Energy-Aware Profit Maximizing Scheduling Algorithm for Heterogeneous Computing Systems

Kyle M. Tarplee[1],
Anthony A. Maciejewski[1], Howard Jay Siegel[1,2]

[1]Department of Electrical and Computer Engineering
[2]Department of Computer Science

Colorado State University
Fort Collins, Colorado, USA

Outline

- problem statement
- optimization problem formulation
- algorithm description
- results

Problem Statement

- static scheduling
  - single bag-of-tasks
  - task assigned to only one machine (task indivisibility)
  - machine runs one task at a time
  - known deterministic execution times
- large number of heterogeneous tasks and machines
- **goal:** maximize profit per time
  - minimize operating cost (energy)
  - minimize makespan: process the next bag-of-tasks after this one

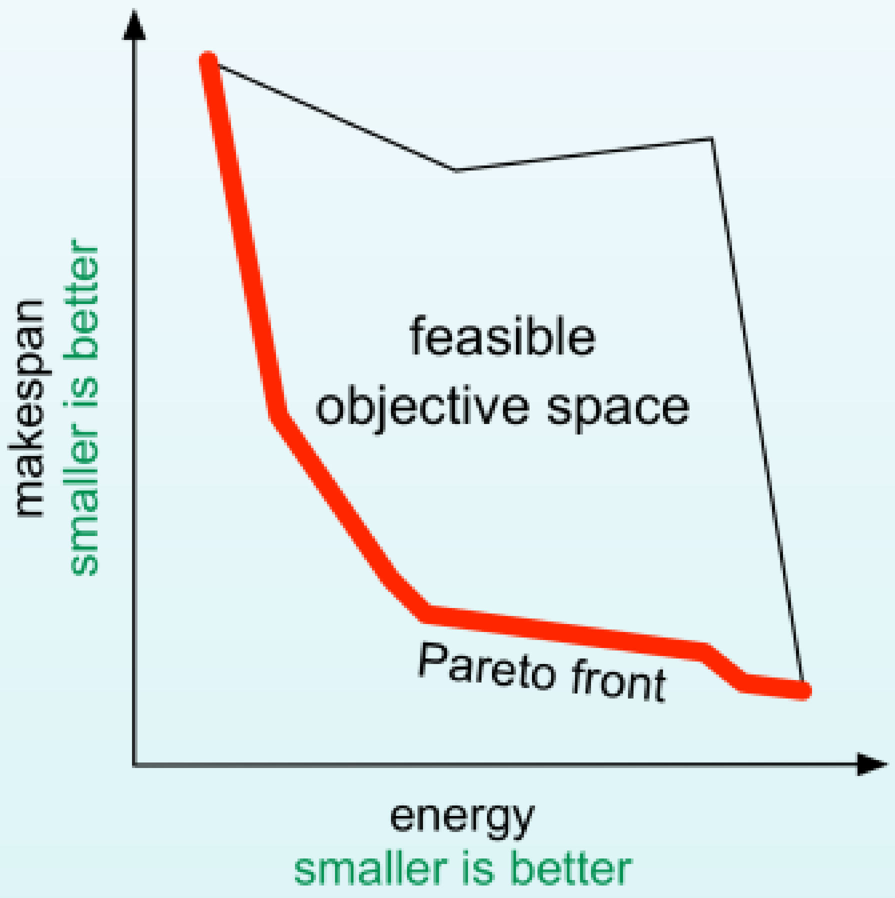
Introduction

- work has been done in minimizing
  - execution time
  - energy consumption
  - reliability
- our focus is on maximizing profit
  - important for businesses
  - combines the makespan, energy, and other cost factors
- contributions
  - monetary model for provider and client HPC
  - algorithm to efficiently find
    - maximum profit schedule
    - bounds on profit

Motivation

- software as a service (SaaS) providers maximize profits by increasing revenue and controlling costs
- example: SaaS web-based video trans-coding
  - charges customers per minute of video converted
  - task execution time is well known due to repetitive tasks executed by the provider
- **objective:** minimize cost of processing workload **and** process tasks as fast as possible

Pareto Fronts



- energy and makespan optimization
- good for system operators
- does not provide a concrete decision space for automated schedulers
- need efficient algorithms for very large-scale systems

Problem Formulation

- let p be the price (revenue) per bag-of-tasks
- let c be the cost per unit of energy
- let E be the energy consumed
- let MS be the makespan
- profit per bag-of-tasks is $p - cE$
- profit per unit time (to be maximized) is $\frac{p-cE}{MS} = \frac{p}{MS} - c\,\frac{E}{MS}$
  - revenue per unit time $-c$ times average power consumption

Problem Formulation

- let p be the price (revenue) per bag-of-tasks
- let c be the cost per unit of energy
- let E be the energy consumed
- let MS be the makespan
- profit per bag-of-tasks is $p - cE$
- profit per unit time (to be maximized) is $\frac{p - cE}{MS} = \frac{p}{MS} - c\frac{E}{MS}$
  - revenue per unit time $-c$ times average power consumption

Problem Formulation

- let p be the price (revenue) per bag-of-tasks
- let c be the cost per unit of energy
- let E be the energy consumed
- let MS be the makespan
- profit per bag-of-tasks is $p - cE$
- profit per unit time (to be maximized) is $\frac{p-cE}{MS} = \frac{p}{MS} - c\frac{E}{MS}$
    - revenue per unit time $-c$ times average power consumption

Solution Approach

- computationally expensive to compute optimal solutions for
  - minimizing makespan
  - maximizing profit (function of makespan)
- need scalable and efficient algorithms to find good schedules

Solution Approach

- computationally expensive to compute optimal solutions for
  - minimizing makespan
  - maximizing profit (function of makespan)
- need scalable and efficient algorithms to find good schedules
- proposed 3 phase algorithm:
  - solve linear optimization problem assuming tasks are divisible
  - round the solution
  - assign tasks to machines

Preliminaries

- simplifying approximation: **each task is divisible among machines**
- T_i — number of tasks of type i
- M_j — number of machines of type j
- ETC_{ij} — estimated time to compute for a task of type i running on a machine of type j
- μ_{ij} — number of tasks of type i assigned to machines of type j
  - matrix $\boldsymbol{\mu}$ is a resource allocation
  - decision variable
  - not binary but integer valued ($\mu_{ij} \gg 1$)
- finishing time of machine type j is (lower bound)

$$F_j = \frac{1}{M_j} \sum_i \mu_{ij} ETC_{ij}$$

Energy and Power

- APC_{ij} — average power consumption for a task of type i running on a machine of type j
- $E_{DL}(\boldsymbol{\mu}) = \sum_i \sum_j \mu_{ij} ETC_{ij} APC_{ij}$
- in the paper E_{DL} includes consideration of idle power
- let P_{max} be the maximum average power consumption
  - models long running average power consumption
  - useful for modeling cooling capacity

