Highly Scalable Algorithms For Scheduling Tasks and Provisioning Machines on Heterogeneous Computing Systems

Kyle M. Tarplee

Outline

- overview of prior work
- Pareto fronts for energy and makespan
- resource provisioning
- future directions

Status

- Fall 2009: started (one class per semester)
- Spring 2012: finished last class
- Summer 2012: started research
- Fall 2012: qualifier
- Spring 2014: prelim
- Spring 2015: final defense

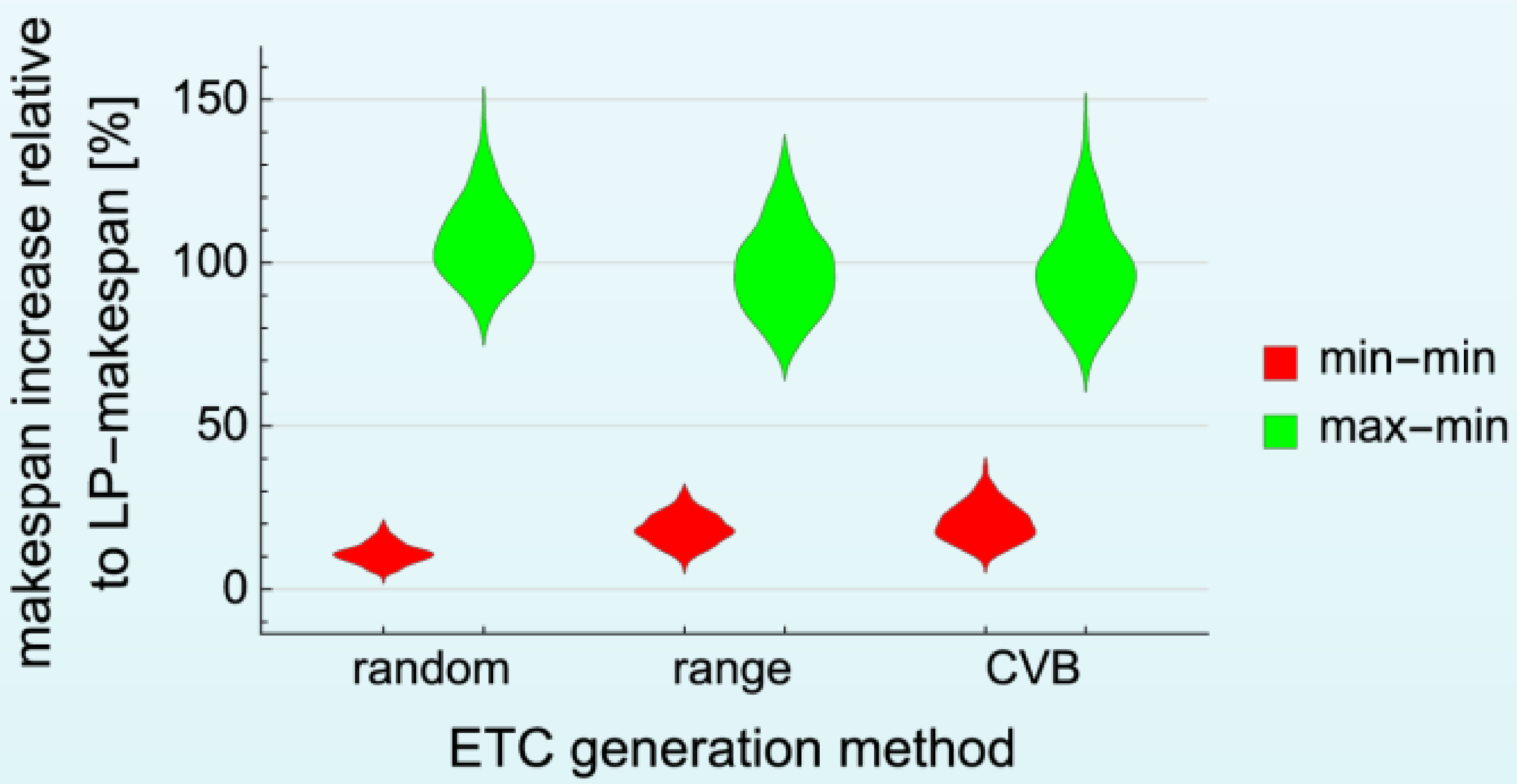
Minimum Energy and Makespan Scheduling

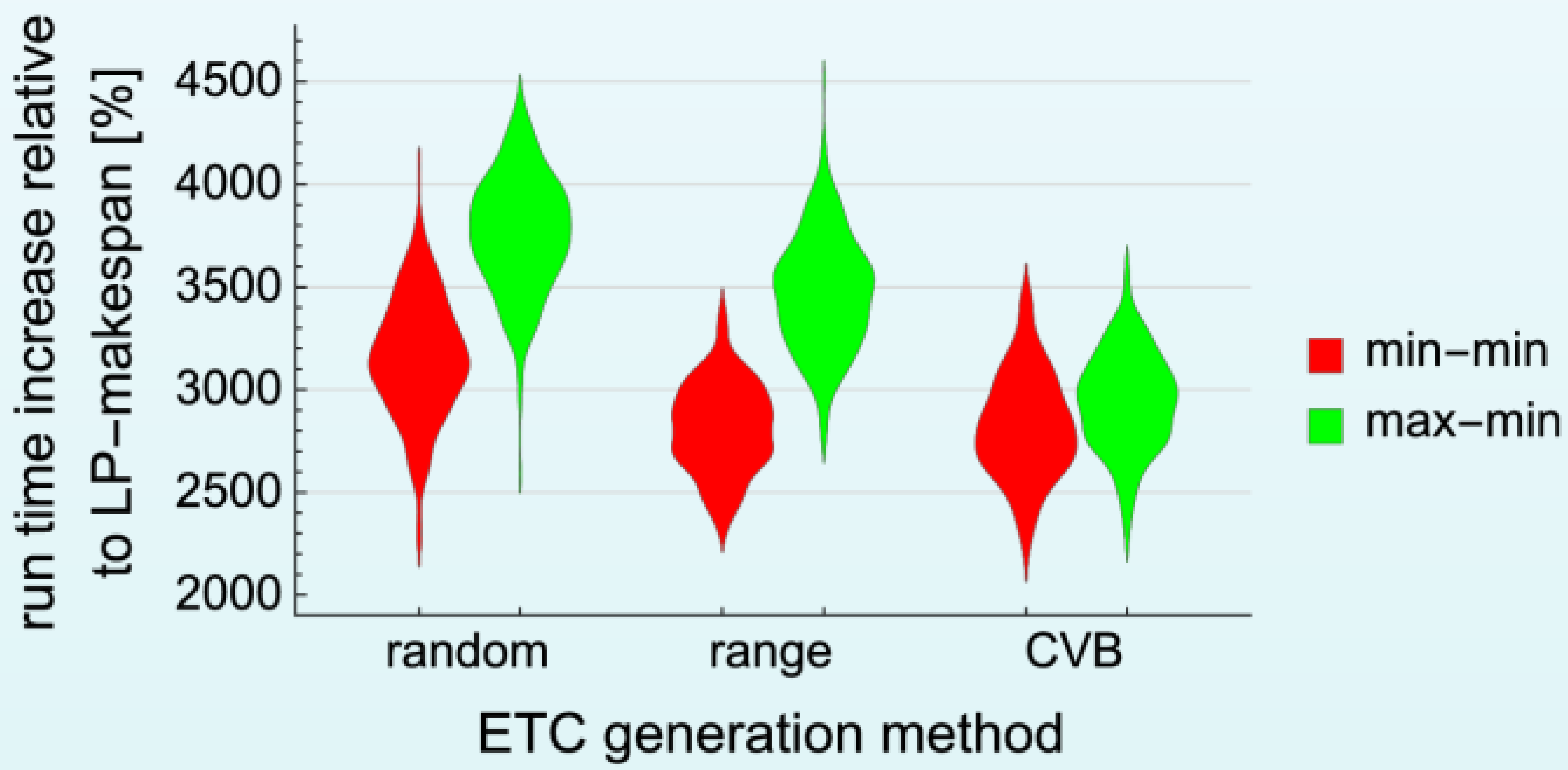
Publications

- **Efficient and Scalable Computation of the Energy and Makespan Pareto Front for Heterogeneous Computing Systems.** Kyle M. Tarplee, Ryan Friese, Anthony A. Maciejewski, and Howard Jay Siegel, 6th Workshop on Computational Optimization (WCO 2013), in the proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS 2013). Krakow, Poland, Sep. 2013.
  - presentation (2013-09-08)
  - best paper award: 2013 Zdzislaw Pawlak Best Paper Award, by the Award Committee of the 8th Symposium on Advances in Artificial Intelligence and Applications
- **Efficient and Scalable Pareto Front Generation for Energy and Makespan in Heterogeneous Computing Systems.** Kyle M. Tarplee, Ryan Friese, Anthony A. Maciejewski, and Howard Jay Siegel, in Recent Advances in Computational Optimization, Studies in Computational Intelligence Series, Springer, 2015
- **Scalable Linear Programming Based Resource Allocation for Makespan Minimization in Heterogeneous Computing Systems.** Kyle M. Tarplee, Ryan Friese, Anthony A. Maciejewski, and Howard Jay Siegel, under review (JPDC), response to reviewers submitted
- **Energy and Makespan Tradeoffs in Heterogeneous Computing Systems using Efficient Linear Programming Techniques.** Kyle M. Tarplee, Ryan Friese, Anthony A. Maciejewski, and Howard Jay Siegel, under review (IEEE TPDS), working on response to reviewers

Makespan and Run Time of Min-Min and Max-Min Relative to LP-makespan

- 200 random environments each
- 10 machine types and 1,000 machines
- 15 task types and 1,000,000 tasks

