Presenter: Kyle M. Tarplee

Colorado State University

Preliminary Examination

Status

- Fall 2009: started (one class per semester)
- Spring 2012: finished last class
- Summer 2012: started research
- Fall 2012: qualifier
- Spring 2014: prelim
- still working for Numerica

Outline

- published research
- ongoing research
- plans for future work

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), cosponsors: Polskie Towaszystwo Informatycznq (PTI), iBS PAN, AGH University of Science and Technology, and Wroclaw University of Economics (UE), 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, 2014 to appear
- journal article submitted soon

Outline

- minimum makespan scheduling
- makespan scheduling simulation results
- comparison to other scheduling algorithms
- scalability analysis and results
- minimum energy/makespan scheduling
- Pareto front generation

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
- heterogeneous tasks and machines
- desire to minimize makespan
- later minimize energy consumption and makespan or maximize profit
- **goal:** efficiently compute high quality schedules for extremely large scale problems

General Approach

- group similar tasks (task types)
- group similar machines (machine types)
- large problems often have many tasks assigned to individual machines
    - small fraction of tasks divided among machines has little impact on makespan

Applicability

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

Makespan Scheduling

Approach

- efficiently compute solutions which bound the makespan
- lower bound uses linear programming (LP) and assumes tasks are **divisible**
  - our approach: determines the **number** of tasks of each type to assign to groups of machines of each type
  - traditional approach: assign **individual** tasks to **individual** machines
  - LP relaxation for traditional approach is intractable
- upper bound is found by recovering a feasible allocation from the LP

Linear Programming Lower Bound

Preliminaries

- simplifying approximation: **tasks are divisible among machines**
- T_i — number of tasks of type i
- M_j — number of machines of type j
- μ_{ij} — number of tasks of type i assigned to a machine of type j
- ETC_{ij} — estimated time to compute for a task of type i running on a machine of type j
- finishing time of machine type j is (lower bound)

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

- makespan (lower bound):

$$MS_{LB} = \max_j F_j$$

Linear Programming Lower Bound

Optimization Problem

$$\underset{\mu_{ij},\, MS_{LB}}{\text{minimize}} \quad MS_{LB}$$

subject to:

$$\forall i \quad \sum_{j} \mu_{ij} = T_i \qquad \text{task constraint}$$

$$\forall j \quad F_j \le MS_{LB} \qquad \text{makespan constraint}$$

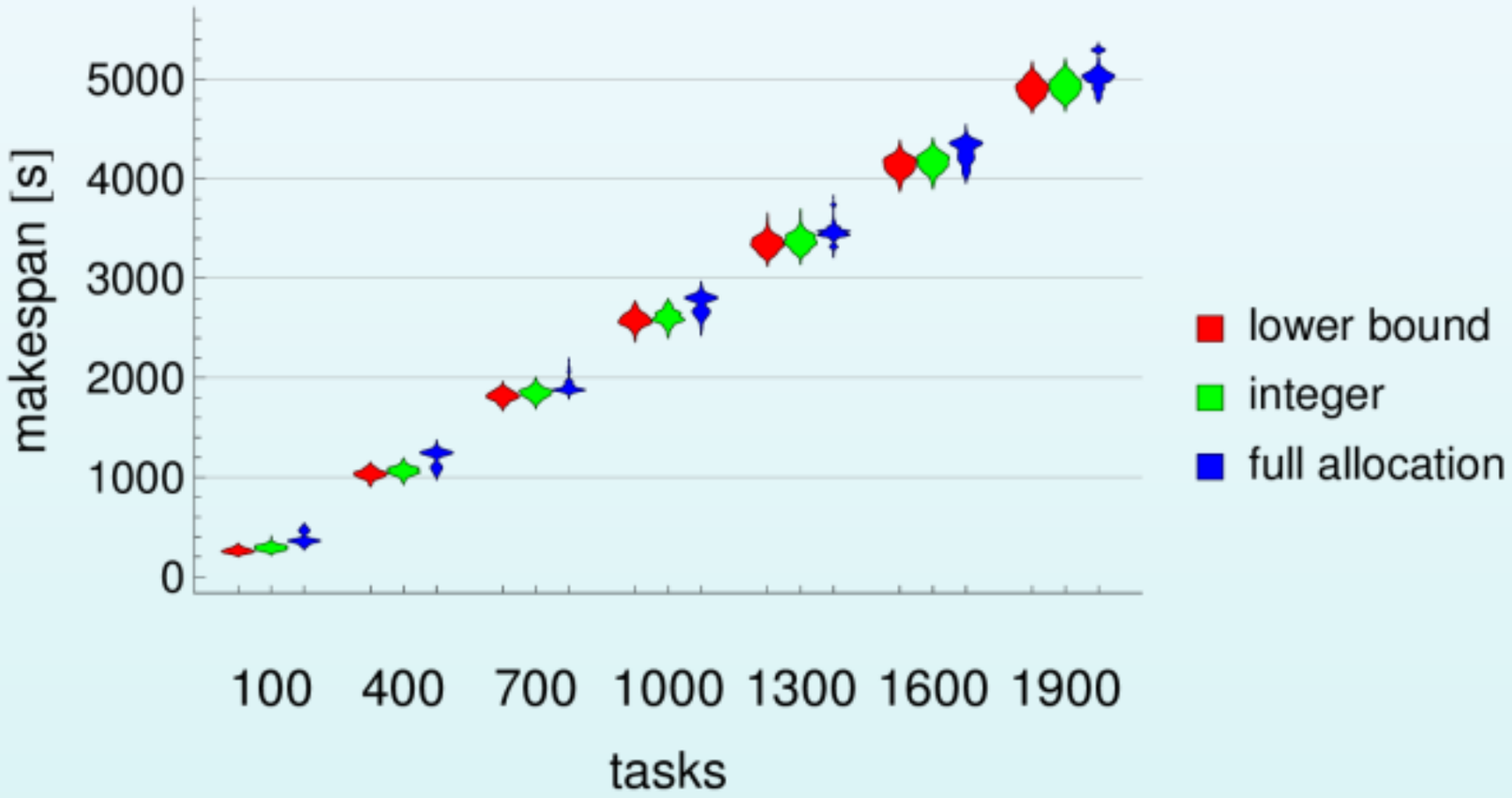
$$\forall i, j \quad \mu_{ij} \ge 0 \qquad \text{assignments must be non-negative}$$

Simulation Setup

- HPC system parameters
  - ETC matrix derived from actual systems benchmarks
  - 9 machine types, 36 machines, 4 machines per type
  - 30 task types, variable number of tasks (task type distribution held constant)
  - 200 Monte Carlo trials
- experiments used single core of a 2009 MacBook Pro laptop
  - using the COIN-OR linear programming solver (third party library in C++)
  - lower bound, rounding, and local assignment phases all implemented in C++
  - min-min and max-min implemented in C++