Optimization Problem

$$\underset{\boldsymbol{\mu},\, MS_{DL}}{\text{maximize}} \quad \frac{p - cE_{DL}(\boldsymbol{\mu})}{MS_{DL}}$$

subject to:

$\forall i$	$\sum_j \mu_{ij} = T_i$	task constraint
$\forall j$	$F_j \leq MS_{DL}$	machine finishing time constraint
$\forall i, j$	$\mu_{ij} \geq 0$	assignments must be non-negativ
	$\frac{E_{DL}}{MS_{DL}} \leq P_{max}$	power constraint (optional)

- recall:
  - F_j is finishing time
  - μ_{ij} is number of tasks

Conversion to a Linear Program

- objective and power constraint are non-linear
- ratios of decision variables, μ_{ij} and MS_{DL}
- constraints can be converted to use ratios of μ_{ij} and MS_{DL}
- variable substitution
  - $z_{ij} \leftarrow \frac{\mu_{ij}}{MS_{DL}}$ is the average tasks per unit time
  - $r \leftarrow \frac{1}{MS_{DL}}$ is the number of bag-of-tasks per unit time
- average power consumption becomes $\bar{P} = \sum_i \sum_j z_{ij} ETC_{ij} APC_{ij}$

Transformed Linear Program

Non-Linear Problem

$$\underset{\mu,\, MS_{DL}}{\text{maximize}} \quad \frac{p - cE_{DL}(\boldsymbol{\mu})}{MS_{DL}}$$

subject to:

$$\forall i \qquad \sum_j \mu_{ij} = T_i$$

$$\forall j \qquad F_j \leq MS_{DL}$$

$$\forall i, j \qquad \mu_{ij} \geq 0$$

$$\frac{E_{DL}}{MS_{DL}} \leq P_{max}$$

Transformed Linear Program

Non-Linear Problem

$$\underset{\mu,\, MS_{DL}}{\text{maximize}} \quad \frac{p - cE_{DL}(\mu)}{MS_{DL}}$$

subject to:

$$\forall i \qquad \sum_j \mu_{ij} = T_i$$

$$\forall j \qquad F_j \leq MS_{DL}$$

$$\forall i, j \qquad \mu_{ij} \geq 0$$

$$\frac{E_{DL}}{MS_{DL}} \leq P_{max}$$

$\Longrightarrow$

Linear Problem

$$\underset{z,\, r}{\text{maximize}} \quad pr - c\bar{P}$$

subject to:

$$\forall i \qquad \sum_j z_{ij} = T_i r$$

$$\forall j \qquad \frac{1}{M_j} \sum_i z_{ij} ETC_{ij} \leq 1$$

$$\forall i, j \qquad z_{ij} \geq 0$$

$$r \geq 0$$

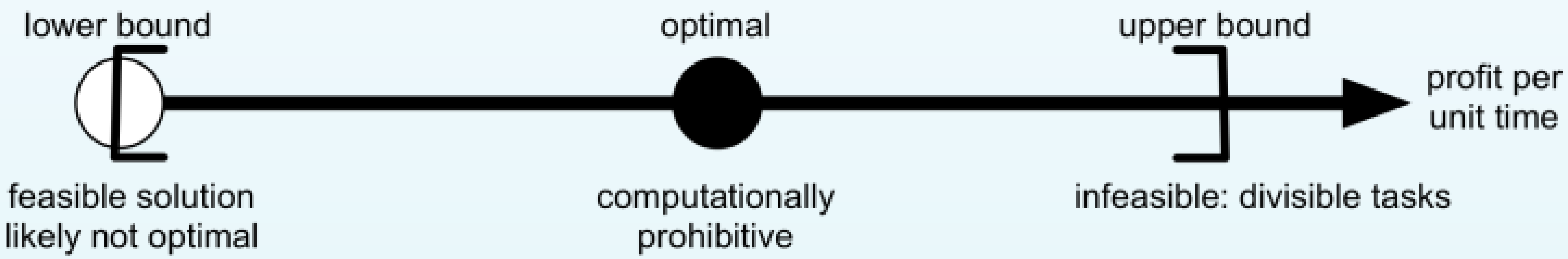
$$\bar{P} \leq P_{max}$$

Algorithm

- solve linear program and compute $\mu_{ij} = \frac{z_{ij}}{r}$ and $MS_{DL} = \frac{1}{r}$
- rounding algorithm (per task type)
  - rounds the number of tasks of each type assigned to each machine type
  - find nearest integer solution while satisfying constraints
- local assignment algorithm (per machine type)
  - assign tasks to individual machines
  - greedy algorithm to minimize makespan

- recall:
  - z_{ij} is average tasks per unit time
  - r is number of bag-of-tasks per unit time

Bounds



- upper bound
  - many Pareto efficient solutions to the relaxation of the energy and makespan optimization problem
  - maximum profit of all those solutions

Alternative Objective Formulation

- collapse p and c into a single intuitive parameter
- let E_{min} be the energy of the minimum energy solution
- let profit ratio $\gamma = \frac{p}{cE_{min}}$
  - γ is unitless
  - $\gamma \geq 0$ is realizable
  - $\gamma > 1 \Rightarrow$ positive profit is achievable

