ICS NEWS
INFORMS Computing Society Newsletter

Inside This Issue

"Where there's a will there's a way!"

Message from the Chair: Sharing Your Cool Toys

Ted Ralphs
Industrial and Systems Engineering
Lehigh University
ted@lehigh.edu

Greetings from "The Big Cheese!" I want to start by thanking all of you for the opportunity to serve as chair of ICS and also by thanking Bill Cook for his excellent guidance of the society over the previous two years. Since I took office, the beer mug on my desk has served as a makeshift "royal sceptre" and reminds me daily of my duty to the society that has been my home and sanctuary within INFORMS for many years now. Obviously, the mug also reminds me daily that a beer would taste mighty good right now!

As chair, I've been thinking about what the mission of ICS really is and what kinds of initiatives would have the greatest impact on that mission. Feel free to drop me a line and let me know what you're hoping to accomplish and what initiatives within ICS would help support that! On some level, the purpose of any academic society is to give everyone the chance to share their cool toys with others in one big communal sandbox. My goal is to make that sandbox as open to all who are interested in playing with us as possible. To that end, I want to make you aware of a couple of the things going on in ICS right now that I hope you will be as enthusiastic about as I am.

- Starting in 2015, **student memberships will be free!** The board decided that since the society is on a good financial footing, this is something we can afford to do and that it is good for the long-term health of the society. Please let your students know that admission to the sandbox is now free!
- We have an excellent conference planned for January 11-13, 2015 in Richmond, Virginia. The organizing and program committees, lead by General Chair Paul Brooks and Program Chair Brian Borchers, have been doing a fantastic job organizing so far! The **pre-conference workshop will be focused on open source tools**, featuring the COIN-OR Optimization Suite and other related open source tools.
- As a new feature at the 2015 ICS Conference, the proceedings will be **published open access!** The proceedings papers will be available on-line for free and a print volume will also be available on demand. The 2011 proceedings will be available open access as well.

The desire to show off our coolest toys and to see what cool toys everyone else has is something we never outgrow. I recently gave a talk entitled "Accessible Analytics" in which I discussed the general theme of "openness" in operations research. It occurred to me that ICS really does a lot enhance the accessibility of the technologies our members develop and that is one of the many things I value about this group. For those who are new, welcome to our sandbox! Play nicely and I look forward to raising a glass with you in San Francisco.

Cheers,
Ted

The 14th INFORMS Computing Society Conference

J. Paul Brooks
Virginia Commonwealth University
Richmond, VA 23284

The INFORMS Computing Society (ICS) is soliciting papers and presentations for its fourteenth conference. ICS is focused on contributions at the interface of computer science, artificial intelligence, operations research, and management science. The conference organizers invite submissions discussing novel theoretical and applied research consistent with the ICS focus. We especially encourage submissions targeting this year's conference theme: Operations Research and Computing: Algorithms and Software for Analytics.

Location: Omni Richmond Hotel in Richmond, Virginia, USA
Dates: January 11-13, 2015

Topics of Interest Include: Computational Optimization and Solvers, Constraint Programming and Hybrid Optimization, Computational Probability and Analysis, Data Mining, Simulation, Modeling Systems and Languages, Heuristic Search, Open Source Software, Computational Stochastic Optimization, Integer Programming, Network Applications, and Mixed Integer Nonlinear Programming.

Organizing Committee:
Brian Borchers, New Mexico Tech, Program Chair
Paul Brooks, Virginia Commonwealth University, General Chair
Xi Chen, Virginia Commonwealth University, Student Poster Session Chair
Jose Dula, Virginia Commonwealth University
Craig Larson, Virginia Commonwealth University, Plenary Chair
Laura McLay, University of Wisconsin-Madison
Yongjia Song, Virginia Commonwealth University

Presentation and Student Poster Session Abstract Submissions
An abstract of no more than 200 words should be submitted through the submission page at https://www.easychair.org/conferences/?conf=ics2015
Names and affiliations of all authors should be provided, with the presenting author listed first. The deadline for submission of presentation and poster abstracts is October 20, 2014.

NSF Student Poster Session Travel Awards
Thanks to the support of the National Science Foundation, support for registration and hotel is available for students who present posters at the conference. Students who wishto apply for travel support must submit an abstract to the ICS 2015 Student Poster Session Abstracts track on EasyChair by October 20, 2014, and indicate they wish to be considered for funding. Students from underrepresented groups are especially encouraged to apply. Space for posters may be limited, and the abstract will be used to decide which posters are accepted, so it is important that the abstract provide a good description of the research to be presented.

For additional information, contact the Conference General Chair Dr. J. Paul Brooks (jpbrooks@vcu.edu).

Message from the Editor

Yongpei Guan
Industrial and Systems Engineering
University of Florida
guan@ise.ufl.edu

It is the time to share the news for the society again and it is my pleasure to put things together. In this letter, please be aware of the updates of the board of directors, ICS 2015 call for papers, the instructions for making a video for IJOC, and the highlights and insights for the 2013 ICS awarding papers (special thanks to all contributors).