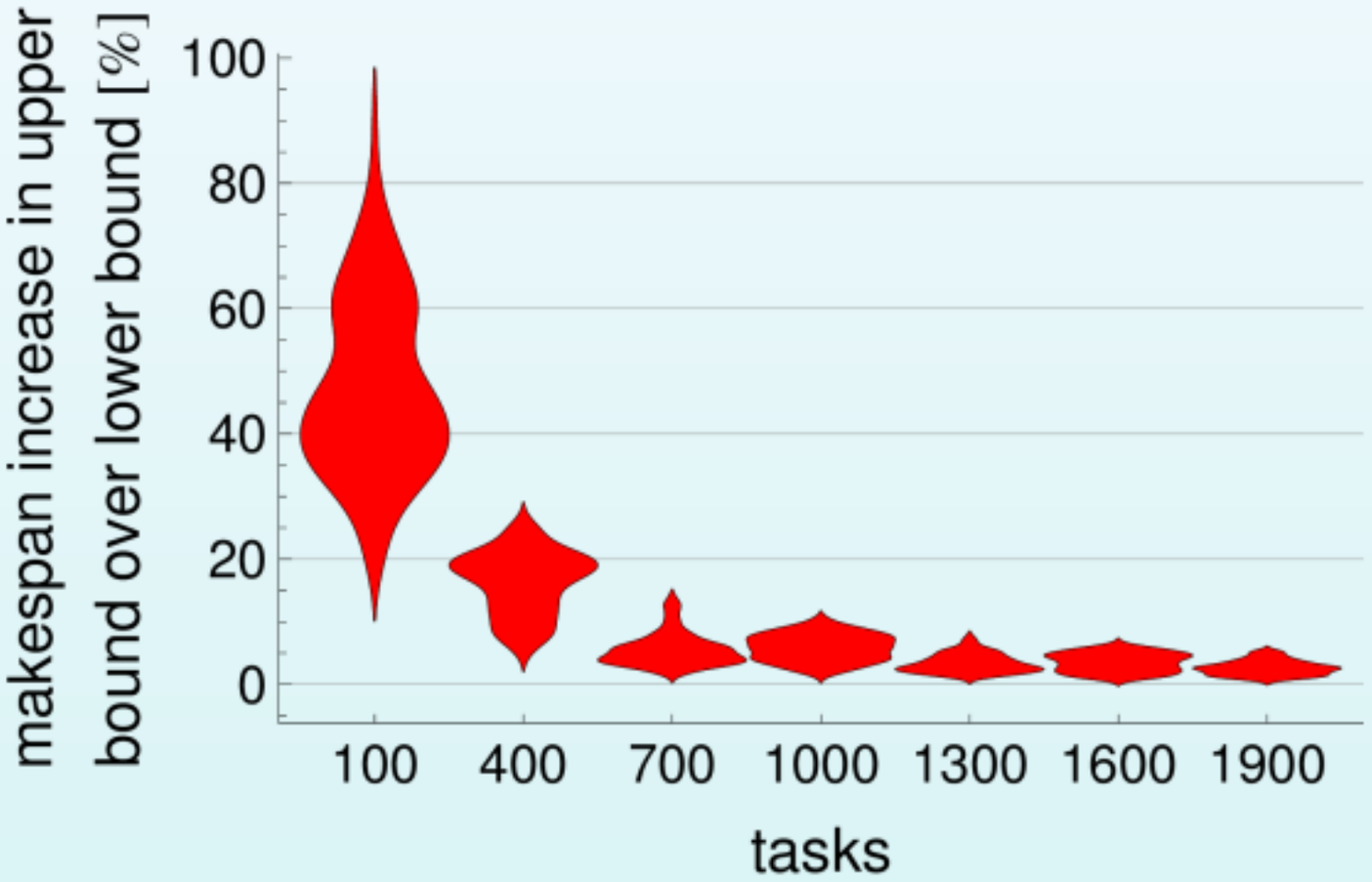
LP-Based Scheduler

Makespan vs Number of Tasks



LP-Based Possible Improvement

Relative Makespan vs Number of Tasks



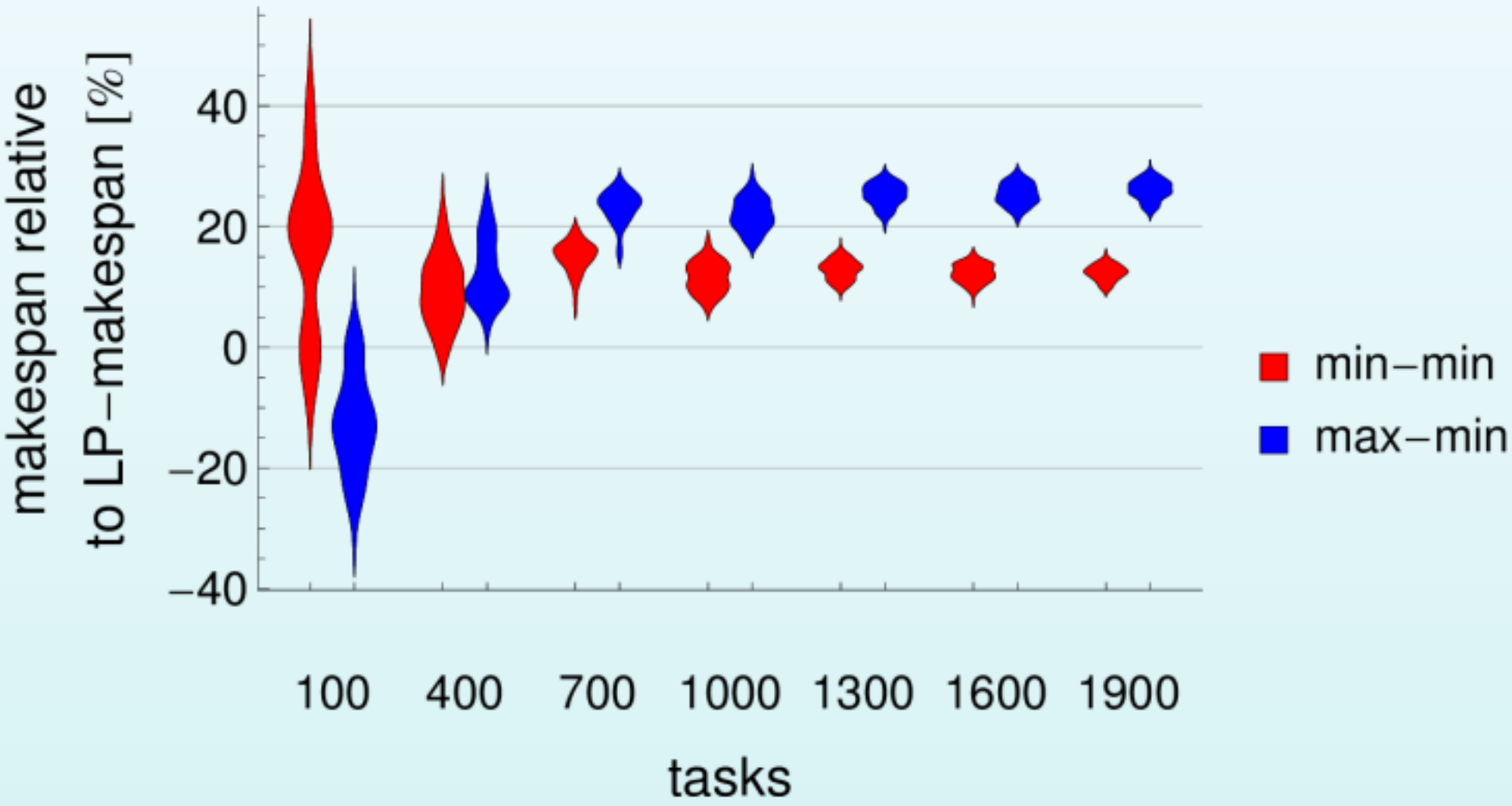
$$\frac{MS_{UB} - MS_{LB}}{MS_{LB}} 100$$

Heuristic Algorithm Comparison

- min-min and max-min are minimum completion time (MCT) based algorithms
- run time optimizations were applied to the vanilla algorithms
  - outer min/max is computed on-the-fly instead of as a post process (second step)
  - takes advantage of task and machine types where possible
  - store best machine assignment for each task type, update only those that are assigned (dirty) last iteration in each pass
  - fixed sized ragged arrays of task counts instead of variable length arrays of task indices

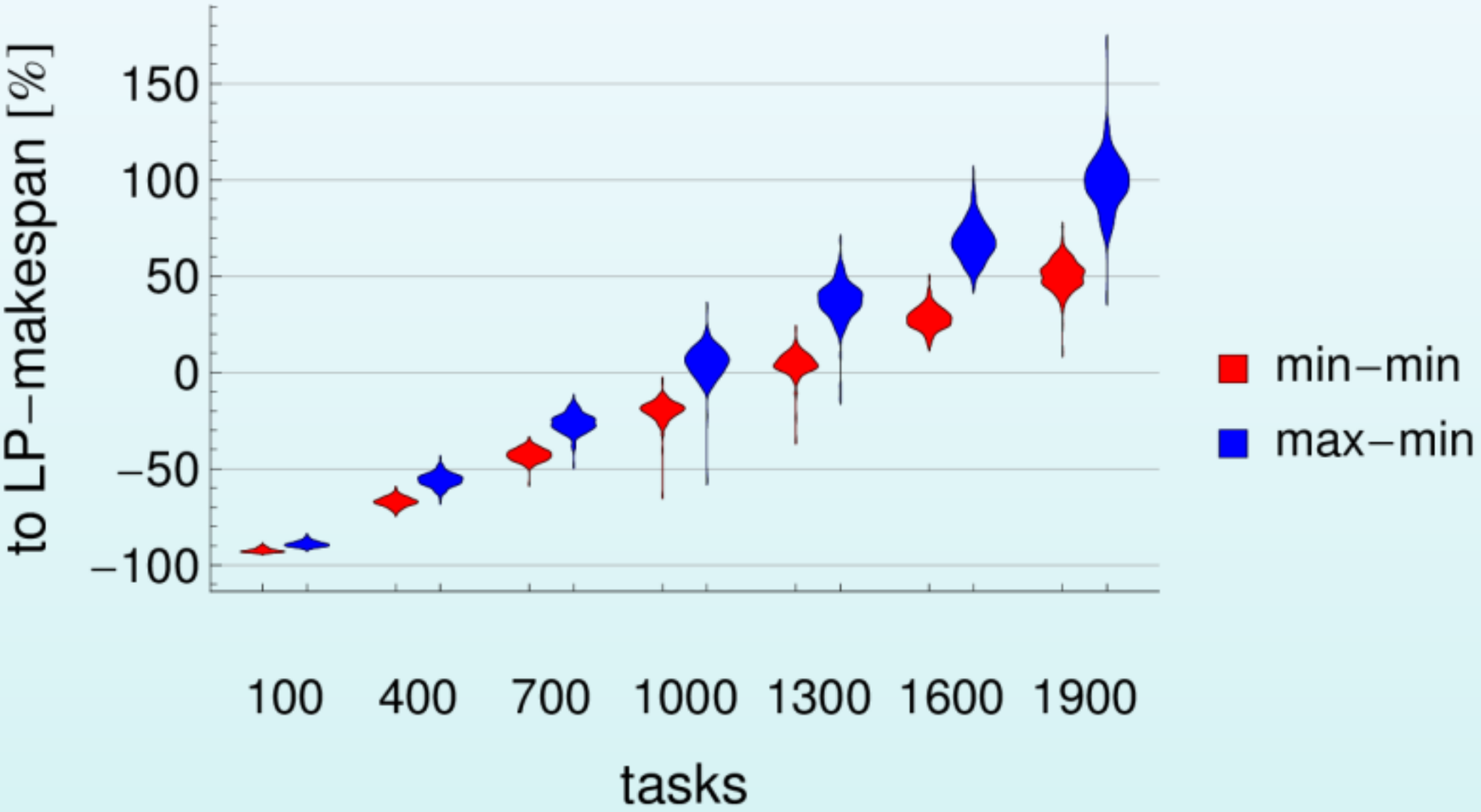
Heuristic Algorithm Comparison

Relative Makespan vs Number of Tasks



Heuristic Algorithm Comparison

Relative Run Time vs Number of Tasks



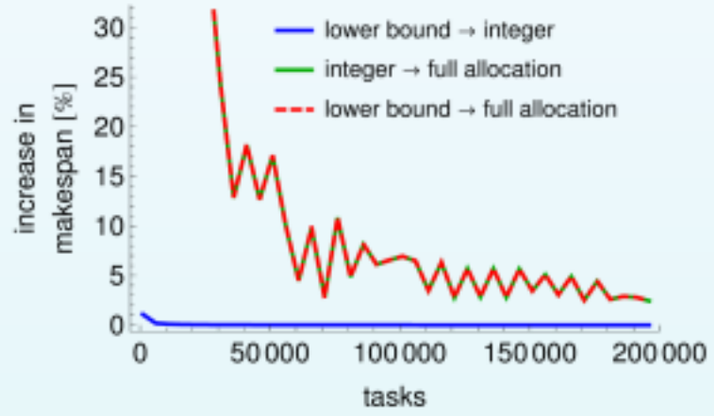
Algorithmic Complexity

- let the ETC matrix be $T \times M$
- linear programming lower bound
  - $T + M$ constraints
  - $TM + 1$ variables
  - average complexity of simplex algorithm is then $(T + M)^2(TM + 1)$
  - **independent** of the number tasks and machines
- rounding step: $T(M \log M)$
  - **independent** of the number tasks and machines
- local assignment step:
  - number of tasks for machine type j is $n_j = \sum_i x_{ij}$
  - worst cast complexity is $M \max_j \left(T \log T + n_j \log M_j\right)$
- complexity of all steps is dominated by either
  - linear programming solver
  - local assignment

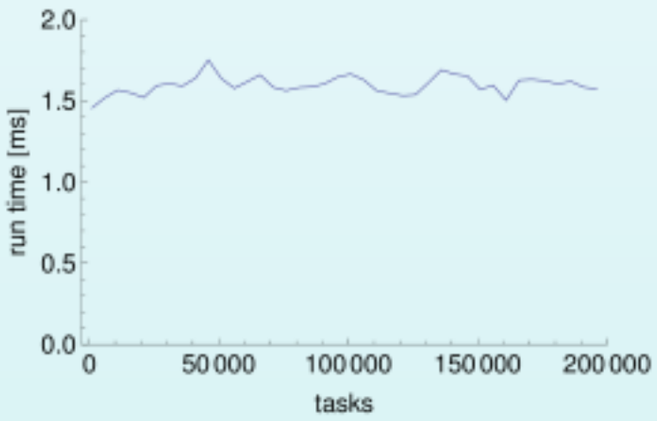
Scalability Results

- nominal system
  - scaled up version of the previous system
  - 3,600 machines composed of 9 machine types
  - 110,000 tasks from 30 task types
- averages of 50 trials are shown
- experiments sweep number of
  - *tasks
  - task types
  - *machines
  - machine types
- 10 million tasks: 1 second