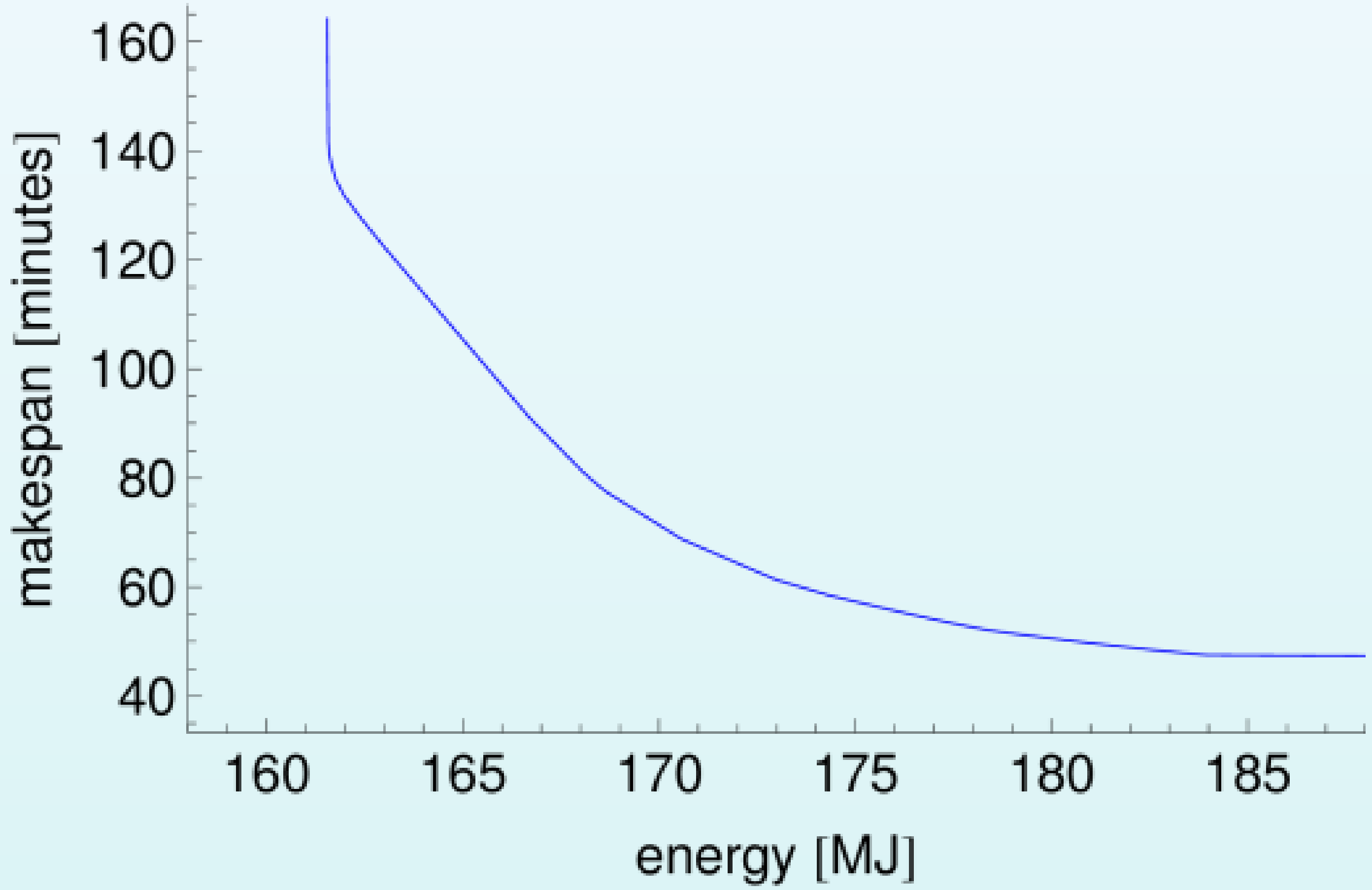
- recall:
  - p is price per bag-of-tasks
  - c is cost per unit of energy

Simulation Setup

- heterogeneous tasks and machines
- 11,000 tasks each is one of 30 task types
- 360 machines each is one of 9 machine types

