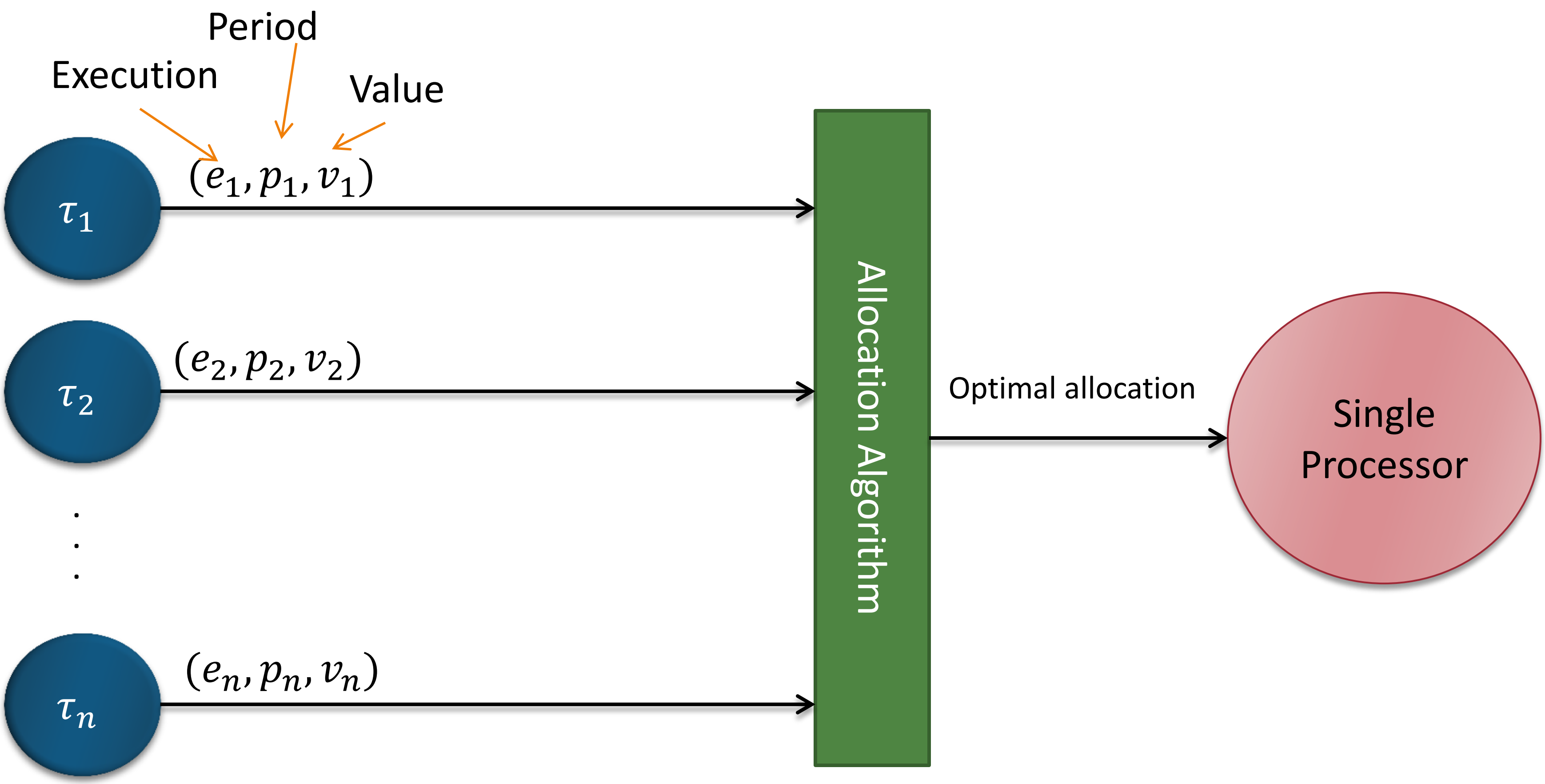
Real-Time Competitive Environments: Truthful Mechanisms for Allocating a Single Processor to Sporadic Tasks

Anwar Mohammadi, Nathan Fisher, and Daniel Grosu
Department of Computer Science
Wayne State University
Detroit, MI, USA

This research has been supported in part by the US National Science Foundation.

