.

How did this collaborative culture develop?

- Cornwall system
  - Lean's Reporter: mechanism to share information.
  - Cost-book incentive system: incentive to share information.
- Miners often held shares of many different mine ventures, as opposed to a single mine.
- Overall profitability of the district was more important than that of individual mines.
- $\uparrow$ average aggregate performance of the steam engine $\rightarrow$ $\uparrow$ overall profitability of the district.

How did this collaborative culture develop?

- Cornwall system
  - Lean's Reporter: mechanism to share information.
  - Cost-book incentive system: incentive to share information.
- Miners often held shares of many different mine ventures, as opposed to a single mine.
- Overall profitability of the district was more important than that of individual mines.
- $\uparrow$ average aggregate performance of the steam engine $\rightarrow$ $\uparrow$ overall profitability of the district.

How did this collaborative culture develop?

- Cornwall system
  - Lean's Reporter: mechanism to share information.
  - Cost-book incentive system: incentive to share information.
- Miners often held shares of many different mine ventures, as opposed to a single mine.
- Overall profitability of the district was more important than that of individual mines.
- $\uparrow$ average aggregate performance of the steam engine $\rightarrow$ $\uparrow$ overall profitability of the district.

Research question

- Goal of the experiment:
  - Provide experimental evidence that markets spread knowledge to overcome computational complexity
- Parallels
  1. Complex task: Steam engine efficiency ⇔ Knapsack problem
  2. Information sharing: Lean's reporter ⇔ Market order book
  3. Incentive system: Cost-book system ⇔ Asset payoff

Research question

- Goal of the experiment:
  - Provide experimental evidence that markets spread knowledge to overcome computational complexity
- Parallels
  1. Complex task: Steam engine efficiency ⇔ Knapsack problem
  2. Information sharing: Lean's reporter ⇔ Market order book
  3. Incentive system: Cost-book system ⇔ Asset payoff

Research question

- Goal of the experiment:
  - Provide experimental evidence that markets spread knowledge to overcome computational complexity
- Parallels
  1. Complex task: Steam engine efficiency $\Leftrightarrow$ Knapsack problem
  2. Information sharing: Lean's reporter $\Leftrightarrow$ Market order book
  3. Incentive system: Cost-book system $\Leftrightarrow$ Asset payoff

Research question

- Goal of the experiment:
  - Provide experimental evidence that markets spread knowledge to overcome computational complexity
- Parallels
  1. Complex task: Steam engine efficiency $\Leftrightarrow$ Knapsack problem
  2. Information sharing: Lean's reporter $\Leftrightarrow$ Market order book
  3. Incentive system: Cost-book system $\Leftrightarrow$ Asset payoff

Research question

- Goal of the experiment:
  - Provide experimental evidence that markets spread knowledge to overcome computational complexity
- Parallels
  1. Complex task: Steam engine efficiency $\Leftrightarrow$ Knapsack problem
  2. Information sharing: Lean's reporter $\Leftrightarrow$ Market order book
  3. Incentive system: Cost-book system $\Leftrightarrow$ Asset payoff

Research question

- Goal of the experiment:
  - Provide experimental evidence that markets spread knowledge to overcome computational complexity
- Parallels
  1. Complex task: Steam engine efficiency $\Leftrightarrow$ Knapsack problem
  2. Information sharing: Lean's reporter $\Leftrightarrow$ Market order book
  3. Incentive system: Cost-book system $\Leftrightarrow$ Asset payoff

Overview

- Participants solve instances of the Knapsack Problem while trading in an experimental marketplace.
- The traded security yields the maximal value discovered across all participants by the end of the period.
- Participants are compensated only for their **trading performance**:
  - *buy* at price below estimated payoff
  - *sell* at price above estimated payoff
- Logistics
  - 18 KP instances of varying complexity
  - 6 instances/periods per session
  - 5 sessions with 20 participants each
  - Run at the CEBEG lab of the University of Cambridge

Overview

- Participants solve instances of the Knapsack Problem while trading in an experimental marketplace.
- The traded security yields the maximal value discovered across all participants by the end of the period.
- Participants are compensated only for their **trading performance**:
  - *buy* at price below estimated payoff
  - *sell* at price above estimated payoff
- Logistics
  - 18 KP instances of varying complexity
  - 6 instances/periods per session
  - 5 sessions with 20 participants each
  - Run at the CEBEG lab of the University of Cambridge

Overview

- Participants solve instances of the Knapsack Problem while trading in an experimental marketplace.
- The traded security yields the maximal value discovered across all participants by the end of the period.
- Participants are compensated only for their **trading performance**:
  - *buy* at price below estimated payoff
  - *sell* at price above estimated payoff
- Logistics
  - 18 KP instances of varying complexity
  - 6 instances/periods per session
  - 5 sessions with 20 participants each
  - Run at the CEBEG lab of the University of Cambridge

Overview

- Participants solve instances of the Knapsack Problem while trading in an experimental marketplace.
- The traded security yields the maximal value discovered across all participants by the end of the period.
- Participants are compensated only for their **trading performance**:
  - *buy* at price below estimated payoff
  - *sell* at price above estimated payoff
- Logistics
  - 18 KP instances of varying complexity
  - 6 instances/periods per session
  - 5 sessions with 20 participants each
  - Run at the CEBEG lab of the University of Cambridge

Overview

- Participants solve instances of the Knapsack Problem while trading in an experimental marketplace.
- The traded security yields the maximal value discovered across all participants by the end of the period.
- Participants are compensated only for their **trading performance**:
  - *buy* at price below estimated payoff
  - *sell* at price above estimated payoff
- Logistics
  - 18 KP instances of varying complexity
  - 6 instances/periods per session
  - 5 sessions with 20 participants each
  - Run at the CEBEG lab of the University of Cambridge

Overview

- Participants solve instances of the Knapsack Problem while trading in an experimental marketplace.
- The traded security yields the maximal value discovered across all participants by the end of the period.
- Participants are compensated only for their **trading performance**:
  - *buy* at price below estimated payoff
  - *sell* at price above estimated payoff
- Logistics
  - 18 KP instances of varying complexity
  - 6 instances/periods per session
  - 5 sessions with 20 participants each
  - Run at the CEBEG lab of the University of Cambridge

Overview

- Participants solve instances of the Knapsack Problem while trading in an experimental marketplace.
- The traded security yields the maximal value discovered across all participants by the end of the period.
- Participants are compensated only for their **trading performance**:
  - *buy* at price below estimated payoff
  - *sell* at price above estimated payoff
- Logistics
  - 18 KP instances of varying complexity
  - 6 instances/periods per session
  - 5 sessions with 20 participants each
  - Run at the CEBEG lab of the University of Cambridge

Overview

- Participants solve instances of the Knapsack Problem while trading in an experimental marketplace.
- The traded security yields the maximal value discovered across all participants by the end of the period.
- Participants are compensated only for their **trading performance**:
  - *buy* at price below estimated payoff
  - *sell* at price above estimated payoff
- Logistics
  - 18 KP instances of varying complexity
  - 6 instances/periods per session
  - 5 sessions with 20 participants each
  - Run at the CEBEG lab of the University of Cambridge

Overview

- Participants solve instances of the Knapsack Problem while trading in an experimental marketplace.
- The traded security yields the maximal value discovered across all participants by the end of the period.
- Participants are compensated only for their **trading performance**:
  - *buy* at price below estimated payoff
  - *sell* at price above estimated payoff
- Logistics
  - 18 KP instances of varying complexity
  - 6 instances/periods per session
  - 5 sessions with 20 participants each
  - Run at the CEBEG lab of the University of Cambridge

Overview

- Participants solve instances of the Knapsack Problem while trading in an experimental marketplace.
- The traded security yields the maximal value discovered across all participants by the end of the period.
- Participants are compensated only for their **trading performance**:
  - *buy* at price below estimated payoff
  - *sell* at price above estimated payoff
- Logistics
  - 18 KP instances of varying complexity
  - 6 instances/periods per session
  - 5 sessions with 20 participants each
  - Run at the CEBEG lab of the University of Cambridge