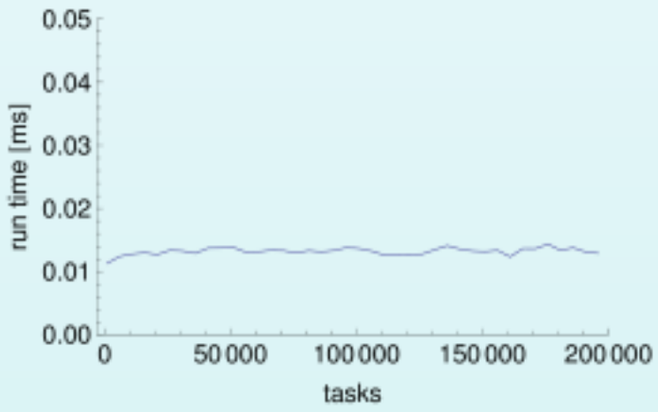
Impact of the Number of Tasks



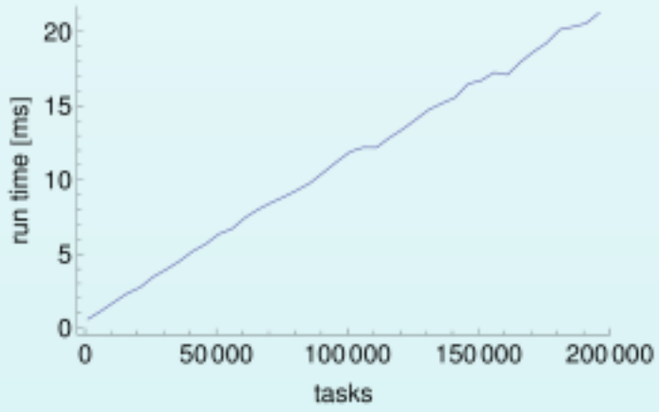
LP



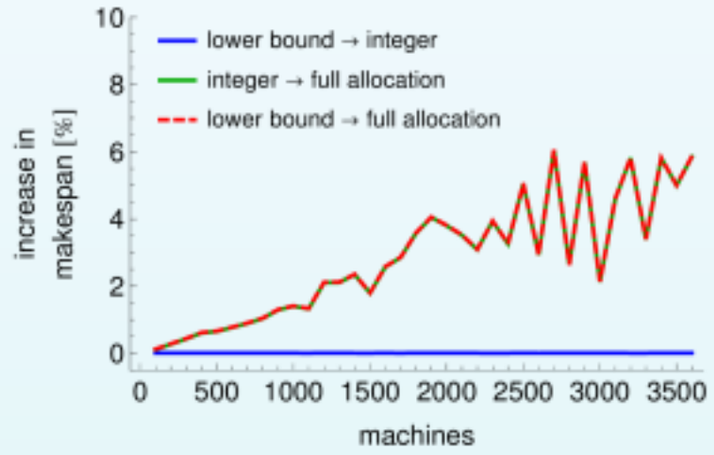
round near



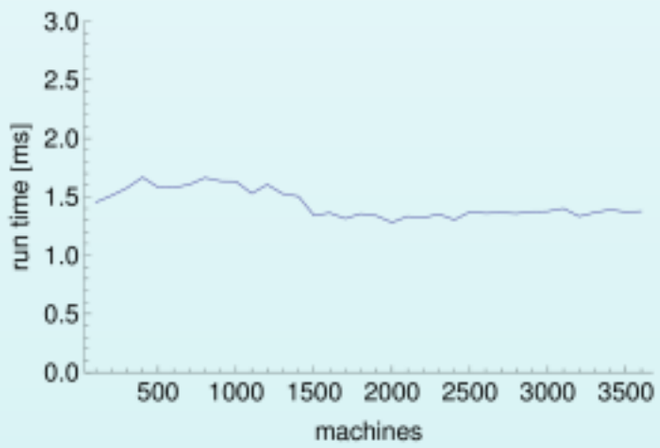
local assignment



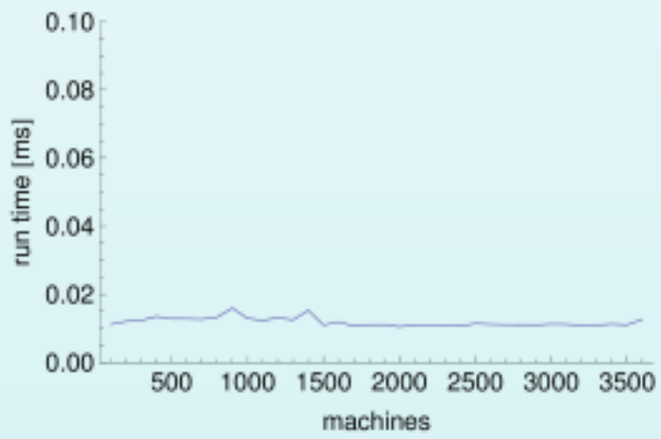
Impact of the Number of Machines



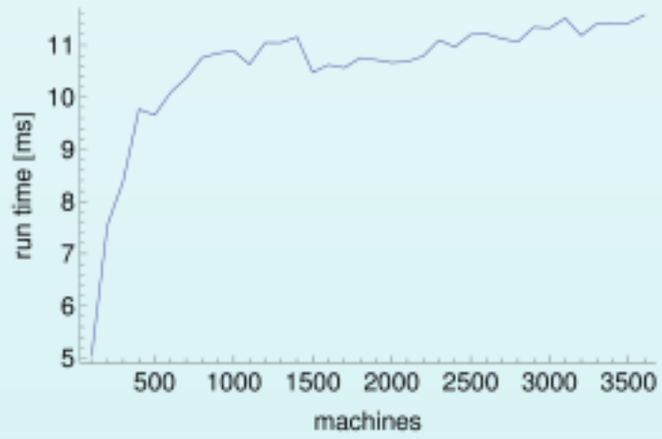
LP



round near



local assignment



Energy-Aware Scheduling

Preliminaries

- simplifying approximation: **tasks are divisible among machines**
- APC_{ij} — average power consumption for a task of type i running on a machine of type j
- $APC_{øj}$ — idle power consumption for a machine of type j
- energy consumed by the bag-of-tasks (lower bound):

$$\begin{aligned} E_{LB} &= \text{execution energy} + \text{idle energy} \\ &= \sum_i \sum_j \mu_{ij} APC_{ij} ETC_{ij} + \sum_j M_j APC_{øj} (MS_{LB} - F_j) \\ &= \sum_i \sum_j \mu_{ij} ETC_{ij} \left(APC_{ij} - APC_{øj}\right) + \sum_j M_j APC_{øj} MS_{LB} \end{aligned}$$

- note that energy is a function of makespan when we have non-zero idle power consumption

Bi-Objective Lower Bound

Vector Optimization

$$\underset{\mu_{ij},\, MS_{LB}}{\text{minimize}} \quad \begin{pmatrix} E_{LB} \\ MS_{LB} \end{pmatrix}$$

subject to:

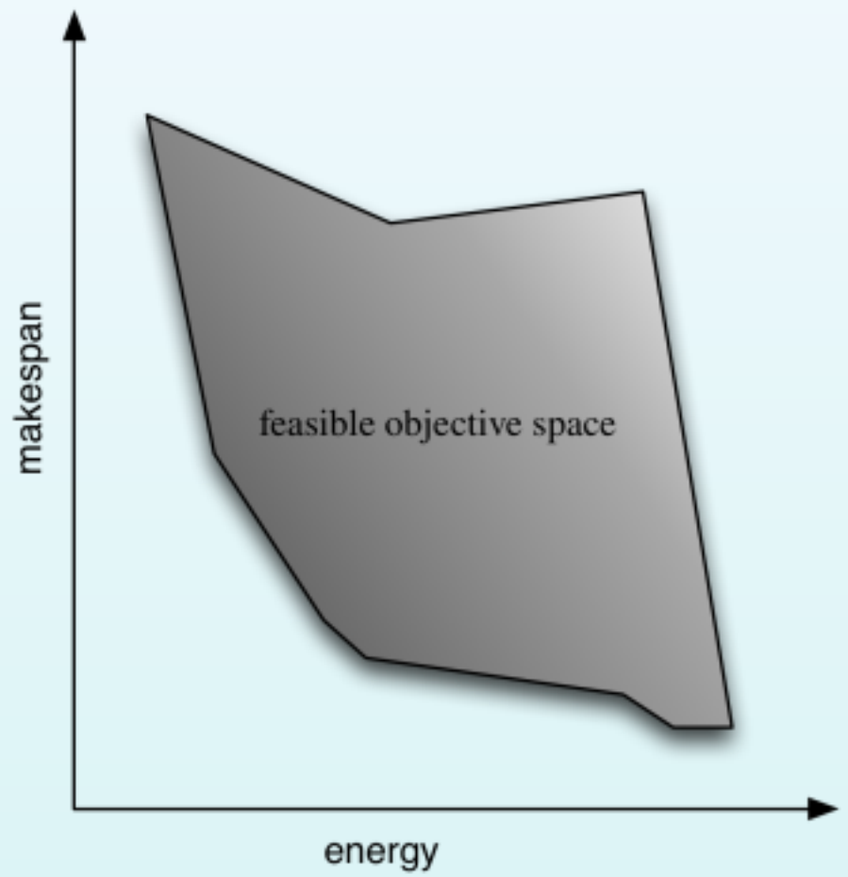
$$\forall i \quad \sum_{j} \mu_{ij} = T_i \qquad \text{task constraint}$$

$$\forall j \quad F_j \le MS_{LB} \qquad \text{makespan constraint}$$

$$\forall i, j \quad \mu_{ij} \ge 0 \qquad \text{assignments must be non-negative}$$

Nomenclature

- objective space: Pareto efficient points, Pareto outcomes, Pareto front, Pareto surface
- solution space: efficient points, efficient solutions



Weighted Sum Scalarization Algorithm

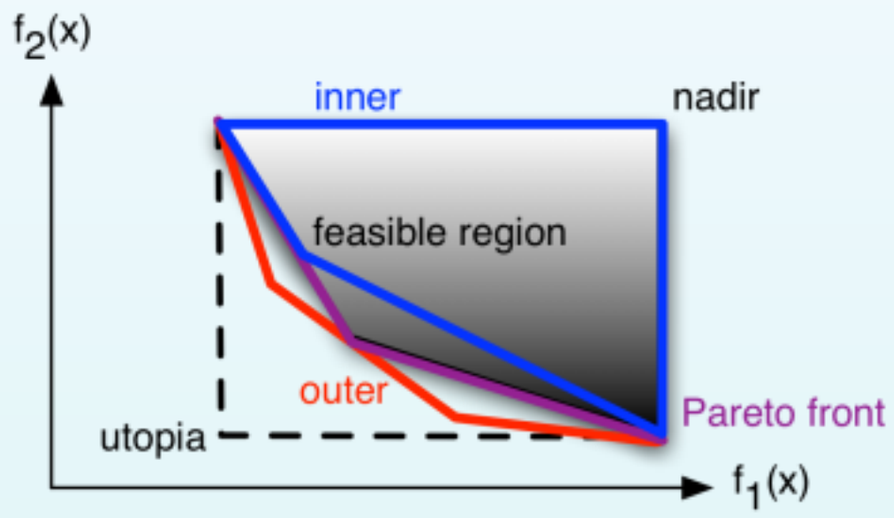
- **step 1** find the utopia (ideal) and nadir (non-ideal) points
- **step 2** sweep α between 0 and 1.
  - at each step solve the scalar LP problem:

$$\min \frac{\alpha}{\Delta E_{LB}} E_{LB} + \frac{1-\alpha}{\Delta MS_{LB}} MS_{LB}$$

    subject to all original constraints
  - from iteration to iteration only the objective changes slightly
  - only need a few more primal simplex steps to achieve optimality
- **step 3** remove duplicates (they are consecutive)
- linear objective functions and convex constraints
  - convex, lower bound Pareto front

Outer vs. Inner Approximation

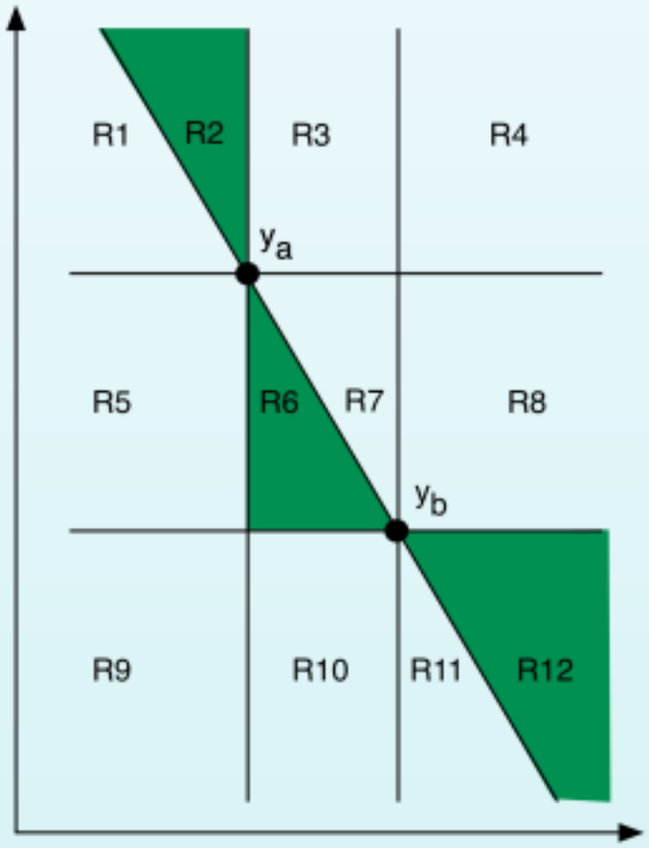
- outer approximation is a polytope that encloses 🐰
  - not all vertices are feasible solutions
- inner approximation is a polytope that is fully enclosed by 🐰
  - all vertices are feasible solutions
  - weighted sum solutions are vertices