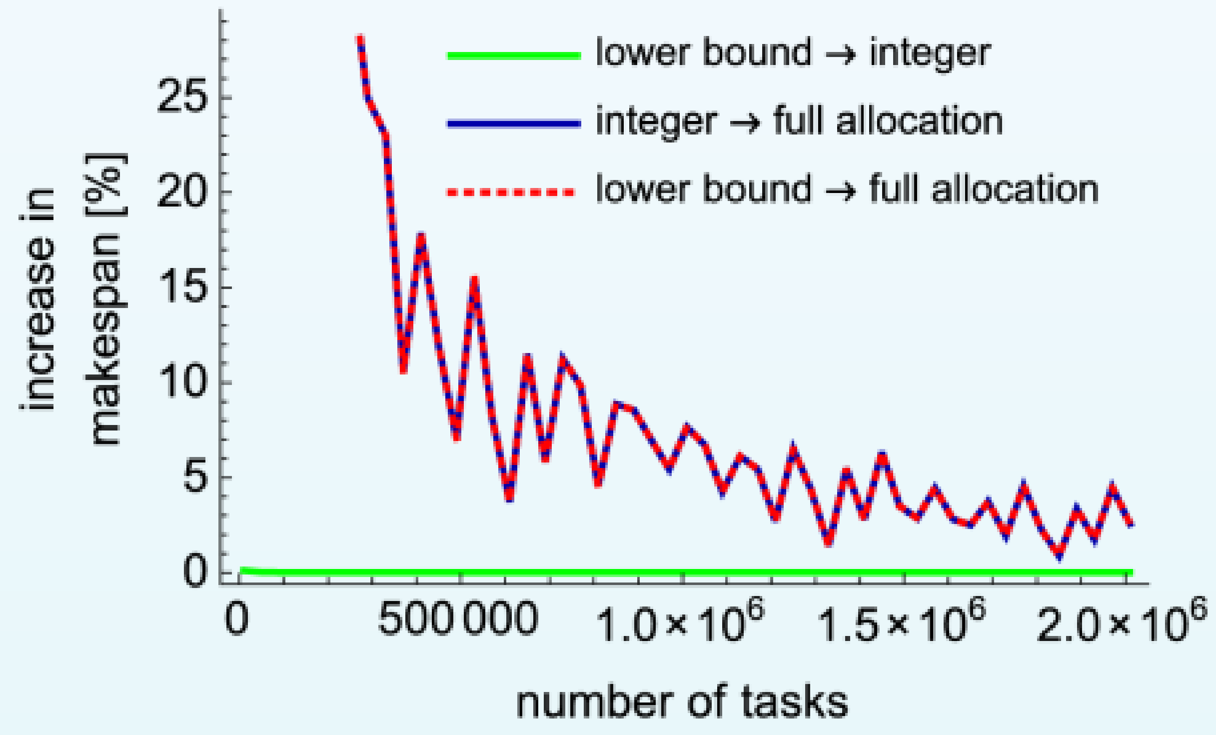




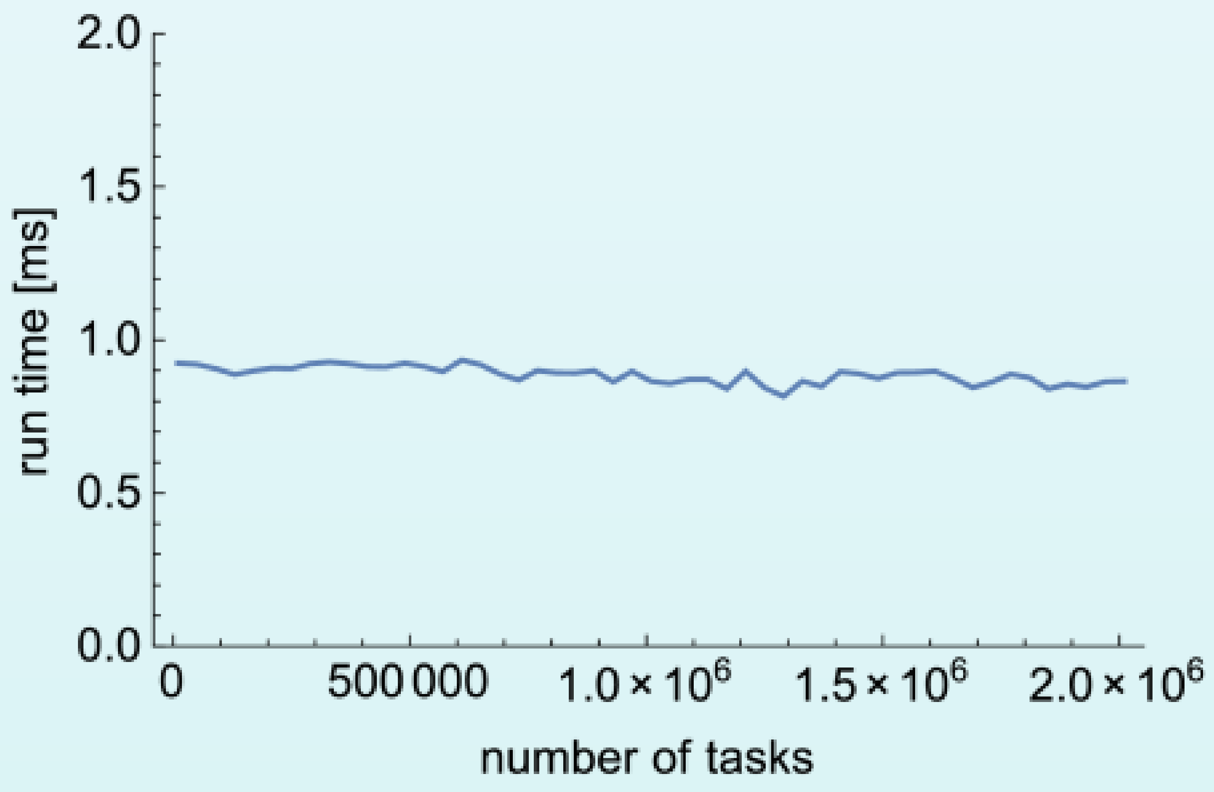
- LP-makespan algorithm takes 64 ms
- for ten million tasks and ten thousand machines
    - LP-makespan takes 0.87 s
    - min-min takes 476 s
    - min-min makespan is longer than LP-makespan

Impact of the Number of Tasks

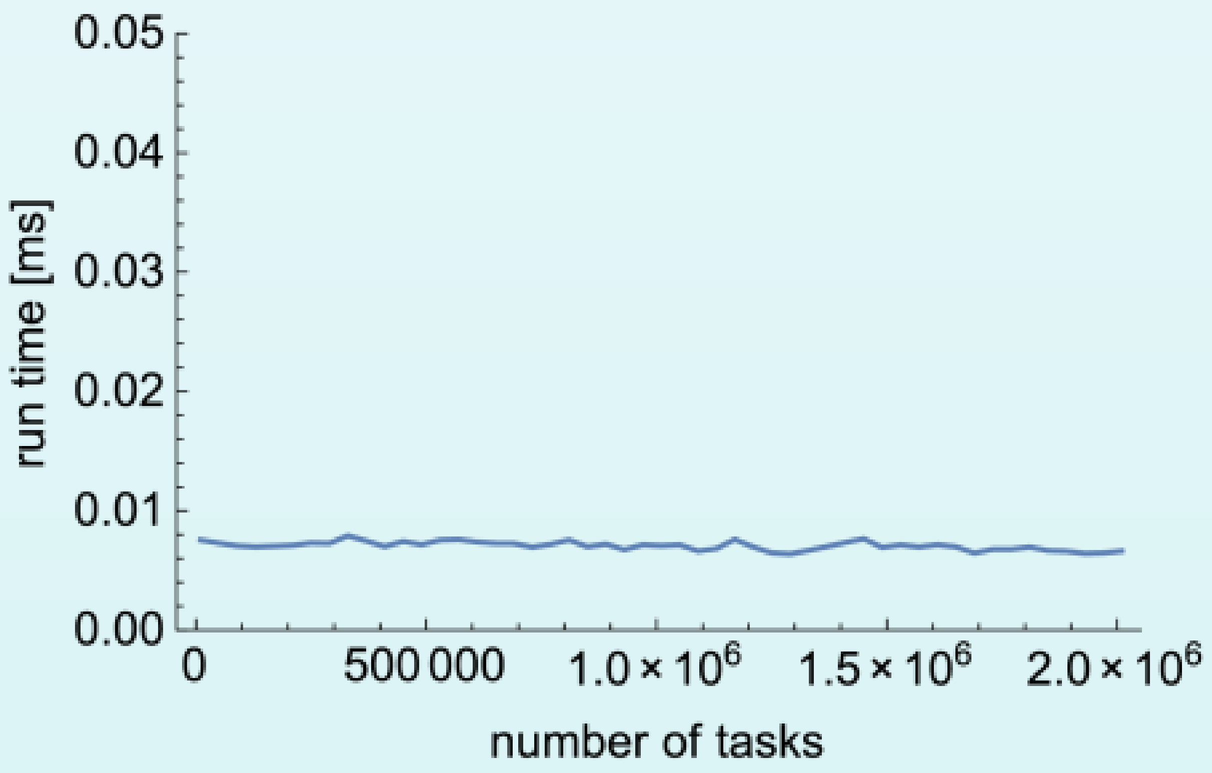
- 9 machine types and 36,000 machines
- 30 task types and 1,100,000 tasks
- averages of 50 trials
- (not shown) 100 million tasks: 8.4 s