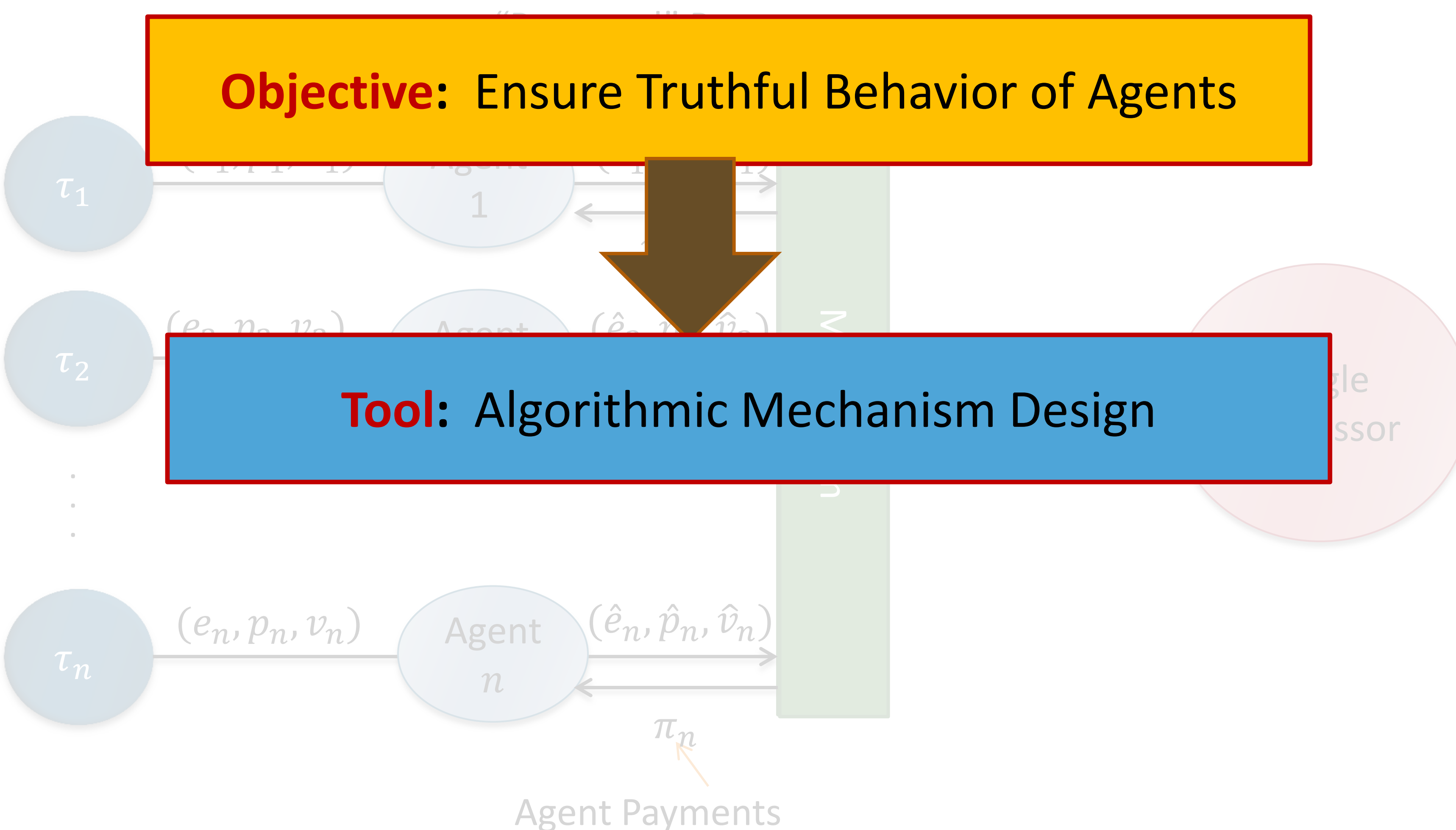
Non-Competitive vs. Competitive



Non-Competitive vs. Competitive

Objective: Ensure Truthful Behavior of Agents

Tool: Algorithmic Mechanism Design



Mechanism Design

What is Mechanism Design?

