An Evolutionary Games Analysis of Bidding Strategies in a Scheduling Auction

Jeff MacKie-Mason

Arthur W. Burks Collegiate Professor of Information and Computer Science

Professor of Economics

School of Information,
University of Michigan

Michael P. Wellman

Professor of Computer Science

Artificial Intelligence Lab,
University of Michigan

Scheduling Problems

- Scheduling optimization with full information is hard:
  - discrete
  - complementarities
  - even with public information it's typically a knapsack problem
- In addition, often have **autonomous agents** with private local information
  - Need scheduling methods that respect autonomy and private information
  - I.e., **decentralized mechanisms**

No decentralized scheduling mechanisms are ideal

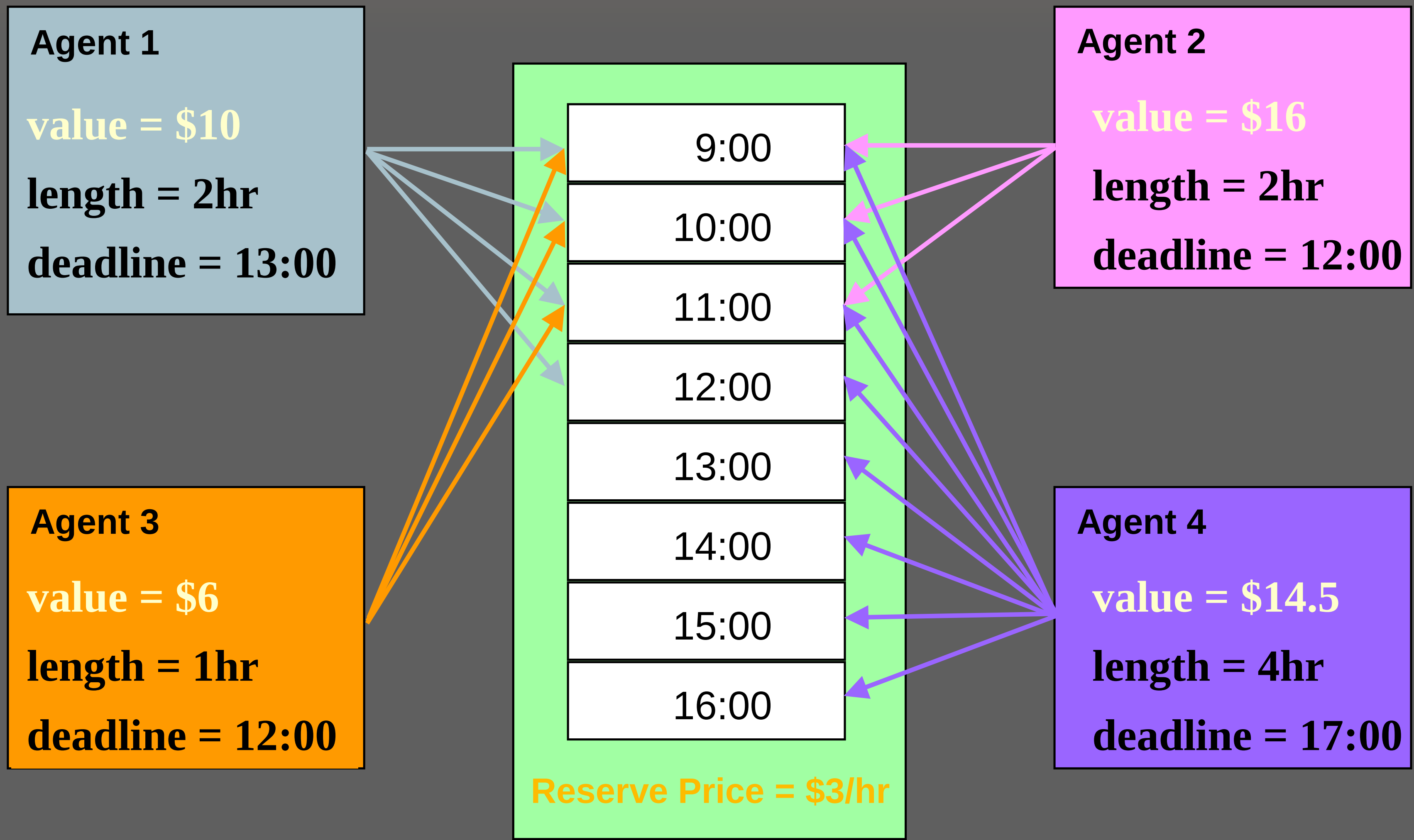
- “Ideal” mechanism satisfies (at least):
  - **Pareto efficiency**: No feasible alternative allocation benefits at least one agent without harming at least one other agent
  - **Participatory efficiency**: willingness; budget balance
  - **Agent strategies are rational**
- Impossibility theorems rule out satisfying all three