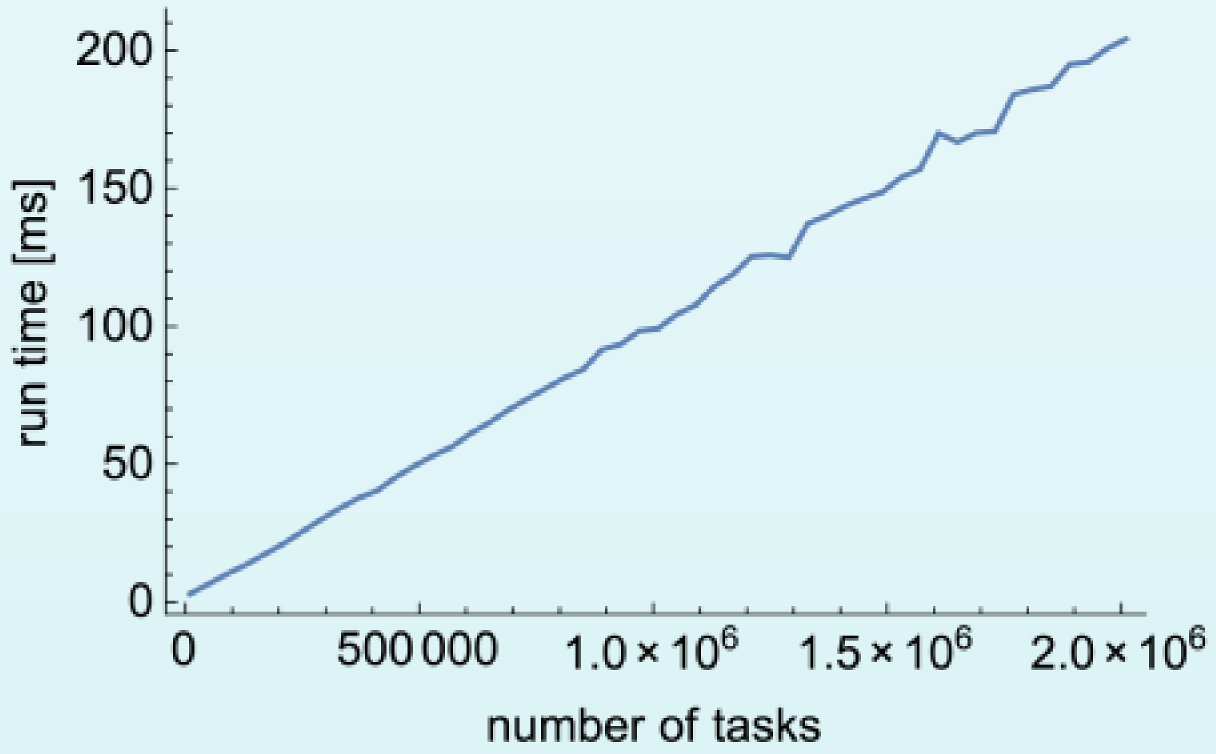
LP



round near



local assignment



Pareto Fronts

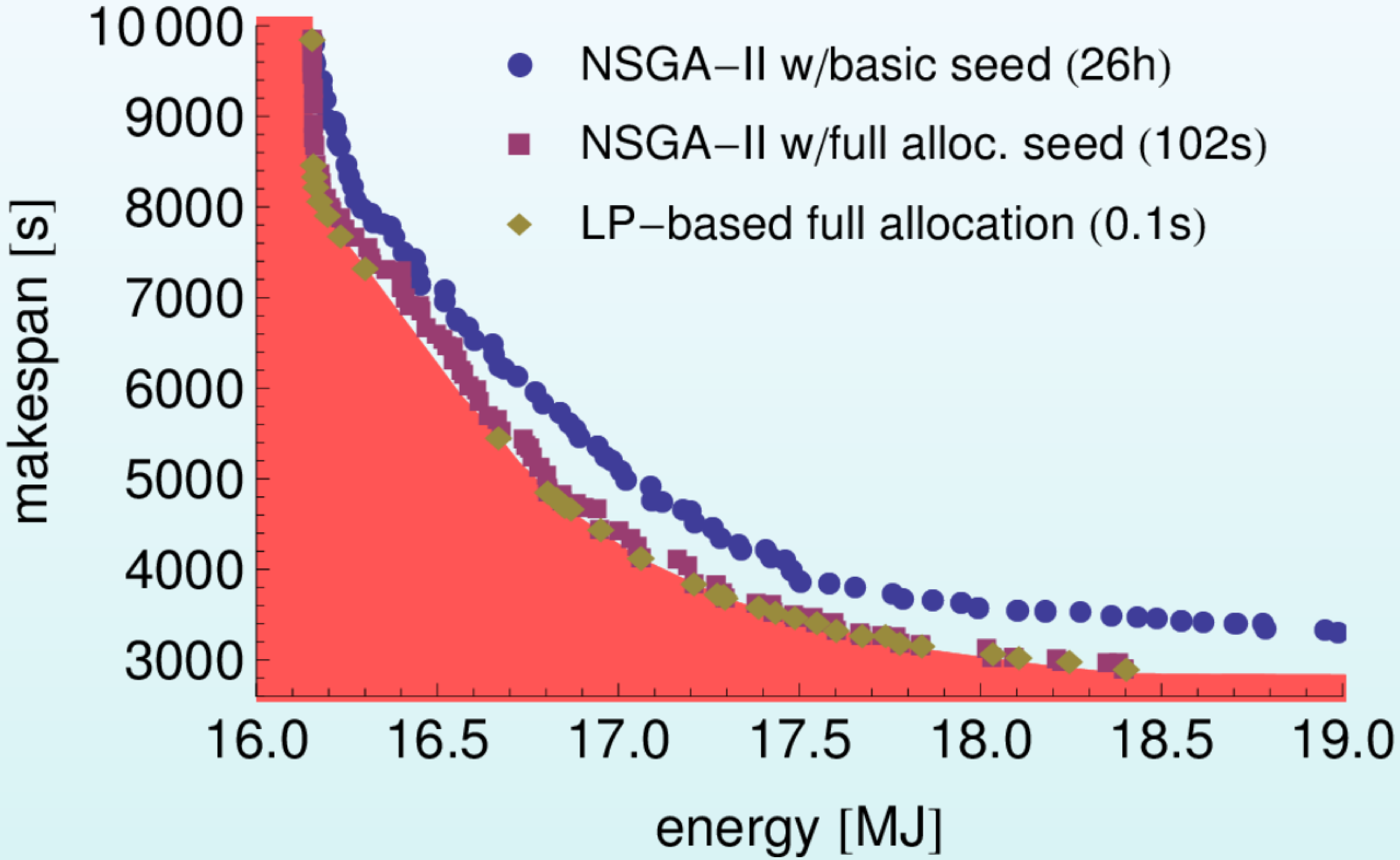
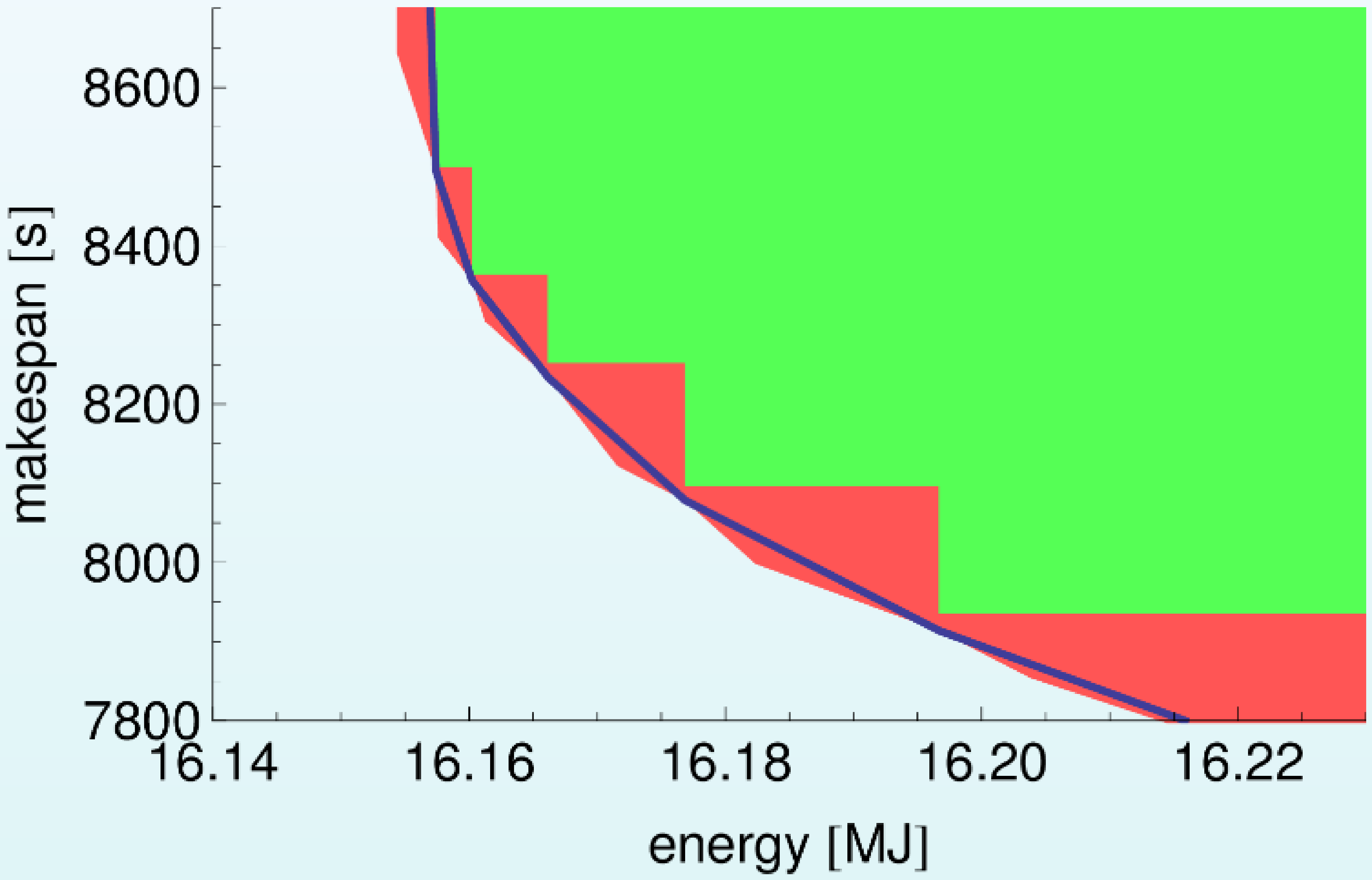
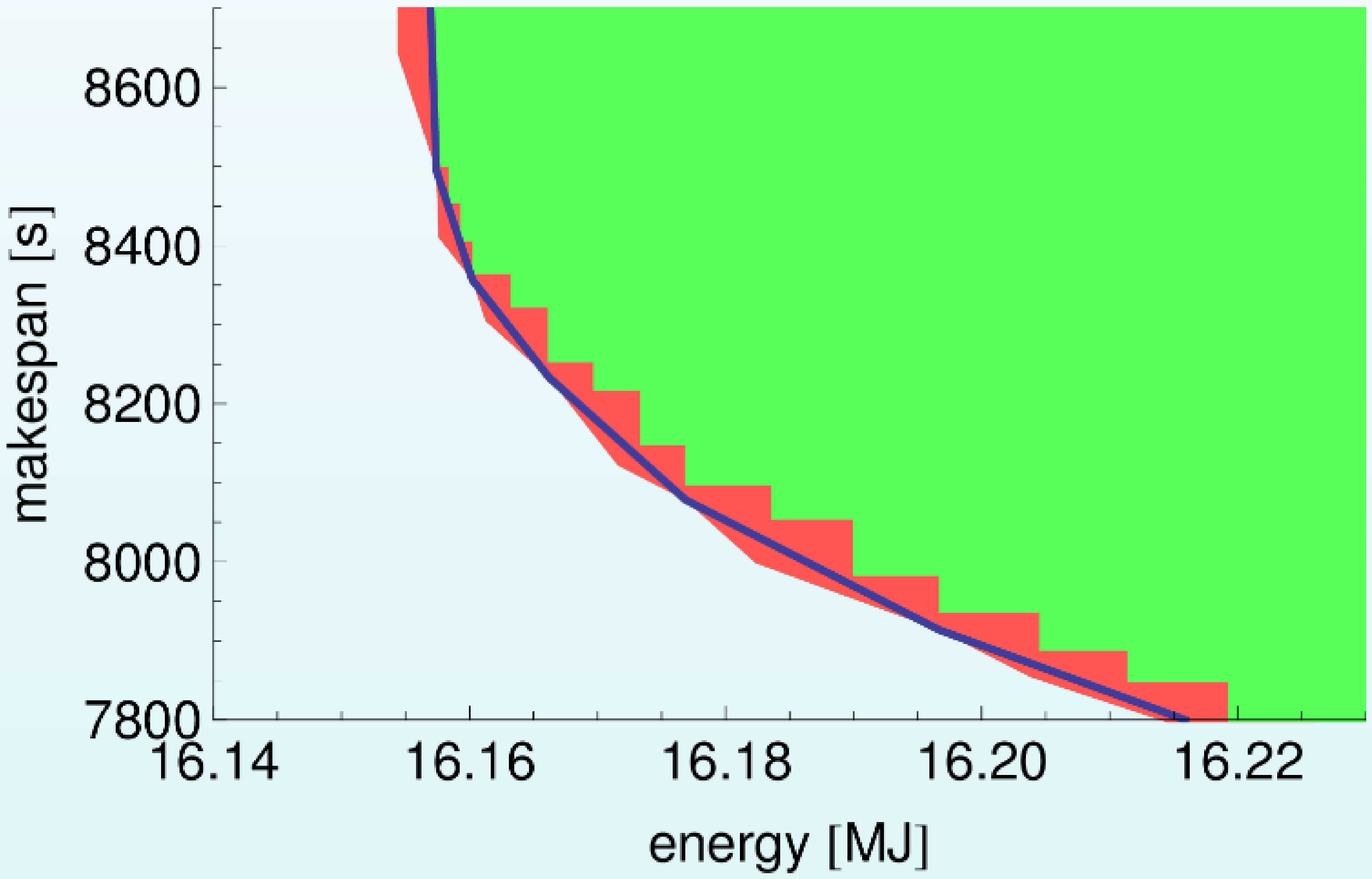