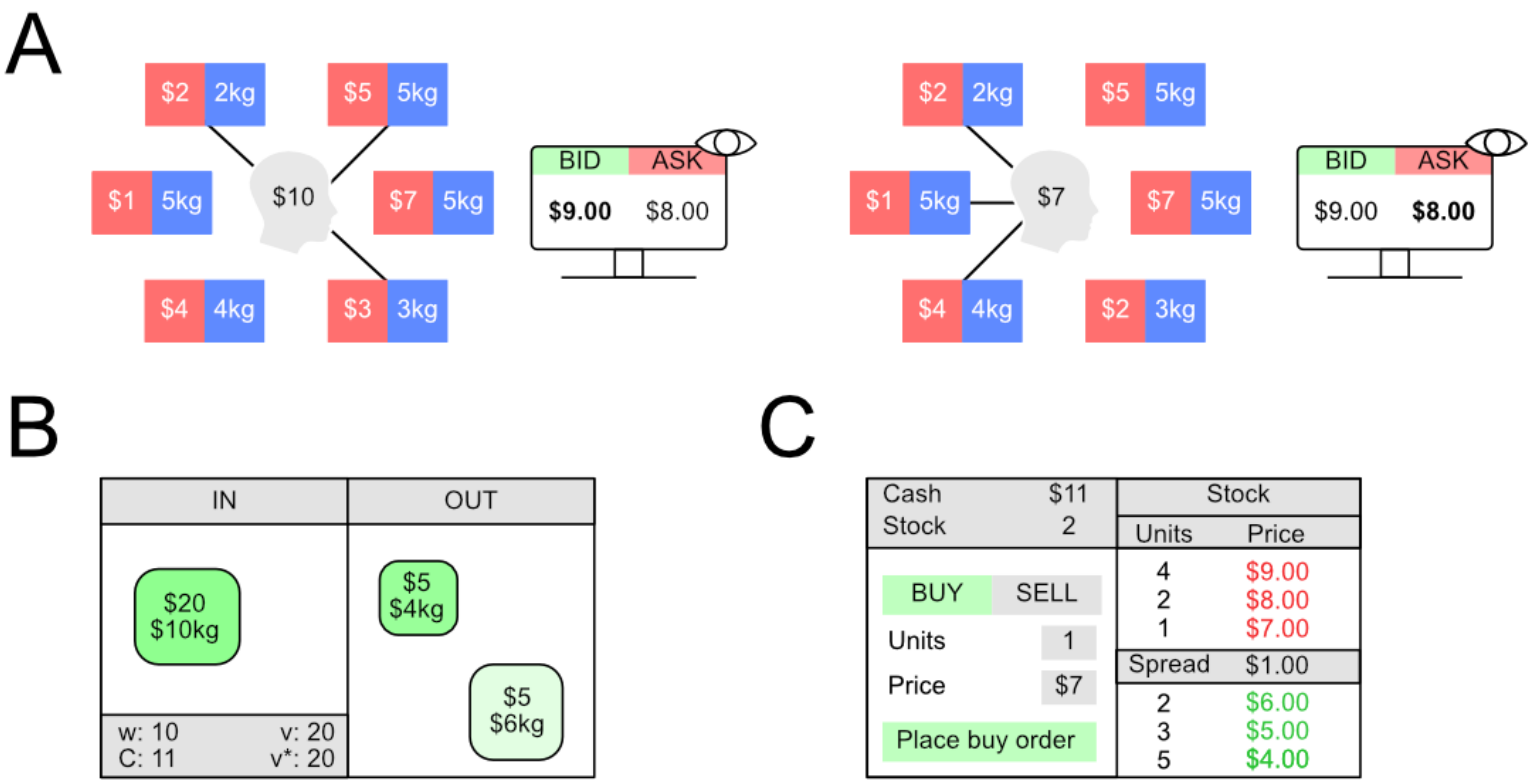
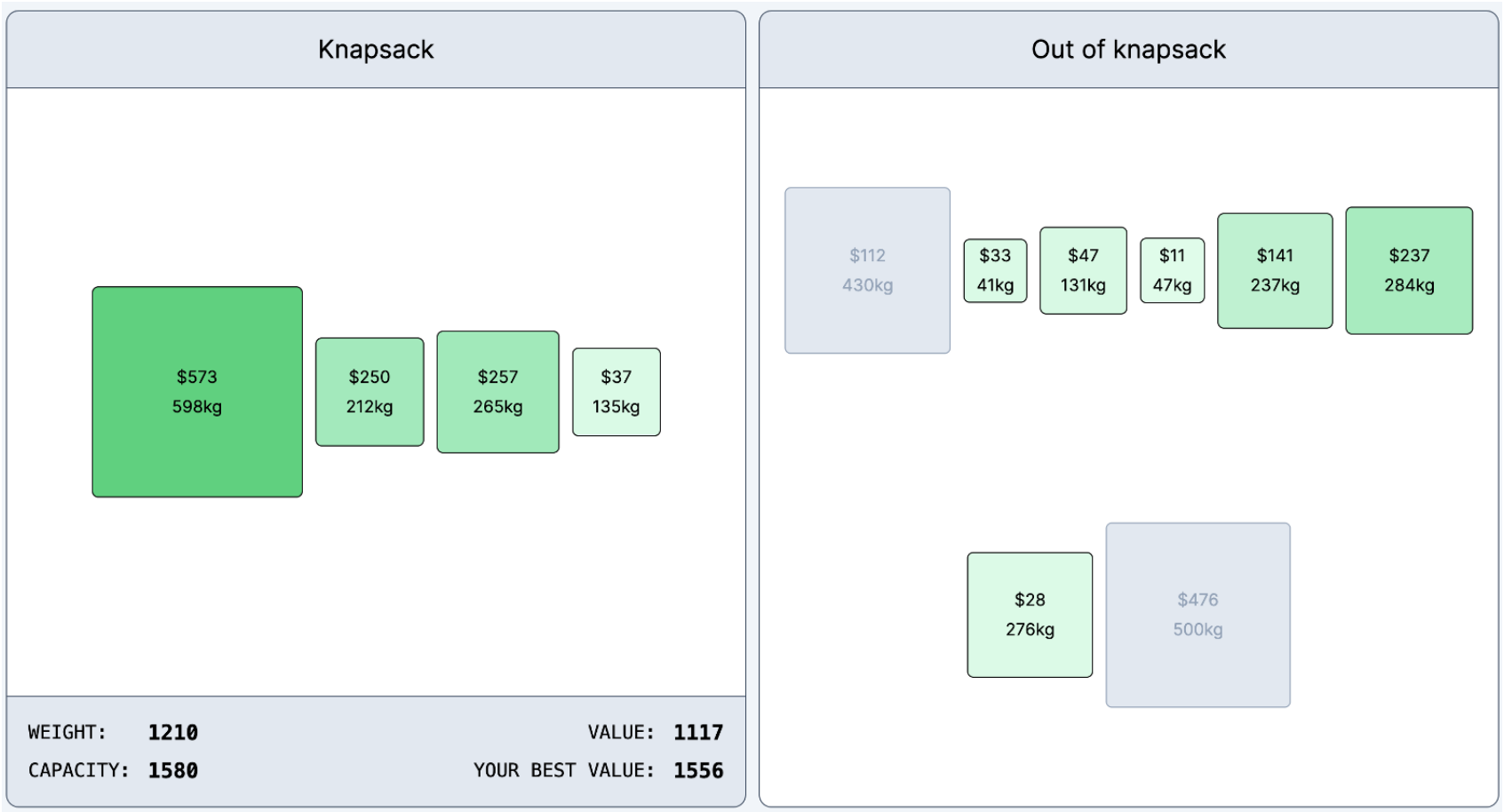
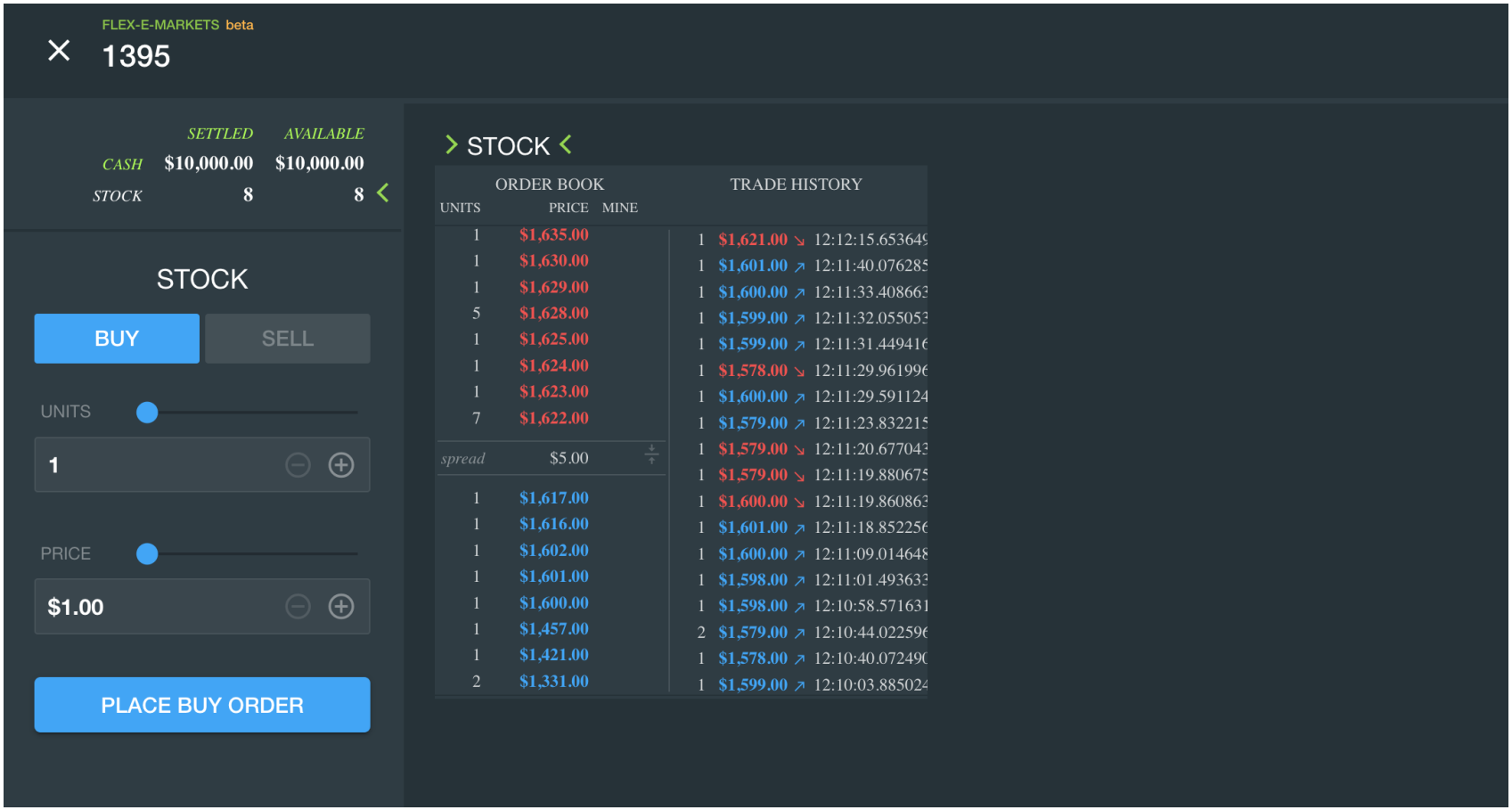


Figure: **(A)** Market experiment. **(B)** Knapsack problem interface. **(C)** Trading interface.

Knapsack Problem interface



Flex-E-Markets trading interface



Greedy and Sahni-k algorithms for Knapsack Problem

1. Sahni-k = 0 (greedy)
   - Calculate the ratio (value/weight) for each item.
   - Sort all the items in decreasing order of the ratio.
   - For every item in the sorted order:
     - If the weight of the current item is less than or equal to the remaining capacity, then add the value of that item into the result.
     - If not, stop. (improved version: skip current item, check next)
2. Sahni-k = 1
   - Add 1 item and then apply greedy algorithm.
3. Sahni-k = 2
   - Add 2 items and then apply greedy algorithm.