The art of designing rules in a competitive environment to achieve

- ***Truthfulness*** ← Design rules such that reporting **true task**
- ***Efficiency*** ← Optimizing some **system-wide objective funtion**.

Why Mechanism Design for Real-Time Systems?

- Real-time systems are becoming more **open**.
- **Many applications in computer science**:
  - network routing
  - human-computer interaction
  - parallel & distributed systems (e.g., grid/cluster computing)
  - internet advertisements
- **Spectacular commercial success:**
  - Google: $6 billion in 2005!
  - Yahoo!: $2.6 billion in 2005!

Related Work

- **Value/Utility Allocation in Non-Competitive Real-Time Systems**:
  - Baruah et al. [1991]: On-line scheduling in the presence of overload.
  - Rajkumar et al. [1997]: QoS-based Resource Allocation Model (QRAM).
  - Aydin et al. [2001]: Optimal reward-based scheduling.
  - …
- **Non-Real-Time Mechanism Design**:
  - Initiated by Nisan and Ronen [2001]
  - Aggarwal [2006] studied knapsack auctions.
- **Game theory in real-time systems**:
  - Sheikh et al. [2011]: Multiprocessor periodic scheduling using game-theoretic concepts
  - Porter [2004]: Mechanism design for online real-time scheduling
  - **None of these prior works on scheduling considers *traditional recurring tasks* in competitive environments.**

Model (Non-Competitive)

Implicit-deadline Sporadic Task System (with value)

Relative Deadline

- $\boldsymbol{T} = \{T_1, T_2, \dots, T_n\}$
- Each task is denoted by $T_i = (e_i, d_i, p_i, v_i)$
- Metrics:
  - Task utilization: $u_i = \frac{e_i}{p_i}$.
  - System utilization: $U(\boldsymbol{T}) = \sum_{T_i \in \boldsymbol{T}} u_i$.
- Implicit-Deadlines: $d_i = p_i$.
- Tasks are scheduled by earliest-deadline-first (EDF).
- Schedulability Test: $U(\boldsymbol{T}) \leq 1$.

Implication: Each task is completely characterized by utilization u_i.

Problem Statement

EDF-MAXVAL Problem:

Maximize $\sum_{i=1}^{n} v_i x_i$

Economics: "Social Welfare"

Subject to: $\sum_{i=1}^{n} u_i x_i \leq 1$

$x_i \in \{0,1\}$

$x_i = 1$ if task T_i is selected and $x_i = 0$, otherwise.

Model (Competitive)

Competitive Environment

- Each task T_i is owned by Agent i.
- Each Agent i is characterized by a ***type*** $\theta_i = (u_i, v_i)$.
- Agent i may have chose a different **declared type** $\hat{\theta}_i = (\hat{u}_i, \hat{v}_i)$.
- Set of agents $N = \{1,2, \ldots, n\}$.
- **Efficiency Assumption**: *Resource owner* is trying to maximize social welfare.

Motivational Example

Agent 1
$u_1 = 0.1$
$v_1 = 2$

Agent 2
$u_2 = 0.2$
$v_2 = 7$

Agent 3
$u_3 = 0.4$
$v_3 = 8$

Agent 4
$u_4 = 0.6$
$v_4 = 9$

Agent 5
$u_5 = 0.7$
$v_5 = 11$

	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
True Parameters	(0.1, 2)	(0.2, 7)	(0.4, 8)	(0.6, 9)	(0.7, 11)
Declared Parameters	(0.1,2)	(0.2,7)	(0.4,8)	(0.6,9) / (0.6,20)	(0.7,11) / (0.8,11)

Allocation Algorithm

Winners:

Implication: Lying by agents can affect overall system objective!