Illustration of the Regions

LP-Based



LP-Based with Convex Fill



Results

Area Between Bounds

algorithm	9 machine type	6 machine type	2 machine type	10 machine type
NSGA-II	2149 MJ s	1351 MJ s	115 MJ s	2655 KJ s
LP-based	684 MJ s	339 MJ s	63 MJ s	1011 KJ s
NSGA-II seeded	436 MJ s	306 MJ s	53 MJ s	851 KJ s
LP-based with convex fill	231 MJ s	238 MJ s	38 MJ s	762 KJ s

- NSGA-II seeded with LP-based improves on LP-based
- convex fill produces the tightest bound

Maximum Profit Scheduling

Publications

- **Energy-Aware Profit Maximizing Scheduling Algorithm for Heterogeneous Computing Systems.** Kyle M. Tarplee, Anthony A. Maciejewski, and Howard Jay Siegel, Extreme Green and Energy Efficiency in Large Scale Distributed Systems Workshop (ExtremeGreen 2014), cosponsors: IEEE Computer Society and the ACM, in the proceedings of the 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2014), Chicago, IL, May 2014
    - presentation (2014-05-26)

- contributions:
    - model for two-party monetary-based HPC environment
    - scalable and efficient algorithm for computing near-optimal maximum profit schedule
    - bounds on the achievable profit for a given HPC environment
- extended by a researcher in China (he read my paper and liked it!)

Resource Provisioning

Publications

- **Robust Performance-Based Resource Provisioning Using a Steady-State Model for Multi-Objective Stochastic Programming.** Kyle M. Tarplee, Anthony A. Maciejewski, and Howard Jay Siegel, under review (IEEE Special Issue on Many-Task Computing)

Resource Provisioning

Motivation

- traditional strategies are insufficient
  - machine utilization → in over subscribed systems all machines will have full utilization
  - number of tasks executed → not optimal if the scheduler is imperfect
  - price/performance → ignores aspects of workload and hardware
  - more of the same → current workload can be different than desired/future workload
- desire an efficient algorithm to "optimally" and "robustly" determine how many of each type of new machines to add to the system
- applications:
  - purchasing physical machines
  - cloud resource provisioning (instantiating virtual machines)

Resource Provisioning

Problem Statement

- **given** high-level HPC system description and prices of machine types
- **find** number of each machine type that should be purchased (or sold)

Approach

- multiobjective optimization problem
  - maximize reward (performance)
  - minimize cost
  - minimize failure rate (maximize reliability)
  - minimize power

- build on steady-state problem formulation from Linear Programming Affinity Scheduling (LPAS)
- steady-state schedule is a by-product of the optimization
- stochastic optimization used to handle uncertainty in parameters

Problem Formulation

- let λ_i be the arrival rate tasks of type i
- let $\tilde{p_{ij}}$ is the number of machines running tasks of type i on machines of type j
- let r_i be the reward for processing a task of type i
- $\frac{1}{ETC_{ij}}$ is the expected computation speed (tasks per second)
- task execution rate for task type i is given by $\sum_j \frac{1}{ETC_{ij}} \tilde{p_{ij}}$

Problem Formulation

- for machines of type j let
  - $M_j^{cur} \rightarrow$ current number
  - $M_j^{min} \rightarrow$ minimum desired number
  - $M_j^{max} \rightarrow$ maximum desired number
  - $M_j^{B} \rightarrow$ number to buy
  - $M_j^{S} \rightarrow$ number to sell
- total number of machines of type j is $M_j = M_j^{cur} + M_j^{B} - M_j^{S}$
- let M^{min} and M^{max} be the limits on the total number of machines allowed

Machine Pricing

- let β_j^B and β_j^S be the buying (purchase) and selling price of a machine of type j
- $\beta_j^B > \beta_j^S$
- can be easily adapted to cloud based computing models (i.e., renting resources)

Modeling

ETC

- let there be L abstract computational operation types
- let η_{il} be the number of operations of type l for tasks of type i
- let τ_{lj} be the seconds per operation of type l on a machine of type j
- then $\mathrm{ETC}_{\mathrm{ij}} = \sum_{\mathrm{l}} \eta_{\mathrm{il}} \tau_{\mathrm{lj}}$ and in matrix form $\mathbf{ETC} = \boldsymbol{\eta}\boldsymbol{\tau}$
- properties sufficient for heterogeneous computing systems
  - $\mathbf{ETC}$ can have nonzero task easiness homogeneity (TEH) and machine performance homogeneity (MPH)
  - for $\mathrm{L} > 1$ it can have nonzero TMA
- given an $\mathbf{ETC}$, compute $\boldsymbol{\eta}$ and $\boldsymbol{\tau}$ via rank L non-negative matrix factorization (NNMF)

Modeling

APC

- let $\mathrm{APC}_{\emptyset j}$ be the static power for a machine of type j
- let ψ_{lj} be the dynamic energy to execute an operation of type l on a machine of type j
- energy of type l operations is $\eta_{il}\psi_{lj}$
- total energy is $\sum_{l} \eta_{il}\psi_{lj}$
- total average dynamic power is $\mathrm{APC}_{ij} = \frac{\sum_{l} \eta_{il}\psi_{lj}}{\sum_{l} \eta_{il}\tau_{lj}}$
- find the model for $\mathbf{ETC}$, then use least squares to find ψ_{lj}

Modeling

Reliability

- control the system failure rate
- machine failures when not executing a task have no affect
- let ν_j be the failure rate of a machine of type j
- system failure rate (to be minimized) is $\sum_i \sum_j \nu_j \tilde{p_{ij}}$
- side note: cost of machine failures (repairs and replacements) can be incorporated into β_j^B

Linear Vector Optimization Problem

Objectives

$$
\underset{\mathbf{M}^B,\mathbf{M}^S,\tilde{\mathbf{p}}}{\text{minimize}}
\begin{pmatrix}
-\sum_i r_i \sum_j \frac{1}{ETC_{ij}} \tilde{p_{ij}} & \text{reward rate} \\
\sum_j M_j^B \beta_j^B - \sum_j M_j^S \beta_j^S & \text{upgrade cost} \\
\sum_i \sum_j \nu_j \tilde{p_{ij}} & \text{failure rate} \\
\sum_i \sum_j APC_{ij} \tilde{p_{ij}} + \sum_j APC_{\emptyset j} M_j & \text{power}
\end{pmatrix}
$$

- $\tilde{p_{ij}}$ is the number of machines running tasks of type i on machines of type j
- β_j^B and β_j^S are the buying and selling price

Linear Vector Optimization Problem

Constraints

$\forall j \quad M_j^{min} \le M_j \le M_j^{max}$	per type quantity limitations
$M^{min} \le \sum_j M_j \le M^{max}$	overall quantity limitation
$\forall j \quad M_j^B \ge 0 \wedge M_j^S \ge 0$	buy and sell non-negativity
$\forall i \quad \sum_j \frac{1}{ETC_{ij}} p_{\tilde{ij}} \le \lambda_i$	task arrival rate
$\forall j \quad \sum_i p_{\tilde{ij}} \le M_j$	machine utilization
$\forall i, j \quad 0 \le p_{\tilde{ij}}$	non-negative schedule

- $p_{\tilde{ij}}$ is the number of machines running tasks of type i on machines of type j
- $M_j = M_j^{cur} + M_j^B - M_j^S$ is the number of machines of type j

Linear Vector Optimization Problem

Extra Constraints

$\sum_j M_j^B \beta_j^B - \sum_j M_j^S \beta_j^S \leq \beta$	budget
$\sum_i \sum_j \nu_j \tilde{p_{ij}} \leq \nu_{max}$	failure rate
$\sum_i \sum_j APC_{ij} \tilde{p_{ij}} + \sum_j APC_{\emptyset j} M_j \leq P_{max}$	power

- β is the budget
- ν_{max} is the maximum system failure rate
- P_{max} is the maximum power consumption

Stochastic Model

- uncertain parameters: $\boldsymbol{\lambda}$, $\mathbf{ETC}$, and $\mathbf{APC}$
- three random matrices define $\mathbf{ETC}$ and $\mathbf{APC}$
  - $\boldsymbol{\eta} \rightarrow$ property of the tasks
  - $\boldsymbol{\tau}$ and $\boldsymbol{\psi} \rightarrow$ property of the machines
- η_{il}, τ_{lj}, and ψ_{lj} are modeled as independent uniform random variables
- optimization needs PDF of $\frac{1}{\mathrm{ETC}_{ij}} = \frac{1}{\sum_l \eta_{il}\tau_{lj}}$
  - nearly impossible to compute in closed form
  - need to sample the distributions

Stochastic Programming

- uncertainty affects the optimal solution
- want a solution that is robust against uncertainty in the parameters
- distributional assumption
- multi-stage stochastic program
    - first stage: "here-and-now" decision based on available data
    - second stage: some random variables are realized, a "recourse" decision is made
    - third stage: more random variable are realized, another "recourse" decision is made
    - and so on...
    - last stage: all random variables are realized, last "recourse" decision

Stochastic Programming

linear program	stochastic program with recourse
$\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{c}^T\mathbf{x}$ subject to: $\mathbf{A}\mathbf{x} = \mathbf{b}$ $\mathbf{x} \geq 0$	$\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{c}^T\mathbf{x} + E_\xi[Q(\mathbf{x},\xi)]$ subject to: $\mathbf{A}\mathbf{x} = \mathbf{b}$ $\mathbf{x} \geq \mathbf{0}$ where: $Q(\mathbf{x},\xi) = \min_{\mathbf{y}} \ \mathbf{q}(\xi)^T\mathbf{y}$ such that: $\mathbf{T}(\xi)\mathbf{x} + \mathbf{W}(\xi)\mathbf{y} = \mathbf{h}(\xi)$ $\mathbf{y} \geq \mathbf{0}$

- ξ is a random vector representing the uncertain parameters
- $Q(\mathbf{x},\xi)$ is called the value function
- expected value function is $E_\xi[Q(\mathbf{x},\xi)]$