Greedy and Sahni-k algorithms for Knapsack Problem

1. Sahni-k = 0 (greedy)
   - Calculate the ratio (value/weight) for each item.
   - Sort all the items in decreasing order of the ratio.
   - For every item in the sorted order:
     - If the weight of the current item is less than or equal to the remaining capacity, then add the value of that item into the result.
     - If not, stop. (improved version: skip current item, check next)
2. Sahni-k = 1
   - Add 1 item and then apply greedy algorithm.
3. Sahni-k = 2
   - Add 2 items and then apply greedy algorithm.

Greedy and Sahni-k algorithms for Knapsack Problem

1. Sahni-k = 0 (greedy)
   - Calculate the ratio (value/weight) for each item.
   - Sort all the items in decreasing order of the ratio.
   - For every item in the sorted order:
     - If the weight of the current item is less than or equal to the remaining capacity, then add the value of that item into the result.
     - If not, stop. (improved version: skip current item, check next)
2. Sahni-k = 1
   - Add 1 item and then apply greedy algorithm.
3. Sahni-k = 2
   - Add 2 items and then apply greedy algorithm.

Greedy and Sahni-k algorithms for Knapsack Problem

1. Sahni-k = 0 (greedy)
   - Calculate the ratio (value/weight) for each item.
   - Sort all the items in decreasing order of the ratio.
   - For every item in the sorted order:
     - If the weight of the current item is less than or equal to the remaining capacity, then add the value of that item into the result.
     - If not, stop. (improved version: skip current item, check next)
2. Sahni-k = 1
   - Add 1 item and then apply greedy algorithm.
3. Sahni-k = 2
   - Add 2 items and then apply greedy algorithm.

Greedy and Sahni-k algorithms for Knapsack Problem

1. Sahni-k = 0 (greedy)
   - Calculate the ratio (value/weight) for each item.
   - Sort all the items in decreasing order of the ratio.
   - For every item in the sorted order:
     - If the weight of the current item is less than or equal to the remaining capacity, then add the value of that item into the result.
     - If not, stop. (improved version: skip current item, check next)
2. Sahni-k = 1
   - Add 1 item and then apply greedy algorithm.
3. Sahni-k = 2
   - Add 2 items and then apply greedy algorithm.

Greedy and Sahni-k algorithms for Knapsack Problem

1. Sahni-k = 0 (greedy)
   - Calculate the ratio (value/weight) for each item.
   - Sort all the items in decreasing order of the ratio.
   - For every item in the sorted order:
     - If the weight of the current item is less than or equal to the remaining capacity, then add the value of that item into the result.
     - If not, stop. (improved version: skip current item, check next)
2. Sahni-k = 1
   - Add 1 item and then apply greedy algorithm.
3. Sahni-k = 2
   - Add 2 items and then apply greedy algorithm.

Greedy and Sahni-k algorithms for Knapsack Problem

1. Sahni-k = 0 (greedy)
   - Calculate the ratio (value/weight) for each item.
   - Sort all the items in decreasing order of the ratio.
   - For every item in the sorted order:
     - If the weight of the current item is less than or equal to the remaining capacity, then add the value of that item into the result.
     - If not, stop. (improved version: skip current item, check next)
2. Sahni-k = 1
   - Add 1 item and then apply greedy algorithm.
3. Sahni-k = 2
   - Add 2 items and then apply greedy algorithm.

Greedy and Sahni-k algorithms for Knapsack Problem

1. Sahni-k = 0 (greedy)
   - Calculate the ratio (value/weight) for each item.
   - Sort all the items in decreasing order of the ratio.
   - For every item in the sorted order:
     - If the weight of the current item is less than or equal to the remaining capacity, then add the value of that item into the result.
     - If not, stop. (improved version: skip current item, check next)
2. Sahni-k = 1
   - Add 1 item and then apply greedy algorithm.
3. Sahni-k = 2
   - Add 2 items and then apply greedy algorithm.

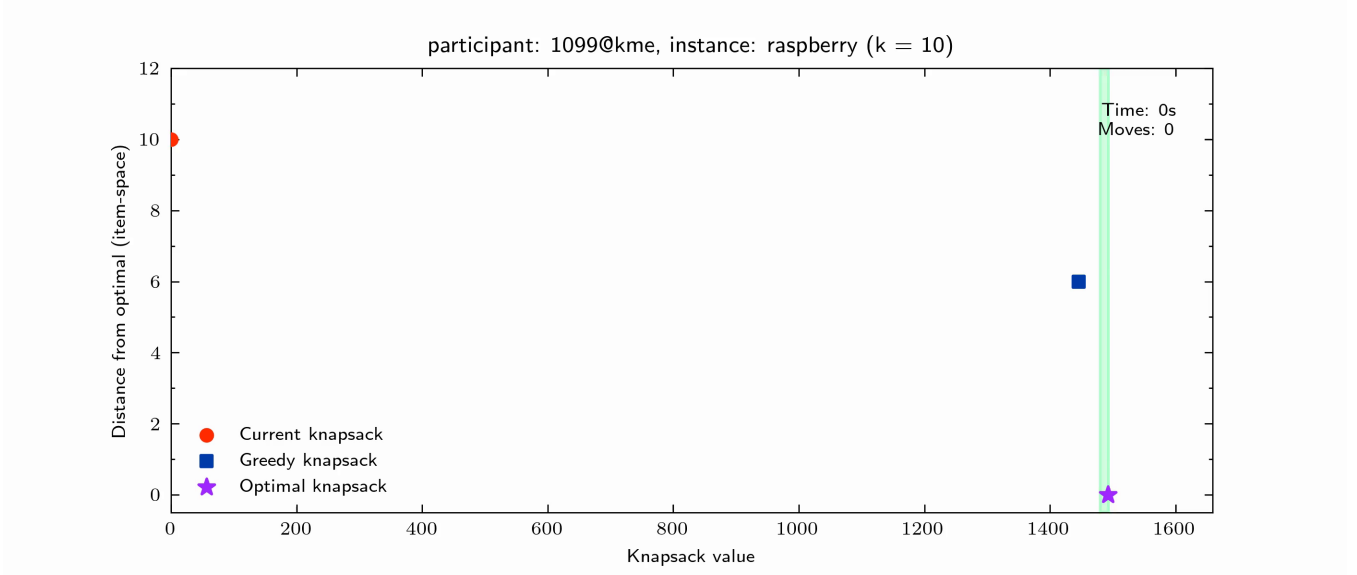
Greedy and Sahni-k algorithms for Knapsack Problem

1. Sahni-k = 0 (greedy)
   - Calculate the ratio (value/weight) for each item.
   - Sort all the items in decreasing order of the ratio.
   - For every item in the sorted order:
     - If the weight of the current item is less than or equal to the remaining capacity, then add the value of that item into the result.
     - If not, stop. (improved version: skip current item, check next)
2. Sahni-k = 1
   - Add 1 item and then apply greedy algorithm.
3. Sahni-k = 2
   - Add 2 items and then apply greedy algorithm.

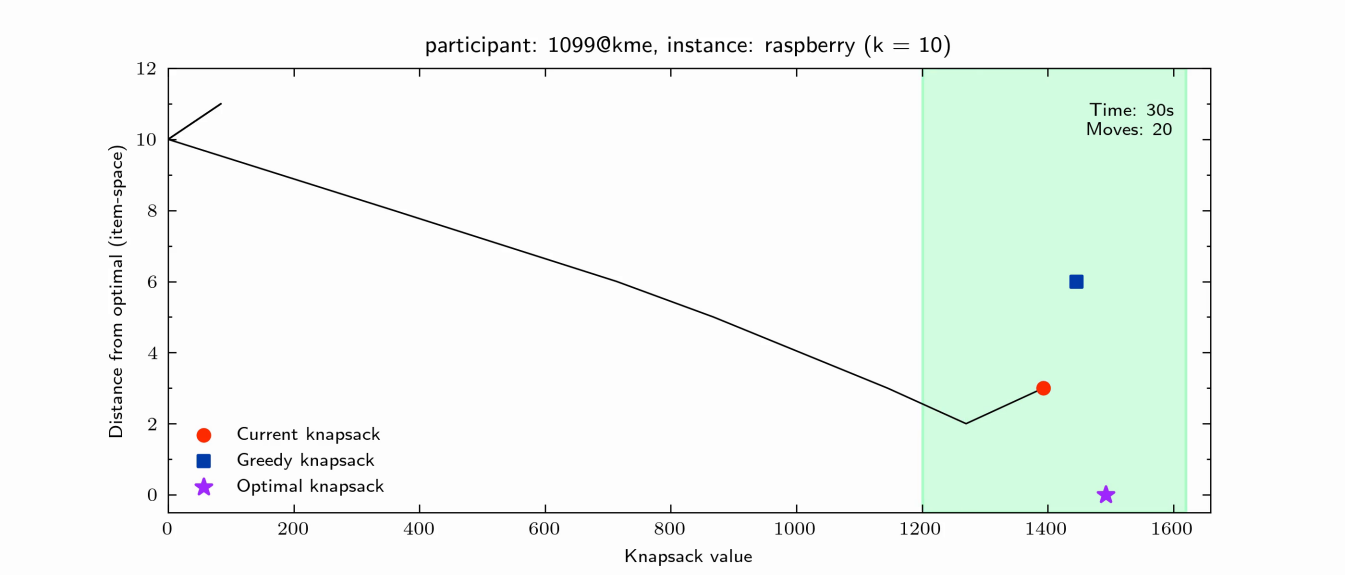
Greedy and Sahni-k algorithms for Knapsack Problem

1. Sahni-k = 0 (greedy)
   - Calculate the ratio (value/weight) for each item.
   - Sort all the items in decreasing order of the ratio.
   - For every item in the sorted order:
     - If the weight of the current item is less than or equal to the remaining capacity, then add the value of that item into the result.
     - If not, stop. (improved version: skip current item, check next)
2. Sahni-k = 1
   - Add 1 item and then apply greedy algorithm.
3. Sahni-k = 2
   - Add 2 items and then apply greedy algorithm.

