

Manipulation, Bribery and Control in Dagstuhl Seminar Seating Arrangements



Manipulation

Manipulation has been defined as changing the outcome of an allocation by individually acting insincerely:

Manipulation

Manipulation has been defined as changing the outcome of an allocation by individually acting insincerely:



Or, in our case: Sitting at the wrong table.

Limits of Manipulation

Theorem

Any manipulation can be detected in $O(n)$ time.

Limits of Manipulation

Theorem

Any manipulation can be detected in $O(n)$ time.

Proof.

Distributed algorithm: Every participant finds his or her own place, one is bound to find the manipulator sitting in his or her place. \square

False Name Manipulation

... or how to ensure two people are always seated together:

False Name Manipulation

... or how to ensure two people are always seated together:



False Name Manipulation

... or how to ensure two people are always seated together:



Bribery

- Wonder why the same people get seated together over and over again?

Bribery

- Wonder why the same people get seated together over and over again?
- Ever wonder why I am always served food first?

Bribery

- Wonder why the same people get seated together over and over again?
- Ever wonder why I am always served food first?



Dagstuhl-Style Bribery

The easy way:

Stamps	
Briefmarken/Stamps	
	- Please specify € - Amount
Snacks, Toiletries, Samsa, Souvenirs, Postcards, Pads	50€ Bribe for

de/Gastinfo or to the information fold

Control

Three methods of Control

- Adding Participants
- Removing Participants
- Partitioning Participants

Control

Three methods of Control

- Adding Participants
- Removing Participants
- Partitioning Participants

Theorem

All Forms of control are possible and computable in linear time if and only if the manipulator has early access to dining room!

Adding Participants



Removing Participants



$O(n)$ Algorithm for Control by Partitioning Participants

- 1 Prepare a list F of 4 people you would like to sit with.

$O(n)$ Algorithm for Control by Partitioning Participants

- 1 Prepare a list F of 4 people you would like to sit with.
- 2 Add yourself to F : $F \leftarrow F \cup \{u\}$.

$O(n)$ Algorithm for Control by Partitioning Participants

- 1 Prepare a list F of 4 people you would like to sit with.
- 2 Add yourself to F : $F \leftarrow F \cup \{u\}$.
- 3 For every blue tag on the table:
 - 1 If the tag contains any name in F :
 - 1 If $|L| = 4$, go to step 3.
 - 2 Pick up the tag, keep in T .
 - 3 Keep the location of the tag in a list L .

$O(n)$ Algorithm for Control by Partitioning Participants

- 1 Prepare a list F of 4 people you would like to sit with.
- 2 Add yourself to F : $F \leftarrow F \cup \{u\}$.
- 3 For every blue tag on the table:
 - 1 If the tag contains any name in F :
 - 1 If $|L| = 4$, go to step 3.
 - 2 Pick up the tag, keep in T .
 - 3 Keep the location of the tag in a list L .
- 4 Let t be the table containing the last tag found.

$O(n)$ Algorithm for Control by Partitioning Participants

- ① Prepare a list F of 4 people you would like to sit with.
- ② Add yourself to F : $F \leftarrow F \cup \{u\}$.
- ③ For every blue tag on the table:
 - ① If the tag contains any name in F :
 - ① If $|L| = 4$, go to step 3.
 - ② Pick up the tag, keep in T .
 - ③ Keep the location of the tag in a list L .
- ④ Let t be the table containing the last tag found.
- ⑤ Pick up all tags in table t , let T' be the tags picked up.

$O(n)$ Algorithm for Control by Partitioning Participants

- 1 Prepare a list F of 4 people you would like to sit with.
- 2 Add yourself to F : $F \leftarrow F \cup \{u\}$.
- 3 For every blue tag on the table:
 - 1 If the tag contains any name in F :
 - 1 If $|L| = 4$, go to step 3.
 - 2 Pick up the tag, keep in T .
 - 3 Keep the location of the tag in a list L .
- 4 Let t be the table containing the last tag found.
- 5 Pick up all tags in table t , let T' be the tags picked up.
- 6 Drop the tags in T at table t .

$O(n)$ Algorithm for Control by Partitioning Participants

- 1 Prepare a list F of 4 people you would like to sit with.
- 2 Add yourself to F : $F \leftarrow F \cup \{u\}$.
- 3 For every blue tag on the table:
 - 1 If the tag contains any name in F :
 - 1 If $|L| = 4$, go to step 3.
 - 2 Pick up the tag, keep in T .
 - 3 Keep the location of the tag in a list L .
- 4 Let t be the table containing the last tag found.
- 5 Pick up all tags in table t , let T' be the tags picked up.
- 6 Drop the tags in T at table t .
- 7 Drop the tags in T' at the locations in L .

Results Table

All results are summarized in the following table:

Results Table

All results are summarized in the following table:



Thank You

I would like to thank Ulle, Jerome, Francesca, Tuomas, and all the people at Dagstuhl for an excellent conferece.