*SLAC and SLACER: What happens when you try to be "smart" about who you reject_*

Emma Norling

_Centre for Policy Modelling,
Manchester (UK)

Outline

- Scenario
- SLAC and SLACER
- Trying to be smart
- Discussion and conclusions

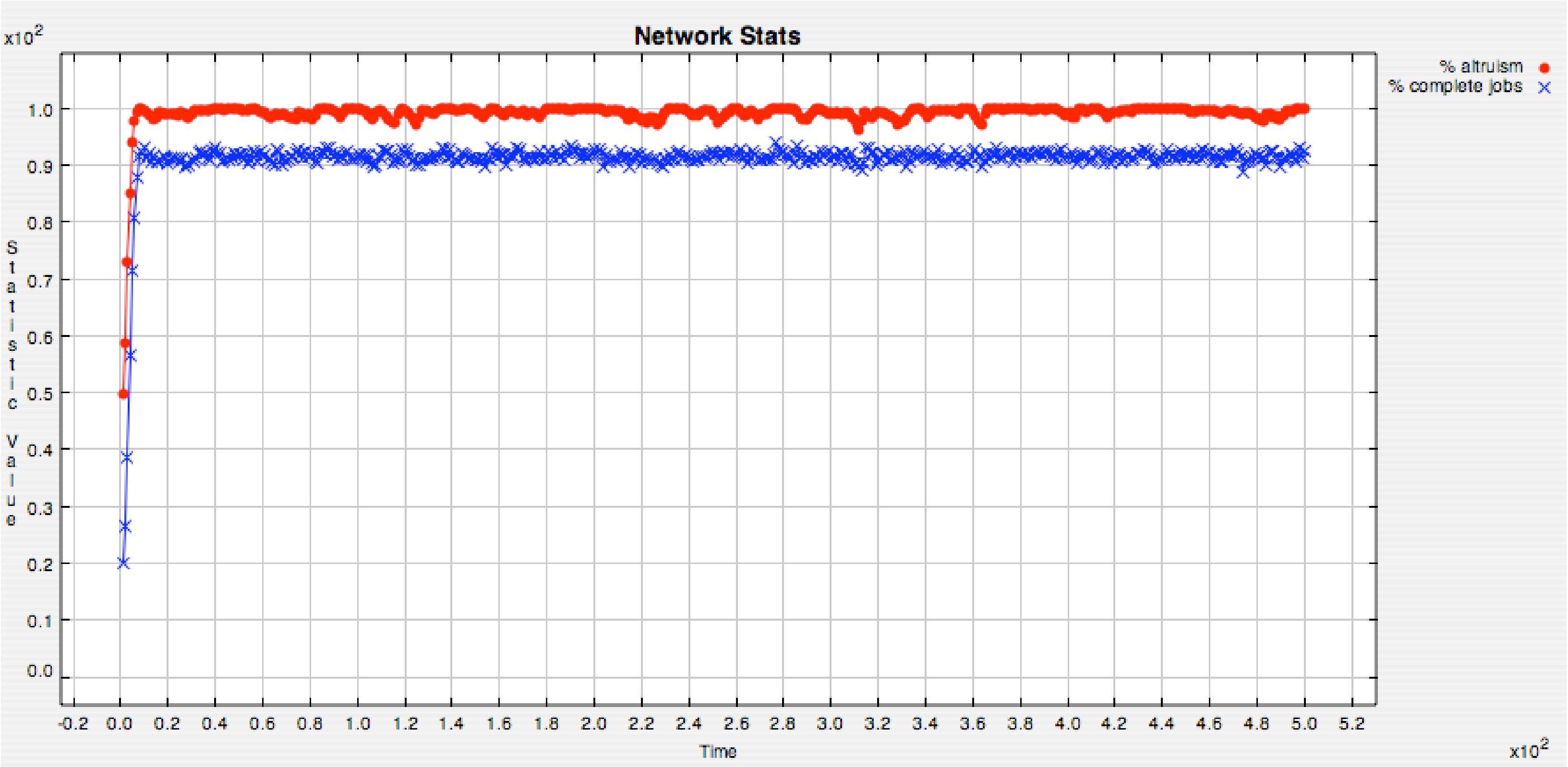
Scenario

- Each individual has ability to process jobs of a certain type
- Each individual receives jobs, some of which it cannot process
  - ⇒Individuals must request assistance of others in order to maximise reward
- An individual may accept or reject a request for assistance

SLAC (recap)