Max Profit Solutions

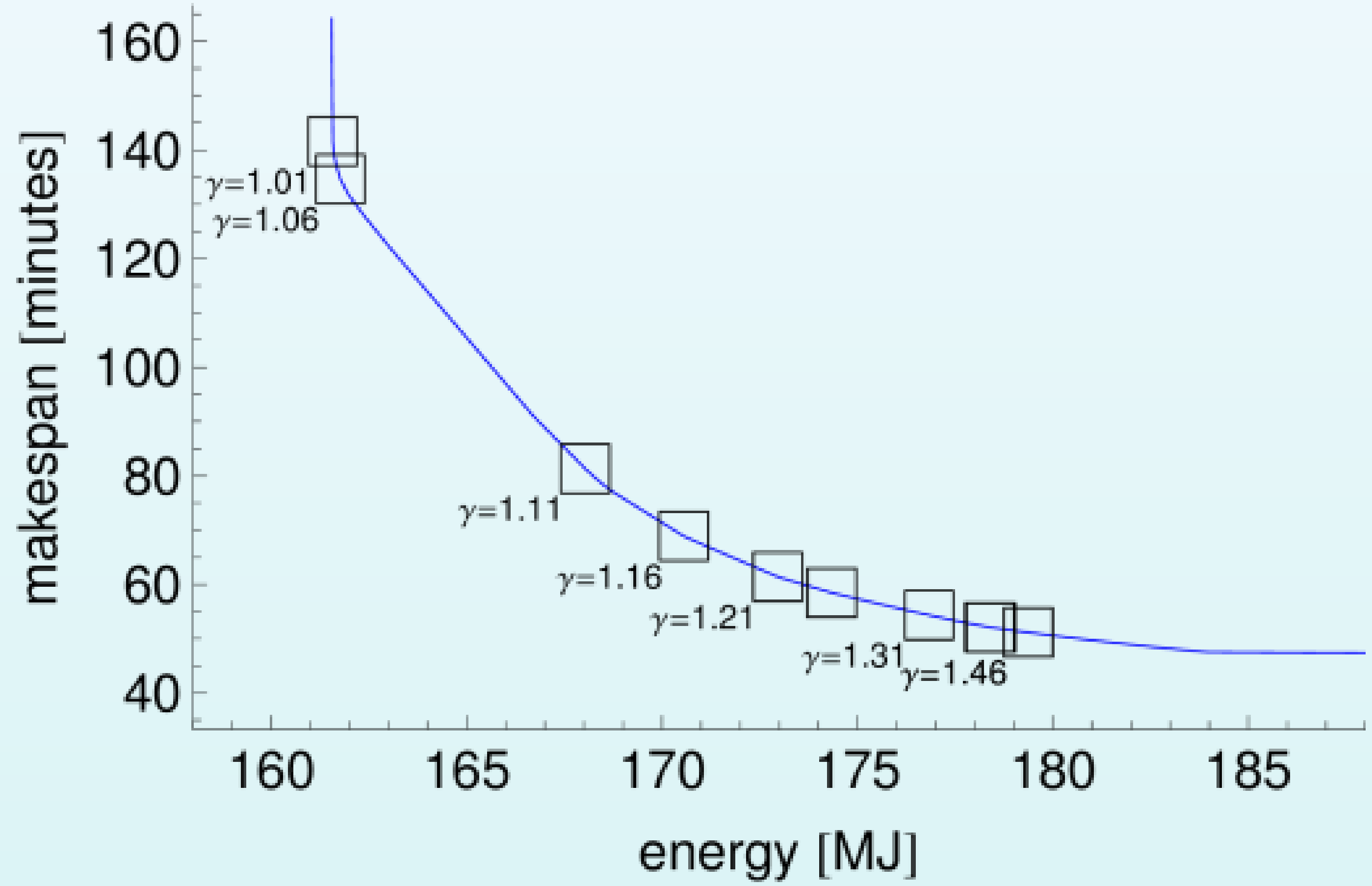
Sweeping Profit Ratio



– Pareto front lower bound

Max Profit Solutions

Sweeping Profit Ratio



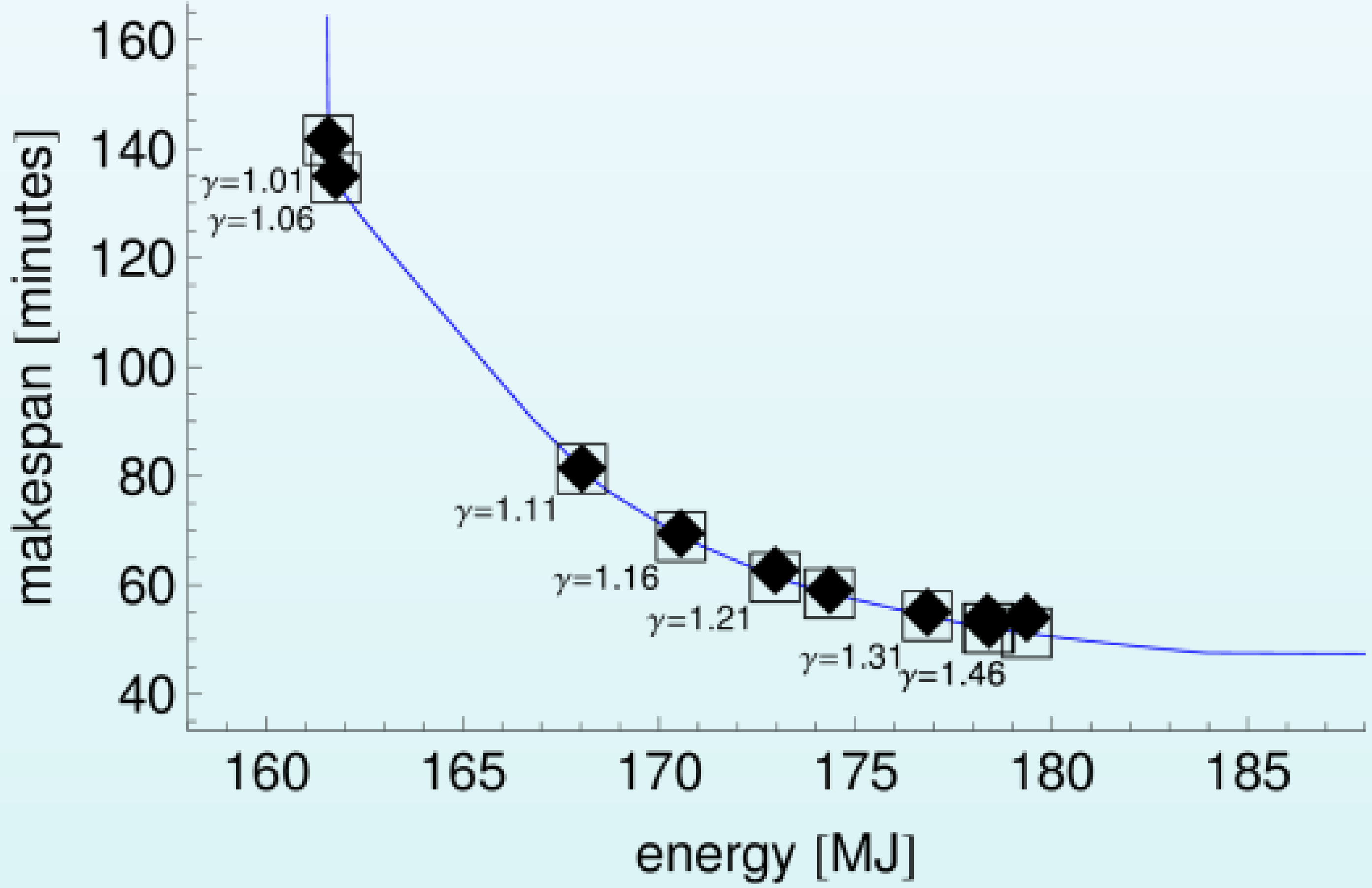
– Pareto front lower bound

□ profit upper bound (infeasible)

- recall: $\gamma = \frac{p}{cE_{min}}$ is the profit ratio

Max Profit Solutions

Sweeping Profit Ratio



– Pareto front lower bound

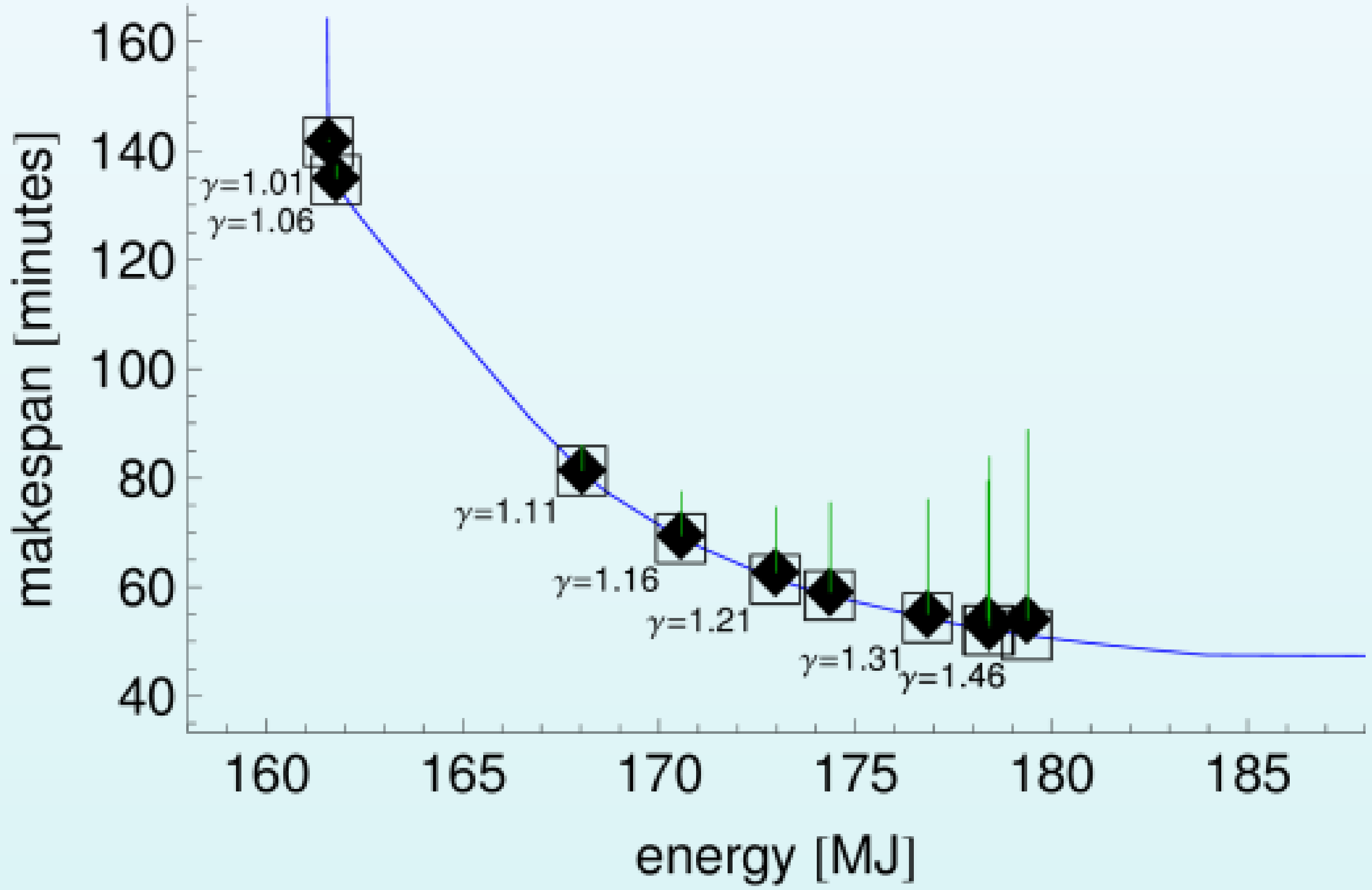
□ profit upper bound (infeasible)