Designing good market mechanisms is immature science

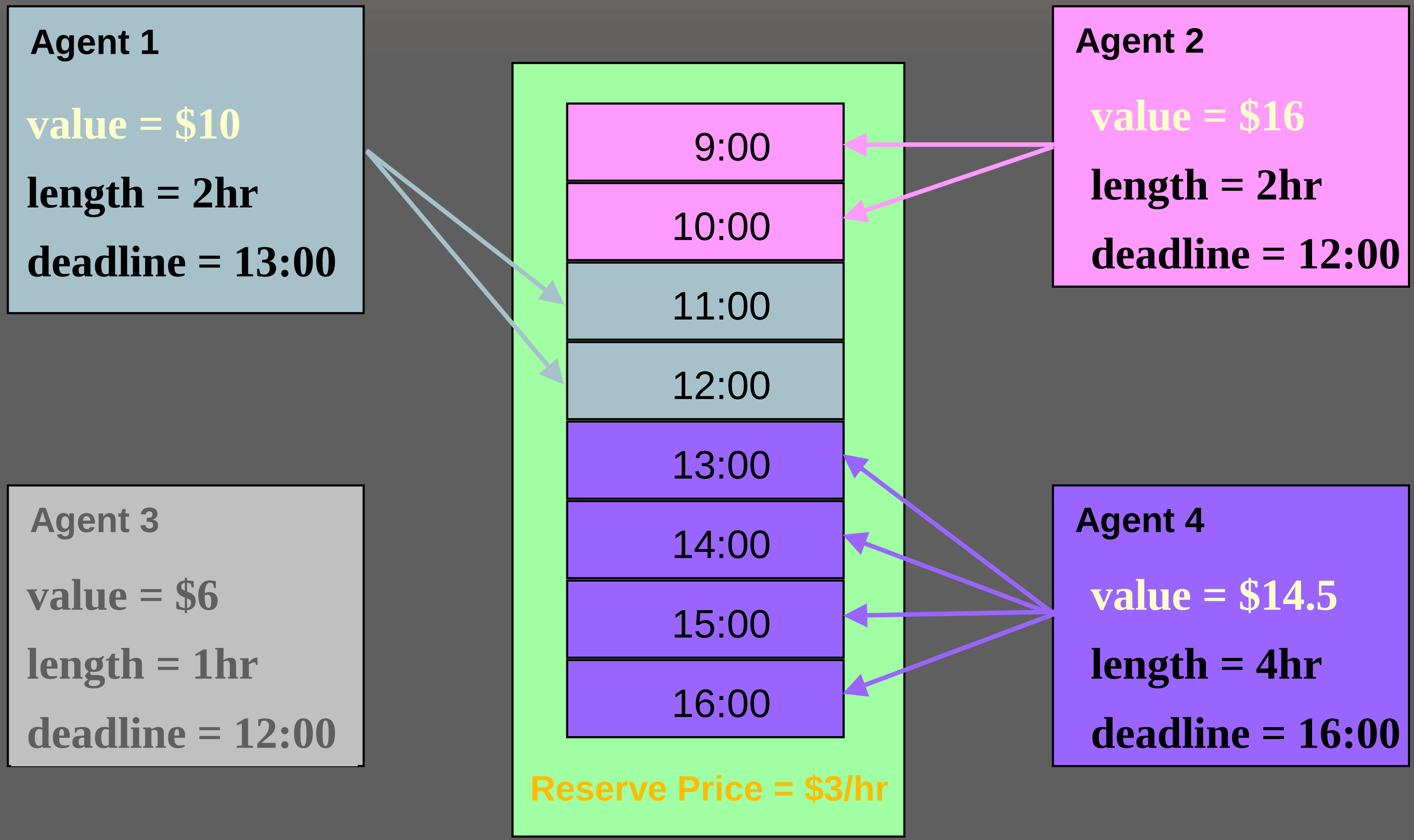
- Need to search for “good enough” mechanisms in large space of those that are not ideal
- To evaluate a mechanism, need to know how agents will interact with it (their **strategies**)
- Typically not possible to analytically derive optimal strategies
- **HOW TO EVALUATE PRACTICAL MECHANISMS?**