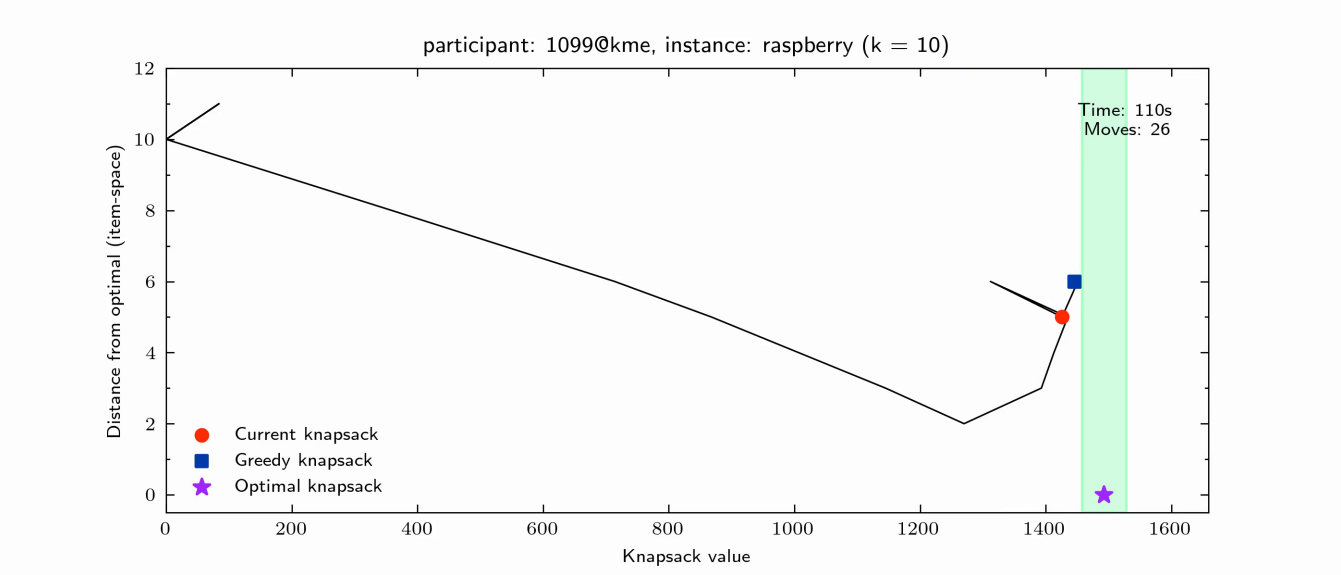
Being greedy



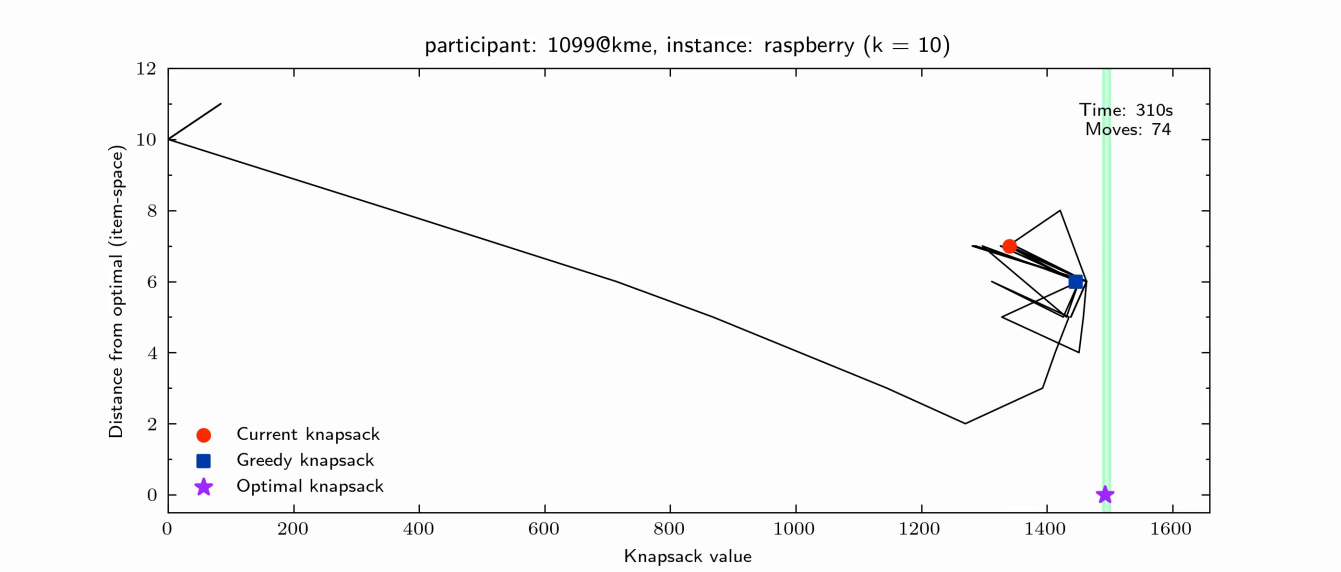
Being greedy



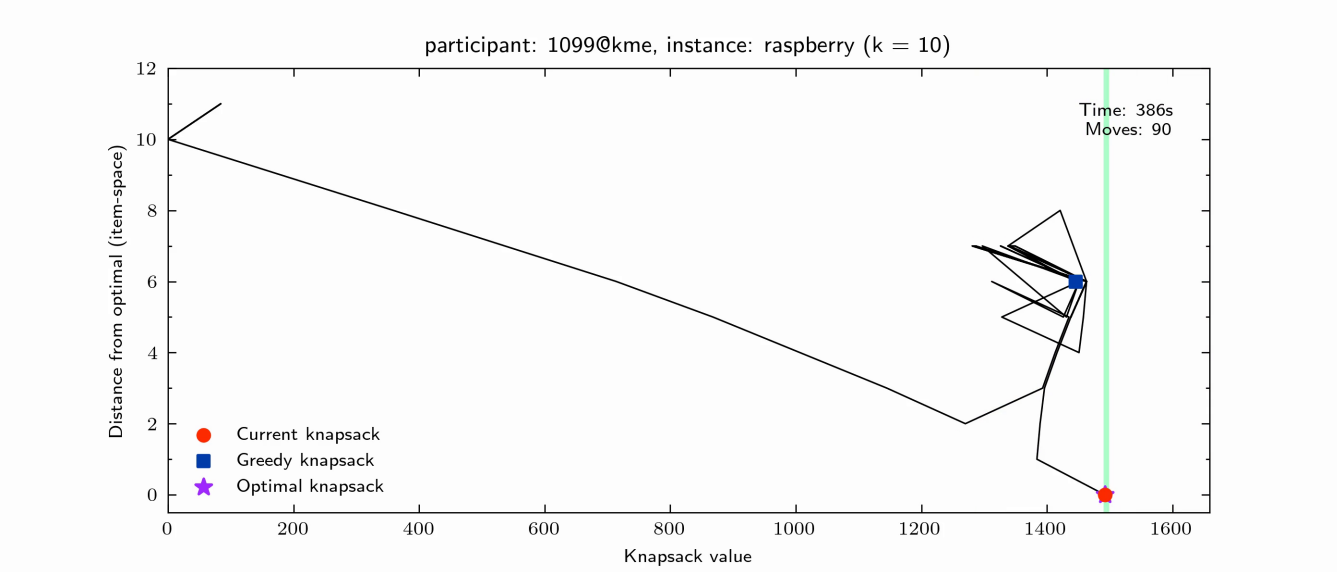
Being greedy



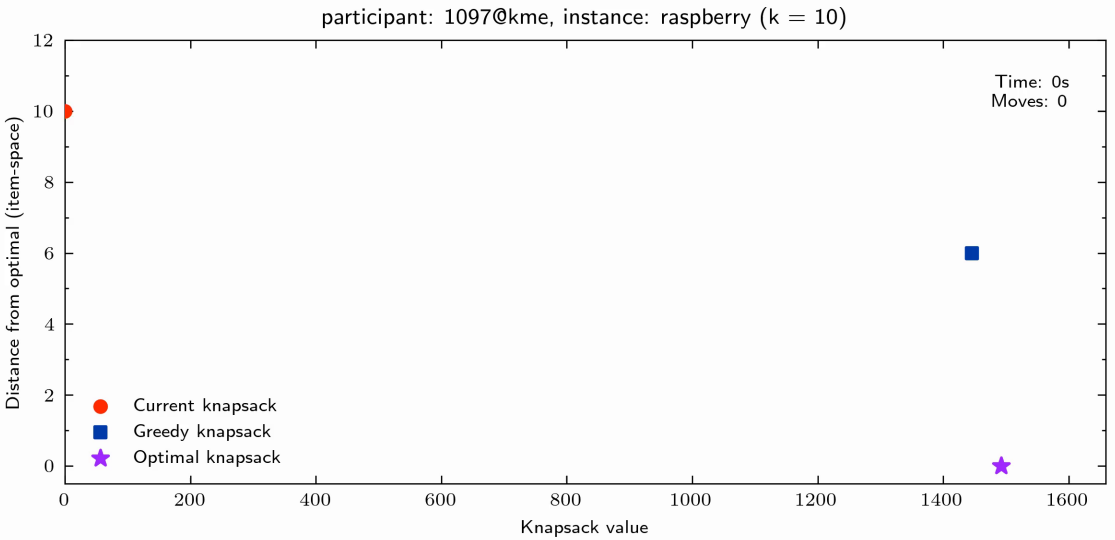
Being greedy



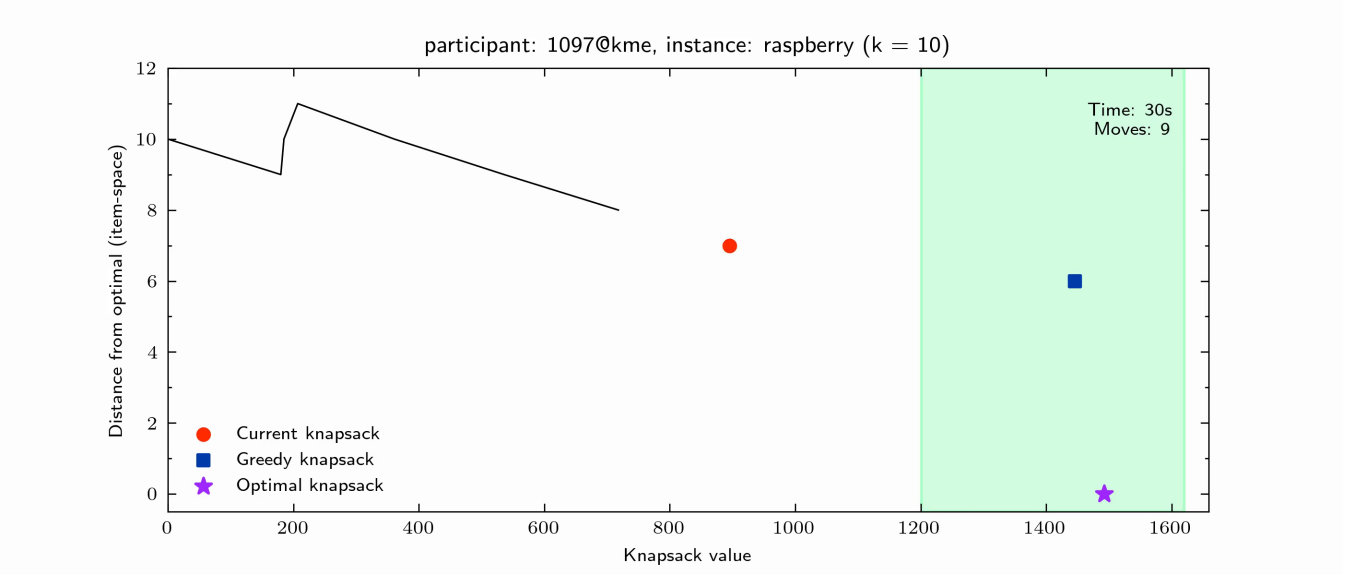
Being greedy



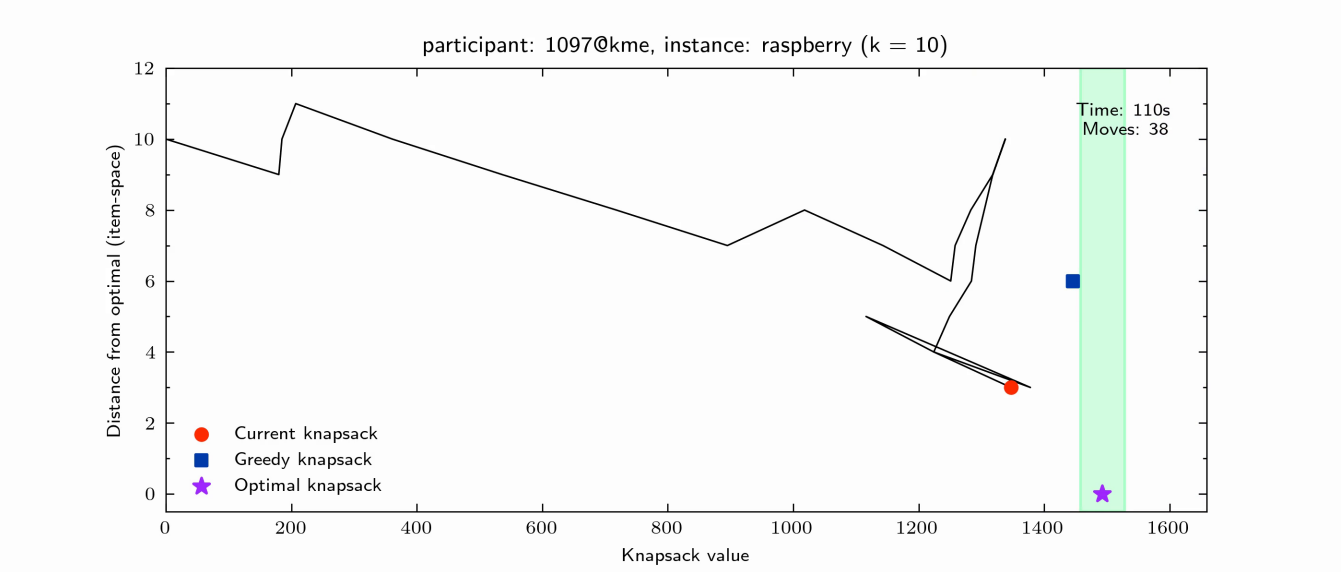
Being stuck


participant: 1097@kme, instance: raspberry (k = 10)
Time: 0s
Moves: 0
Distance from optimal (item-space)
Knapsack value
Current knapsack
Greedy knapsack
Optimal knapsack
0
2
4
6
8
10
12
0
200
400
600
800
1000
1200
1400
1600