Truthful Social Welfare: 20

Social Welfare: 20 / 17 / 18

Mechanism Design Concepts

Mechanism

A mechanism is composed of

- **Allocation Algorithm $\boldsymbol{A}$**: determines which agents obtain the processor according to efficiency assumption.
- **Payment Scheme $\boldsymbol{\pi} = (\boldsymbol{\pi_1}, \dots, \boldsymbol{\pi_n})$**: calculates the payment of each agent.

Agent's strategy

- The strategy of an agent is her declared type.
- Agent's utility is $\mu_j = \begin{cases} v_j - \pi_j, & \text{if Agent j is a winner,} \\ -\pi_j, & \text{otherwise.} \end{cases}$
- **Selfish Assumption:** Each agent tries to maximize her utility.
- An agent may strategically declare a different type from her true type to maximize her utility.

Truthful Mechanisms

A mechanism is called **truthful** if for each agent, truthful revelation is a **dominant strategy**, that is agents maximize their utilities by reporting their true types.

Vickrey-Clarke-Groves (VCG) Mechanism: a truthful mechanism given an **optimal** allocation algorithm.

Truthful Mechanisms

EDF-MAXVAL-VCG Mechanism

- **Allocation algorithm:** The pseudo-polynomial-time algorithm EDF-MAXVAL-DP.
- **Payment scheme:**

$$\pi_j^{VCG} = \sum_{i \in A(\hat{\boldsymbol{\theta}}_{-j})} \hat{v}_i - \sum_{i \in A(\hat{\boldsymbol{\theta}}),\, i \neq j} \hat{v}_i$$

VCG Payment ≈ Total marginal loss of value of including Agent j (w.r.t. other agents).

Computable in Pseudo-Polynomial Time (dependent upon maximum task value).

Approximate Mechanisms

- Applying VCG payments to standard knapsack approximation algorithm [Kim & Ibarra, 1975] is <u>not</u> truthful!
  - **Reason**: VCG requires a **monotonic** allocation algorithm.

If Agent j wins (using allocation algorithm A) declaring $\hat{\theta}_i = (\hat{u}_i, \hat{v}_i)$, then she should also win declaring $\hat{\theta}'_i = (\hat{u}'_i, \hat{v}'_i)$ where

$$\hat{u}'_i \leq \hat{u}_i \text{ and } \hat{v}'_i \geq \hat{v}_i.$$

- We apply technique by Briest et al. [2005] to obtain truthful approximation called EDF-MAXVAL-APROX.

Approximate Mechanisms

EDF-MAXVAL-APROX Mechanism (FPTAS)

For any $\varepsilon > 0$, EDF-MAXVAL-APROX is **truthful** and returns an allocation with social welfare **no less than** $(1 - \varepsilon)$ times the optimal obtainable social welfare in time polynomial in $1/\varepsilon$ and n.

Reserve Prices

Theorem: EDF-MAXVAL-APROX remains truthful even if the **resource owner requires** that each Agent j report $\hat{v}_j \geq C\hat{u}_j$ for some constant C. ← **Reserve Price per Unit**

Frugality Metric

Measurement of over/under-payment by agents.

The frugality ratio of a mechanism is the **total payments** divided by the **second disjoint optimum value**. [Talwar, 2003]

Upper Bound (EDF-MAXVAL-VCG)

Theorem: Given k winning agents, the maximum frugality ratio is k.

The bound is "tight".

Upper Bound (EDF-MAXVAL-APROX)

Theorem: Given k winning agents and $\varepsilon > 0$, the maximum frugality ratio is at most $\big(1 + \varepsilon(n+2)\big)k$.