Resource Provisioning via Stochastic Programming

	First Stage	Second Stage
Stochastic	$\boldsymbol{\lambda}$, $\mathbf{ETC}$, and $\mathbf{APC}$	nothing
Decision Variable	$\mathbf{x} = \begin{pmatrix} \mathbf{M}^{\mathrm{B}} \\ \mathbf{M}^{\mathrm{S}} \end{pmatrix}$	$\mathbf{y} = \text{flatten}(\tilde{\mathbf{p}})$
Objective	$\mathbf{c}^{\mathrm{T}}\mathbf{x}$	$\mathbf{q}^{\mathrm{T}}\mathbf{y}$
Objective Coefficients	$\mathbf{c}$ is a function of $\boldsymbol{\beta}^{\mathrm{B}}$, $\boldsymbol{\beta}^{\mathrm{S}}$, and $\mathbf{APC}_{\emptyset}$	$\mathbf{q}$ is a function of the reward $\mathbf{r}$ and $\mathbf{ETC}$
Constraints	$\mathbf{Ax} = \mathbf{b}$ $\mathbf{x} \geq \mathbf{0}$	$\mathbf{Tx} + \mathbf{Wy} = \mathbf{h}$ $\mathbf{y} \geq \mathbf{0}$
Constraint Coefficients	$\mathbf{A}$ is a function of $\boldsymbol{\beta}^{\mathrm{B}}$, $\boldsymbol{\beta}^{\mathrm{S}}$ $\mathbf{b}$ is a function of β and machine limits	$\mathbf{h}$ is a function of $\boldsymbol{\lambda}$ $\mathbf{T}$ and $\mathbf{W}$ are functions of $\mathbf{ETC}$ and $\mathbf{APC}$
Action	buy machines $\mathbf{M}^{\mathrm{B}}$ sell machines $\mathbf{M}^{\mathrm{S}}$	use schedule $\tilde{\mathbf{p}}$ for task assignment

Deterministic Equivalent Linear Program

$$
\begin{array}{lllllll}
\underset{\mathbf{x},\,\mathbf{y_k}}{\text{minimize}} & \mathbf{c}^T\mathbf{x} & +p_1\mathbf{q}_1^T\mathbf{y}_1 & +p_2\mathbf{q}_2^T\mathbf{y}_2 & \cdots & +p_K\mathbf{q}_K^T\mathbf{y}_K & \\
\text{subject to:} & \mathbf{A}\mathbf{x} & & & & & = \mathbf{b} \\
 & \mathbf{T}_1\mathbf{x} & +\mathbf{W}_1\mathbf{y}_1 & & & & = \mathbf{h}_1 \\
 & \mathbf{T}_2\mathbf{x} & & +\mathbf{W}_2\mathbf{y}_2 & & & = \mathbf{h}_2 \\
 & \vdots & & & & & \vdots \\
 & \mathbf{T}_K\mathbf{x} & & & & +\mathbf{W}_K\mathbf{y}_K & = \mathbf{h}_K \\
 & \mathbf{x}, & \mathbf{y}_1, & \mathbf{y}_2, & \cdots & \mathbf{y}_K & \geq \mathbf{0}
\end{array}
$$

Value of Information

- let $z(\mathbf{x}, \xi)$ be the optimal objective function value for a given $\mathbf{x}$ and a particular scenario ξ

$$z(\mathbf{x}, \xi) = \mathbf{c}^T \mathbf{x} + \min_{\mathbf{y}} \{ \mathbf{q}(\xi)^T \mathbf{y} \mid \mathbf{T}(\xi)\mathbf{x} + \mathbf{W}(\xi)\mathbf{y} = \mathbf{h}(\xi) \wedge \mathbf{y} \geq \mathbf{0} \}$$

- wait-and-see problem
  - wait until ξ is realized then find optimal solution
  - $\mathrm{WS} = \mathrm{E}_\xi [\min_{\mathbf{x}} z(\mathbf{x}, \xi)]$
  - requires perfect information → unachievable
- recourse problem (stochastic problem)
  - $\mathrm{RP} = \min_{\mathbf{x}} \mathrm{E}_\xi [z(\mathbf{x}, \xi)]$
  - achievable, optimal strategy
- expected value problem (mean value problem)
  - use means of all parameters
  - $\mathrm{EV} = \min_{\mathbf{x}} z\left(\mathbf{x}, \mathrm{E}_\xi[\xi]\right)$
  - achievable, sub-optimal
  - expected value of using the EV solution, $\mathbf{x}_{\mathrm{EV}}$ is $\mathrm{EEV} = \mathrm{E}_\xi [z(\mathbf{x}_{\mathrm{EV}}, \xi)]$
  - uses optimal second, third, etc, stage decisions

Value of Information

- let $z(\mathbf{x},\boldsymbol{\xi})$ be the optimal objective function value for a given $\mathbf{x}$ and a particular scenario $\boldsymbol{\xi}$

$$z(\mathbf{x},\boldsymbol{\xi}) = \mathbf{c}^T\mathbf{x} + \min_{\mathbf{y}}\{\mathbf{q}(\boldsymbol{\xi})^T\mathbf{y} \mid \mathbf{T}(\boldsymbol{\xi})\mathbf{x} + \mathbf{W}(\boldsymbol{\xi})\mathbf{y} = \mathbf{h}(\boldsymbol{\xi}) \wedge \mathbf{y} \geq \mathbf{0}\}$$

- wait-and-see problem
  - wait until $\boldsymbol{\xi}$ is realized then find optimal solution
  - $\mathrm{WS} = \mathrm{E}_{\boldsymbol{\xi}}\left[\min_{\mathbf{x}} z(\mathbf{x},\boldsymbol{\xi})\right]$
  - requires perfect information → unachievable
- recourse problem (stochastic problem)
  - $\mathrm{RP} = \min_{\mathbf{x}} \mathrm{E}_{\boldsymbol{\xi}}\left[z(\mathbf{x},\boldsymbol{\xi})\right]$
  - achievable, optimal strategy
- expected value problem (mean value problem)
  - use means of all parameters
  - $\mathrm{EV} = \min_{\mathbf{x}} z\left(\mathbf{x}, \mathrm{E}_{\boldsymbol{\xi}}[\boldsymbol{\xi}]\right)$
  - achievable, sub-optimal
  - expected value of using the EV solution, $\mathbf{x}_{\mathrm{EV}}$ is $\mathrm{EEV} = \mathrm{E}_{\boldsymbol{\xi}}\left[z(\mathbf{x}_{\mathrm{EV}},\boldsymbol{\xi})\right]$
  - uses optimal second, third, etc, stage decisions

- expected value of perfect information is $\mathrm{EVPI} = \mathrm{RP} - \mathrm{WS} \geq 0$
- value of the stochastic solution is $\mathrm{VSS} = \mathrm{EEV} - \mathrm{RP} \geq 0$

Comparison Purchasing Strategies

- strategies... buy machines of type

H1:	$j^* = \arg\max_j \sum_i \frac{1}{ETC_{ij}}$	highest performing machine
H2:	$j^* = \arg\max_j \frac{1}{\beta_j^B} \sum_i \frac{1}{ETC_{ij}}$	highest performance/price machine
H3:	$j^* = \arg\max_j \frac{1}{\beta_j^B} \sum_i \lambda_i r_i \frac{1}{ETC_{ij}}$	highest relevant performance/price machine

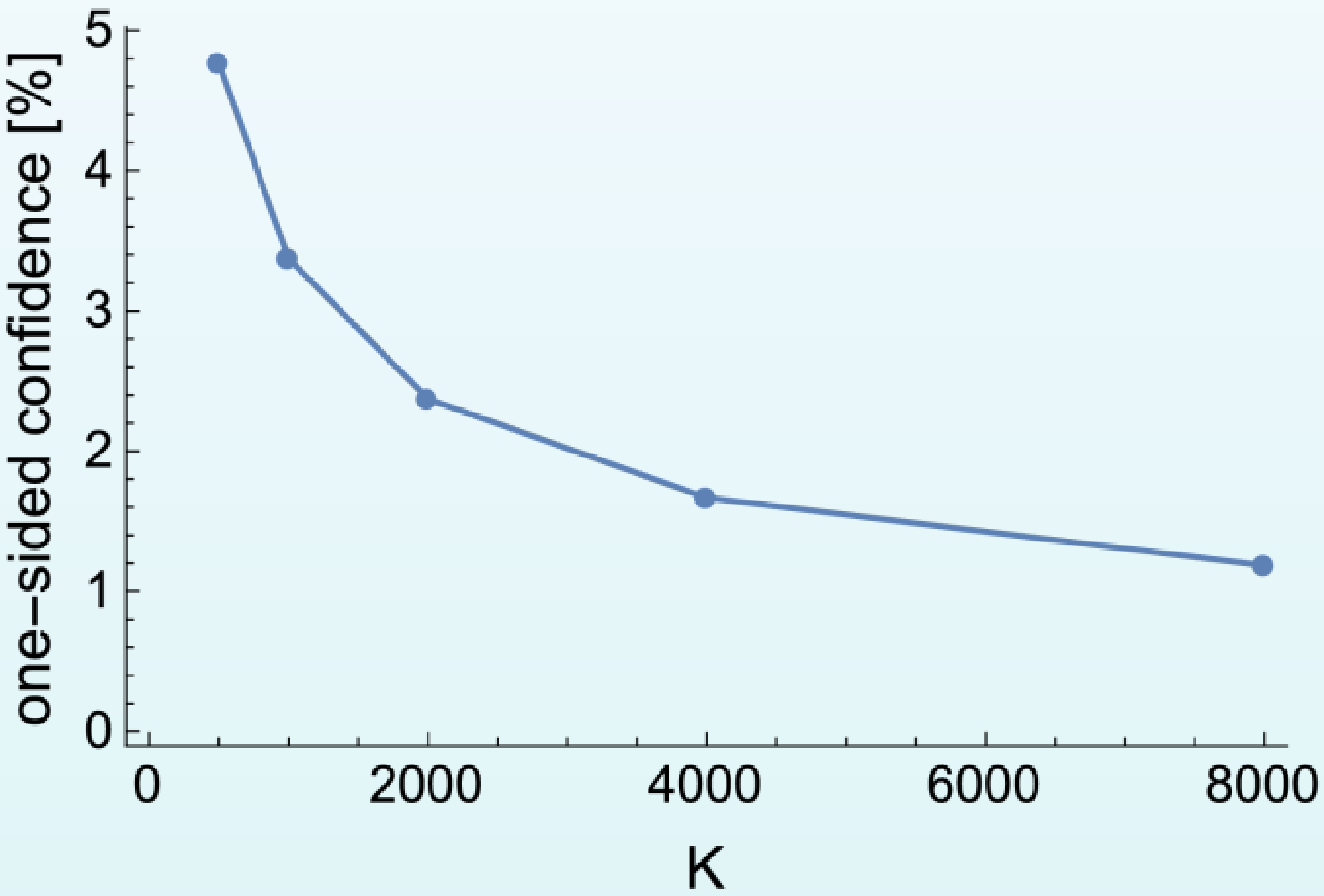
- buy maximum (fractional) number of machines of type j^*
  - satisfy all constraints (such as budget, power, etc.)
  - uses the optimal second stage decision
- reward rate: solve LP using the mean of the parameters (similar to EV)
- expected reward rate: expectation over all scenarios (similar to EEV)