◆ profit lower bound (feasible)

- recall: $\gamma = \frac{p}{cE_{min}}$ is the profit ratio

Max Profit Solutions

Sweeping Profit Ratio



– Pareto front lower bound

□ profit upper bound (infeasible)

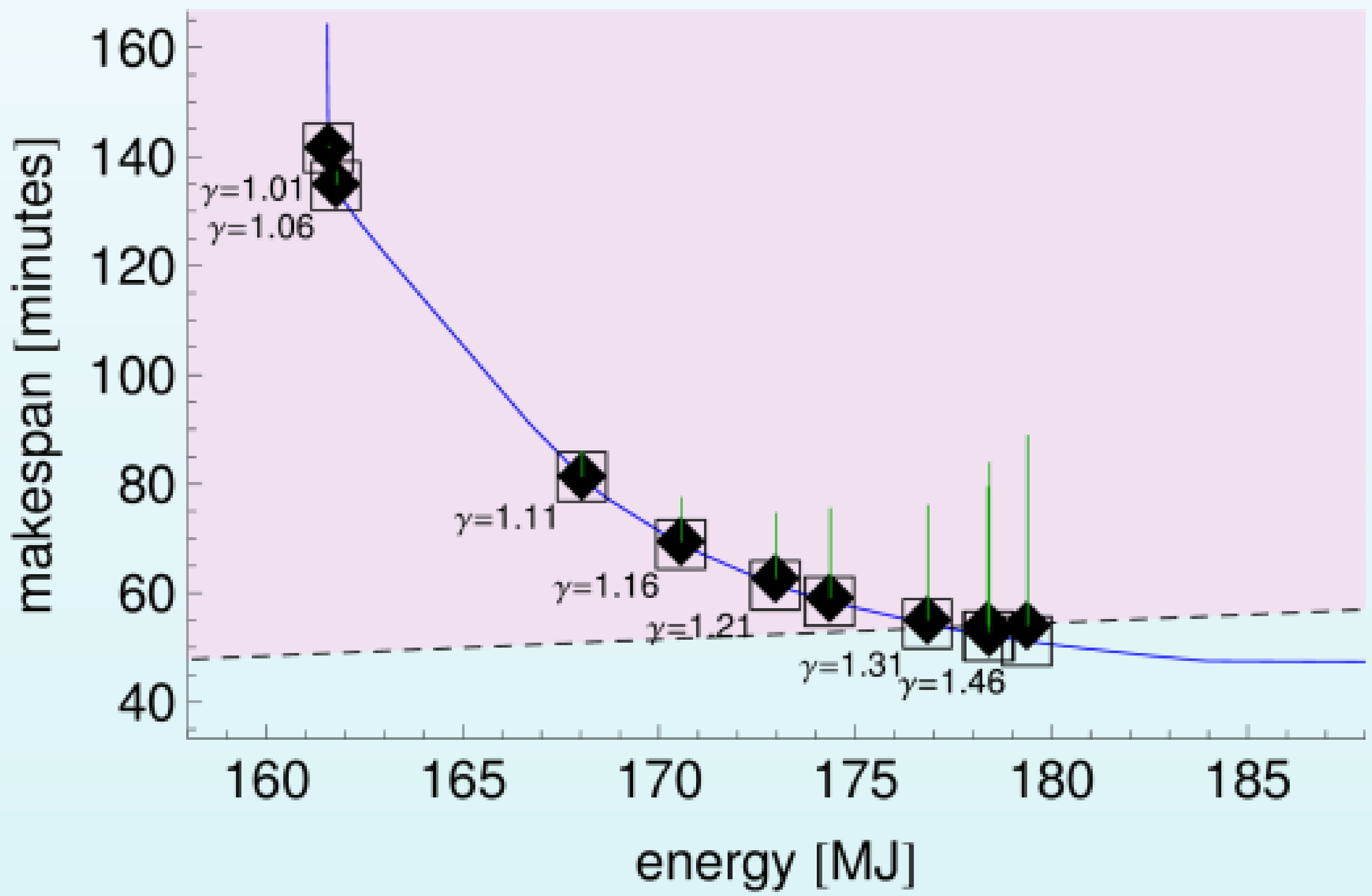
◆ profit lower bound (feasible)

| profit per unit time

- recall: $\gamma = \frac{p}{cE_{min}}$ is the profit ratio

Max Profit Solutions

Sweeping Profit Ratio



– Pareto front lower bound

□ profit upper bound (infeasible)