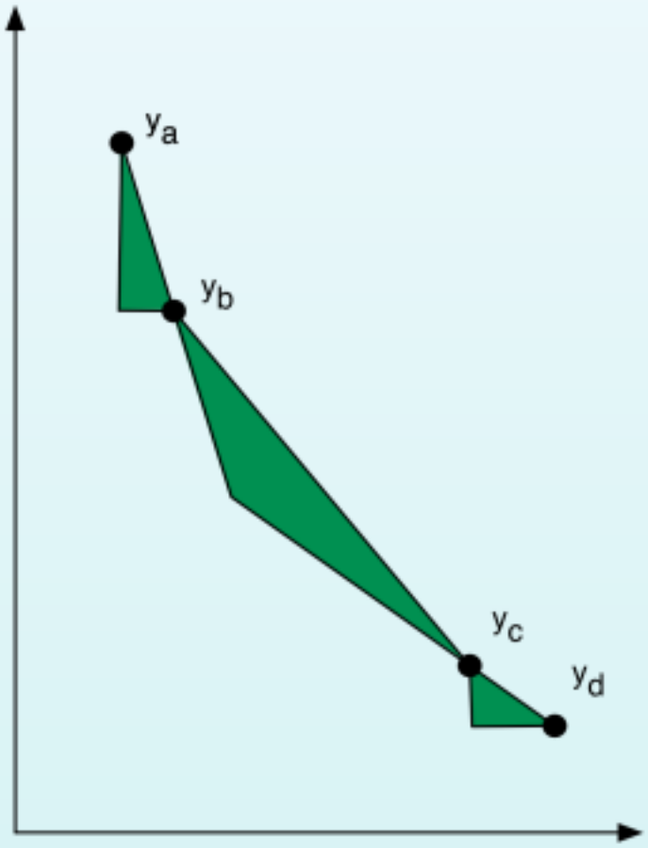
Pareto Front Lower Bound Solutions

Let $y_a, y_b, y_c, y_d \in$ [illegible]$_{ND}$

Two Efficient Points



Four Efficient Points



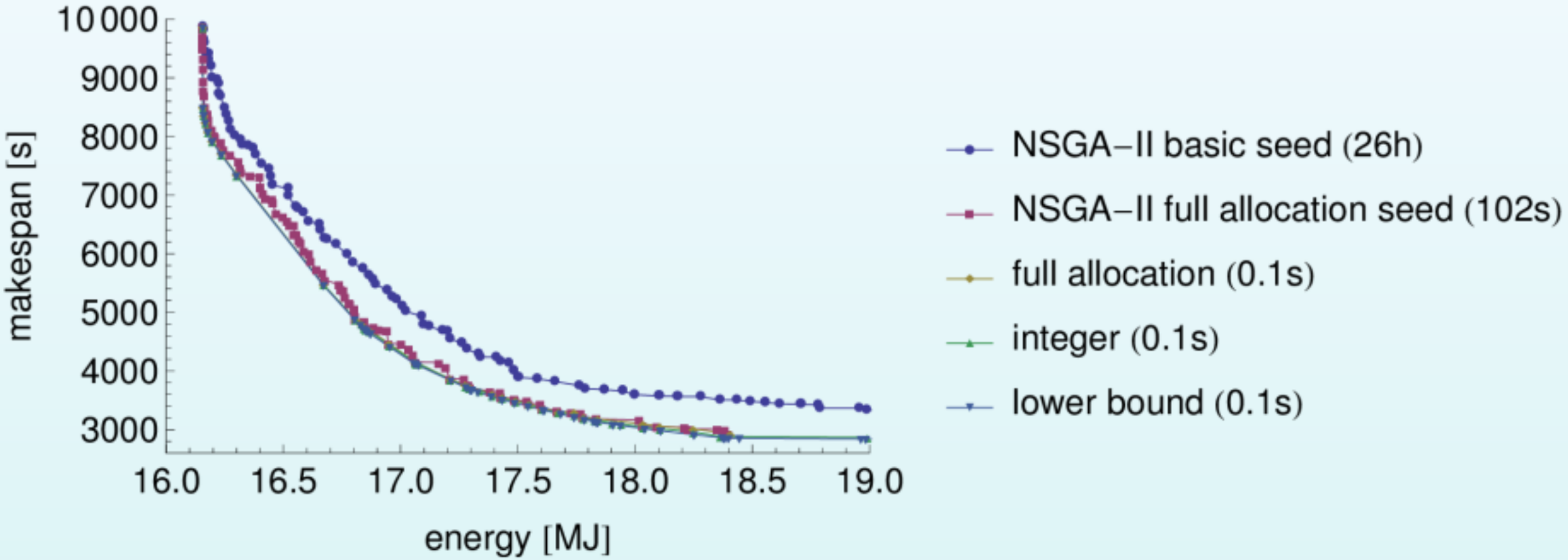
Pareto Front Generation Procedure

- **step 1** weighted sum scalarization
- **step 2** for each solution use "round near"
- **step 3** remove duplicates (they are consecutive)
- **step 4** for each solution use "local assignment"
- remove duplicates and dominated solutions
- full allocation is an upper bound on the true Pareto front

Simulation Results

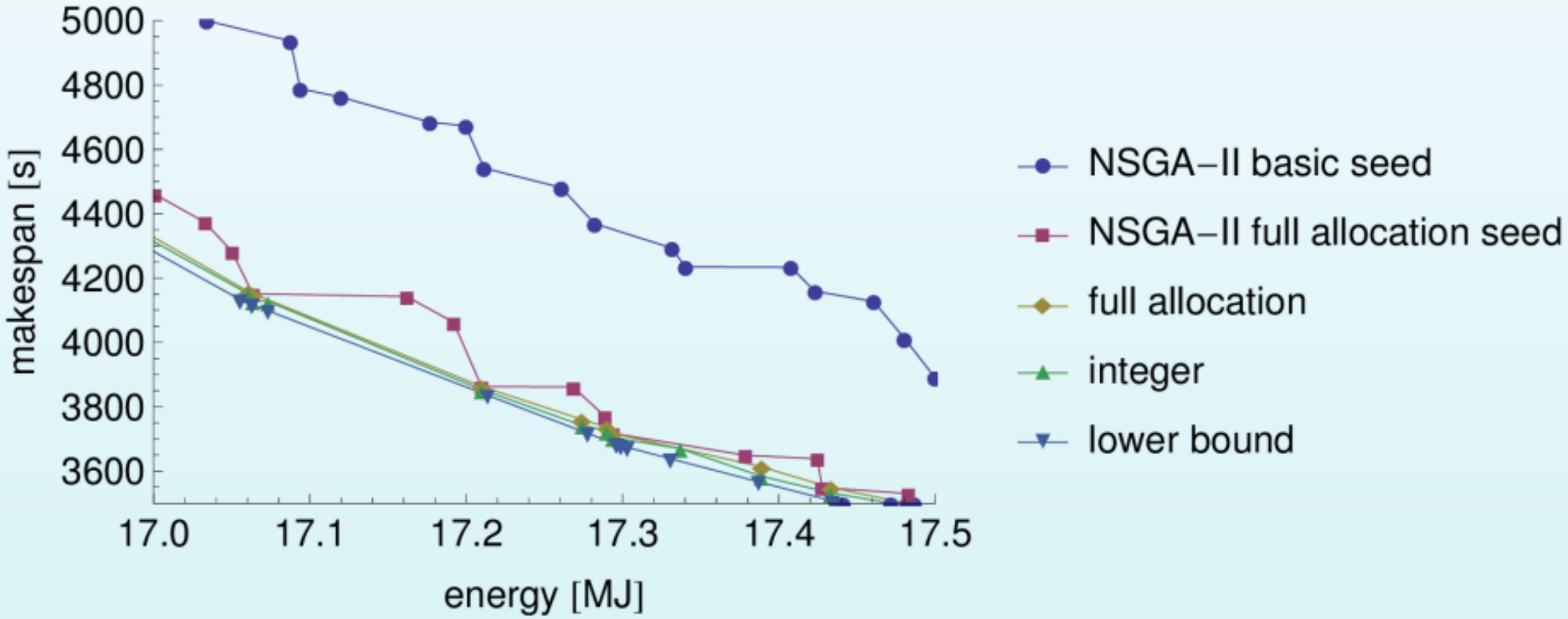
- simulation setup
  - ETC matric derived from actual systems
  - 9 machine types, 36 machines, 4 machines per type
  - 30 task types, 1100 tasks, 11-75 tasks per type
- Non-dominated Sorting Genetic Algorithm II
  - NSGA-II is another algorithm for finding the Pareto front
  - an adaptation of classical genetic algorithms
  - seeds
    - **basic:** min energy, min-min completion time, and random
    - **full allocation:** all solutions from upper bound Pareto front