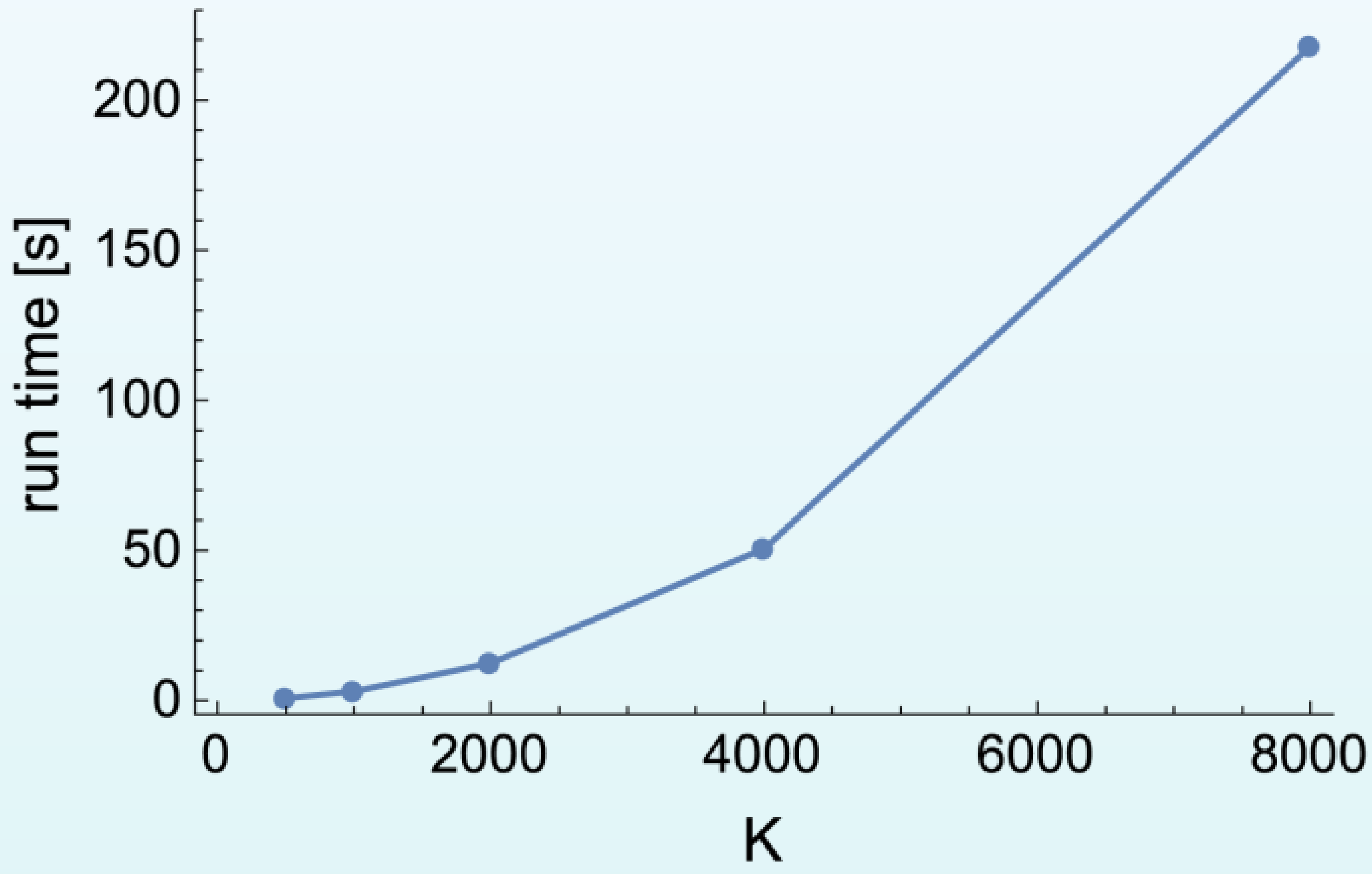
Medium Sized Problem

- T=10, M=5, L=2
- SAA with K=20000
  - 1M variables
  - 340K constraints
  - solved in 5 minutes using one core
  - reasonable run time for an offline algorithm
- constraint matrix is very sparse
  - dense matrix would consume 2.7TB of RAM
  - solved with only 400MB of RAM
  - sparse linear algebra libraries are awesome!
- maximize reward rate

Medium Sized Problem

Solution Quality and Run Time





- average of 10 runs

Medium Sized Problem

- initial system has 10 machines of type 5

ETC

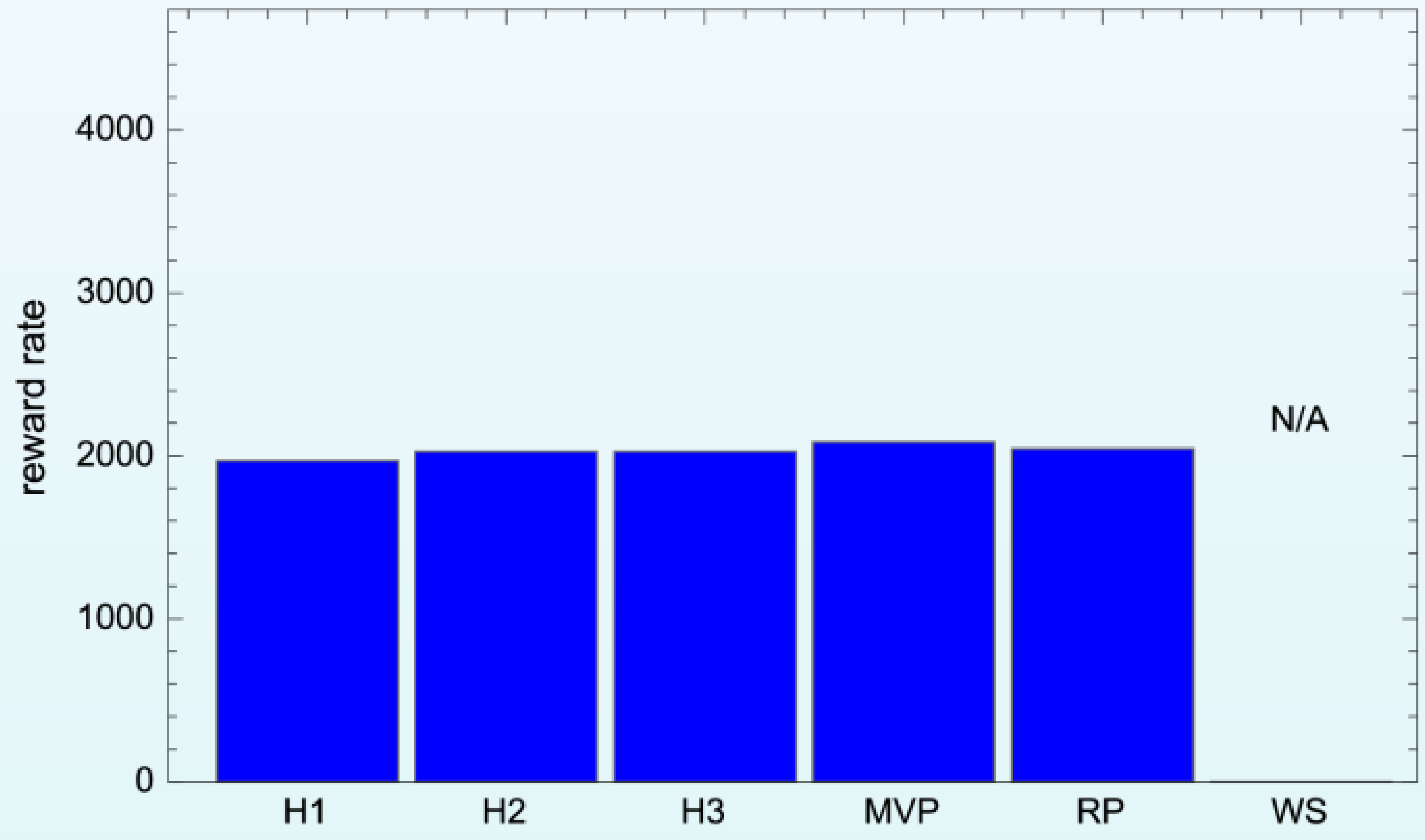
	M1	M2	M3	M4	M5
T1	5	10	10	101	30
T2	15	30	30	300	90
T3	50	101	11	1010	303
T4	50	101	100	1010	303
T5	505	1010	1001	10100	3030
T6	15	30	12	300	90
T7	55	110	110	1100	330
T8	17	34	16	340	102
T9	6	11	10	110	33
T10	5	10	10	100	30