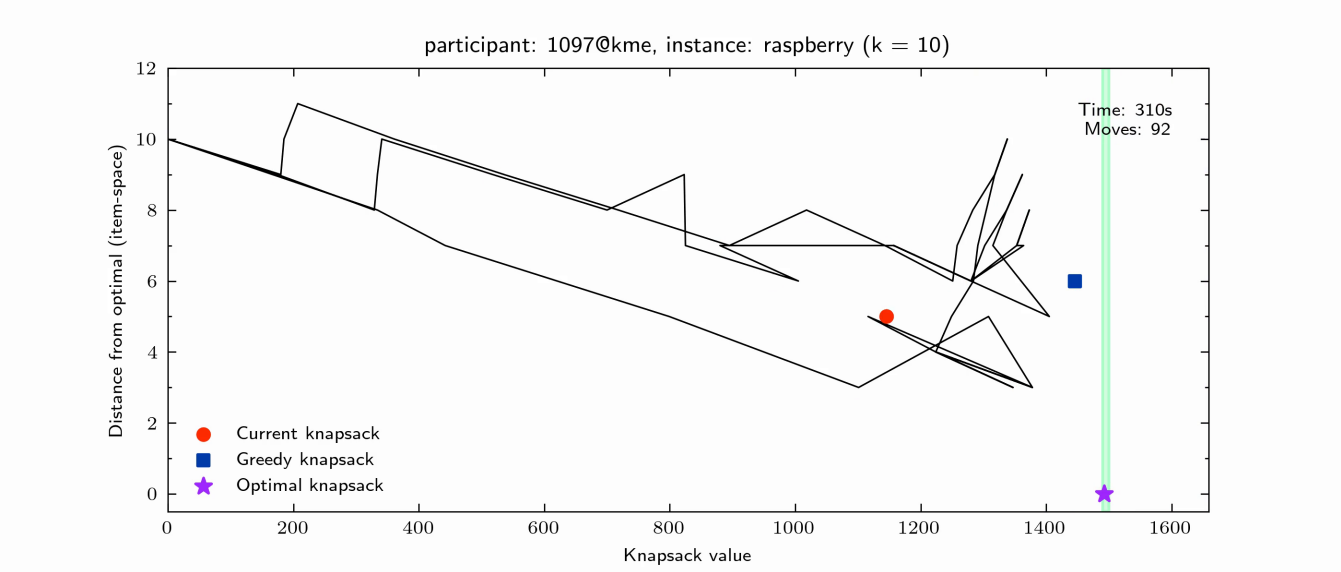
Being stuck


participant: 1097@kme, instance: raspberry (k = 10)
Distance from optimal (item-space)
Knapsack value
Time: 30s
Moves: 9
Current knapsack
Greedy knapsack
Optimal knapsack
0
2
4
6
8
10
12
0
200
400
600
800
1000
1200
1400
1600

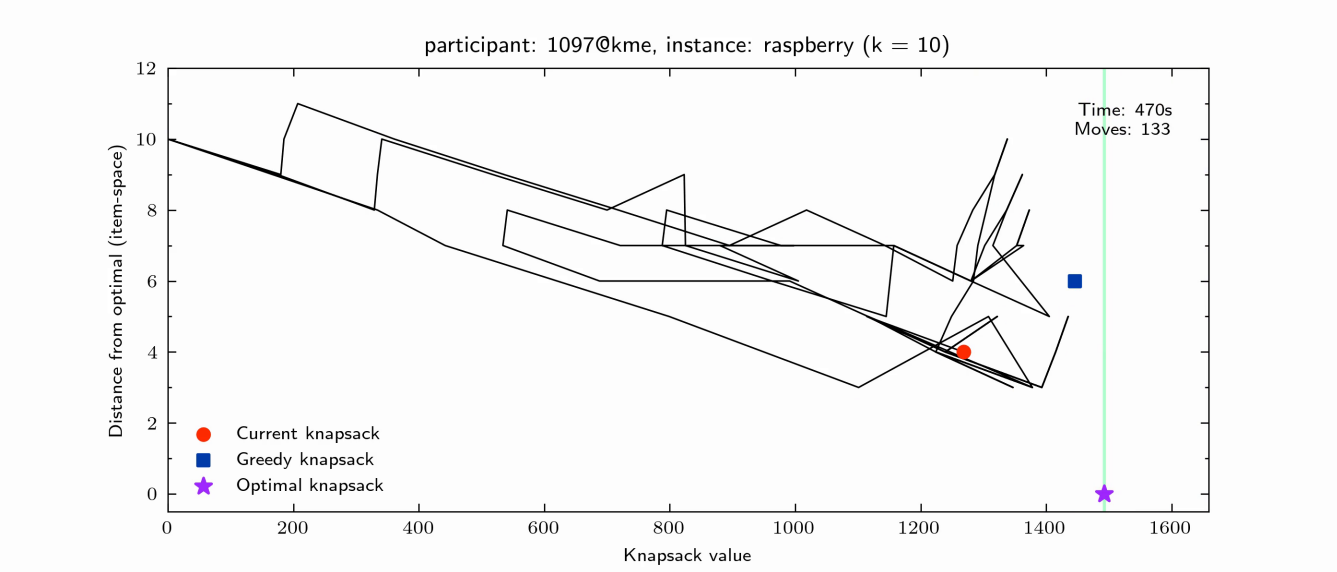
Being stuck


participant: 1097@kme, instance: raspberry (k = 10)
Distance from optimal (item-space)
Knapsack value
Time: 110s
Moves: 38
Current knapsack
Greedy knapsack
Optimal knapsack
0
2
4
6
8
10
12
0
200
400
600
800
1000
1200
1400
1600

