Optimising partner matching for microsimulations of the HIV epidemic

Nathan Geffen

SACEMA Research Day, 24 March 2015

Website for this talk

http://nathangeffen.webfactional.com/partnermatching/partnermatching.html

- Too much to cover in 10 minutes
- This presentation is bird's eye view
- Webpage has details
- Perhaps seminar

Aim

- Microsimulation of HIV epidemic gives rich insights
- We want:
    - Monte Carlo simulation to calculate confidence intervals
    - Convenient high-level language programming
    - Visualisations on the web using Javascript
- BUT: It is too slow
- Partner matching is the bottle-neck
- **Find faster ways of doing partner matching**

Methodology

- Define three partner matching algorithms
- Define two reference algorithms to compare these with:
  - Quality: One very slow producing nearly ideal matches
  - Speed: One very fast producing random matches
- Define measure of quality
- Analyse mathematically
- Compare empirically in multiple tests

Typical discrete time microsimulation

```
for each time-step
    ----------------
    for each event E
        for each agent A
            if E should be applied to A
                apply E to A
    -----------------
```

Algorithm efficiency

- $O(n)$ vs $O(n^2)$

n	$n \log n$	n^2
10	33	100
100	664	10,000
1,000	9,966	1,000,000
10,000	132,877	100,000,000
100,000	1,660,964	10,000,000,000
1,000,000	19,931,569	1,000,000,000,000

Graphically depicted

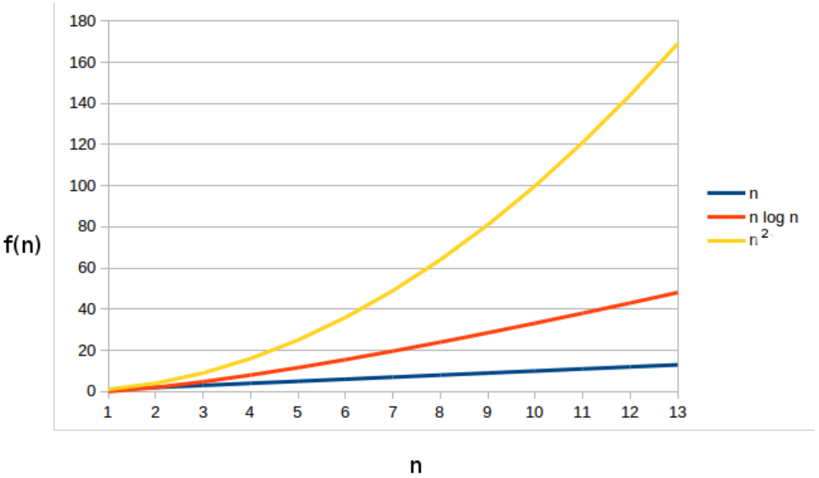

Figure 1:Three efficiency classes

Simulation details

- Simulation time-scale: 1 week
- Typical events: HIV infection, migration, death, Partner matching
- Partner matching attributes
    - age
    - sex
    - desire for new partnership
    - riskiness (including whether agent is a sex worker)
    - relationship status (including whether agent is married)

Euclidean space

- Mapping agents to Euclidean space would help
  - Each attribute (age, sex etc) is Euclidean co-ordinate
- Efficient nearest neighbour approximation algorithms
  - Locality-sensitive hashing
  - Best bin first
  - Balanced box decomposition

Mapping to Euclidean space not possible

Triangle rule of metric spaces violated:

$$d(x,z) \leq d(x,y) + d(y,z)$$

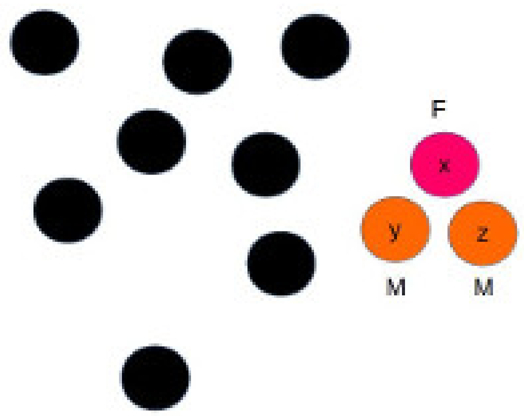

Figure 2:Heterosexual partner matching does not map

Distance function

- Example of distance function on webpage
- Extract from this a **cluster** function

Algorithms

- Brute force (reference: too slow)
- Random match (reference: too hopeless)
- Random match k
- Weighted shuffling
- Cluster shuffling

Brute force

```
brute_force_match(Agents):
// Agents is an array of agents
    shuffle(Agents)
    best = infinity
    for each agent, a, in Agents
        for each unmatched agent, b, after a in Agents
            d = distance(a, b)
            if d < best
                best = d
                best_partner = b
        make a and best_partner partners
```

This is $O(n^2)$.