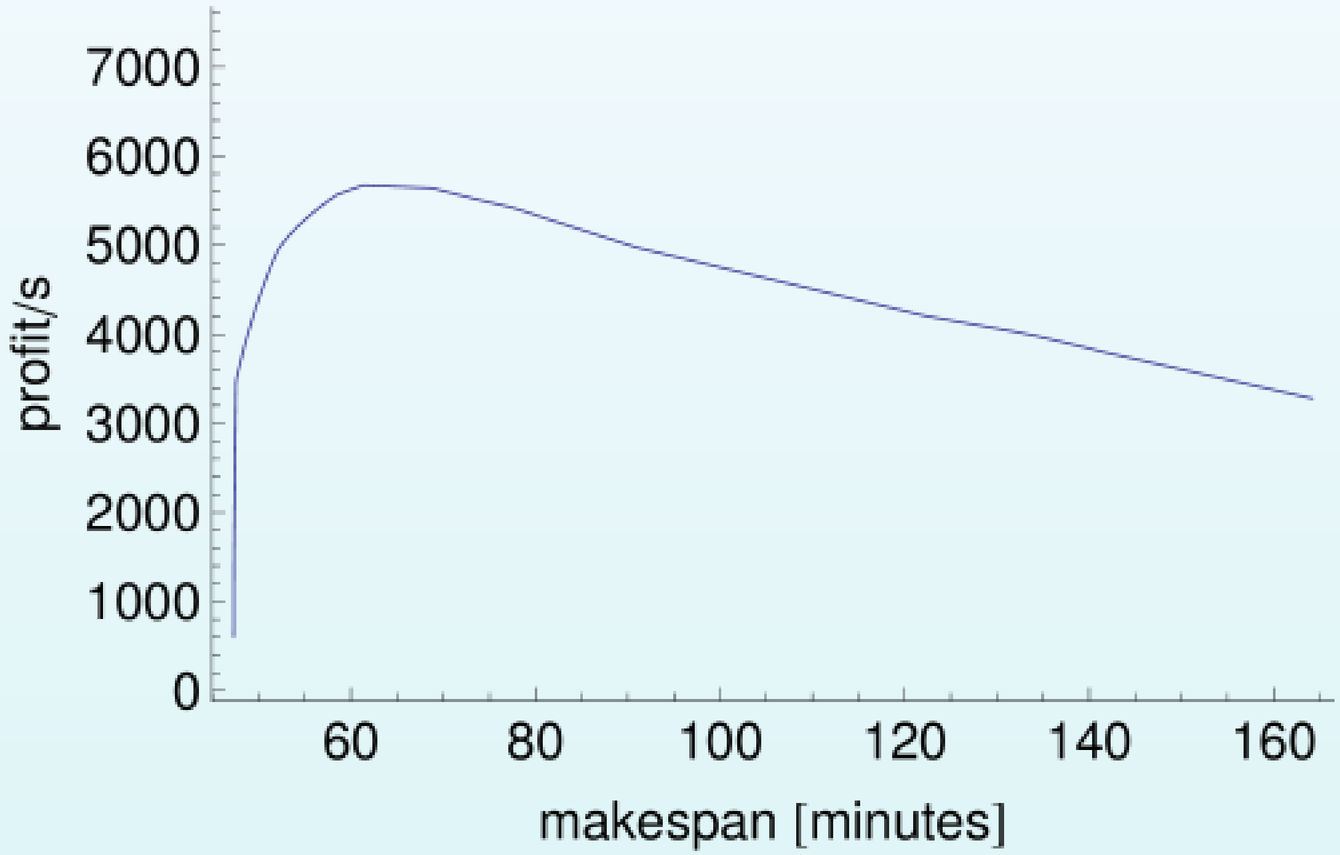
◆ profit lower bound (feasible)

| profit per unit time

-- power constraint (55 KW)

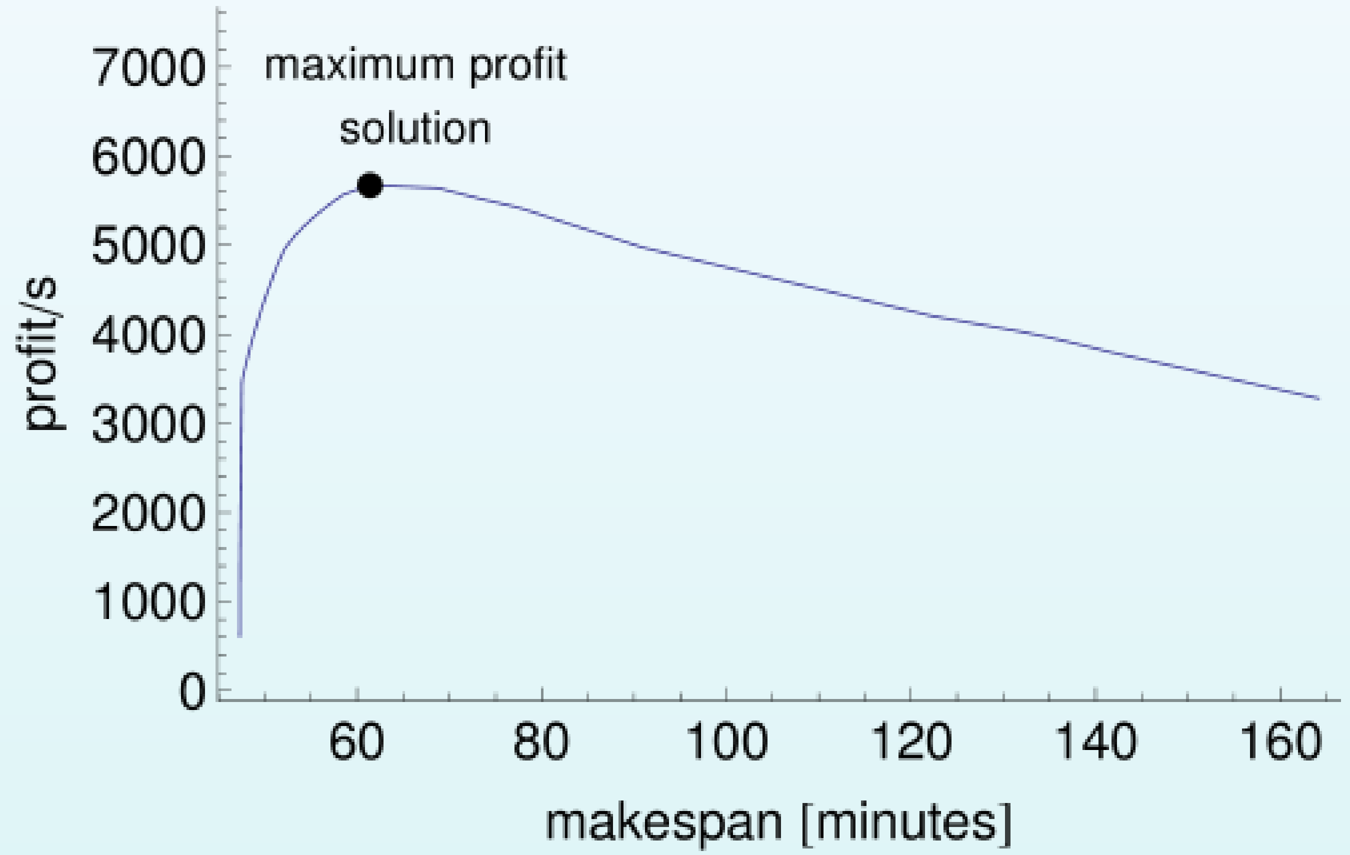
- recall: $\gamma = \frac{p}{cE_{min}}$ is the profit ratio