Factory Scheduling Example

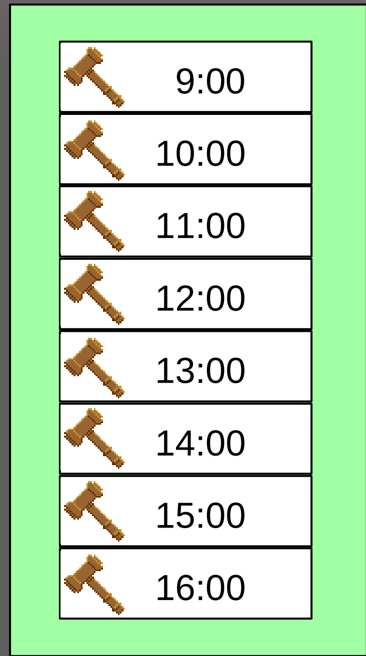
http://auction.eecs.umich.edu/FactoryDemoDocs/factory-demo.html



Efficient Solution



Simple Case:
Ascending Single Good Auctions



- **Goods**: 1 auction for each slot
- **Rules**:
  - min. bid increment ε
  - no bid withdrawal
  - closure when bidding stops
- **Baseline Strategy**:
  - agent j bids for set of slots to max surplus at current p
    - – drop out if no set of slots has positive surplus
  - N.B. For single-slot problem, this is dominant strategy
  - N.B. For multi-slot, not regret proof

Some theoretical results I: Single slot demands

- **Theorem**: A price equilibrium exists
- **Theorem**: Achieved *p* will differ from the min. unique equilibrium price by at most $\kappa\varepsilon$, where κ = min(# slots,# agents)
- **Theorem**: *v(a)* will differ from optimal by at most $\kappa\varepsilon(1+\kappa)$

Some theoretical results II: Multi-slot demands

- *p* can differ from equilibrium by arbitrarily large amount
- *v(a)* can differ from optimal by arbitrarily large amount
- So, apparently need to evaluate alternative mechanisms to find improved performance
- However: mechanisms evaluated against given strategies. **How good are the strategies?**

Strategies sensible?