Cluster shuffle match

```
cluster_shuffle_match(Agents, c, k)
// Agents is an array of agents
// c is the number of clusters
// k is the number of neighbours to search
// cluster_size = number of agents / c

    calculate cluster values for all agents

    sort agents in cluster_value order

    shuffle each cluster

    for each agent
        find best partner from k neighbours
```

This is $O(n \log n)$.

Speed (1)

	Brute	Random	Random k	Cluster	Weighted
Mean (ms)	2,337	2	20	21	22
Speedup	1	1,438	116	112	105

Speed (2)

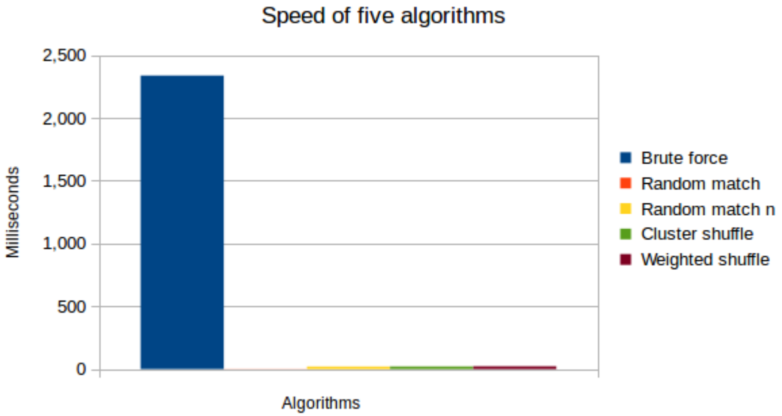

Figure 3:Brute force is two orders of magnitude slower than the three good approximation algorithms.

Quality (1)

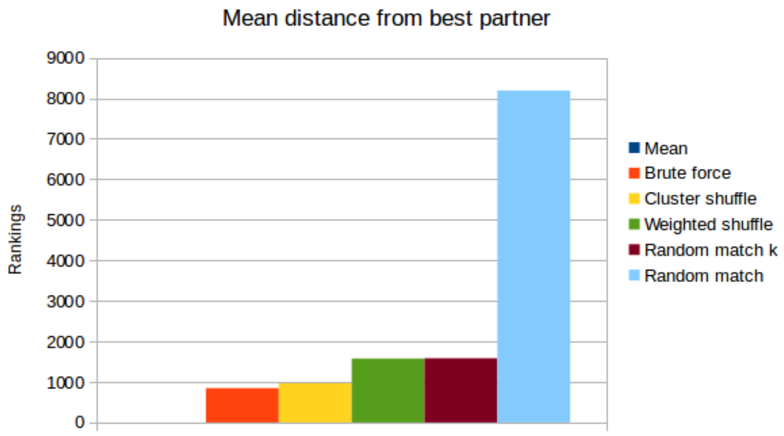

Figure 4:Approximation algorithms perform much better than random matching

Quality (2)

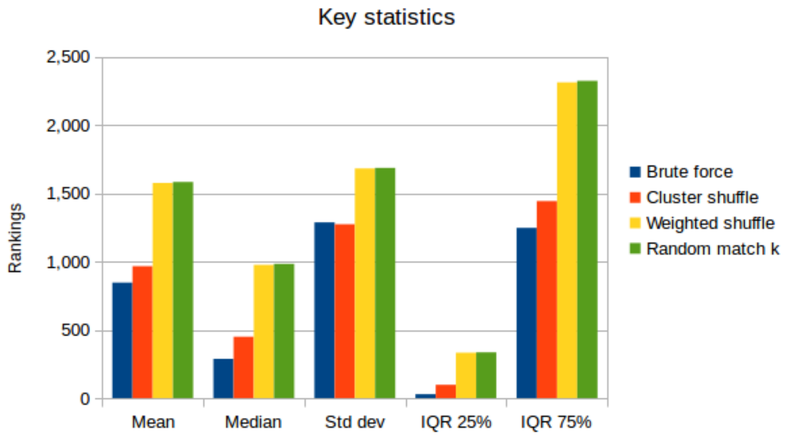

Figure 5:Cluster shuffle does well

Limitations

- Preliminary results - need to be verified.
- Criticism of ranking method I need to consider
- Only tested under one difference function - need to test more

Conclusions

- Cluster shuffle algorithm is good compromise between speed and quality
- Next step: use in actual microsimulation of the HIV epidemic

Acknowledgements

- Michelle Kuttel (supervisor)
- Andrew Boulle (supervisor)
- Leigh Johnson
- Nicoli Nattrass
- SACEMA
- UCT Department of Computer Science
- Centre for Social Science Research
- ICTS High Performance Computing team