Pareto Fronts



Pareto Fronts

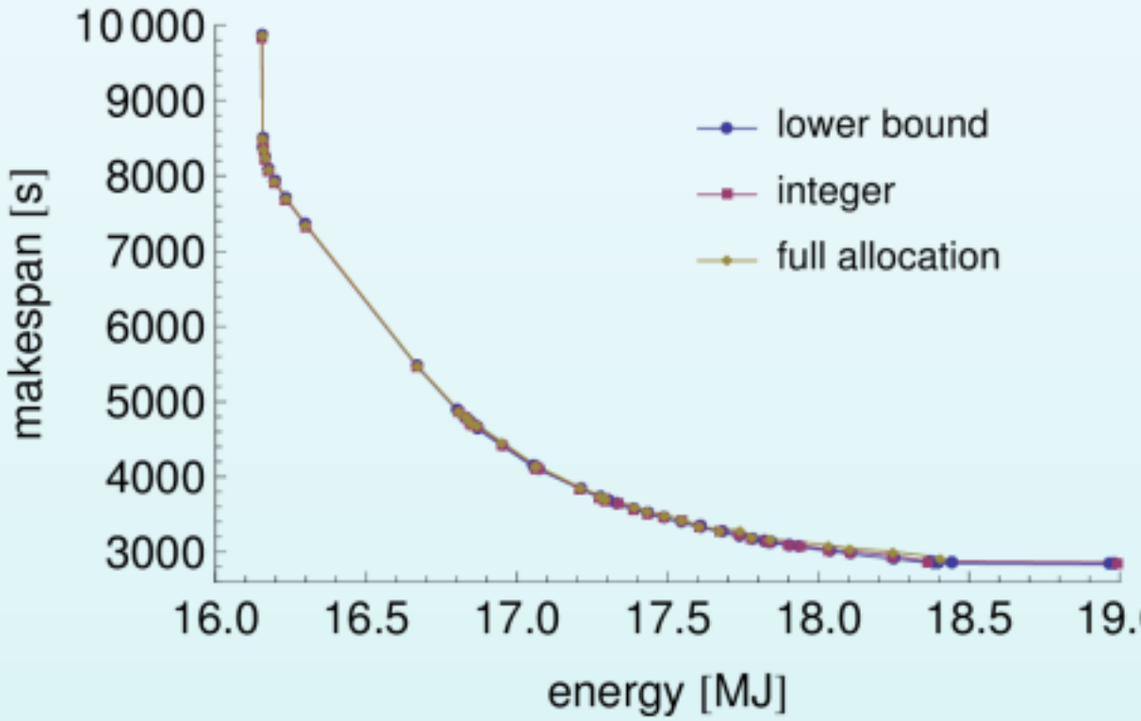
Zoomed into the knee



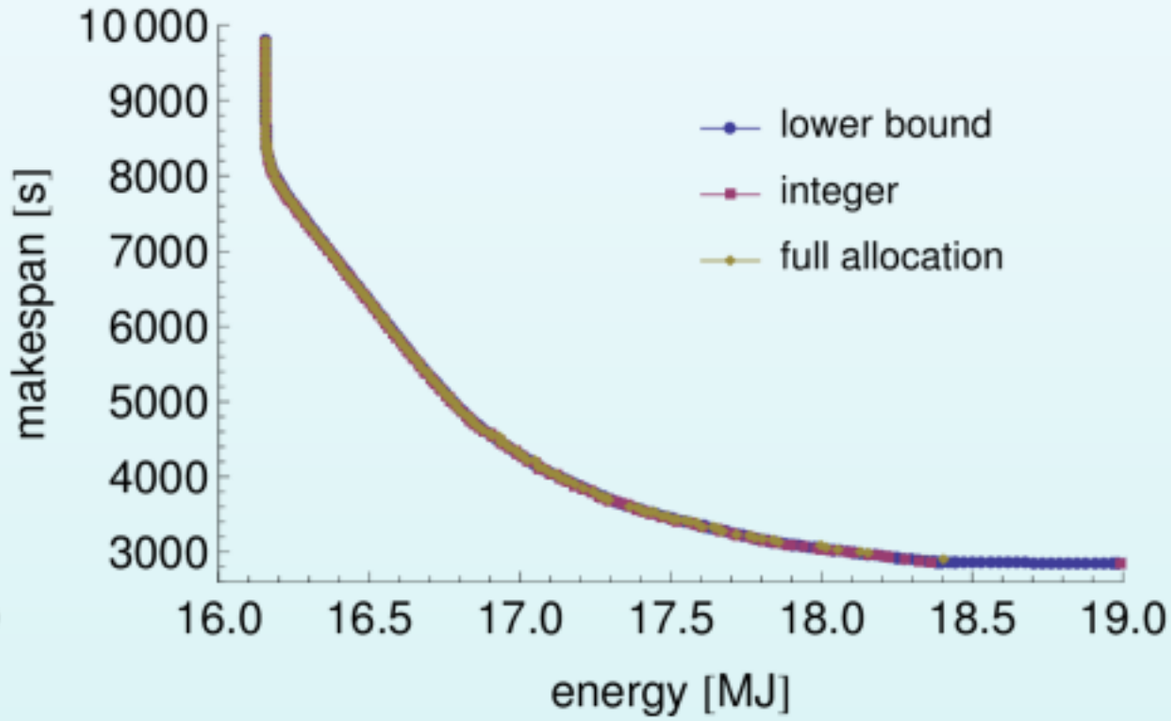
Convex Filling

Pareto Fronts

weighted sum

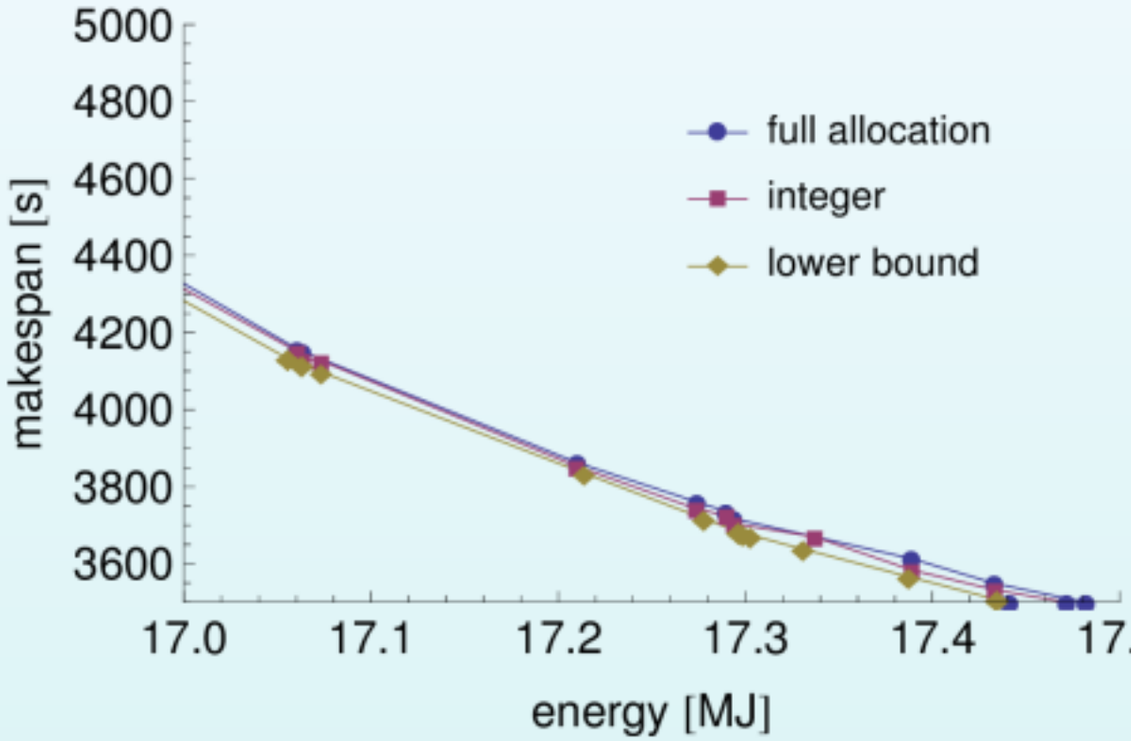


weighted sum with convex filling

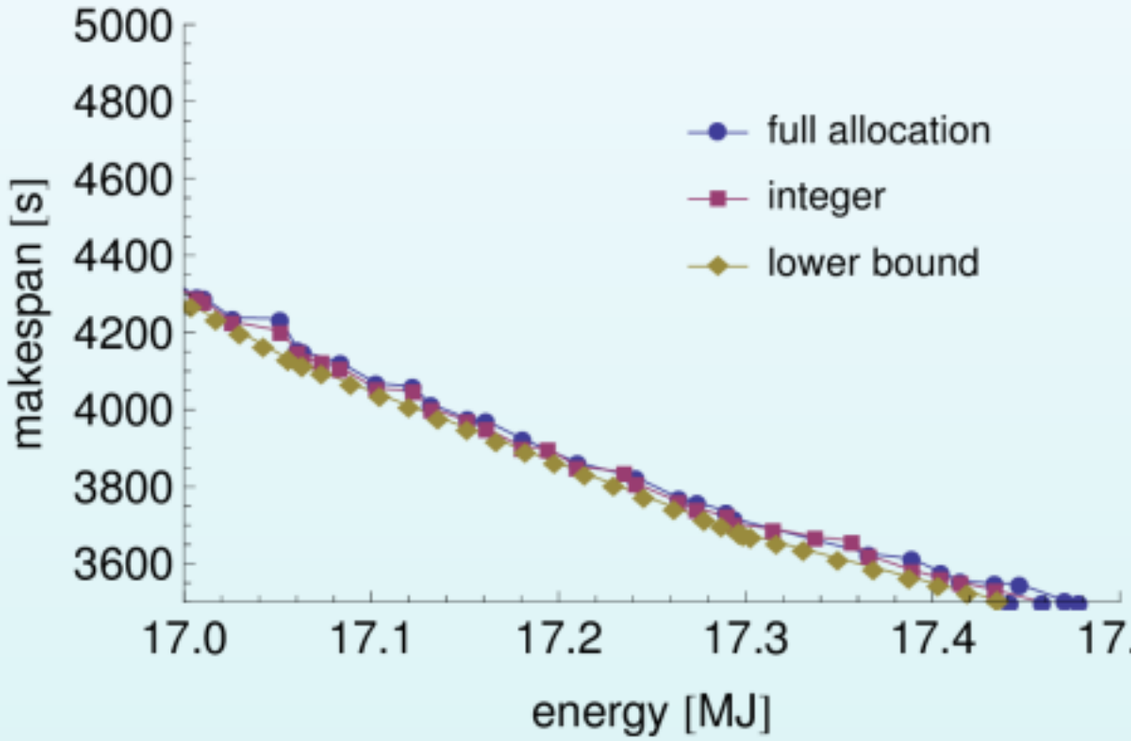


Pareto Fronts (zoomed)