	Length	Deadline	Value
Agent 1	1	1	3
Agent 2	2	2	11

Minimum prices: (1,9)

Suppose:

A_2 bids: $b_1=1$, $b_2=9$

A_1 bids: $b_1=2$

Then: $s_1 \rightarrow A_1$, $s_2 \rightarrow A_2$, $v(f)=3$

But optimum: $s_1 \rightarrow A_1$, $s_2 \rightarrow \emptyset$, $v(f)=12$

- Is it reasonable for A_2 to stop bidding?
  - By bidding $b_1=3$, it can do better than if auction stopped ($v_2 = -1$ rather than $v_2 = -9$)

Approaches to strategy discovery / selection

- Deductive analytics
  - see above
- Human-subject experiments
  - expensive, hard to generalize, limited to simple problems
- Statistical analysis
  - real world experiments few compared to number of possible mechanisms
  - expensive to implement field trials
- **Evolutionary games**
  - can **select** and **evolve** good strategies

Evolutionary games to select strategies

- Set of s=1,…,S strategies
- Population(s) of N agents, each initialized to $s_i \in S$
- Strategy i played by fraction f_i of population
- During a "generation", agents interact through mechanism, each obtains payoff ("fitness") π_i
- Update fraction f_i based on relative fitness
- Iterate

Selection outcomes

- **Monomorphic** population: strategy i^* dominates
- **Polymorphic** equilibrium: mixed strategy equilibrium
  - Note: May have multiple steady states (if any) so initial conditions matter
- Theoretical properties known for some problems:
  - E.g., under fairly general conditions
    Evolutionary Equilibria $\subset$ Nash Equilibria

Discovery by evolving strategies

- Add a method to search through other parts of strategy space
  - E.g., genetic algorithm
- At each generation, invoke new strategies

Our method

- 0. Specify a scheduling problem (N slots), initialize a population with strategy distrib *f*
- 1. Randomly draw agents to participate in a scheduling market ("instance")
- 2. Randomly assign schedule preferences, play instance
- 3. Each generation update population fractions proportional to fitness

Design issues and implications

- 0. **Problem specificity**: Strategy performance may vary by problem
- 1. **Playing the field**: find strategies that succeed on average against distribution of other strategies in population
- 2. **Preference independence**: find strategies that succeed on average across all admissible preferences
  - General in principle, but in practice not the same as preference-specific strategies
- 3. **Update dynamics** may determine number and type of equilibria (and whether found by the algorithm)

Strategies explored

- **Baseline – "sunk unaware"**: agent *j* bids for set of slots to max surplus at current *p*
  - Bids as if incremental cost for slots currently winning is full price
- **Problem**: Agents ignoring "sunk cost" and may stop irrationally early
- **Alternative**: **"sunk aware"**: bids as if incremental cost for slots currently winning is zero

Preliminary results

- Environment 1: “Contentious” – sunkness likely to matter
  - 5 slots available
  - 5 agents with varying length jobs, λ=1,…,5
  - Agents have monotonically decreasing values for later deadlines
  - Job lengths and deadline values drawn randomly
- Environment 2: “Loose”
  - 10 slot schedules
  - 5 agents each with 2-slot jobs
  - Monotonically decreasing random deadline values