Profit Rate vs Makespan



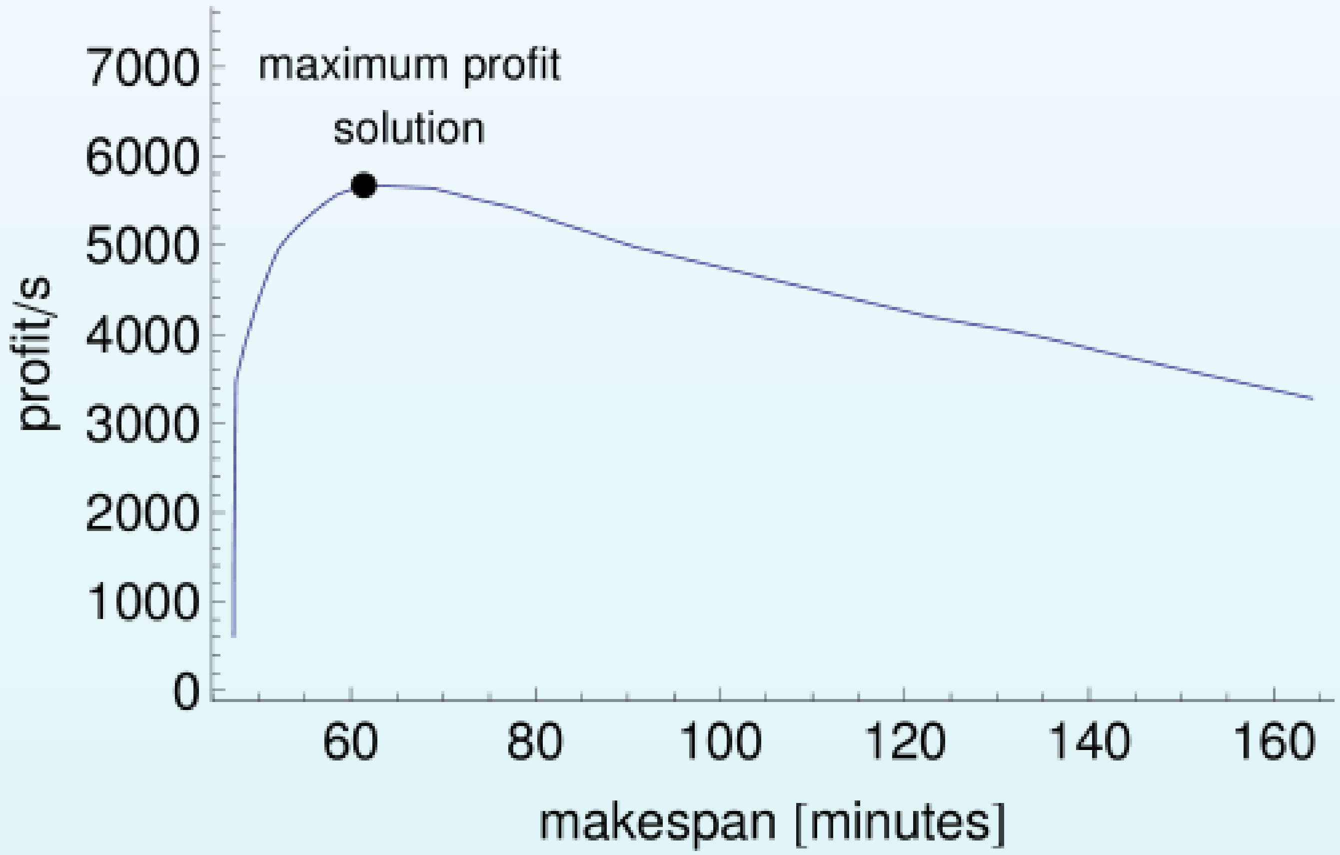
- profit ratio = 1.2
- 11,000 tasks
- Pareto front generation: 173 ms

Profit Rate vs Makespan



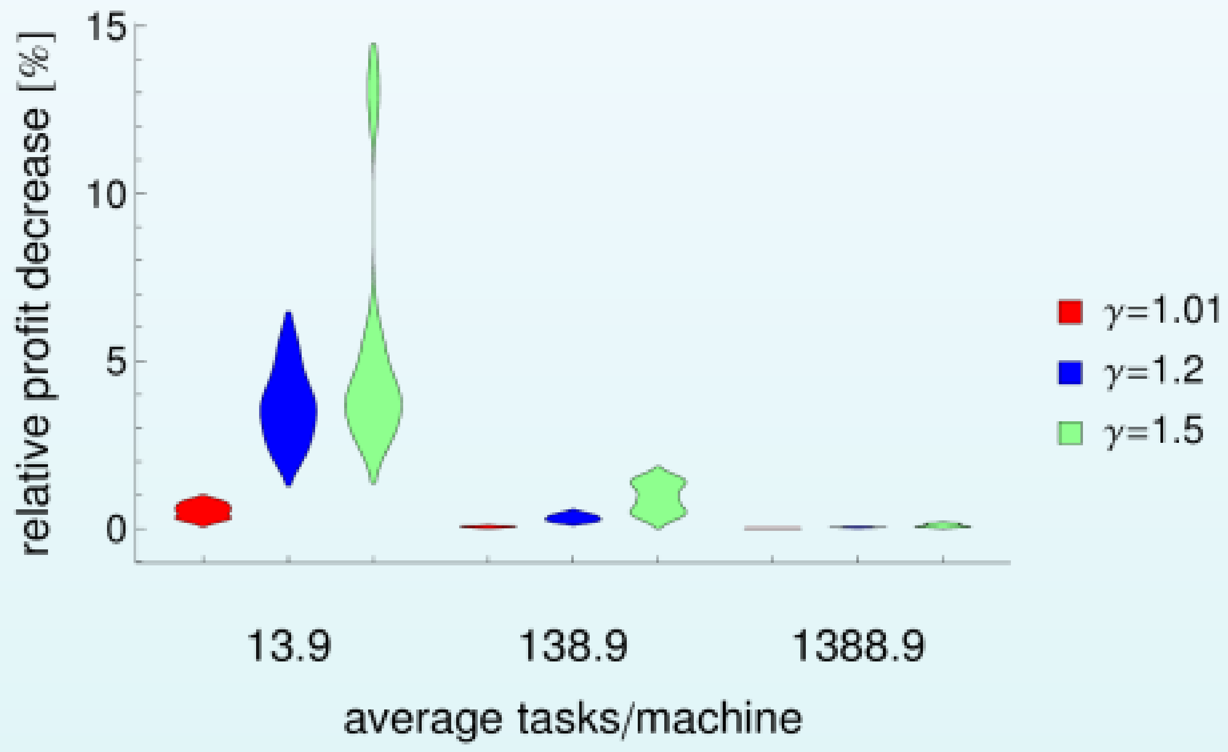
- profit ratio = 1.2
- 11,000 tasks
- Pareto front generation: 173 ms
- single max profit solve: 2 ms