weighted sum

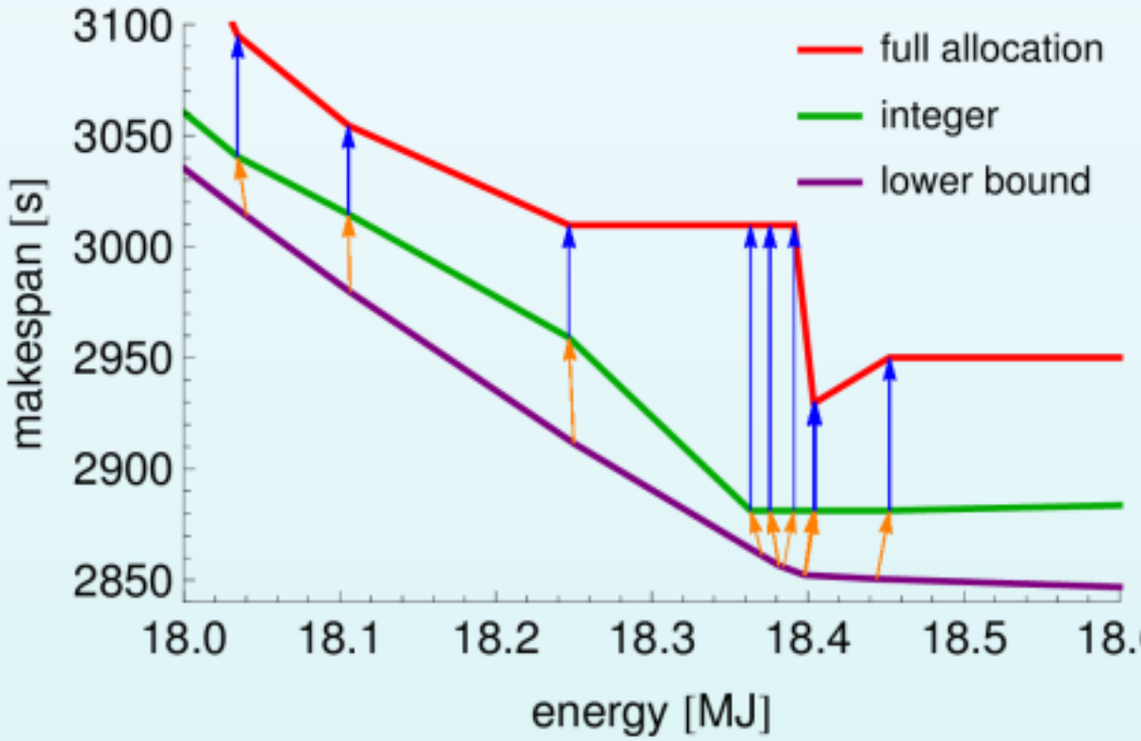


weighted sum with convex filling

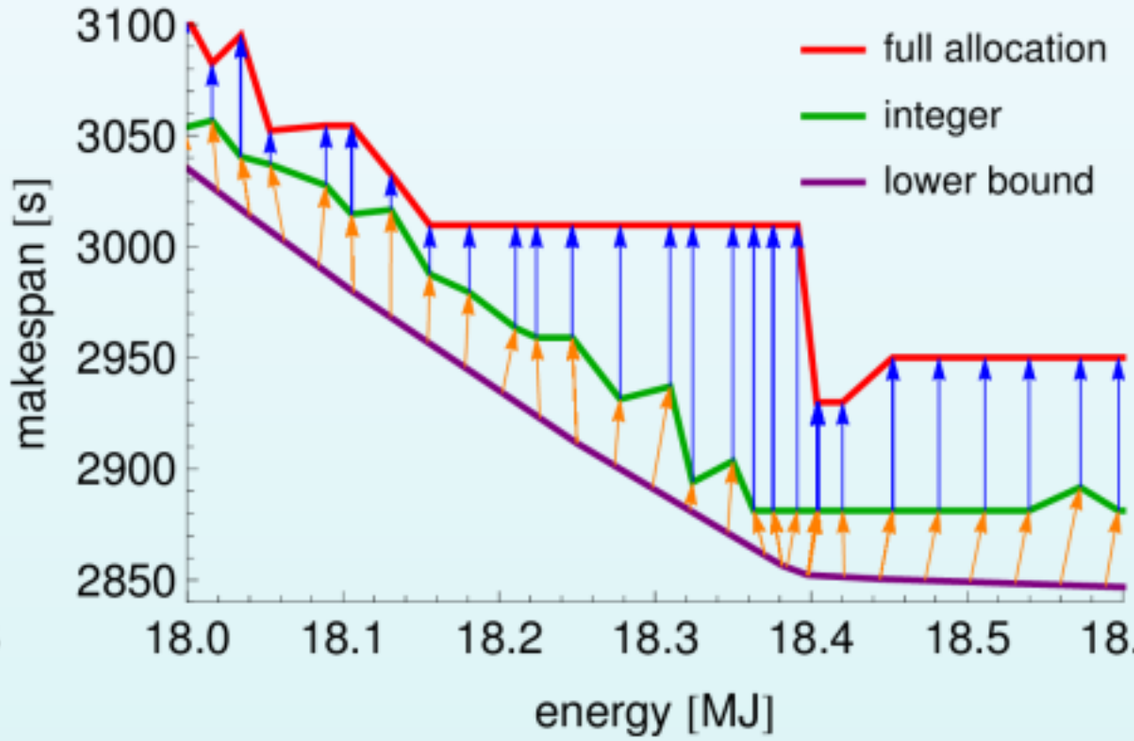


Progression of Solutions

weighted sum



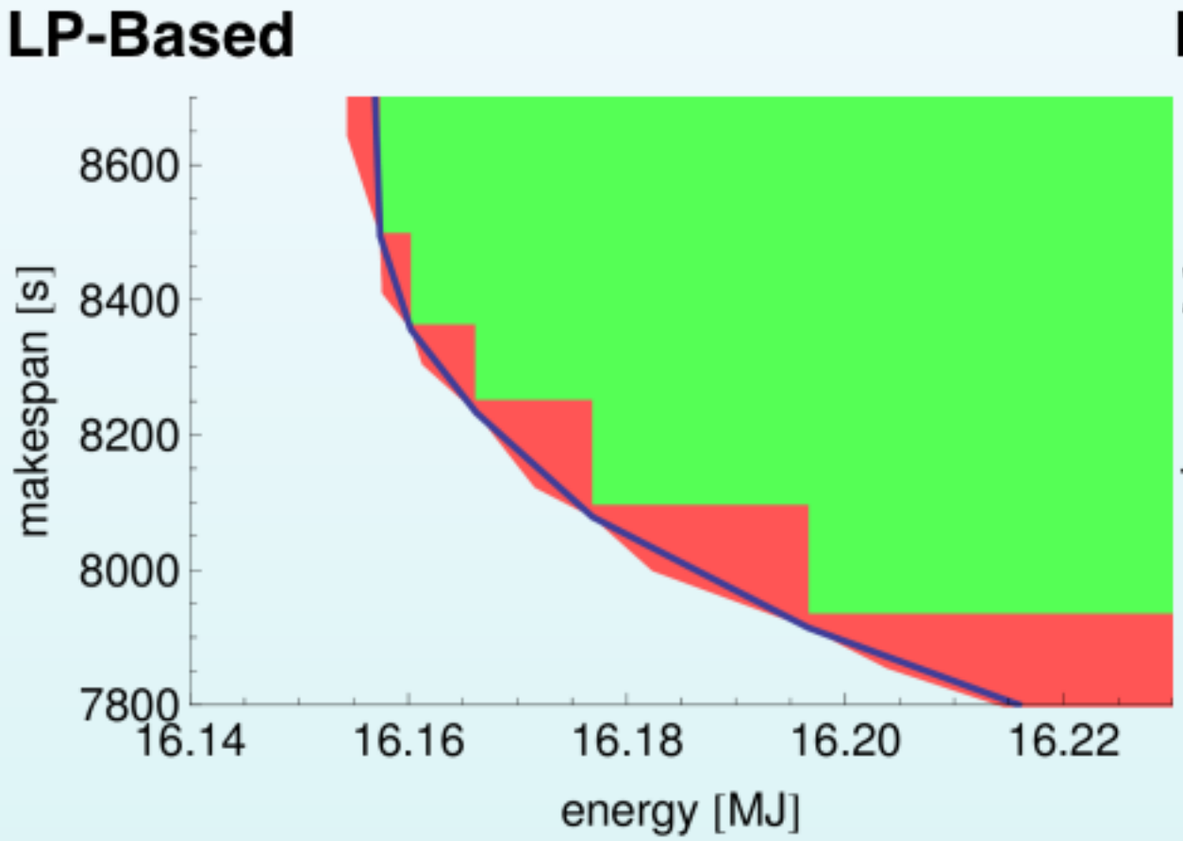
weighted sum with convex filling

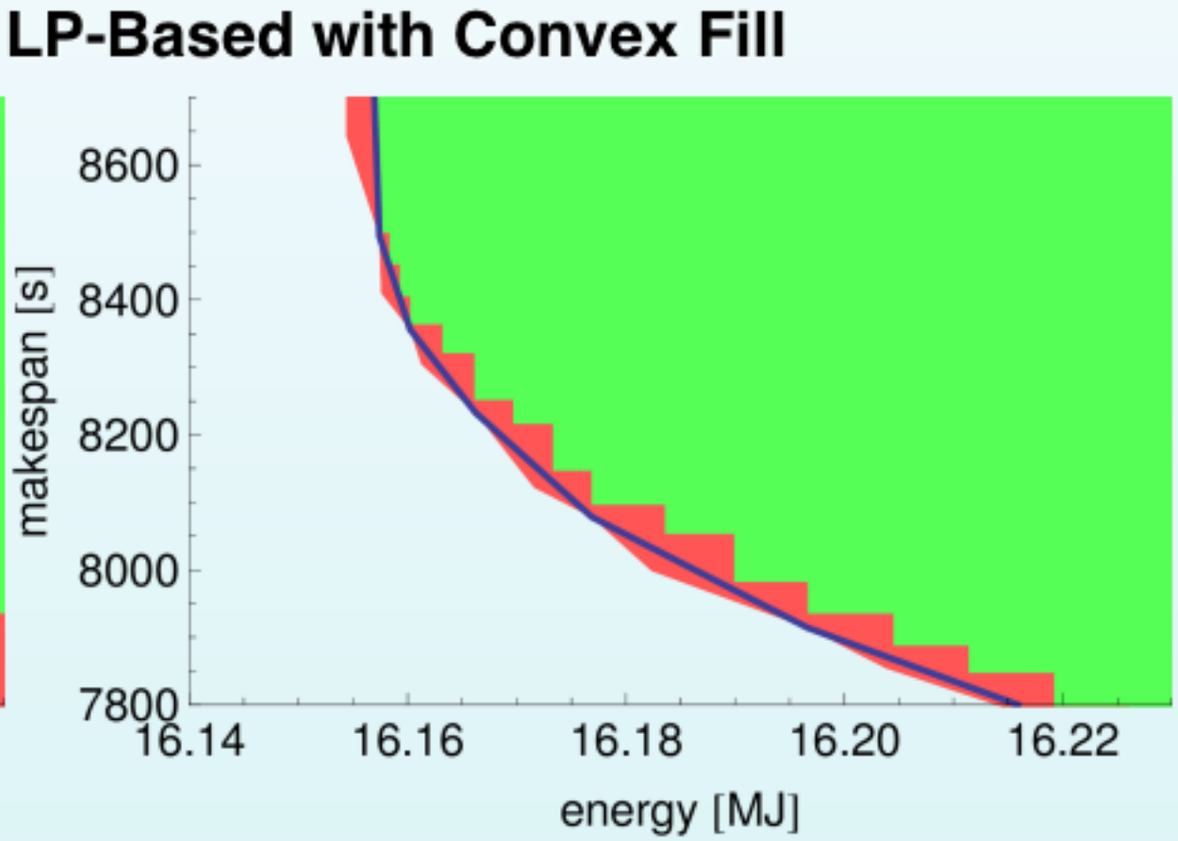


Pareto Front Quality Measure

- Pareto front is between
  - lower bound
  - upper bound (full allocation)
- compute area between these regions
- algorithm
  - compute nadir point of both lower and upper
  - compute area of lower polygon
  - compute area of upper polygon
  - compute area where the true Pareto front can reside as Area(lower) - Area(upper)
- extend the asymptotes of the Pareto front bounds to complete the polygon with the nadir point

Illustration of the Regions





Results

Area Between Bounds

algorithm	9 machine type	6 machine type	2 machine type	10 machine type
nsga	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 seeded	436 MJ s	306 MJ s	53 MJ s	851 KJ s
lp with convex fill	231 MJ s	235 MJ s	38 MJ s	762 KJ s

- nsga seeded with lp-based improves on lp-based
- convex fill improves on lp-based the most

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, to appear.

Problem Formulation

- let p be the price (revenue) per bag-of-tasks
- let c be the cost per unit of energy
- let $E(\mu)$ be the energy consumed with schedule μ
- let $MS(\mu)$ be the makespan of schedule μ
- profit per bag is $p - cE(\mu)$
- profit per unit time (to be maximized) is $\frac{p-cE(\mu)}{MS(\mu)} = \frac{p}{MS(\mu)} - c\,\frac{E(\mu)}{MS(\mu)}$
  - first term is revenue per unit time
  - second term is c times average power consumption
- let P_{max} be the maximum average power consumption
  - corresponds to the cooling capacity allocated to the HPC system
  - long running average is preferred over peak power usage

Optimization Problem

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

subject to:

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

Conversion to a Linear Program

- recall $E_{LB} = \sum_i \sum_j \mu_{ij} ETC_{ij} \left(APC_{ij} - APC_{\emptyset j} \right) + \sum_j M_j APC_{\emptyset j} MS_{LB}$
- objective and the power constraint are non-linear (bad!)
- objective is ratios of decision variables, μ_{ij} and MS_{LB} (good)
- constraints can be converted to ratios of μ_{ij} and MS_{LB}
- variable substitution
  - $z_{ij} \leftarrow \frac{\mu_{ij}}{MS_{LB}}$ is the average tasks per unit time
  - $r \leftarrow \frac{1}{MS_{LB}}$ is the number of bags per unit time
- average power consumption becomes

$$\bar{P} = \sum_i \sum_j z_{ij} ETC_{ij} \left(APC_{ij} - APC_{\emptyset j} \right) + \sum_j M_j APC_{\emptyset j}$$

Linear Program

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

subject to:

$$\forall i \qquad \sum_j z_{ij} = T_i r \qquad \text{equivalent to} \qquad \sum_j \mu_{ij} = T_i$$

$$\forall j \qquad \frac{1}{M_j} \sum_i z_{ij} ETC_{ij} \leq 1 \qquad \text{equivalent to} \qquad \frac{1}{M_j} \sum_i \mu_{ij} ETC_{ij} \leq MS_{LB}$$

$$\forall i, j \qquad z_{ij} \geq 0 \qquad \text{equivalent to} \qquad \mu_{ij} \geq 0$$

$$r \geq 0 \qquad \text{equivalent to} \qquad MS_{LB} \geq 0$$

$$\bar{P} \leq P_{max} \qquad \text{equivalent to} \qquad \frac{E_{LB}}{MS_{LB}} \leq P_{max}$$

Recovering a Feasible Allocation

- once the linear program is solved compute

$$\mu_{ij} = \frac{z_{ij}}{r}$$

$$MS_{LB} = \frac{1}{r}$$

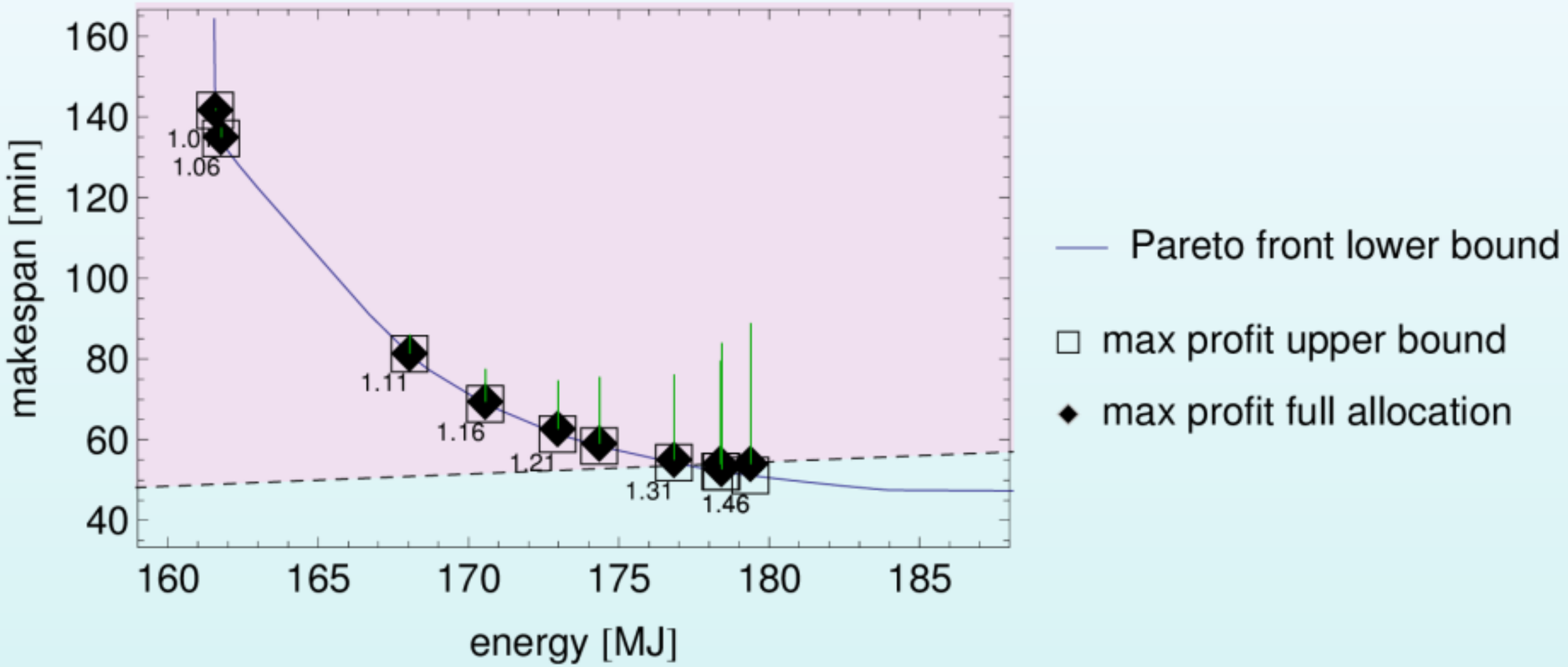
- recover the full allocation (from prior work)
  - round near algorithm
  - local assignment algorithm

Simulation Setup

- 11,000 tasks composed of 30 task types
- 360 machines composed of 9 machine types
- Pareto front generated from 1,000 points (weights)