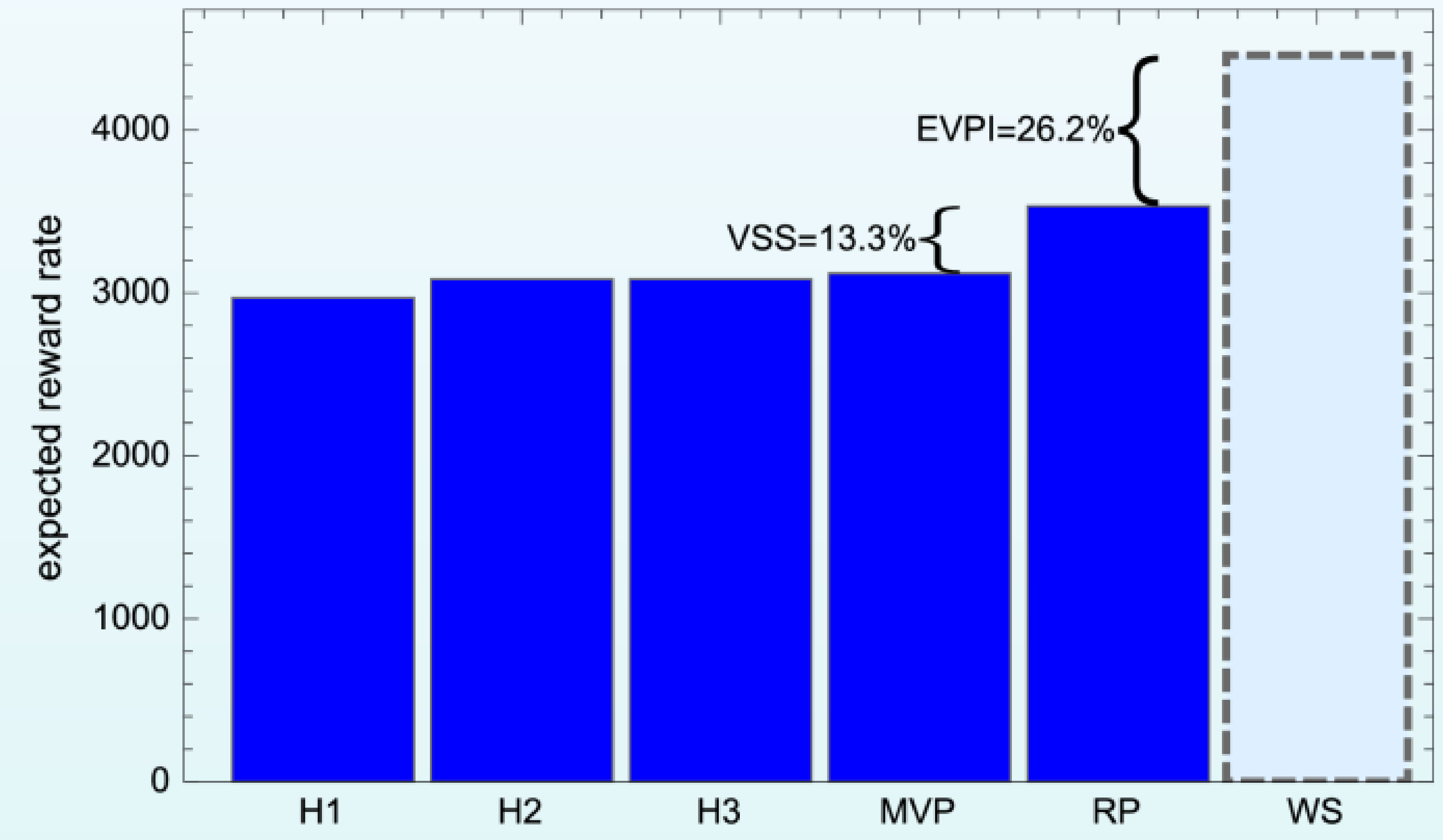
Solution

	H1	H2	H3	MVP	RP
M1	31.8	0	0	0	10.
M2	0	67.3	67.3	8.5	26.9
M3	0	0	0	32.	11.5
M4	0	0	0	0	0
M5	0	0	0	-10.	-6.7

Medium Sized Problem

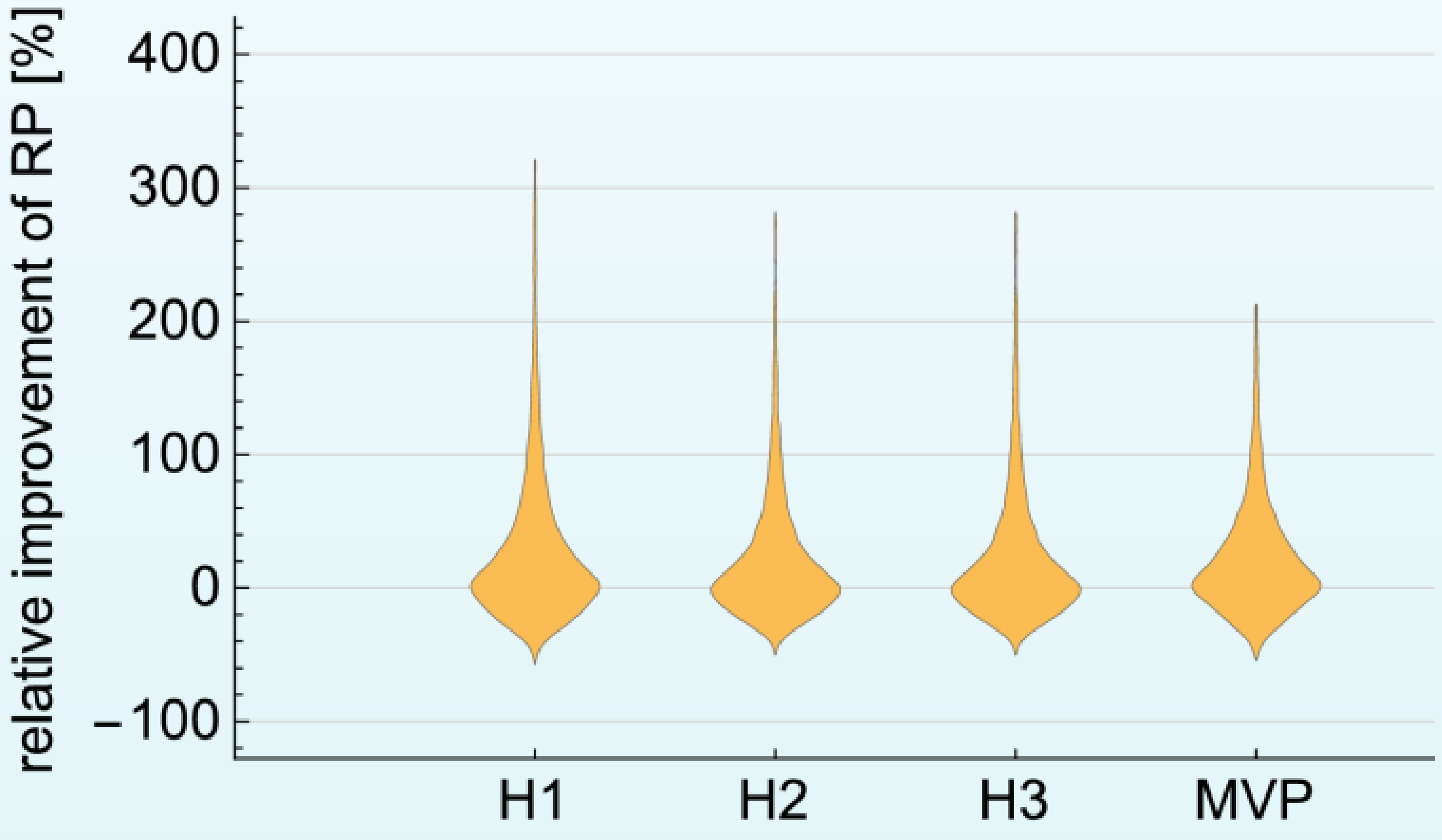
Comparison



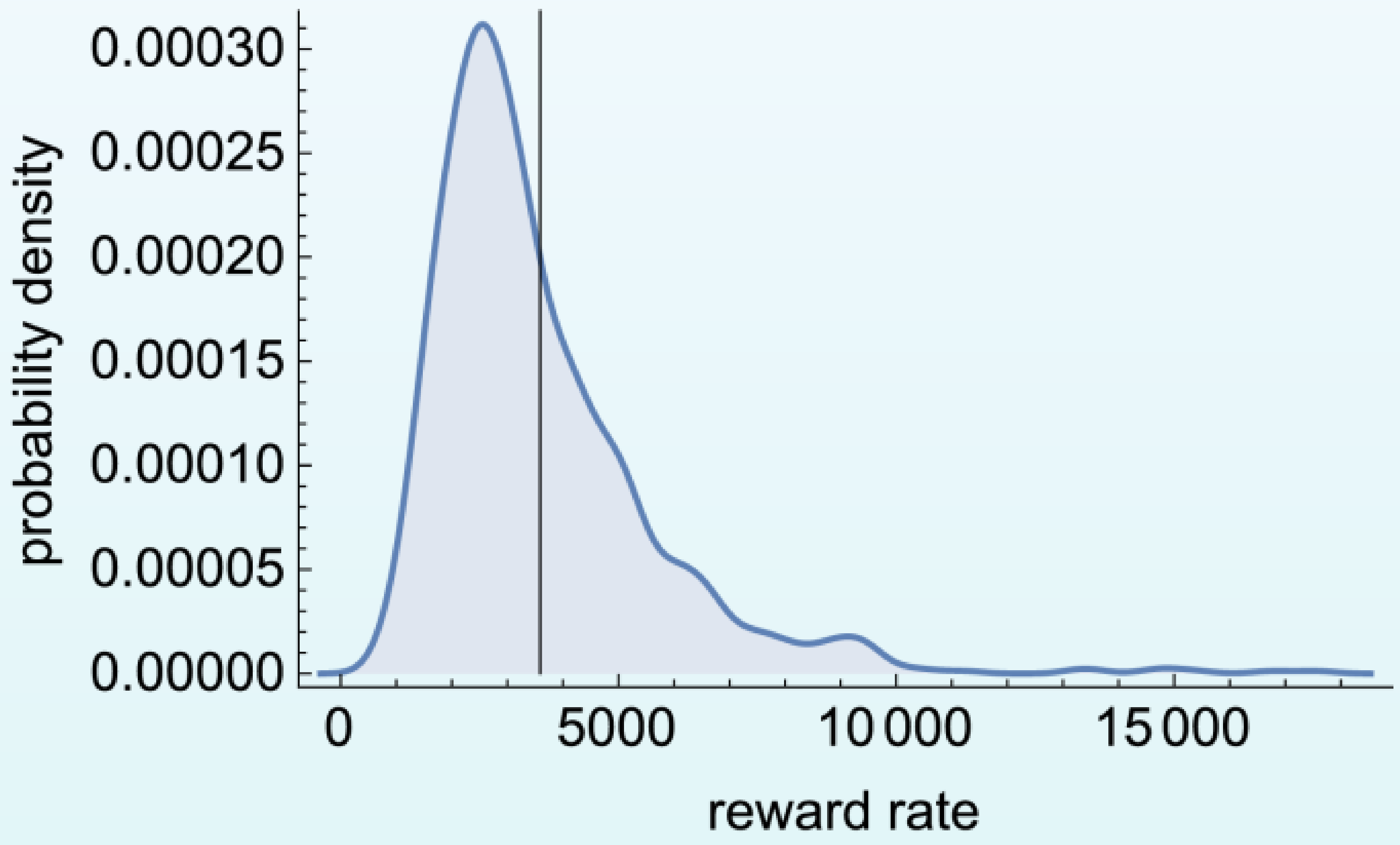


Medium Sized Problem

Relative Performance per Scenario



RP Objective PDF



Nine Machine Type Environment

- based on benchmarks
- T=10, M=9, L=3
    - coefficient of variance (CoV) of 25% used to compute the variances
    - uniform distribution for arrival rates and τ (seconds per operation)
    - given ETC and APC, computed η, τ, and ψ with NNMF and least squares
    - given CoV, computed the variance of τ via least squares
- using uniform distributions for τ and reducing variance (as necessary) to keep it non-negative
- budget is $400K
- primary objective: maximize reward rate
- secondary objective: minimize cost (not at the expense of reward rate)