Evaluation

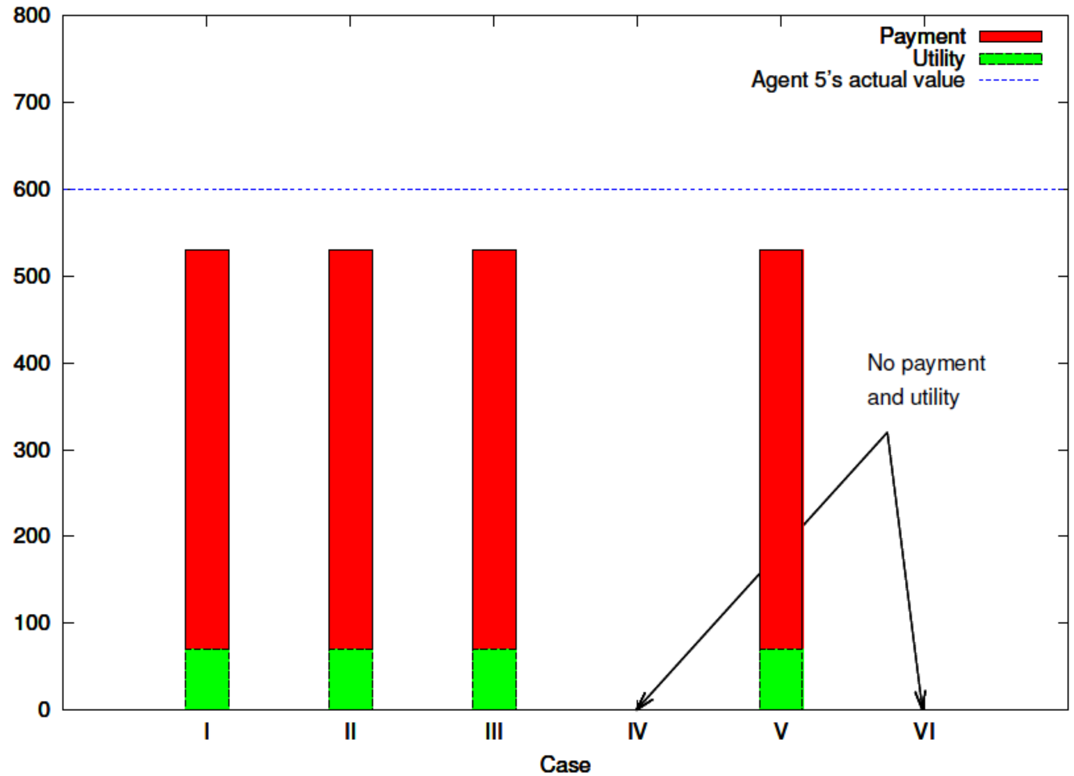
Type of Comparisons:

- Non-Truthful Type Declaration (Case Analysis)
- Frugality Ratios (Simulation)
- Execution Time (Simulation)

Non-Truthful Type Declaration (Case Analysis)

- We consider an environment of 10 agents.
- All agents are truthful except Agent 5.
- We consider 6 cases:
  - Case I: Agent 5 is truthful.
  - Cases II, III, IV: Agent 5 is declaring non-true values
  - Cases V and VI: Agent 5 is declaring non-true utilization

Non-Truthful Type Declaration (Case Analysis)