- Agent A1 processes job (if possible)
- Mutation step:
  - A1 randomly selects another agent, A2
  - If wealth A2 > wealth A1, A1 copies A2:
    - Drops existing links
    - Links to A2
    - Copies links of A2
    - Copies altruism of A2
    - With small chance, mutates link (random rewire)
    - With smaller chance, mutates behaviour

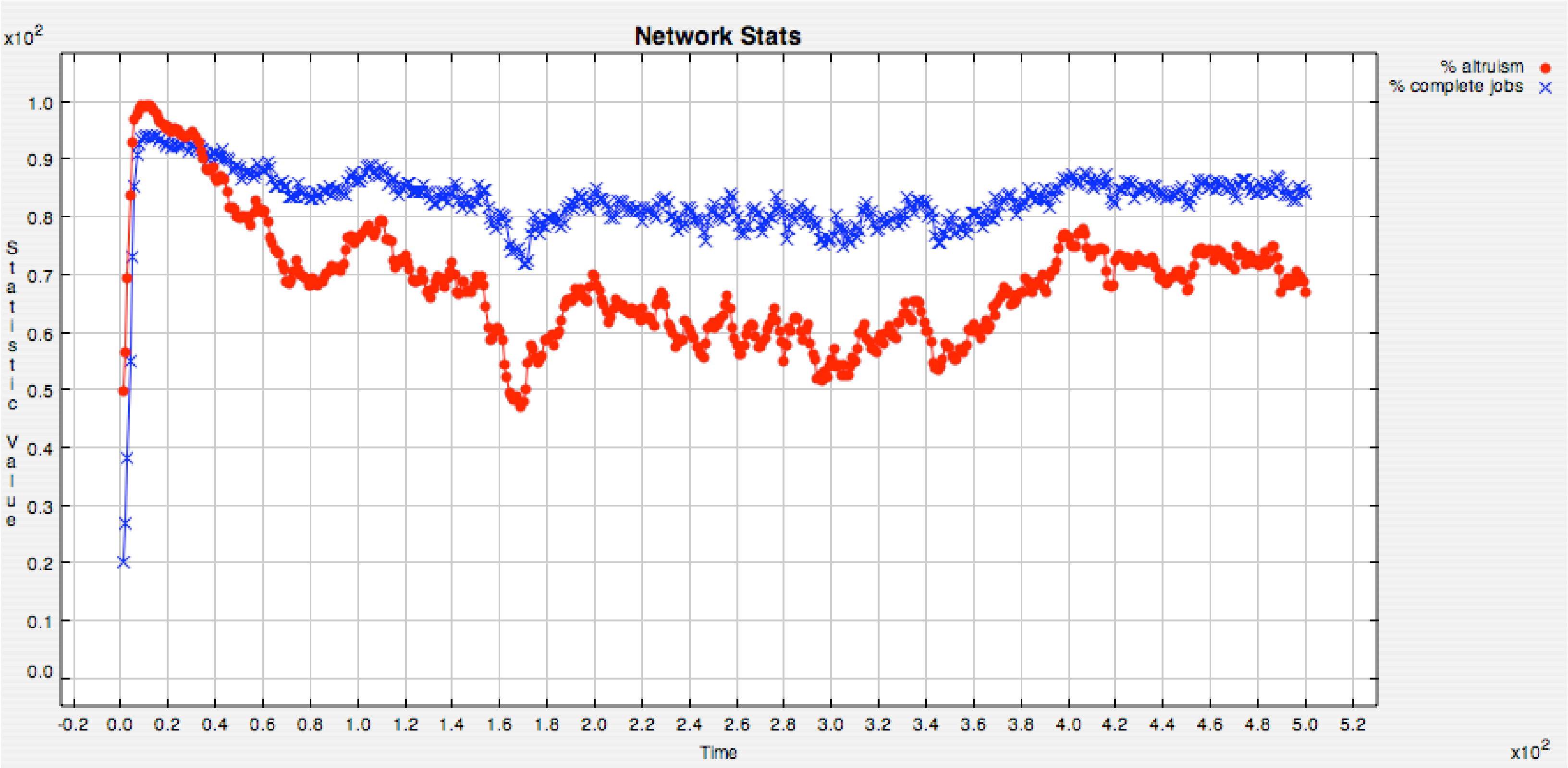
Typical Results



SLACER (recap)

- As with SLAC, but instead of dropping all existing links, probabilistically drops links (typical $p = 0.95$; $p = 1.0 \equiv$ SLAC)
- Leads to small world network