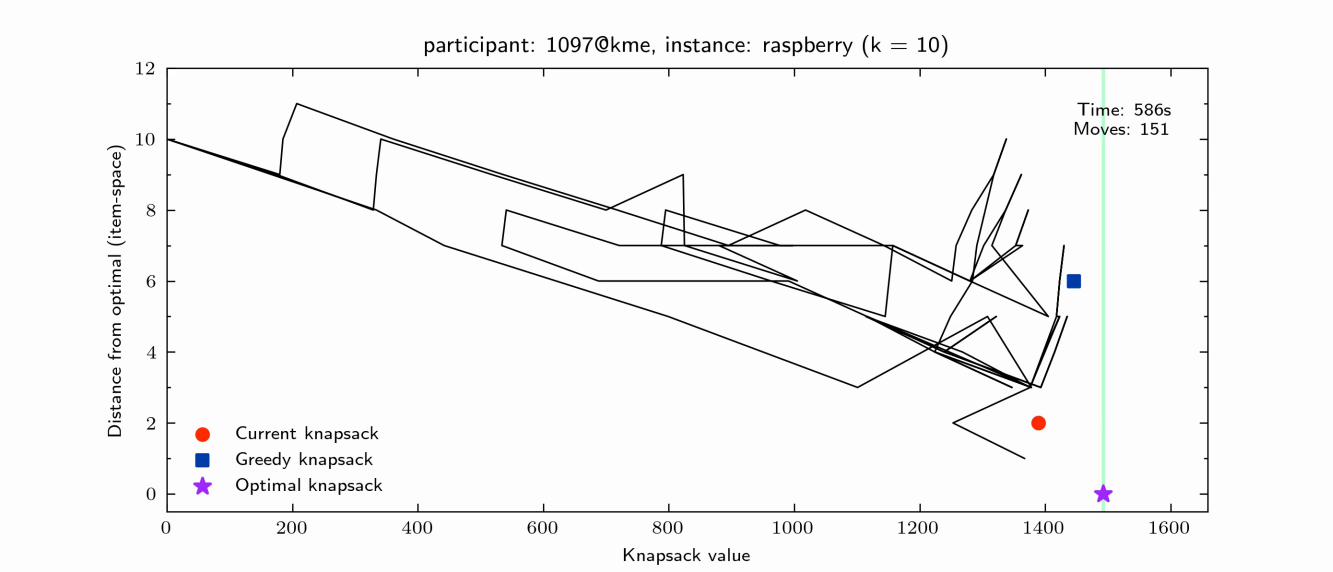
Being stuck


participant: 1097@kme, instance: raspberry (k = 10)
Time: 310s
Moves: 92
Distance from optimal (item-space)
Knapsack value
Current knapsack
Greedy knapsack
Optimal knapsack
0
2
4
6
8
10
12
0
200
400
600
800
1000
1200
1400
1600

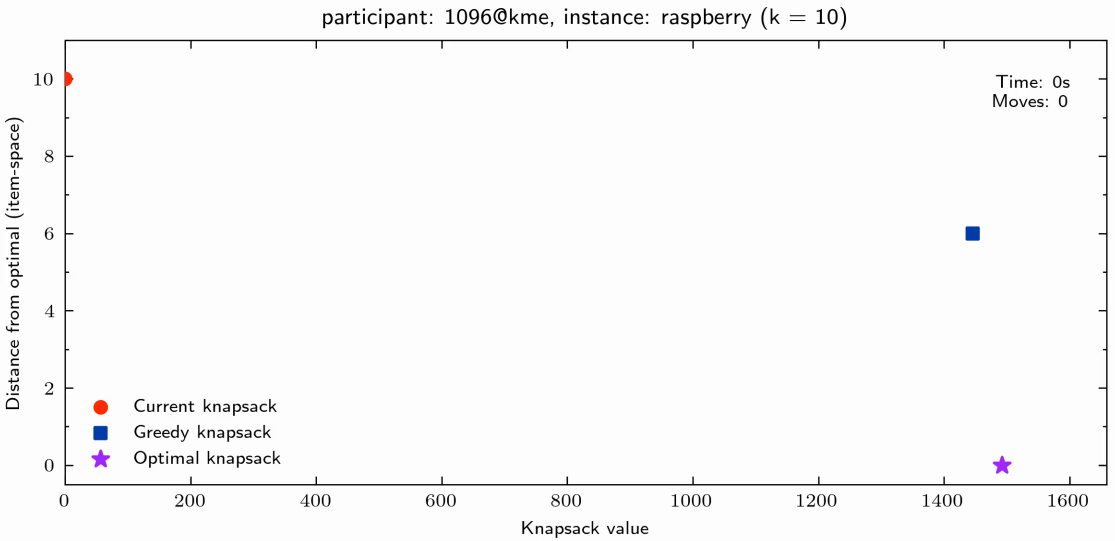
Being stuck


participant: 1097@kme, instance: raspberry (k = 10)
Time: 470s
Moves: 133
Current knapsack
Greedy knapsack
Optimal knapsack
Distance from optimal (item-space)
Knapsack value
0
2
4
6
8
10
12
0
200
400
600
800
1000
1200
1400
1600

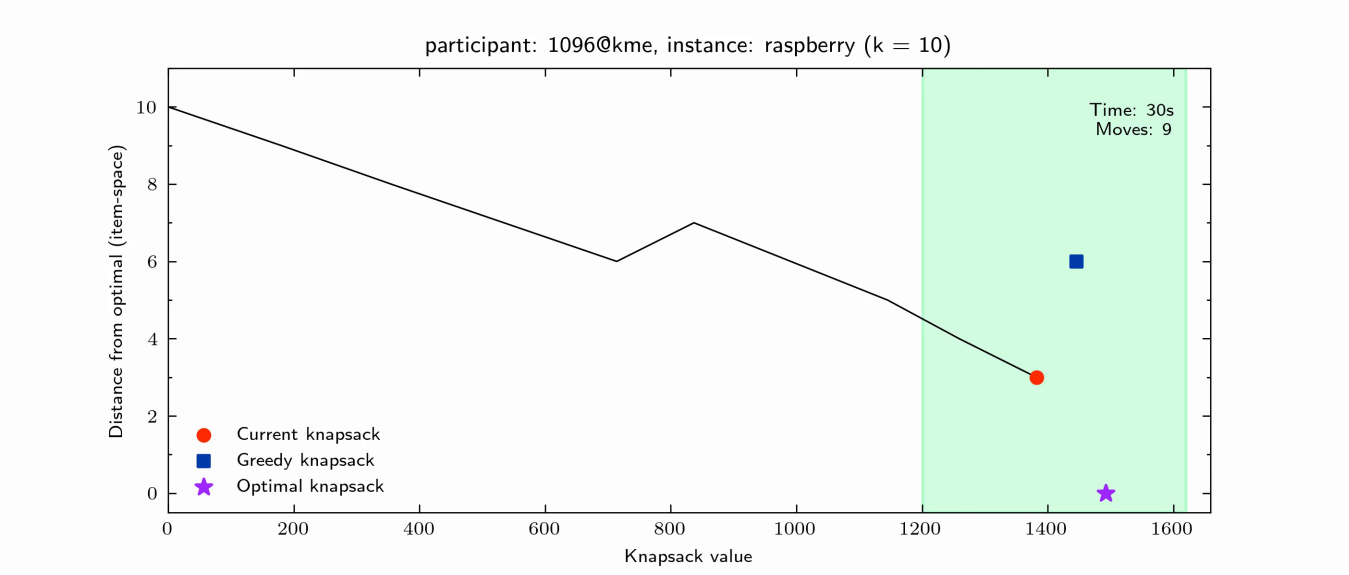
Being stuck


participant: 1097@kme, instance: raspberry (k = 10)
Time: 586s
Moves: 151
Current knapsack
Greedy knapsack
Optimal knapsack
Distance from optimal (item-space)
Knapsack value
0
2
4
6
8
10
12
0
200
400
600
800
1000
1200
1400
1600