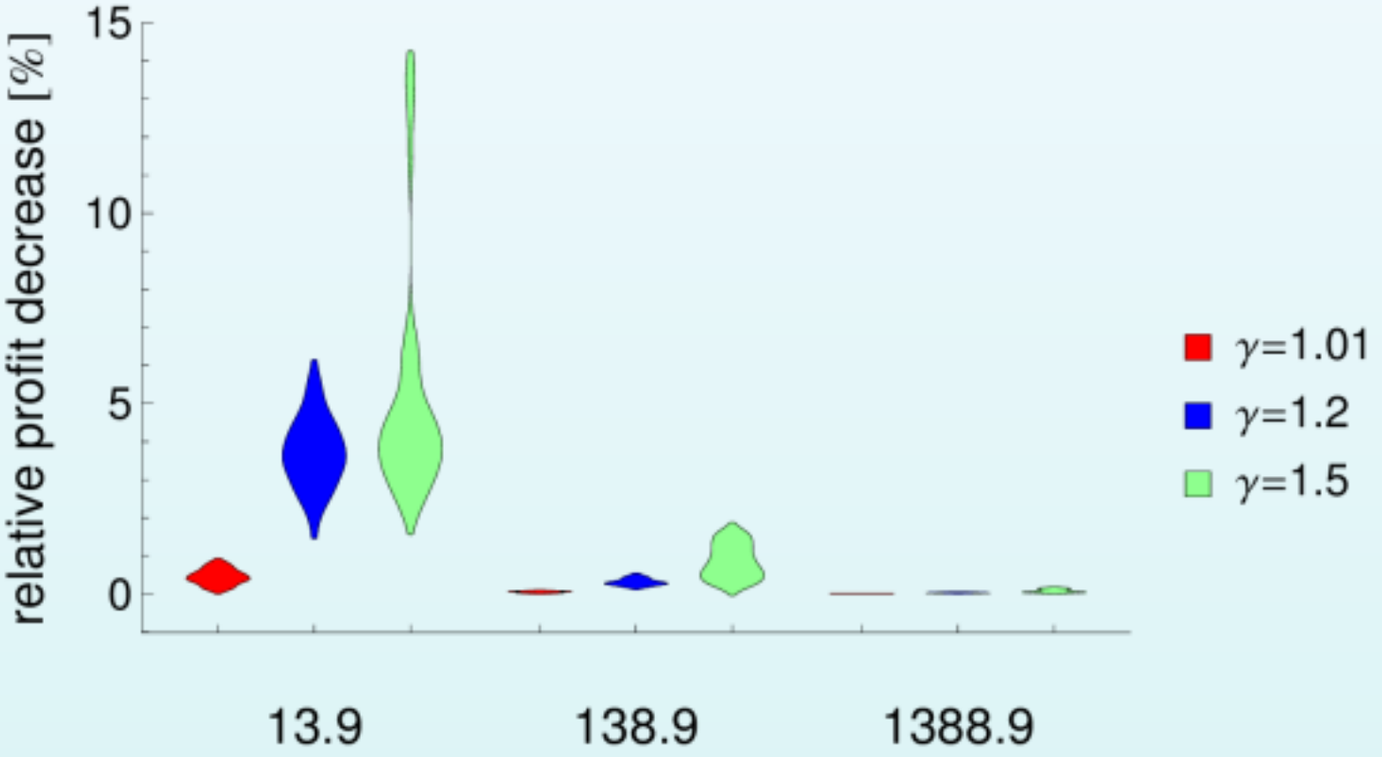
Max Profit Solutions

Sweeping Price per Bag



Relative Profit Rate vs Number of Tasks

No Idle Power, 100 Monte Carlo Runs over the Bag



$$\text{relative decrease in profit} = \frac{\text{profit}_{\text{real}} - \text{profit}_{\text{full}}}{\text{profit}_{\text{real}}} 100$$

Ongoing Research

Operating Cost Objective

- add an objective to the existing problem to optimize operating cost
- similar to common cloud computing cost models:
- let C_j be the operating cost per time unit of a machine of type j
- then the operating cost is:

$$\text{Cost} = \sum_i \sum_j \mu_{ij} \text{ETC}_{ij} C_j$$

  - Amazon EC2 charges a given rate for each machine type, rounding up the number of hours used
  - this model charges a given rate for each machine type without rounding
- if $C_j = 1$ then this reduces to total computing time
- this cost objective can be re-cast as a budget constraint

Machine Failure Model Assumptions

- machine failures are independent
- time between failures is independent of prior failures
- failure of any machine causes the bag-of-tasks to fail
- from "Optimizing Performance and Reliability on Heterogeneous Parallel Systems: Approximation Algorithms and Heuristics"
  - adapted to use the lower bound on completion time (F_j) for groups of tasks and machines

Reliability Metric Derivation

- let t_f be the time of the machine failure
- let λ_j be the MTTF of machine type j (exponential distribution)

$$\begin{aligned}\Pr(\text{machine completes work}) &= \Pr(t_f > F_j) \\ &= 1 - \Pr(t_f \le F_j) \\ &= 1 - \left(1 - e^{\lambda_j F_j}\right) \\ &= e^{-\lambda_j F_j}\end{aligned}$$

- let machine failures be independent then

$$\begin{aligned}\Pr(\text{bag-of-tasks completes}) &= \prod_j \left(e^{-\lambda_j F_j}\right)^{M_j} \\ &= \prod_j e^{-M_j \lambda_j F_j} \\ &= e^{-\sum_j M_j \lambda_j F_j}\end{aligned}$$

Reliability as an Objective

- the reliability index (to be minimized) is

$$\begin{aligned} \text{rel} &= -\ln \Pr(\text{bag completes}) \\ &= \sum_j M_j \lambda_j F_j \\ &= \sum_i \sum_j \lambda_j ETC_{ij} \mu_{ij} \end{aligned}$$

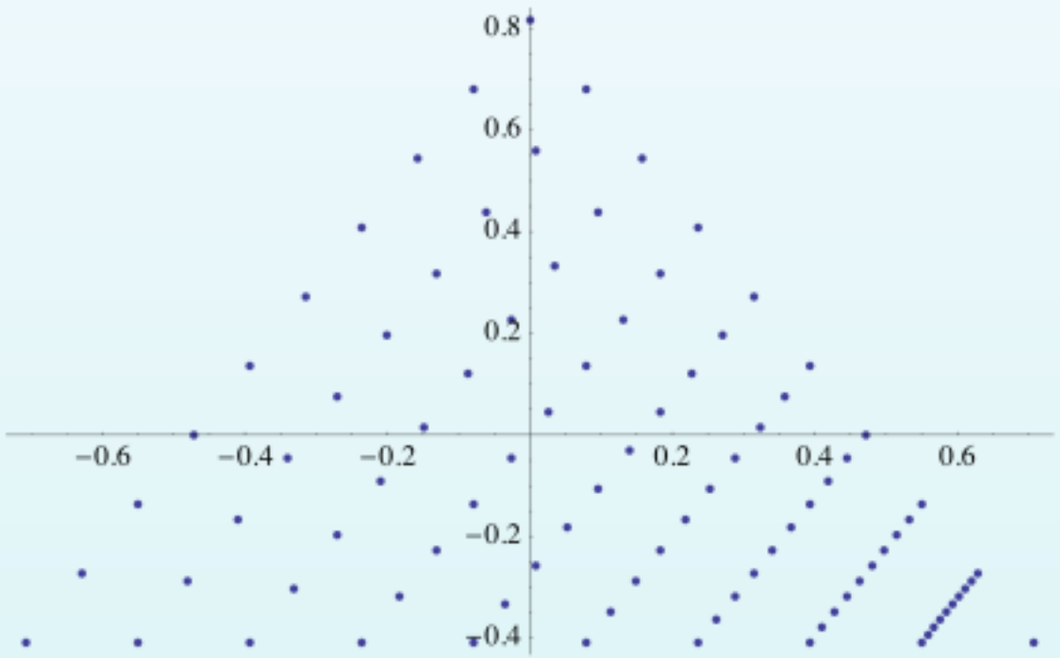
- rel is linear in μ_{ij} so it can be used as another objective function along side energy and makespan
- good schedules will try to reduce rel to increase the probability of the bag of tasks completing with no machine failures
- decreasing makespan and/or choosing more reliable machines are the scheduler's degrees of freedom
- this is really a specialization of the classic "makespan with cost" problem

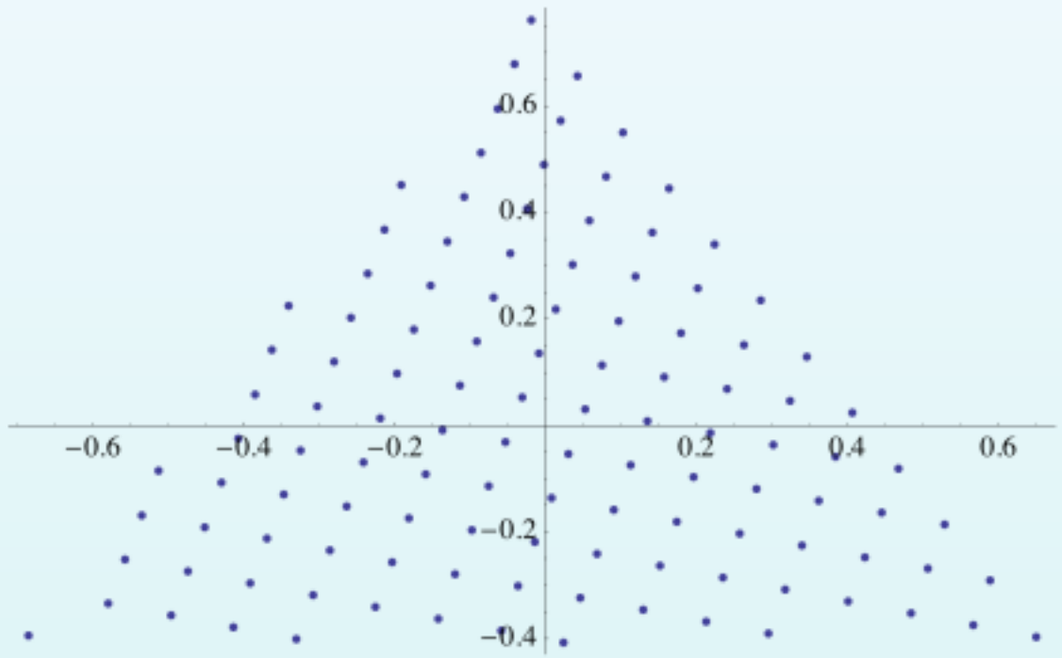
Multi-Dimensional Sweeping Algorithms

- consider a m-objective vector optimization problem
- want to scalarize the problem via weighted sums
- let the weigths be $\boldsymbol{\omega}$
- must exclude $\boldsymbol{\omega} = \mathbf{0}$ from the set of weights so we impose a constraint $\sum_{i=1}^{m} \omega_i = 1$
- ω is thus in a $m - 1$ dimensional linear subspace
- two algorithms
  - recursively combine pairs of objective functions while adding a new sweep variable
    - all sweep variables are swept independently from 0 to 1
    - produces duplicate weight vectors
    - non-uniform sweeping in the subspace defined by $\sum_{i=1}^{m} \omega_i = 1$
  - find an orthnormal basis (spanning set) for the null space of $\mathbf{1}_m$ and sweep independently in the $m - 1$ dimensional space defined by the basis vectors
    - this sweeps the subspace uniformly (doesn't prefer any objective to any other)
    - to ensure the whole space is swept one must sweep from $-\Delta$ to $+\Delta$ where $\Delta = \sqrt{1 - \frac{1}{m}}$

Comparision of Sweeping Algorithms

Recursive (left), Subspace (right)





Optimal Capacity Planning

- over subscribed systems eliminate the obvious strategies
  - machine utilization: all machines will have full utilization
  - number of tasks executed: sub-optimal scheduler
  - price/performance: does not take into account all aspects of workload and hardware
- overarching problem: current workload is different than desired/future workload
- quick algorithm to "optimally" determine how many of each type of new machines to add to a system to either
  - maximize throughput subject to a budget constraint
  - given desired throughput minimize the monetary upgrade cost

Approach

- build on steady-state problem formulation from Linear Programming Affinity Scheduling (LPAS)
- build a steady-state model of the system and find the theoretical optimal performance
- steady-state schedule is a by-product of the optimization

Steady-State Problem Formulation

- let w_i be the probability that a task of type i that will arrive
  - w_i is a PMF so $\sum_i w_i = 1$
- let p_{ij} be the fraction of time a machine of type j should process tasks of type i
- task execution rate for task type i is given by $\sum_j M_j \frac{1}{ETC_{ij}} p_{ij}$
- let λ_T be the average rate of task execution by the whole system
- let the time utilization for machine j be $p_j = \sum_i p_{ij}$

Problem Formulation

- let β_j be the cost of machine type j
- let the budget be given by γ
- for machines of type j let
  - M_j^{cur} be the current number
  - M_j^{min} be the minimum desired number
  - M_j^{max} be the maximum desired number

Maximize Throughput Subject to a Budget Constraint

Non-Linear Optimization Problem

$$\underset{p_{ij},\lambda_T,M_j}{\text{maximise}} \quad \lambda_T$$

$$\text{subject to:} \quad \forall i \quad \lambda_T w_i \le \sum_j M_j \frac{1}{ETC_{ij}} p_{ij} \qquad \text{(task constraint)}$$

$$\forall j \quad p_j \le 1 \qquad \text{(machine constraint)}$$

$$\forall i,j \quad 0 \le p_{ij} \le 1 \qquad \text{(decision variable constraint)}$$

$$\forall j \quad M_j^{min} \le M_j \le M_j^{max} \qquad \text{(decision variable constraint)}$$

$$\sum_j \left(M_j - M_j^{cur}\right)\beta_j \le \gamma \qquad \text{(budget constraint)}$$

