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

  - 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


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
Pareto Fronts

- NSGA-II w/basic seed (26h)
- NSGA-II w/full alloc. seed (102s)
- LP-based full allocation (0.1s)
Illustration of the Regions

LP-Based

LP-Based with Convex Fill
## Results

### Area Between Bounds

<table>
<thead>
<tr>
<th>algorithm</th>
<th>9 machine type</th>
<th>6 machine type</th>
<th>2 machine type</th>
<th>10 machine type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>2149 MJ s</td>
<td>1351 MJ s</td>
<td>115 MJ s</td>
<td>2655 KJ s</td>
</tr>
<tr>
<td>LP-based</td>
<td>684 MJ s</td>
<td>339 MJ s</td>
<td>63 MJ s</td>
<td>1011 KJ s</td>
</tr>
<tr>
<td>NSGA-II seeded</td>
<td>436 MJ s</td>
<td>306 MJ s</td>
<td>53 MJ s</td>
<td>851 KJ s</td>
</tr>
<tr>
<td>LP-based with convex fill</td>
<td>231 MJ s</td>
<td>238 MJ s</td>
<td>38 MJ s</td>
<td>762 KJ s</td>
</tr>
</tbody>
</table>

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

Publications

  - 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

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 $\lambda_i$ be the arrival rate tasks of type $i$
- let $p_{ij}^r$ 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 $i$
- $\frac{1}{\text{ETC}_{ij}}$ is the expected computation speed (tasks per second)
- task execution rate for task type $i$ is given by $\sum_j \frac{1}{\text{ETC}_{ij}} p_{ij}^r$
Problem Formulation

- for machines of type $j$ let
  - $M_j^{\text{cur}} \rightarrow$ current number
  - $M_j^{\text{min}} \rightarrow$ minimum desired number
  - $M_j^{\text{max}} \rightarrow$ 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 $M_j = M_j^{\text{cur}} + M_j^B - M_j^S$

- let $M^{\text{min}}$ and $M^{\text{max}}$ be the limits on the total number of machines allowed
Machine Pricing

- let $\beta_j^B$ and $\beta_j^S$ 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)
ETC

- let there be $L$ abstract computational operation types
- let $n_{il}$ be the number of operations of type $l$ for tasks of type $i$
- let $\tau_{ij}$ be the seconds per operation of type $l$ on a machine of type $j$

then $ETC_{ij} = \sum_i n_{il} \tau_{ij}$ and in matrix form $ETC = \eta \tau$

- properties sufficient for heterogeneous computing systems
  - $ETC$ can have nonzero task easiness homogeneity (TEH) and machine performance homogeneity (MPH)
  - for $L > 1$ it can have nonzero TMA

- given an $ETC$, compute $\eta$ and $\tau$ via rank $L$ non-negative matrix factorization (NNMF)
Modeling

APC

- let $APC_{\theta j}$ be the static power for a machine of type $j$
- let $\psi_{ij}$ be the dynamic energy to execute an operation of type $i$ on a machine of type $j$
- energy of type $i$ operations is $\eta_{ii} \psi_{ij}$
- total energy is $\sum_i \eta_{ii} \psi_{ij}$
- total average dynamic power is $APC_{ij} = \frac{\sum_i \eta_{ii} \psi_{ij}}{\sum_i \eta_{ii} \nu_{ij}}$
- find the model for ETC, then use least squares to find $\psi_{ij}$
Modeling

Reliability

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

Objectives

\[
\begin{aligned}
&\text{minimize} \quad M^B, M^S, \hat{\beta} \\
&\quad \left( \begin{array}{ccc}
- \sum_i r_i \sum_j \frac{1}{ETC_{ij}} \tilde{p}_{ij} \\
\sum_j M_j^B \beta_j^B - \sum_j M_j^S \beta_j^S \\
\sum_i \sum_j \nu_j \tilde{p}_{ij} \\
\sum_i \sum_j APC_{ij} \tilde{p}_{ij} + \sum_j APC_{\beta j} M_j
\end{array} \right) \\
&\quad \text{reward rate} \\
&\quad \text{upgrade cost} \\
&\quad \text{failure rate} \\
&\quad \text{power}
\end{aligned}
\]

- $\tilde{p}_{ij}$ is the number of machines running tasks of type $i$ on machines of type $j$
- $\beta_j^B$ and $\beta_j^S$ are the buying and selling price
Linear Vector Optimization Problem

Constraints

\[ \forall j \quad M_j^{\text{min}} \leq M_j \leq M_j^{\text{max}} \quad \text{per type quantity limitations} \]
\[ M_j^{\text{min}} \leq \sum_j M_j \leq M_j^{\text{max}} \quad \text{overall quantity limitation} \]
\[ \forall j \quad M_j^B \geq 0 \land M_j^S \geq 0 \quad \text{buy and sell non-negativity} \]
\[ \forall i \quad \sum_j \frac{1}{ETC_{ij}} p_{ij} \leq \lambda_i \quad \text{task arrival rate} \]
\[ \forall j \quad \sum_i p_{ij} \leq M_j \quad \text{machine utilization} \]
\[ \forall i, j \quad 0 \leq p_{ij} \quad \text{non-negative schedule} \]