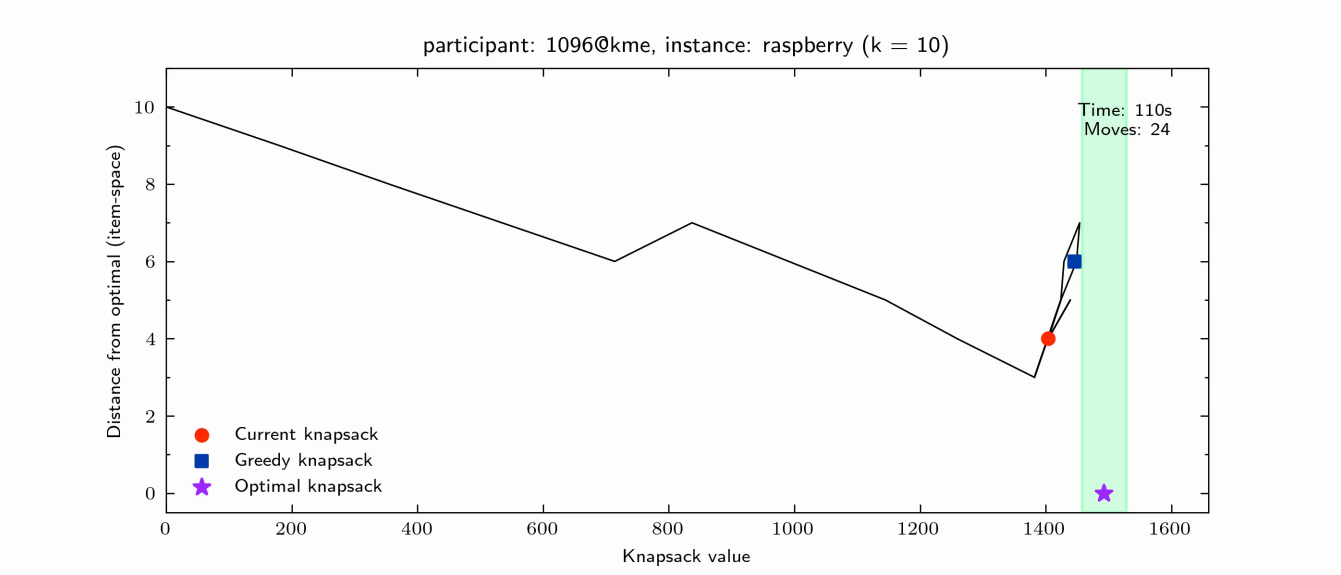
Being unaware



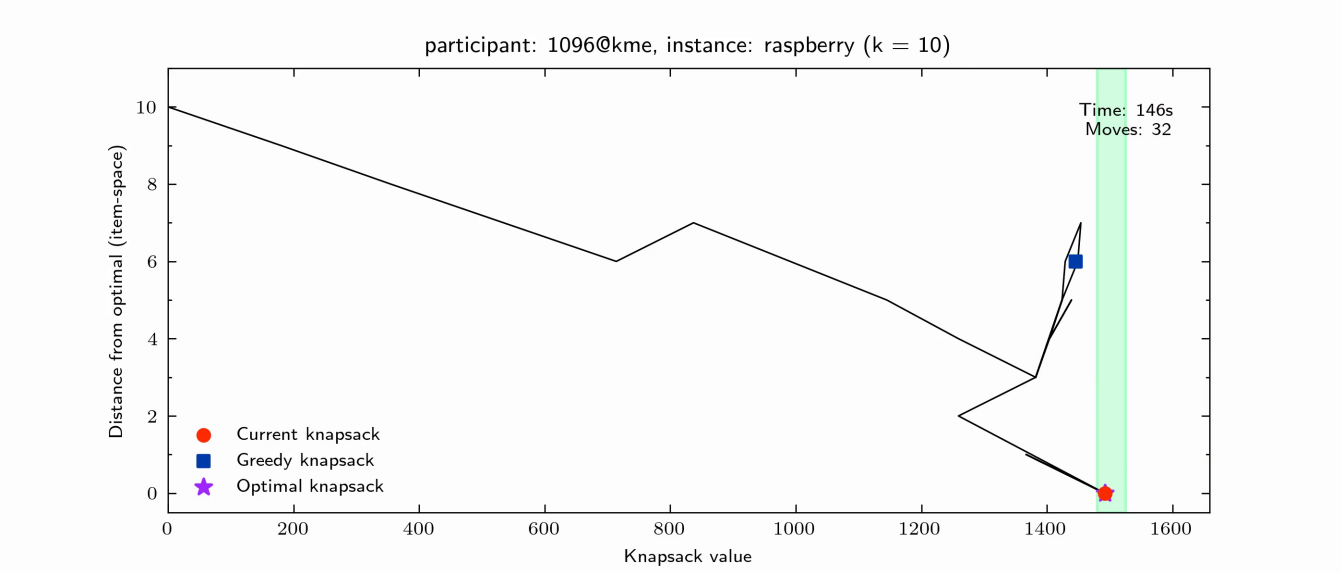
Being unaware



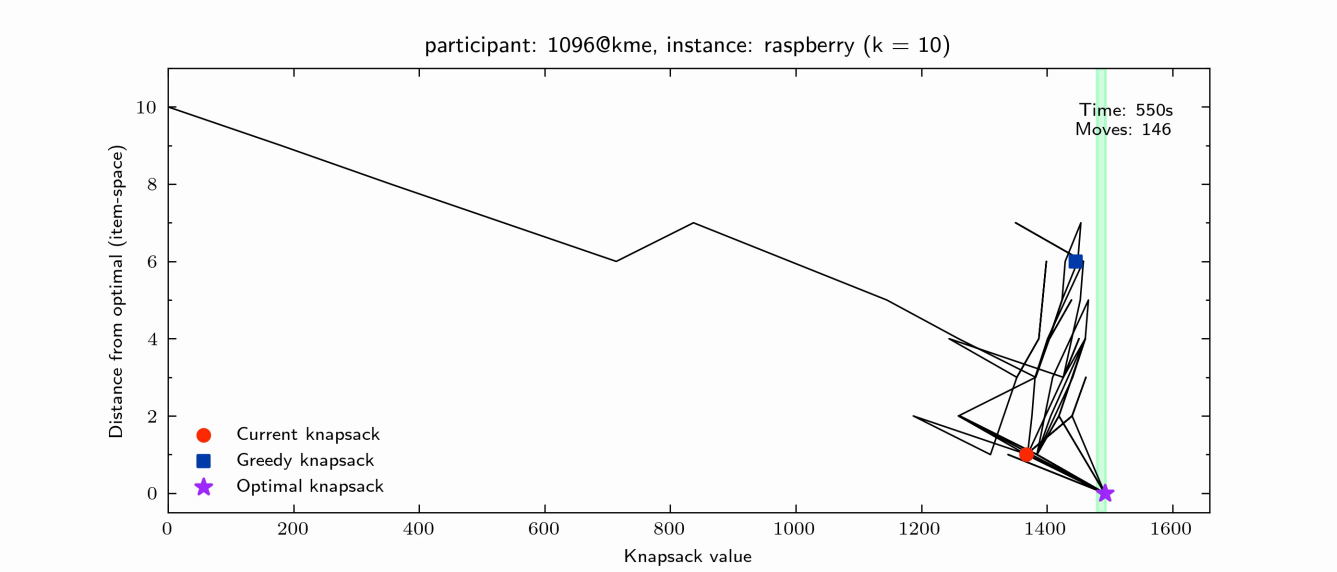
Being unaware



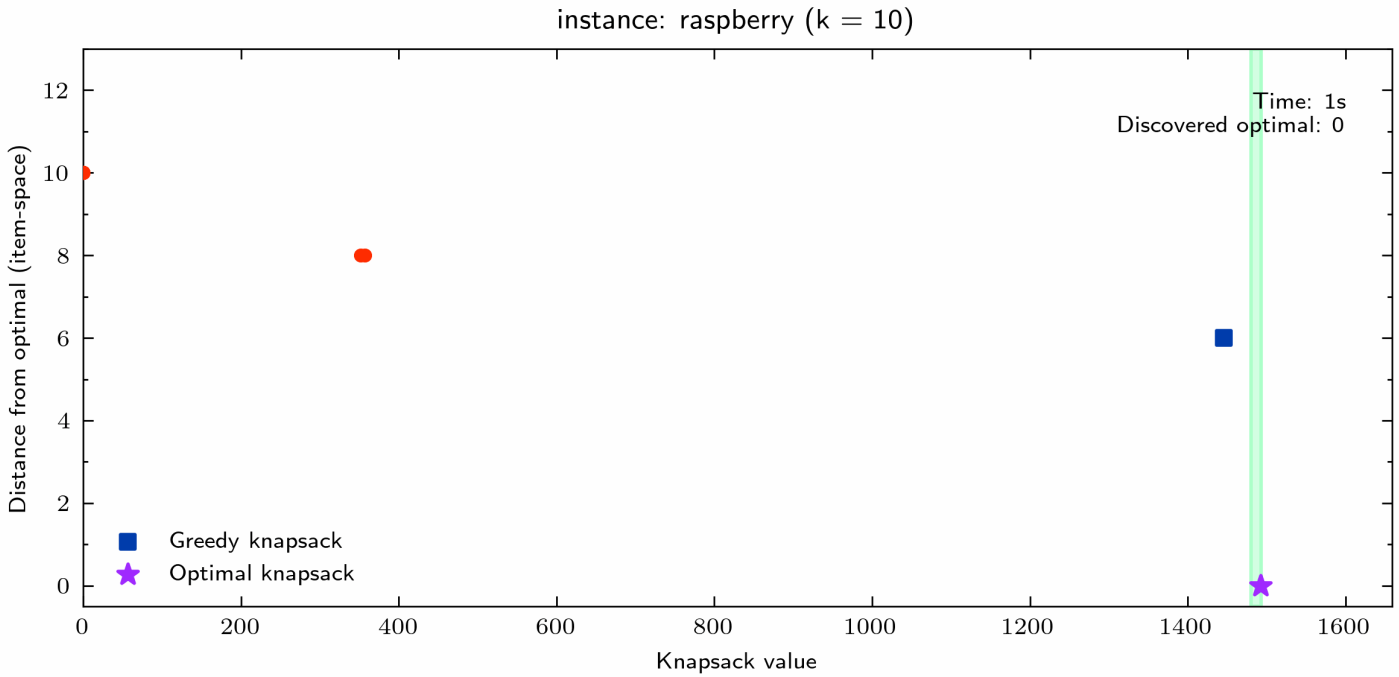
Being unaware



Being unaware

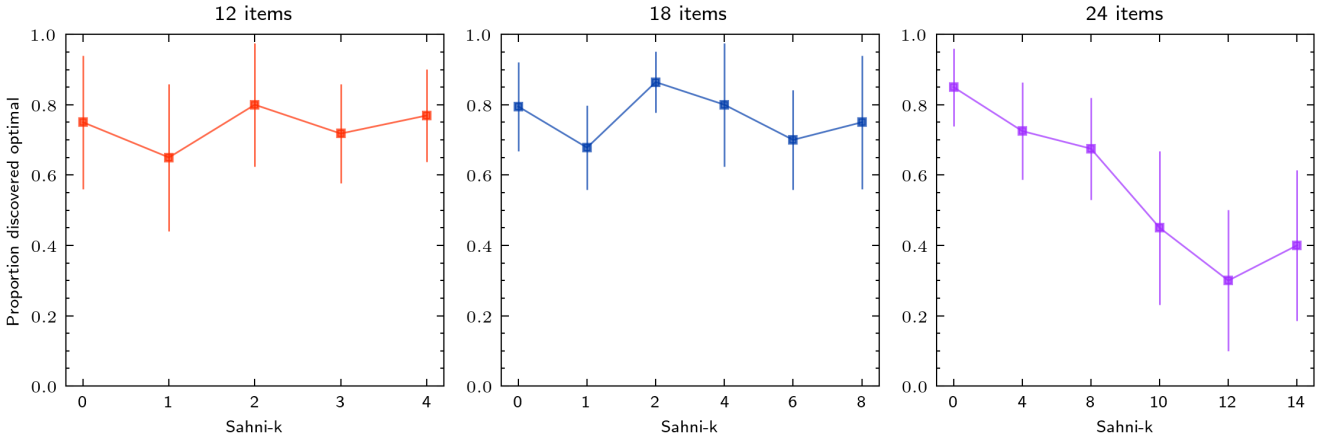


Full trading period

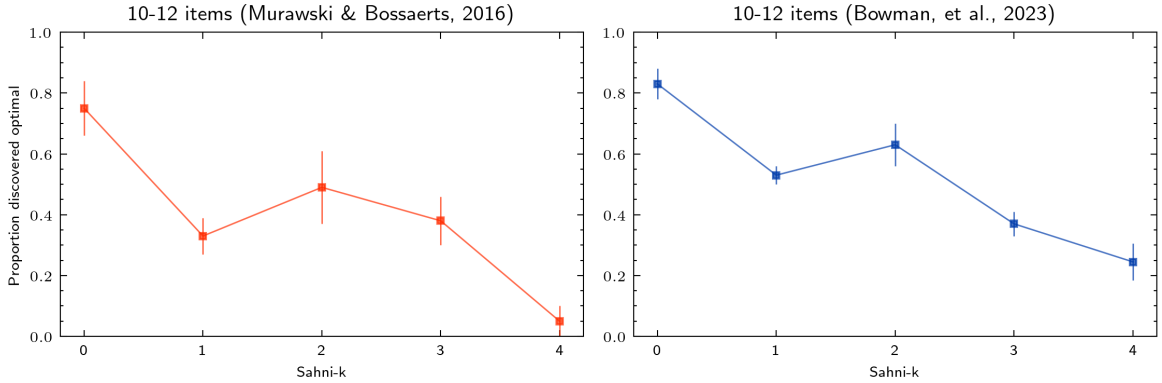


Performance in Knapsack

With the market



Without the market



Summary

Experimental evidence that markets:

1. spread knowledge
2. help individuals solve computationally complex tasks
3. improve performance compared to patents (?)

Summary

Experimental evidence that markets:

1. spread knowledge
2. help individuals solve computationally complex tasks
3. improve performance compared to patents (?)

Summary

Experimental evidence that markets:

1. spread knowledge
2. help individuals solve computationally complex tasks
3. improve performance compared to patents (?)

Individual experiment

- Find out how well people perform in the absence of a market (individually)
- Conditions:
  1. Information signal condition: solve KP after learning the optimal value
  2. Control condition: solve KP without any extra information
- Incentives: rewarded for obtaining the optimal solution

Individual experiment

- Find out how well people perform in the absence of a market (individually)
- Conditions:
  1. Information signal condition: solve KP after learning the optimal value
  2. Control condition: solve KP without any extra information
- Incentives: rewarded for obtaining the optimal solution

Individual experiment

- Find out how well people perform in the absence of a market (individually)
- Conditions:
  1. Information signal condition: solve KP after learning the optimal value
  2. Control condition: solve KP without any extra information
- Incentives: rewarded for obtaining the optimal solution

Individual experiment

- Find out how well people perform in the absence of a market (individually)
- Conditions:
  1. Information signal condition: solve KP after learning the optimal value
  2. Control condition: solve KP without any extra information
- Incentives: rewarded for obtaining the optimal solution

Individual experiment

- Find out how well people perform in the absence of a market (individually)
- Conditions:
  1. Information signal condition: solve KP after learning the optimal value
  2. Control condition: solve KP without any extra information
- Incentives: rewarded for obtaining the optimal solution

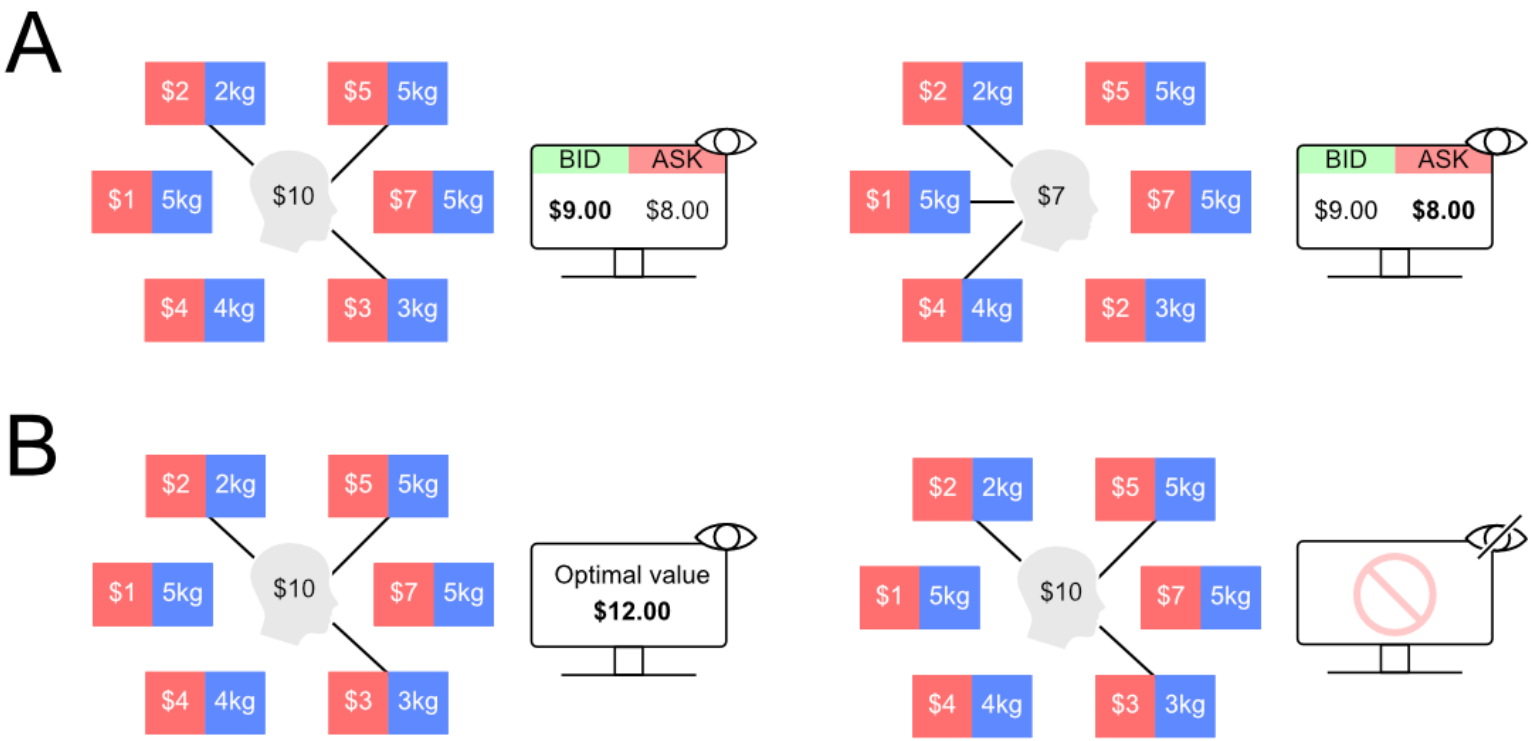


Figure: **(A)** Market experiment. **(B)** Individual experiment.