Maximize Throughput Subject to a Budget Constraint

Conversion to Linear Optimization Problem

- change of variables, $p_{ij}M_j \rightarrow \tilde{p}_{ij}$
- let $\lambda_i = \sum_j \frac{1}{ETC_{ij}} \tilde{p}_{ij}$
- machine constraint can be rewritten as:

$$p_j = \sum_i p_{ij} = \sum_i \frac{\tilde{p}_{ij}}{M_j} = \frac{\tilde{p}_j}{M_j}$$

where: $\tilde{p}_j = \sum_i \tilde{p}_{ij}$ $M_j \geq 0$ and $p_{ij} \geq 0$ thus we can write:

$$p_j \leq 1 \implies \frac{\tilde{p}_j}{M_j} \leq 1 \implies \tilde{p}_j \leq M_j$$

- machine constraint upper bound, $p_{ij} \leq 1$ becomes $\tilde{p}_{ij} \leq M_j$

Maximize Throughput Subject to a Budget Constraint

Linear Programming Problem

$$\underset{\tilde{p}_{ij},\,\lambda_T,\,M_j}{\text{maximise}} \quad \lambda_T$$

subject to:

$$\forall i \quad \lambda_T w_i \le \sum_j \frac{1}{ETC_{ij}} \tilde{p}_{ij} \qquad \text{(task constraint)}$$

$$\forall j \quad \tilde{p}_j \le M_j \qquad \text{(machine constraint)}$$

$$\forall i, j \quad 0 \le \tilde{p}_{ij} \le M_j \qquad \text{(decision variable constraint)}$$

$$\forall j \quad M_j^{min} \le M_j \le M_j^{max} \qquad \text{(decision variable constraint)}$$

$$\sum_j \left(M_j - M_j^{cur}\right) \beta_j \le \gamma \qquad \text{(budget constraint)}$$

Minimize Cost Subject to a Throughput Constraint

Linear Programming Problem

$$\underset{\tilde{p}_{ij},\, M_j}{\text{minimize}} \quad \sum_j \left(M_j - M_j^{cur}\right)\beta_j$$

subject to:

$$\forall i, \quad \lambda_D w_i = \sum_j \frac{1}{ETC_{ij}}\tilde{p}_{ij} \qquad \text{(task constraint)}$$

$$\forall j, \quad \tilde{p}_j \le M_j \qquad \text{(machine constraint)}$$

$$\forall i, j \quad 0 \le \tilde{p}_{ij} \le M_j \qquad \text{(decision variable constraint)}$$

$$\forall j, \quad M_j^{min} \le M_j \le M_j^{max} \qquad \text{(decision variable constraint)}$$

Heterogeneity Measures

Overview

- goal: find measures of heterogeneity that "best" characterize systems
- conjectures:
  - workload + hardware = system
  - characterizations are useful for:
    - guiding/choosing scheduling algorithms
    - generating test systems

Definitions

- let ETC be the estimated time to compute matrix
- need weights (relative importance) for the rows and columns of ETC
  - let T_i be the importance (probability, arrival rate, number) of a task type i
  - let M_j be the importance (number of machines) of a machine of type j
- task easiness

$$TE_i = T_i \sum_j M_j \frac{1}{ETC_{ij}}$$

- machine performance

$$MP_j = M_j \sum_i T_i \frac{1}{ETC_{ij}}$$

Definitions

Abstraction

- focus on task (rows) and machine (column) heterogeneity measures
- measures for task heterogeity can be applied to machine heterogeneity (and vice versa)
- let x_i be the i^{th} task easiness or machine performance
- larger values of x_i are generally better
- likewise let ω_i be the i^{th} task or machine importance
- $\boldsymbol{x}$ and $\boldsymbol{\omega}$ are vectors of length N
- goal: measure the heterogeneity of $(\boldsymbol{x}, \boldsymbol{\omega})$

"Desirable" Properties (1 of 2)

- let $\mathrm{f}(\boldsymbol{x}, \boldsymbol{\omega})$ be some measure of heterogeneity:
- invariant to scale in $\boldsymbol{x}$ and $\boldsymbol{\omega}$: $\forall \boldsymbol{x}, \boldsymbol{\omega}, \alpha > 0, \beta > 0 \quad \mathrm{f}(\boldsymbol{x}, \boldsymbol{\omega}) = \mathrm{f}(\alpha \boldsymbol{x}, \beta \boldsymbol{\omega})$
  - $\mathrm{f}((1,2,3), \boldsymbol{\omega}) = \mathrm{f}((2,4,6), \boldsymbol{\omega})$
  - $\mathrm{f}(\boldsymbol{x}, (1,2,3)) = \mathrm{f}(\boldsymbol{x}, (2,4,6))$
  - requires measure to be unitless
- invariant to permutation in $\boldsymbol{x}$: $\forall \boldsymbol{x}, \boldsymbol{\omega}$, permutation matrix $\boldsymbol{P} \quad \mathrm{f}(\boldsymbol{x}, \boldsymbol{\omega}) = \mathrm{f}(\boldsymbol{P}\boldsymbol{x}, \boldsymbol{P}\boldsymbol{\omega})$
  - $\mathrm{f}((1,2,3), \boldsymbol{\omega}) = \mathrm{f}((3,1,2), \boldsymbol{\omega}) = \mathrm{f}((3,2,1), \boldsymbol{\omega})$
- perfect homogeneity: $\mathrm{f}(\mathbf{1}, \boldsymbol{\omega}) = 0$
  - or the weaker condition: $\forall \boldsymbol{y} \quad \mathrm{f}(\mathbf{1}, \boldsymbol{\omega}) \leq \mathrm{f}(\boldsymbol{y}, \boldsymbol{\omega})$
  - equality iff $\boldsymbol{y}$ has all identical elements

"Desirable" Properties (2 of 2)

- bimodal should be more heterogeneous: $\forall a > 0, b > 0 \quad f((a, a, b), \mathbf{1}) > f((a, \frac{a-b}{2}, b), \mathbf{1})$
  - $f((1, 1, 3), \mathbf{1}) > f((1, 2, 3), \mathbf{1})$
- homogenization by weighting: $\forall a \neq b, \alpha > 0, \beta > 0, \gamma > \frac{\beta}{\alpha}, \quad f((a, b), (\alpha, \beta)) > f((a, b), (\gamma\alpha, \beta))$
  - $f((1, 2), (3, 4)) > f((1, 2), (6, 4))$
  - $\gamma = 0$ also homogenizes the system
  - what are all the values of γ that homogenize?
- invariant to combination: $\forall a, b, \alpha, \beta, \gamma \quad f((a, a, b), (\alpha, \beta, \gamma)) = f((a, b), (\alpha + \beta, \gamma))$
  - $f((1, 1, 16), (1, 1, 1)) = f((1, 16), (2, 1))$
  - identical machine or task types can be safely collapsed into one type
  - likewise they can be safely replicated into different types

Empirical Property Tests

	existing measures					new measures		
	$1 - R$	$\frac{1}{R} - 1$	COV	$1 - h_g$	$1 - h$	WSD speed	WSD time	WSD log
scale in x	pass	pass	pass	pass	pass	fail	fail	pass
scale in ω	pass	pass	pass	pass	pass	fail	pass	pass
permute	pass	pass	pass	pass	pass	pass	pass	pass
combination	fail	fail	pass	fail	fail	pass	pass	pass
homogeneity	pass	pass	pass	pass	pass	pass	pass	pass
homogeneity with ω	fail	fail	pass	fail	fail	pass	pass	pass
bimodal	fail	fail	pass	fail	fail	pass	pass	pass
homogenization	pass	pass	fail	pass	pass	pass	pass	pass

Future Work

Goals

- publish unfinished work
- understand relationships between different approaches (tie together)
- expand applicability of research

Current Research

- journal for the energy/makespan scheduling work
  - review and submit (and repeat)
- optimal capacity planning
  - ties back into steady-state scheduling

Incomplete Research

Profit

- online (or steady-state) max-profit scheduling algorithm
- simulations to evaluate performance
- experiment with the non-energy costs and power constraint
- simulate dynamic price and energy costs

Incomplete Research

- reliability as a third objective
  - subspace method to sweep the free variables in weighted sum
  - resolve other issues relating to >2D Pareto front
- online pull-based (on-demand) scheduling
- improvements to LPAS for steady-state scheduling
  - recreated all (and more) of my qualifier paper's work
  - developed a randomized LPAS algorithm that out performs LPAS and MCT
  - needs more testing and development
- estimate ETC using current and prior task's execution data
  - feedback of the estimated remaining time to compute
  - number of completed ticks (N/M)
  - profile common computational benchmarks to demonstrate applicability

Incomplete Research

Stochastic ETC (and APC)

- goal: minimal model that represents the key properties of a system (i.e. has TMA)
- model a machine as "resources" (CPU, IO bandwidth)
- model a task as "consuming" or "requiring" differing amounts of "resources"
- derive the probability distribution for ETC for a given task and machine (joint distribution)
- evaluate (statistical) heterogeneity measures against model
- generate test systems from model

Incomplete Research

Heterogeneity Measures

- prove/disprove "desirable" properties for each measure
- develop applicable task/machine affinity measure
- relate these measures back to scheduling algorithm performance
- parameterized tasks
  - infinite number of task types
  - need a joint PDF over parameter and ETC
  - heterogeneity measures via "WSD" or "WSD log" are still possible

Proposed Timeline

- Spring 2014:
  - submit LP-scheduling journal paper
  - complete research on capacity planning
- Summer 2014:
  - reliability
  - online pull-based scheduling with the LP
- Fall/Spring 2014:
  - stochastic ETC
  - heterogeneity measures

Possible Work Products

- conference: LP-based Pareto front (WCO 2013, published, best paper)
- book chapter: LP-based Pareto front book chapter (Springer, accepted)
- conference: maximum profit scheduling (ExtremeGreen 2014 during CCGrid, accepted)
- journal: LP-based scheduling (Pareto, Min-Min, more systems) (IEEE Transactions PDS, nearly complete)
- conference: optimal capacity planning for HPC
- conference: reliability
- journal: LPAS, LP batch, and others online pull-based scheduling
- conference: ETC estimation via instrumented tasks
- conference: stochastic ETC/APC model
- journal: stochastic ETC/APC and heterogeneity measures

Questions?

Goals

- publish unfinished work
- understand relationships between different approaches (tie together)
- expand applicability of research

Current Research

- journal for the WCO work
  - review and submit (and repeat)
- optimal capacity planning
  - ties back into steady-state scheduling work

Incomplete Research

Profit

- online (or steady-state) max-profit scheduling algorithm
- simulations to evaluate performance
- experiment with the non-energy costs and power constraint
- simulate dynamic price and energy costs

Incomplete Research

- reliability as a third objective
  - subspace method to sweep the free variables in weighted sum
  - resolve other issues relating to >2D Pareto front
- online pull-based (on-demand) scheduling
- improvements to LPAS for steady-state scheduling
  - recreated all (and more) of my qualifier paper's work
  - developed a randomized LPAS algorithm that out performs LPAS and MCT
  - needs more testing and development
- estimate ETC using current and prior task's execution data
  - feedback of the estimated remaining time to compute
  - number of completed ticks (N/M)
  - profile common computational benchmarks to demonstrate applicability

Incomplete Research

Stochastic ETC (and APC)

- goal: minimal model that represents the key properties of a system (i.e. has TMA)
- model a machine as "resources" (CPU, IO bandwidth)
- model a task as "consuming" or "requiring" differing amounts of "resources"
- derive the probability distribution for ETC for a given task and machine (really need the joint distribution)
- evaluate (statistical) heterogeneity measures against this model
- generate test systems from model

Incomplete Research

Heterogeneity Measures

- prove/disprove "desirable" properties for each measure
- develop applicable task/machine affinity measure
- relate these measures back to scheduling algorithm performance
- parameterized tasks
  - infinite number of task types
  - need a joint PDF over parameter and ETC
  - heterogeneity measures via "WSD" or "WSD log" are still possible

Proposed Timeline

- Spring 2014:
  - submit LP-scheduling journal paper
  - complete research on capacity planning
- Summer 2014:
  - reliability
  - online pull-based scheduling with the LP
- Fall/Spring 2014:
  - stochastic ETC
  - heterogeneity measures

Possible Work Products

- conference: LP-based Pareto front (WCO 2013, published, best paper)
- book chapter: LP-based Pareto front book chapter (Springer, accepted)
- conference: maximum profit scheduling (ExtremeGreen 2014 during CCGrid, accepted)
- journal: LP-based scheduling (Pareto, Min-Min, more systems) (IEEE Transactions PDS, nearly complete)
- conference: optimal capacity planning for HPC
- conference: reliability
- journal: LPAS, LP batch, and others online pull-based scheduling
- conference: ETC estimation via instrumented tasks
- conference: stochastic ETC/APC model
- journal: stochastic ETC/APC and heterogeneity measures

Questions?