Utilities and payments of Agent 5

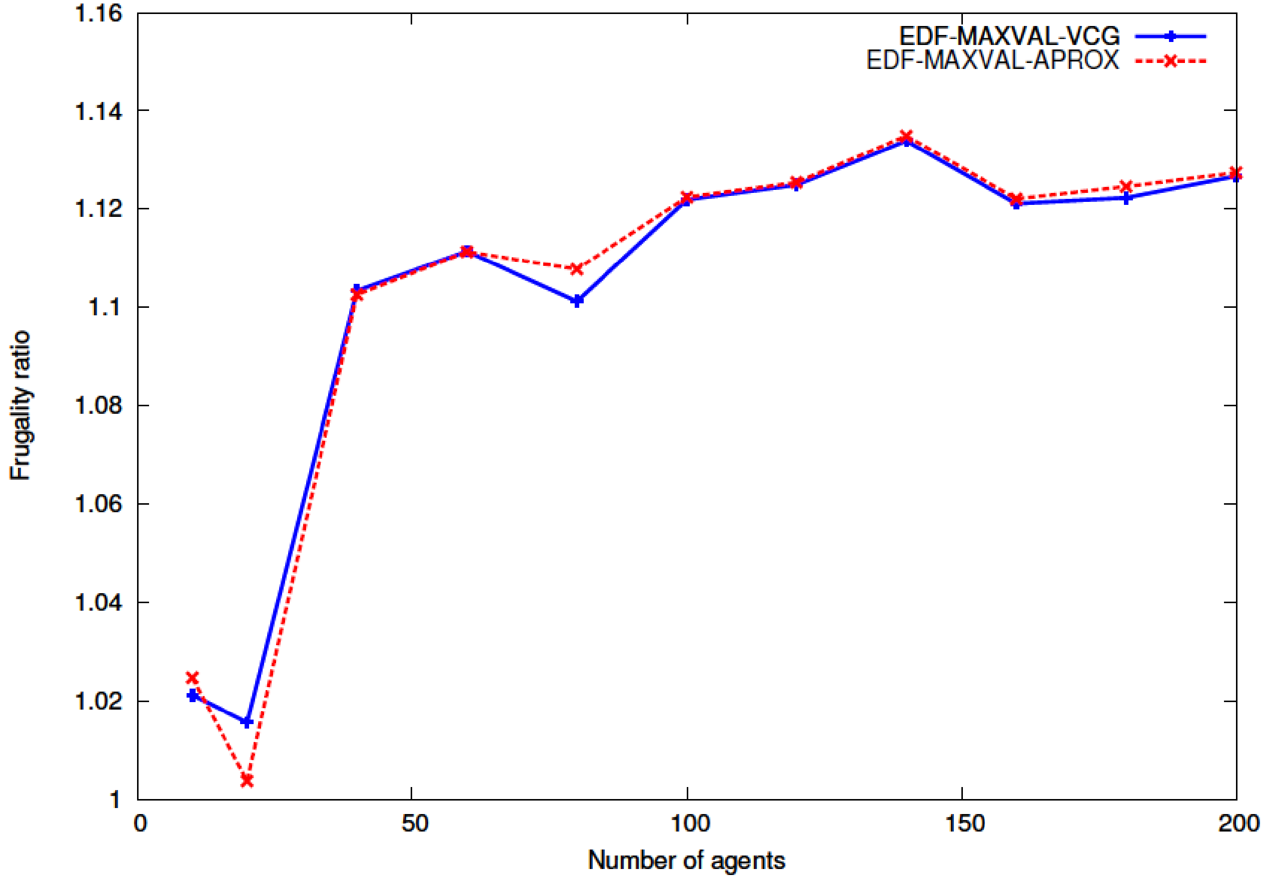
Agent 5 can not obtain better utility by lying.