Nine Machine Type Environment

ETC

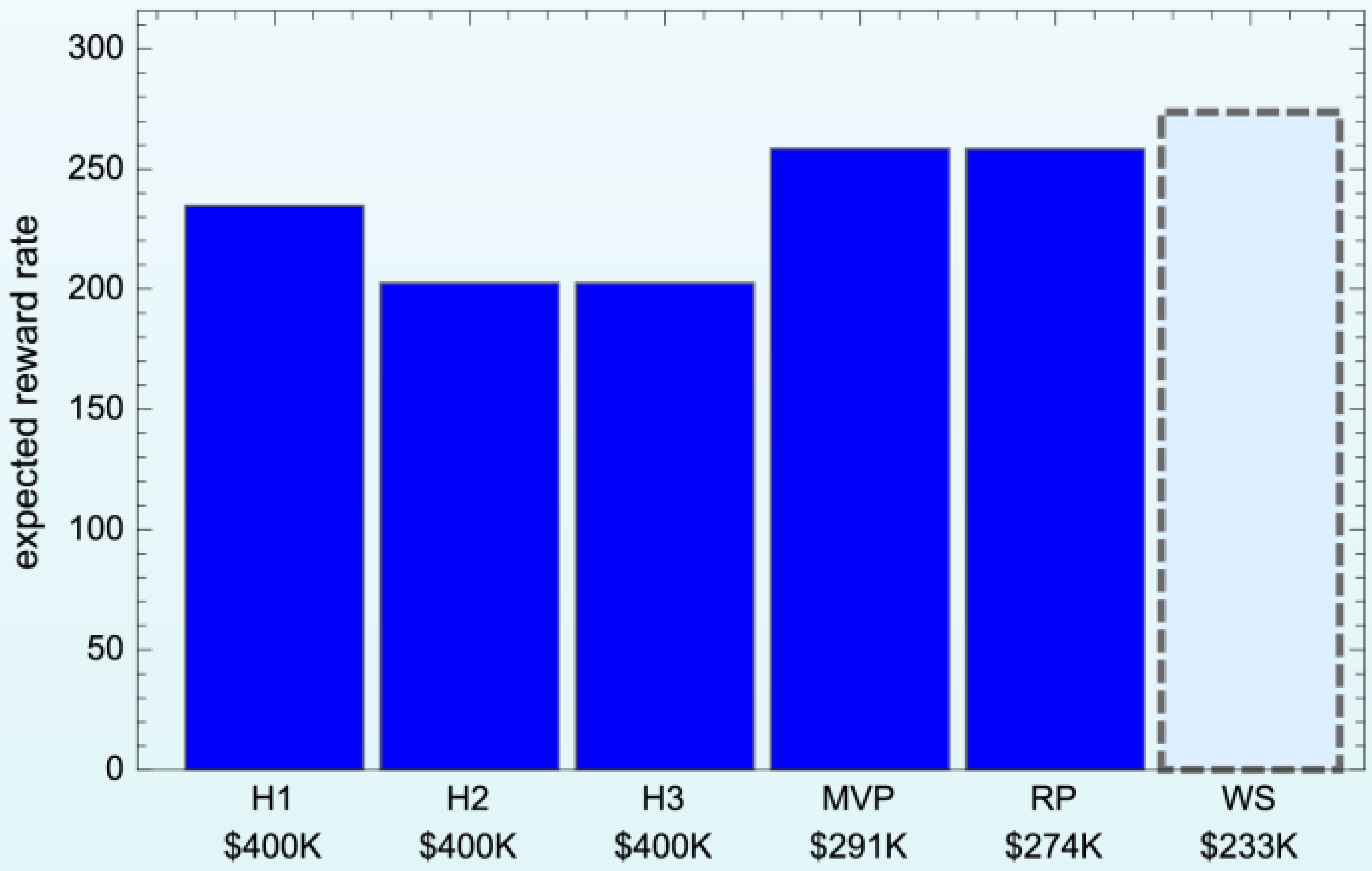
	M1	M2	M3	M4	M5	M6	M7	M8	M9
T1	57	28	72	45	41	19	27	28	26
T2	98	50	120	77	70	37	49	49	45
T3	463	303	362	342	314	311	303	290	264
T4	165	113	113	120	111	122	114	108	98
T5	167	91	185	129	118	74	90	88	81
T6	162	87	185	125	114	68	85	84	77
T7	45	22	57	36	33	15	22	22	20
T8	57	28	74	45	41	18	27	27	25
T9	59	36	54	44	41	34	36	35	32
T10	39	22	41	30	27	19	21	21	19

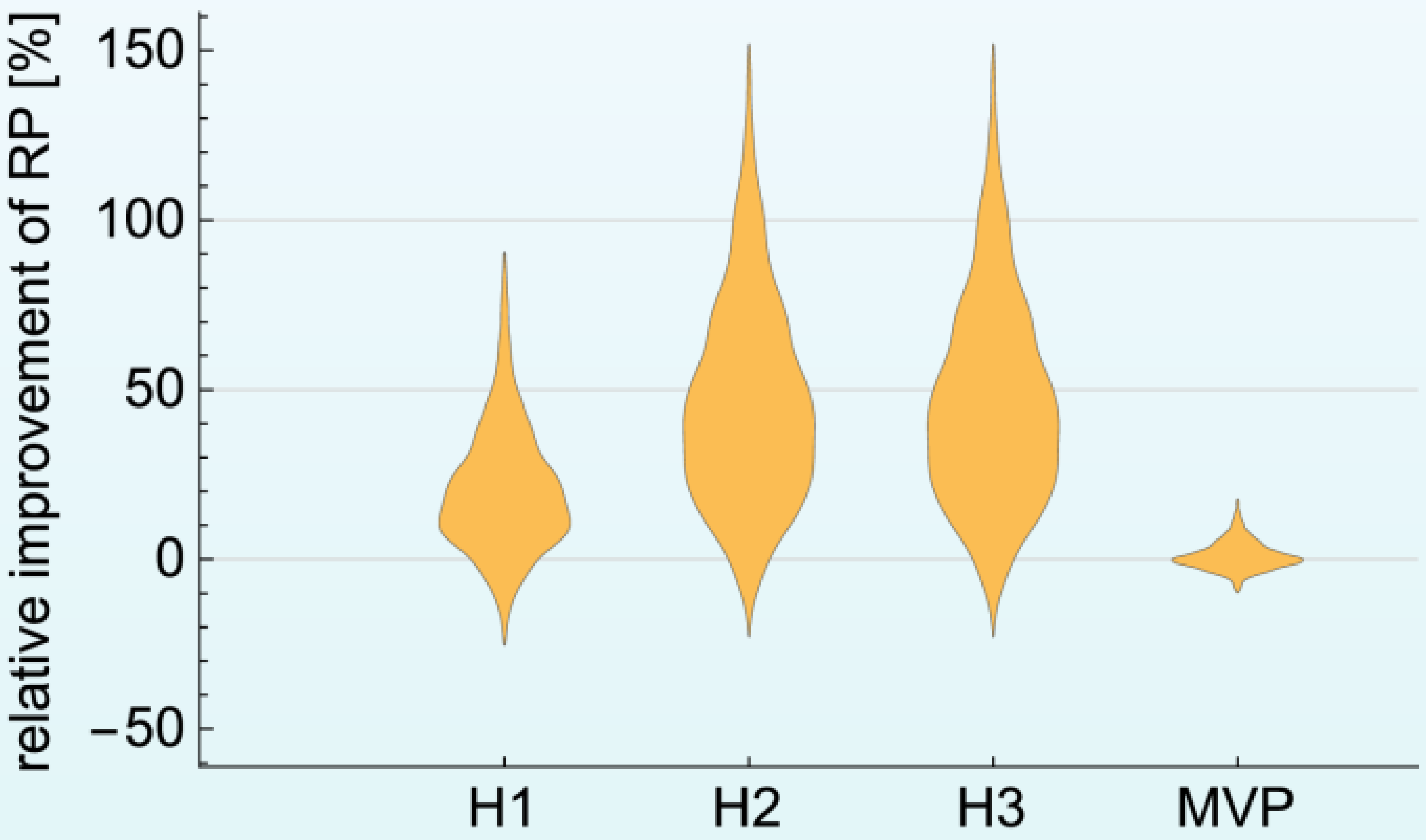
Solution

	H1	H2	H3	MVP	RP
M1	0	0	0	0	0
M2	0	168.8	168.8	0	0
M3	0	0	0	0	4.3
M4	0	0	0	0	0
M5	0	0	0	0	0
M6	121.2	0	0	0	12.7
M7	0	0	0	0	4.5
M8	0	0	0	100.	73.8
M9	0	0	0	0	1.1

Nine Machine Type Environment

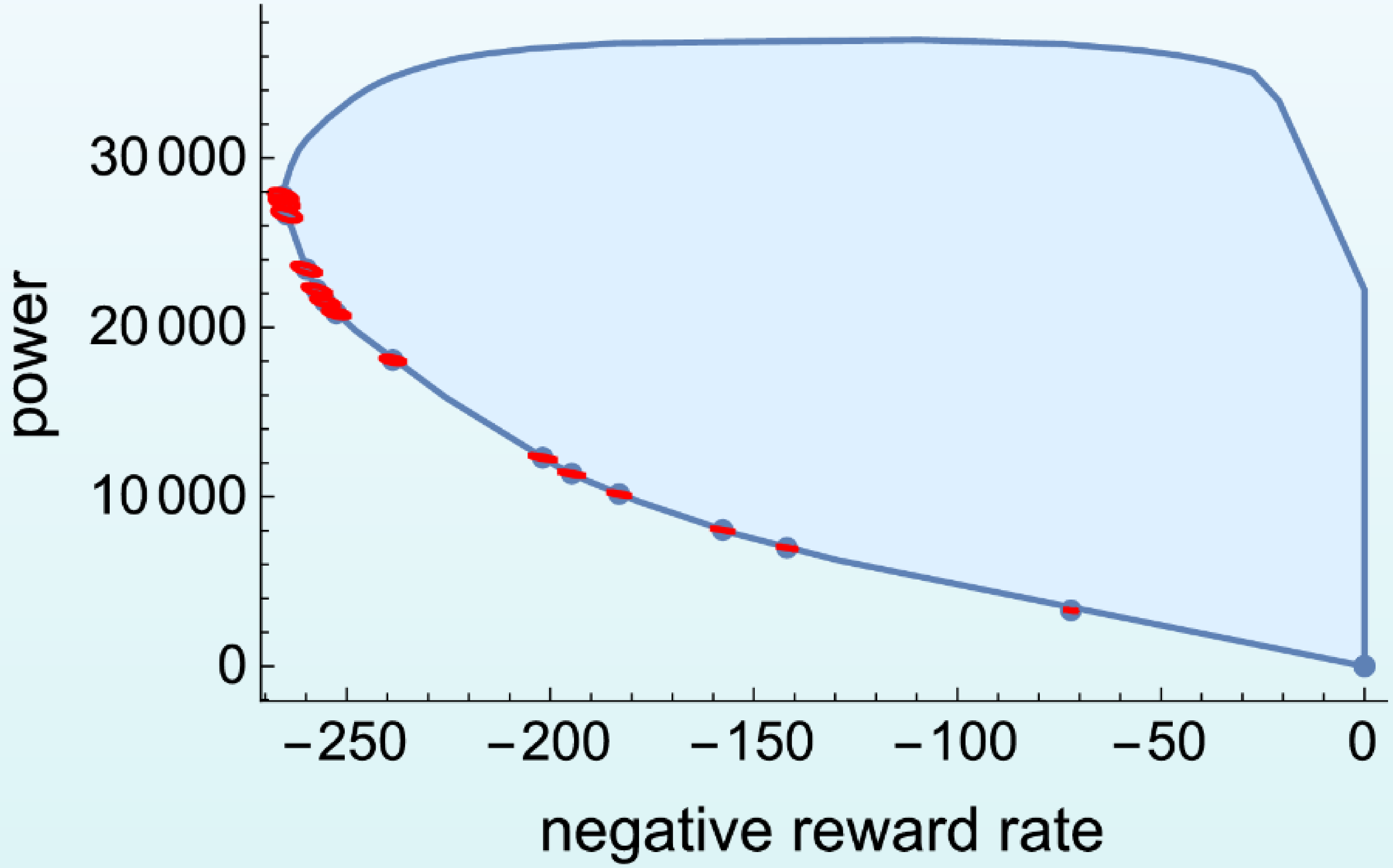
Comparison





Nine Machine Type Environment

Pareto Front and Feasible Region



Future Directions

- stochastic programming
    - risk-averse formulations
    - improve/develop more modeling tools
    - apply to LP in energy and makespan scheduling
    - use AWS EC2 instance types and map their properties to abstract workloads
    - use Ryan's data to evaluate accuracy of the $\mathbf{ETC}$ and $\mathbf{APC}$ models for small L
- design improved TMA measure then publish improved heterogeneity measures and TMA
- batch mode scheduling
    - adapt algorithms
    - evaluate performance with discrete event simulations

Questions