Contentious

Surprise: "Unaware" does better

Contentious

"Unaware" seems to have higher avg. fitness

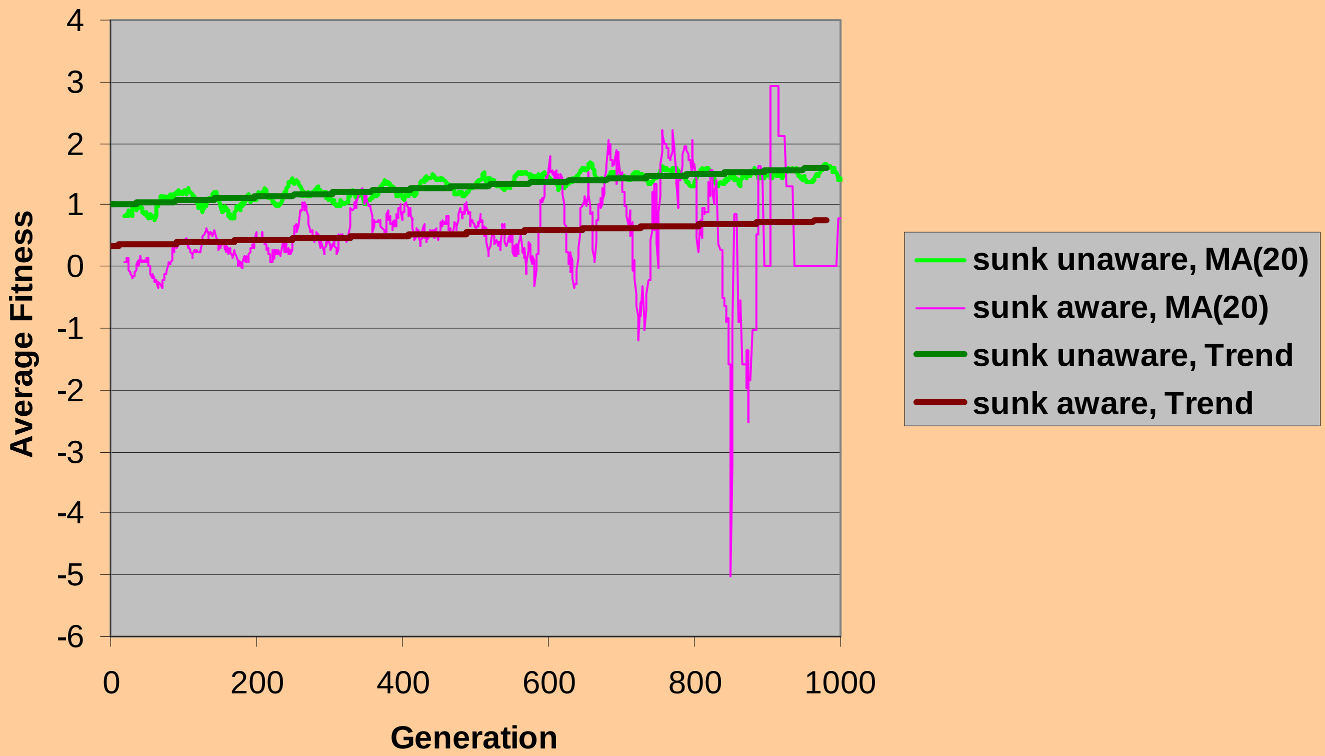
But most striking is higher variance for "aware"

Contentious

"Unaware" really does have higher avg. fitness

Oddly, they both perform better as population goes monomorphic

Agent Moving Average Fitness
5 slots, 5 agents w/varying schedule lengths
10 schedules/generation
2 strategy types: sunk {aware | unaware}
Fitness averaged over 10 epochs
Moving averaged over 20 generations
"Trend" from regression on linear time trend