Simulation Settings

U > 1

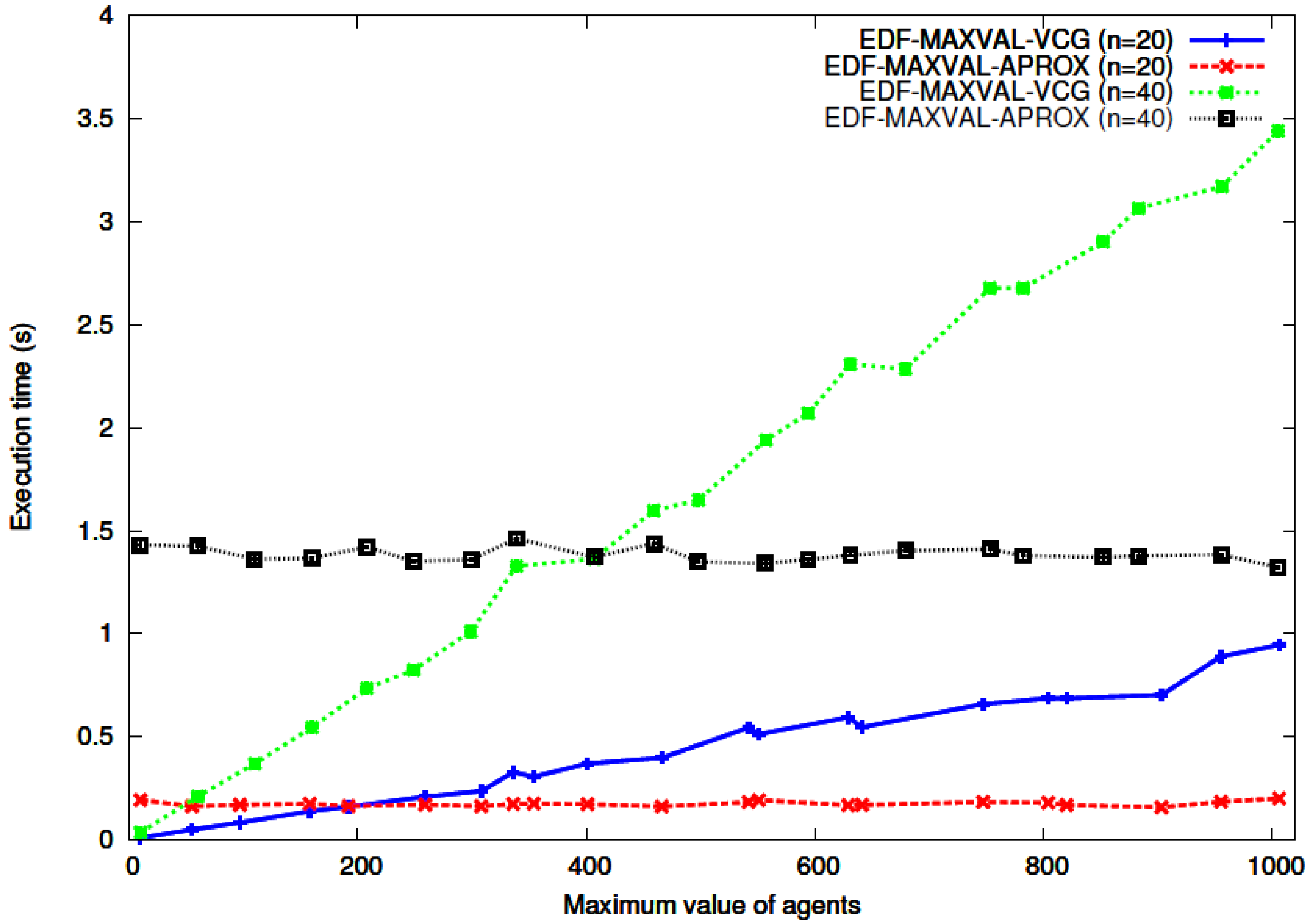
- Generate the utilizations using **UUniFast-Discard** [Davis & Burns 2009].
- Generate values using a random uniform number generator within [1, 1000].
- MATLAB environment on an 8-core Intel Core i7 (1.73GHz) machine was used.
- **Approximation Error:** $\varepsilon = 0.1$.

Frugality Ratios



- More competition, higher frugality ratio.
- Frugality ratios for exact and approx mechanism are close.

Execution Time of Mechanisms



Conclusion

- **Goal**: Introduce notion of *competition* to real-time scheduling/allocation.
  - **Reason**: systems are becoming more *open*.
  - **Challenge**: game theory often “well-behaved” utility functions.
- We extended existing algorithms to obtain **truthful** exact and approximate mechanisms with
  - *bounded* frugality ratios
  - reserve prices
- **Future Plans**:
  - Different resource owner objectives
  - Compositional systems
  - Multi-processor settings
  - Group-strategy proof mechanisms

Questions?

Thank you!

fishern@cs.wayne.edu