Disentangling effects

Markets experiment	Individual experiment (information condition)	Individual experiment (control condition)
Interface ↑	Interface ↑	Interface ↑
Instances ↑	Instances ↑	Instances ↑
Information signal (prices) ↑	Information signal (exogenous) ↑	
Tournament effects* ↑		
Incentives (trading) ↓	Incentives (finding optimal) ↑	Incentives (optimal) ↑

* Or any other factor unique to markets that increases performance

Disentangling effects

Markets experiment	Individual experiment (information condition)	Individual experiment (control condition)
Interface ↑	Interface ↑	Interface ↑
Instances ↑	Instances ↑	Instances ↑
Information signal (prices) ↑	Information signal (exogenous) ↑	
Tournament effects* ↑		
Incentives (trading) ↓	Incentives (finding optimal) ↑	Incentives (optimal) ↑

* Or any other factor unique to markets that increases performance

Disentangling effects

Markets experiment	Individual experiment (information condition)	Individual experiment (control condition)
Interface ↑	Interface ↑	Interface ↑
Instances ↑	Instances ↑	Instances ↑
Information signal (prices) ↑	Information signal (exogenous) ↑	
Tournament effects* ↑		
Incentives (trading) ↓	Incentives (finding optimal) ↑	Incentives (optimal) ↑

* Or any other factor unique to markets that increases performance

Disentangling effects

Markets experiment	Individual experiment (information condition)	Individual experiment (control condition)
Interface ↑	Interface ↑	Interface ↑
Instances ↑	Instances ↑	Instances ↑
Information signal (prices) ↑	Information signal (exogenous) ↑	
Tournament effects* ↑		
Incentives (trading) ↓	Incentives (finding optimal) ↑	Incentives (optimal) ↑

* Or any other factor unique to markets that increases performance

Disentangling effects

Markets experiment	Individual experiment (information condition)	Individual experiment (control condition)
Interface ↑	Interface ↑	Interface ↑
Instances ↑	Instances ↑	Instances ↑
Information signal (prices) ↑	Information signal (exogenous) ↑	
Tournament effects* ↑		
Incentives (trading) ↓	Incentives (finding optimal) ↑	Incentives (optimal) ↑

* Or any other factor unique to markets that increases performance

Disentangling effects

Markets experiment	Individual experiment (information condition)	Individual experiment (control condition)
Interface ↑	Interface ↑	Interface ↑
Instances ↑	Instances ↑	Instances ↑
Information signal (prices) ↑	Information signal (exogenous) ↑	
Tournament effects* ↑		
Incentives (trading) ↓	Incentives (finding optimal) ↑	Incentives (optimal) ↑

* Or any other factor unique to markets that increases performance