Profit Rate vs Makespan



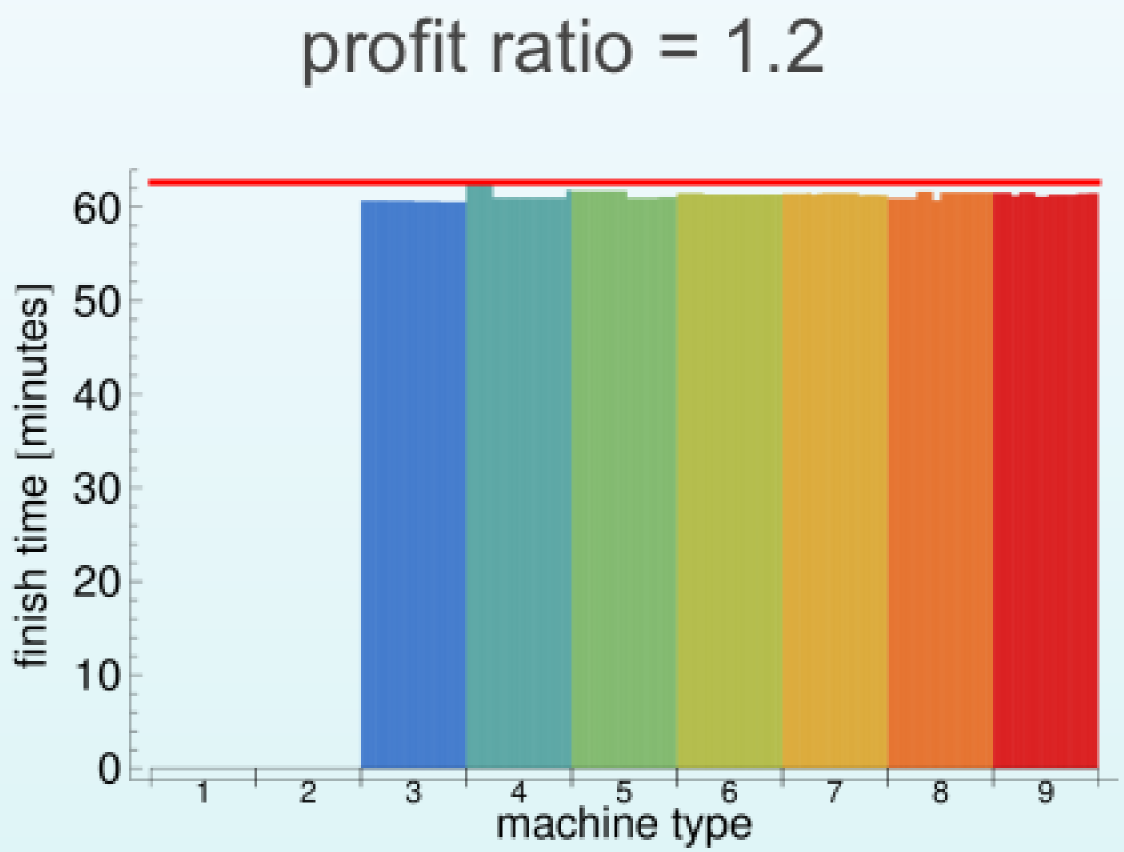
- profit ratio = 1.2
- 11,000 tasks
- Pareto front generation: 173 ms
- single max profit solve: 2 ms
  - if 1 million tasks: 70 ms

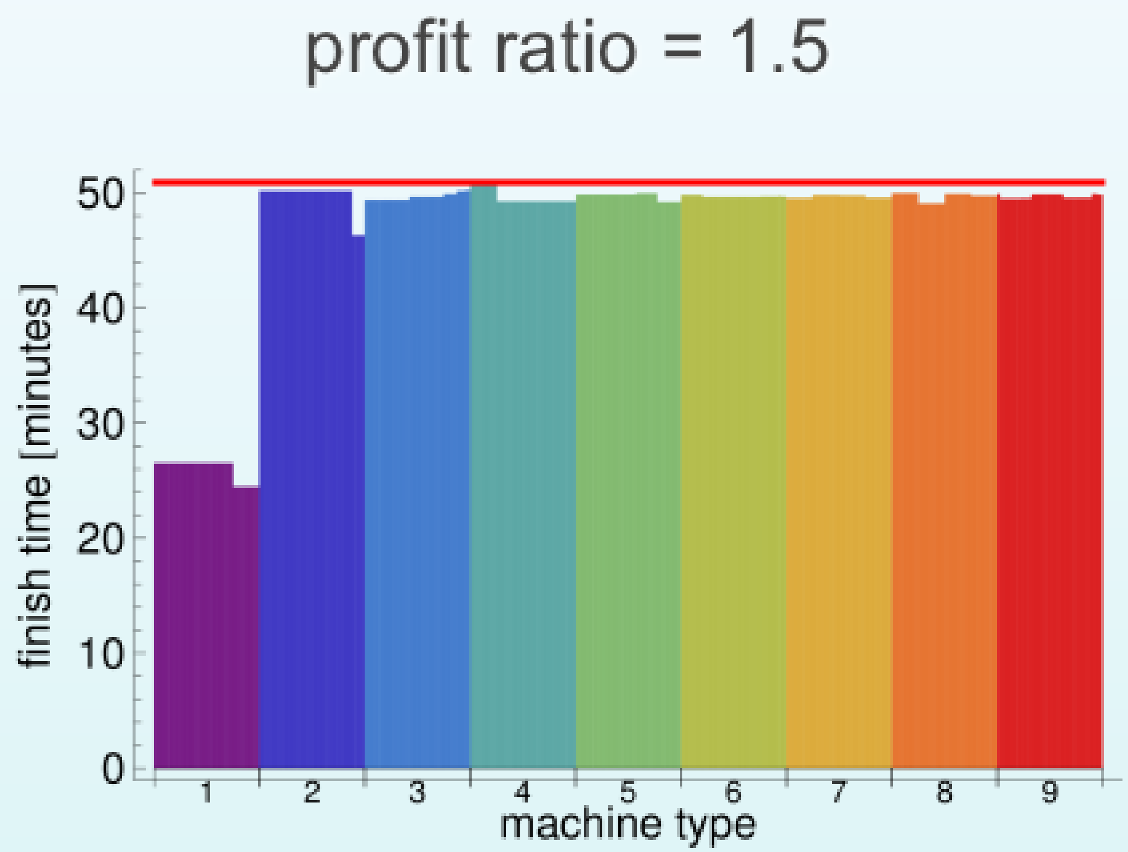
Relative Profit Rate vs Number of Tasks



- 100 Monte Carlo runs sampling over the task types
- relative decrease in profit = $100 \frac{\text{profit}_{DL} - \text{profit}_{full}}{\text{profit}_{DL}}$
- width of glyphs represent the probability density

Finishing Times





- machine type 1 and 2 are less power efficient
- use when price is high thus emphasizing makespan minimization

Applicability

- millions of tasks and tens of thousands of machines
- scheduler execution times are sub-second
- online batch mode scheduling for uncertain
  - execution times
  - arrival rates
- schedule to cores instead of machines

Conclusions

- fast profit maximizing scheduling algorithm
- scalable to very large systems
- provides tight lower and upper bounds on the profit per unit time
- easily applied to online batch mode scheduling

Questions