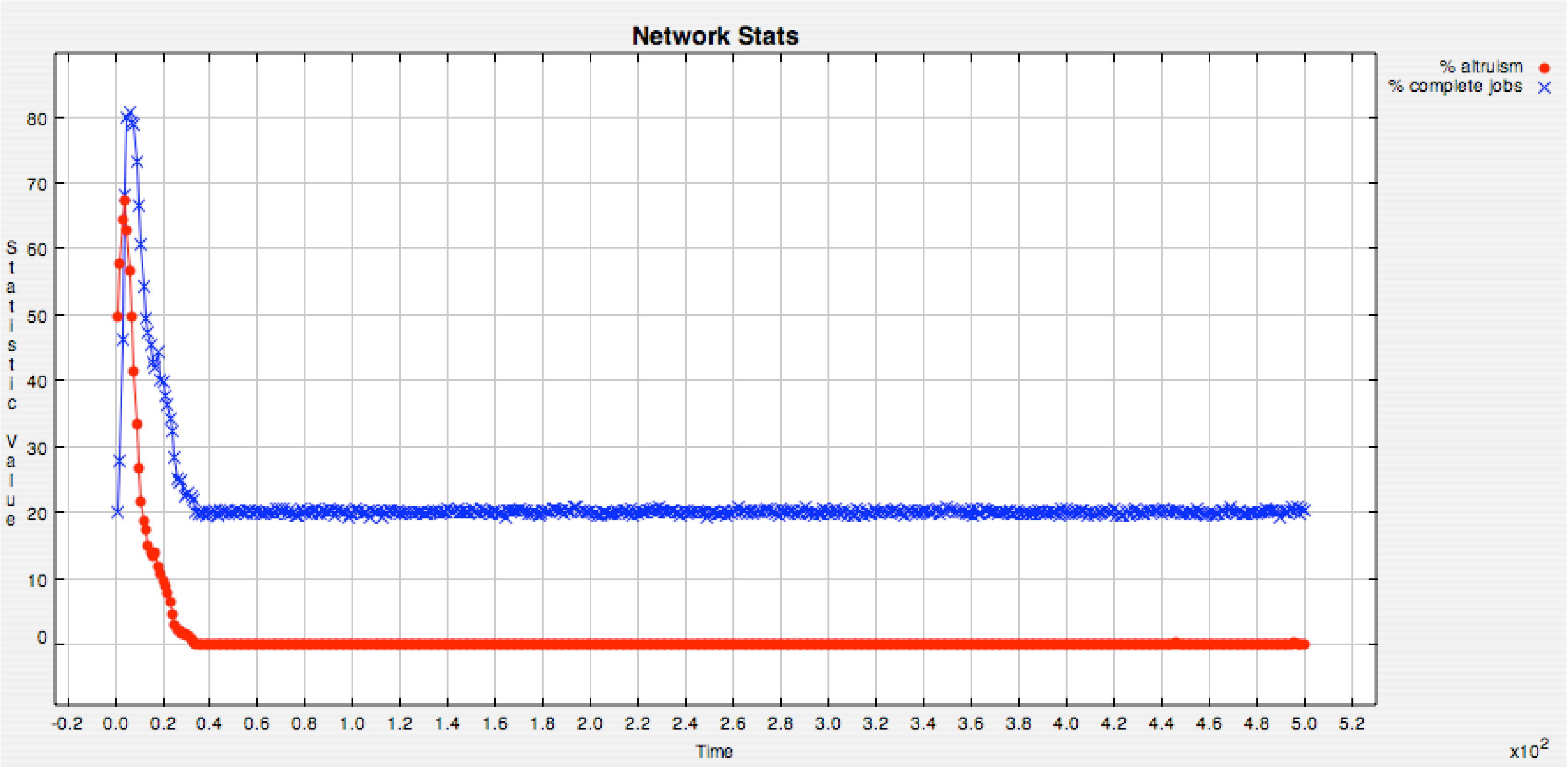
Typical Results (SLACER)



Trying to be smart

- What if, instead of probabilistically dropping links, drop links that have not been helpful?
- Naïve approach:
  - Keep track of who did / did not assist with requests, drop those who did not assist.
- Surprising outcome…

Typical Results



What happened?

- Cheaters do better in the short term, because they gain from altruists but do not incur the cost of helping others ⇒ agents tend to copy cheaters
- BUT because cheaters don’t help others, links to them get dropped ⇒ agents lose the benefits of copying these cheaters

Discussion

- A naïve approach to smart networking does not work
  - Maybe it is better to assume agents with low wealth are more likely to help?
    - *NO*
  - Maybe cost of figuring out a better way of networking is not worth it?
    - *SLACER doesn’t do too badly, and is low cost*

Ongoing Work

- Many jobs require several tasks, either with or without sequence
  - Can we extend these algorithms to deal with such jobs?
    - Time may play an important role