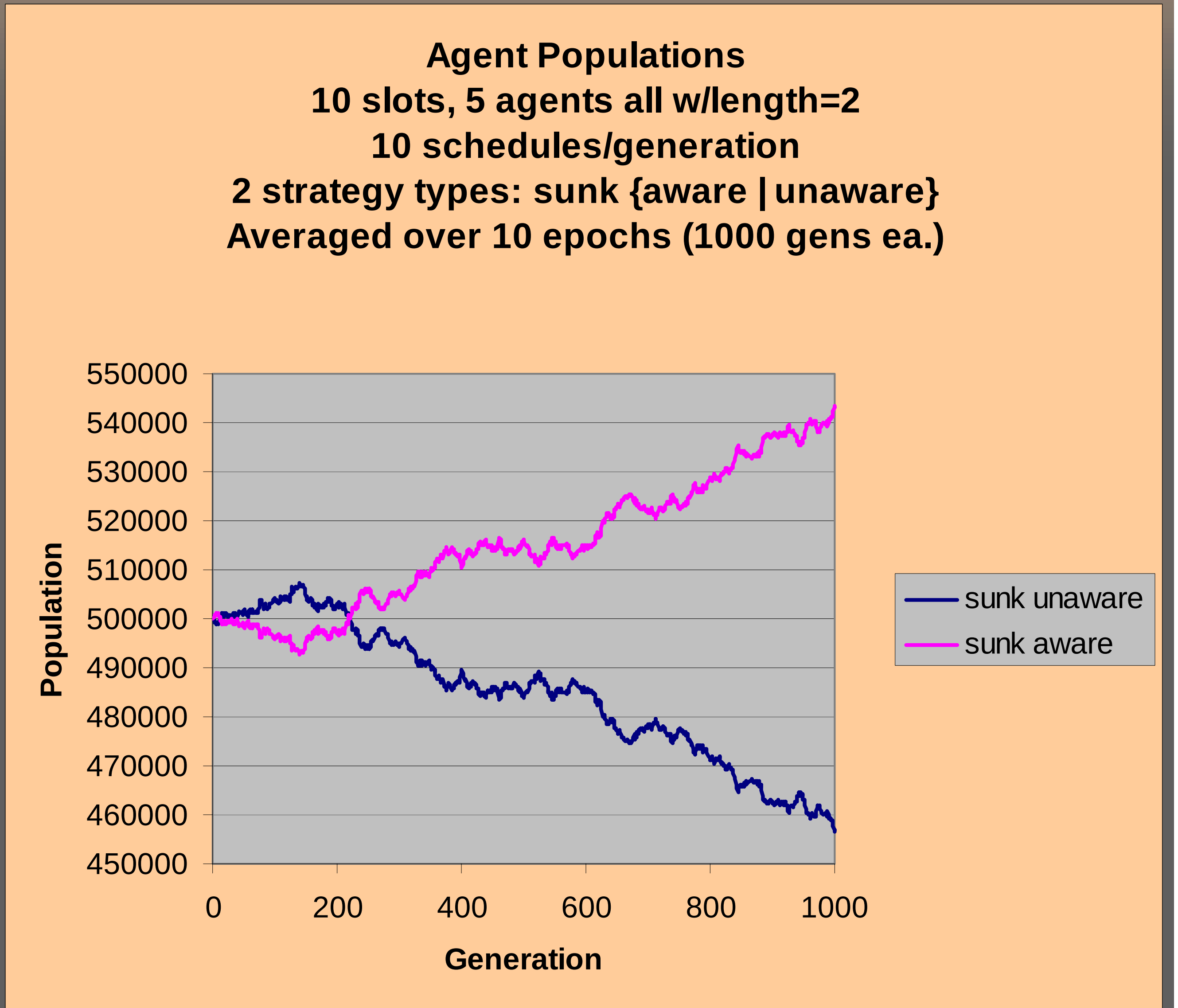


What's wrong with "aware" strategy?

- "Aware" strategy bids as if agent believes it must pay for currently winning slots with certainty, so full current price is sunk cost in expectation
- But non-zero probability currently winning slots will be bid away
- So may be <u>too</u> aggressive: too often lose slots that got agent in trouble in exchange for getting <u>new</u> slots
  - Sometimes dig a **deeper hole**

Loose

"Unaware" can perform better (problem dependency)