- \( p_{ij} \) is the number of machines running tasks of type \( i \) on machines of type \( j \)
- \( M_j = M_j^{\text{cur}} + M_j^B - M_j^S \) is the number of machines of type \( j \)
Linear Vector Optimization Problem

Extra Constraints

\[ \sum_j M_j^B \beta_j^B - \sum_j M_j^S \beta_j^S \leq \beta \]  \hspace{1cm} \text{budget}

\[ \sum_i \sum_j \nu_{ij} p_{ij} \leq \nu_{\text{max}} \]  \hspace{1cm} \text{failure rate}

\[ \sum_i \sum_j \text{APC}_{ij} p_{ij} + \sum_j \text{APC}_{\emptyset j} M_j \leq P_{\text{max}} \]  \hspace{1cm} \text{power}

- \( \beta \) is the budget
- \( \nu_{\text{max}} \) is the maximum system failure rate
- \( P_{\text{max}} \) is the maximum power consumption
Stochastic Model

- uncertain parameters: $\lambda$, ETC, and APC
- three random matrices define ETC and APC
  - $\eta \rightarrow$ property of the tasks
  - $\tau$ and $\psi \rightarrow$ property of the machines
- $\eta_{il}$, $\tau_{ij}$, and $\psi_{lj}$ are modeled as independent uniform random variables
- optimization needs PDF of $\frac{1}{\text{ETC}_{ij}} = \frac{1}{\sum_i \eta_{il} \tau_{ij}}$
  - 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

<table>
<thead>
<tr>
<th>linear program</th>
<th>stochastic program with recourse</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimize ( c^T x ) with ( x \geq 0 )</td>
<td>minimize ( c^T x + \mathbb{E}_\xi [Q(x, \xi)] ) with ( x \geq 0 )</td>
</tr>
<tr>
<td>subject to: ( Ax = b )</td>
<td>subject to: ( Ax = b )</td>
</tr>
</tbody>
</table>

Where: \( Q(x, \xi) = \min_y q(\xi)^T y \) such that \( T(\xi)x + W(\xi)y = h(\xi) \) and \( y \geq 0 \)

- \( \xi \) is a random vector representing the uncertain parameters
- \( Q(x, \xi) \) is called the value function
- The expected value function is \( \mathbb{E}_\xi [Q(x, \xi)] \)
### Resource Provisioning via Stochastic Programming

<table>
<thead>
<tr>
<th></th>
<th>First Stage</th>
<th>Second Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stochastic</strong></td>
<td>$\lambda$, ETC, and APC</td>
<td>nothing</td>
</tr>
<tr>
<td><strong>Decision Variable</strong></td>
<td>$x = \begin{pmatrix} M^B \ M^S \end{pmatrix}$</td>
<td>$y = \text{flatten}(\tilde{p})$</td>
</tr>
<tr>
<td><strong>Objective</strong></td>
<td>$c^T x$</td>
<td>$q^T y$</td>
</tr>
<tr>
<td><strong>Objective Coefficients</strong></td>
<td>$c$ is a function of $\beta^B$, $\beta^S$, and APC₂</td>
<td>$q$ is a function of the reward $r$ and ETC</td>
</tr>
<tr>
<td><strong>Constraints</strong></td>
<td>$Ax = b$, $x \geq 0$</td>
<td>$Tx + Wy = h$, $y \geq 0$</td>
</tr>
<tr>
<td><strong>Constraint Coefficients</strong></td>
<td>$A$ is a function of $\beta^B$, $\beta^S$</td>
<td>$h$ is a function of $\lambda$</td>
</tr>
<tr>
<td></td>
<td>$b$ is a function of $\beta$ and machine limits</td>
<td>$T$ and $W$ are functions of ETC and APC</td>
</tr>
<tr>
<td><strong>Action</strong></td>
<td>buy machines $M^B$, sell machines $M^S$</td>
<td>use schedule $\tilde{p}$ for task assignment</td>
</tr>
</tbody>
</table>
Deterministic Equivalent Linear Program

\[
\begin{align*}
\text{minimize} \quad & \mathbf{c}^T \mathbf{x} + p_1 q_1^T y_1 + p_2 q_2^T y_2 + \cdots + p_K q_K^T y_K \\
\text{subject to:} \quad & A\mathbf{x} = \mathbf{b} \\
& T_1 \mathbf{x} + W_1 y_1 = h_1 \\
& T_2 \mathbf{x} + W_2 y_2 = h_2 \\
& \vdots \\
& T_K \mathbf{x} + W_K y_K = h_K \\
& \mathbf{x}, \ y_1, \ y_2, \ \cdots \ y_K \geq 0
\end{align*}
\]
Value of Information

- Let \( z(x, \xi) \) be the optimal objective function value for a given \( x \) and a particular scenario \( \xi \)

\[
z(x, \xi) = c^T x + \min_y \{ q(\xi)^T y \mid T(\xi)x + W(\xi)y = h(\xi) \land y \geq 0 \}
\]

- Wait-and-see problem
  - Wait until \( \xi \) is realized then find optimal solution
  - \( WS = E_\xi [\min_x z(x, \xi)] \)
  - Requires perfect information \( \rightarrow \) unachievable

- Recourse problem (stochastic problem)
  - \( RP = \min_x E_\xi [z(x, \xi)] \)
  - Achievable, optimal strategy

- Expected value problem (mean value problem)
  - Use means of all parameters
  - \( EV = \min_x z \left( x, E_\xi [\xi] \right) \)
  - Achievable, sub-optimal
  - Expected value of using the EV solution, \( x_{EV} \) is
  - \( E_EV = E_\xi [z(x_{EV}, \xi)] \)
  - Uses optimal second, third, etc, stage decisions
Value of Information

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

\[
z(x, \xi) = c^T x + \min_y \{ q(\xi)^T y \mid T(\xi)x + W(\xi)y = h(\xi) \land y \geq 0 \}
\]

- expected value of perfect information is
  \( \text{EVPI} = \text{RP} - \text{WS} \geq 0 \)

- value of the stochastic solution is
  \( \text{VSS} = \text{EEV} - \text{RP} \geq 0 \)

- wait-and-see problem
  - wait until \( \xi \) is realized then find optimal solution
  - \( \text{WS} = E_\xi [\min_x z(x, \xi)] \)
  - requires perfect information \( \rightarrow \) unachievable

- recourse problem (stochastic problem)
  - \( \text{RP} = \min_x E_\xi [z(x, \xi)] \)
  - achievable, optimal strategy

- expected value problem (mean value problem)
  - use means of all parameters
  - \( \text{EV} = \min_x z(x, E_\xi[\xi]) \)
  - achievable, sub-optimal
  - expected value of using the EV solution, \( x_{\text{EV}} \) is
    \( \text{EEV} = E_\xi[z(x_{\text{EV}}, \xi)] \)
  - uses optimal second, third, etc, stage decisions
Comparison Purchasing Strategies

- strategies... buy machines of type

\[ H1: \quad j^* = \arg \max_j \sum_i \frac{1}{ETC_{ij}} \]  
\[ \text{highest performing machine} \]

\[ H2: \quad j^* = \arg \max_j \frac{1}{\beta_j^B} \sum_i \frac{1}{ETC_{ij}} \]  
\[ \text{highest performance/price machine} \]

\[ H3: \quad j^* = \arg \max_j \frac{1}{\beta_j^B} \sum_i \lambda_i r_i \frac{1}{ETC_{ij}} \]  
\[ \text{highest relevant performance/price machine} \]

- 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

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

### Solution

<table>
<thead>
<tr>
<th></th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
<th>MVP</th>
<th>RP</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>31.8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10.</td>
</tr>
<tr>
<td>M2</td>
<td>0</td>
<td>67.3</td>
<td>67.3</td>
<td>8.5</td>
<td>26.9</td>
</tr>
<tr>
<td>M3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>32.</td>
<td>11.5</td>
</tr>
<tr>
<td>M4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-10.</td>
<td>-6.7</td>
</tr>
</tbody>
</table>
Medium Sized Problem

Comparison

![Comparison Chart](image_url)

- Reward Rate:
  - H1: N/A
  - H2: N/A
  - H3: N/A
  - MVP: N/A
  - RP: N/A
  - WS: N/A

- Expected Reward Rate:
  - H1: 3000
  - H2: 3000
  - H3: 3000
  - MVP: 3000
  - RP: 4000
  - WS: EVPI=26.2%
  - VSS=13.3%
Medium Sized Problem

Relative Performance per Scenario

RP Objective PDF

-100 0 100 200 300 400
relative improvement of RP [%]

H1 H2 H3 MVP

0 0.00000 0.00005 0.00010 0.00015 0.00020 0.00025 0.00030
probability density

0 5000 10000 15000
reward rate
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 $\tau$ (seconds per operation)
  - given ETC and APC, computed $\eta$, $\tau$, and $\psi$ with NNMF and least squares
  - given CoV, computed the variance of $\tau$ via least squares
- using uniform distributions for $\tau$ 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

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

### Solution

<table>
<thead>
<tr>
<th></th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
<th>MVP</th>
<th>RP</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M2</td>
<td>0</td>
<td>168.8</td>
<td>168.8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.3</td>
</tr>
<tr>
<td>M4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12.7</td>
</tr>
<tr>
<td>M7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.5</td>
</tr>
<tr>
<td>M8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100.</td>
<td>73.8</td>
</tr>
<tr>
<td>M9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.1</td>
</tr>
</tbody>
</table>
Nine Machine Type Environment

Comparison

expected reward rate

<table>
<thead>
<tr>
<th></th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
<th>MVP</th>
<th>RP</th>
<th>WS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>$400K</td>
<td>$400K</td>
<td>$400K</td>
<td>$291K</td>
<td>$274K</td>
<td>$233K</td>
</tr>
</tbody>
</table>

relative improvement of RP [%]

<table>
<thead>
<tr>
<th></th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
<th>MVP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

-50  0  50  100  150
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 ETC and 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