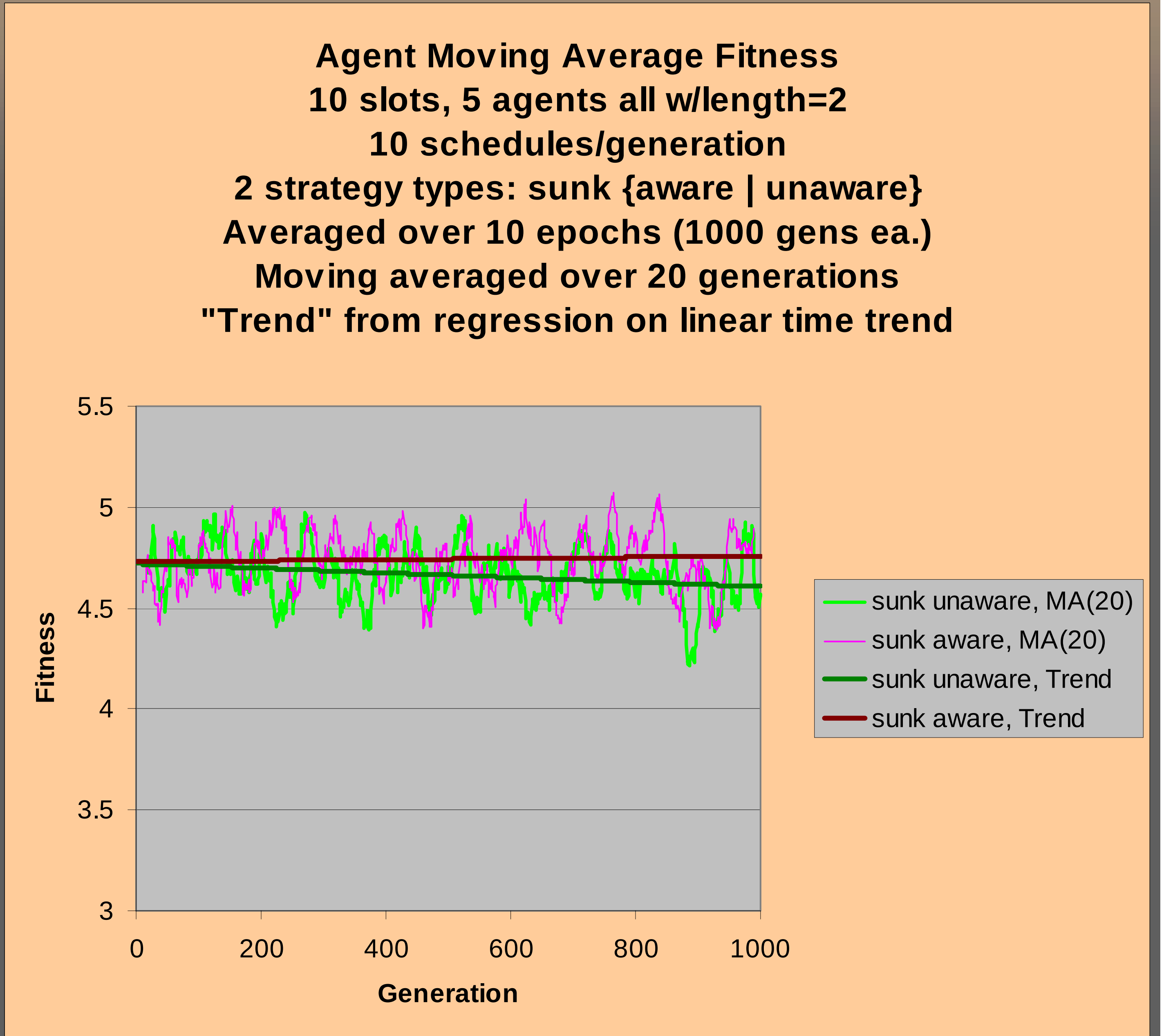


Loose

Agent Average Fitness
10 slots, 5 agents all w/length=2
10 schedules/generation
2 strategy types: sunk {aware | unaware}
Averaged over 10 epochs (1000 gens ea.)
Population
0
1
2
3
4
5
6
7
0
200
400
600
800
1000
Generation
sunk unaware
sunk aware

Loose

Fitness depends only slightly on composition of population



Summary

- Scheduling problems are hard, especially with distributed autonomous agents
- Markets are valuable class of mechanisms for decentralized problems
- Evaluating market performance depends on assumed strategies in play
- **Evolutionary games method is a promising approach for mechanism design evaluation**

For more info...

- http://www-personal.umich.edu/~jmm/
- http://ai.eecs.umich.edu/people/wellman/