Report from the Editor of the INFORMS Journal on Computing

David Woodruff
University of California, Davis
joc@mail.informs.org

If you have a paper published at IJOC, you should make a video. Consider making two videos. I was skeptical at first, too, but the value is now pretty clear.

The journal is encouraging authors to create a five minute video intended for a technical audience and/or a two minute video intended for the general public. We have an example of each type in the online version of the journal. If you go to the page for these papers, you will find a link to the videos just below the abstract (there are also links in the supplements tab).

An example of the five minute technical video is provided by “Complexity and Approximation Results for the Balance Optimization Subset Selection Model for Causal Inference in Observational Studies” and the link is shown as follows: http://pubsonline.informs.org/doi/abs/10.1287/ijoc.2013.0583.

A two minute video is available for “Scalable Heuristics for a Class of Chance-Constrained Stochastic Programs” and the link is shown as follows: http://pubsonline.informs.org/doi/abs/10.1287/ijoc.1090.0372.

The five minute video provides an overview with enough detail to help a reader determine if the paper is relevant for their work and, perhaps more important, it is like seeing the movie before reading the book. Although it ruins the suspense, it is a lot easier to follow what is going on. The two minute video is intended to help provosts, deans, students, and even family members have some idea what is being done and why it has some importance. Both types of videos have the potential to be used by funding agencies as well as for fund-raising.

I have to admit that I was, and am, self-conscious about my somewhat overproduced video. Also, in my effort to make it understandable by the general public, I can now see that I was technically imprecise in some places. On the other hand, my Associate Dean thought it was great. I have found it useful for a number of purposes, and I think you would as well if you made one. Your video does not need to be professionally produced. Almost all universities and companies offer some support for simple, short videos. So, make a video (or make two) and we will attach it to the online version of your paper at IJOC.

2013 ICS Prize Goes to John Gunnar Carlsson from University of Minnesota

The 2013 INFORMS ICS Prize was awarded to John Gunnar Carlsson for his papers

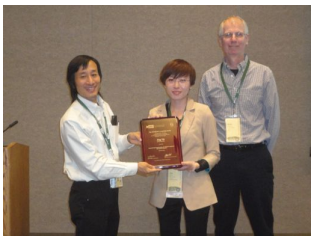
- John Gunnar Carlsson, “Dividing a territory among several vehicles", INFORMS Journal on Computing, Vol. 24, No. 4, Fall 2012, pp. 565Ű577.
- John Gunnar Carlsson and Erick Delage, “Robust partitioning for stochastic multivehicle routing", Operations Research (published online May 24, 2013);
- John Gunnar Carlsson and Raghuveer Devulapalli, “Dividing a territory among several facilities", INFORMS Journal on Computing (published online December 20, 2012).

The awarded set of papers comprise an elegant analysis of service problems in the plane, in the asymptotic limit of large numbers of customers whose location is stochastically distributed under fairly broad assumptions. The goal in one version is to identify a partition of the space such that the stochastic workload in each partition is asymptotically equal. In another version the partitioning will be into regions to be served by facilities, so as to minimize the maximum workload of a facility. In the third version, a robust partitioning problem is considered, where the customer distribution is not completely determined (an ambiguous distribution setting).

To solve these problems the research incorporates novel and elegant analytical and algorithmic tools in order to simultaneously reason about the shape of the partitions and the stochastic work-load allocation objectives. The papers combine rigorous mathematical analysis, detailed empirical/algorithmic work, and an exceptionally clear exposition. In blending fundamental, modern methodology as well as practical considerations, the work should stimulate continued interest in the problems and methodologies investigated.

Readers are referred to the article "Highlights: Segmenting a Geographic Region Optimally" in this newsletter for further highlights and description.

2013 ICS Prize Committee: Chris Beck (University of Toronto), Daniel Bienstock (Columbia University), and Dorit Hochbaum, Chair (University of California, Berkeley)



2013 ICS Best Student Paper Award Goes to Jing Xie at Cornell University

The 2013 Student Paper Award Winner is Jing Xie, Cornell University (advisor Peter Frazier), for the paper, "Sequential Bayes-Optimal Policies for Multiple Comparisons with a Known Standard."

This paper considers the statistical ranking & selection problem of multiple comparisons with a standard in the stochastic simulation setting. Specifically, given a set of alternatives with unknown mean performances, the goal is to find the optimal sequential allocation of simulation replications for determining which of the alternatives' mean performances exceeds a given performance threshold. Under a Bayesian dynamic programming formulation and using techniques from optimal stopping and multi-armed bandit problems, this paper is able to explicitly and efficiently compute the sequential Bayes-optimal for a very general class of sampling distributions: the well-known exponential family, which includes the most common continuous and discrete distributions such as normal, gamma, Poisson, geometric, and binomial. Computational experiments comparing the policy with other sampling policies in the literature demonstrate the effectiveness of the implemented sequential algorithm. Overall, the paper is well written and makes important contributions to both the theory and practice of simulation optimization by using a rigorous modeling framework that leads to useful implementable algorithms.

Readers are referred to the article "Highlights: Sequential Bayes-Optimal Policies for Multiple Comparisons with a Known Standard" in this newsletter for further highlights and description.

2013 ICS Best Student Paper Runner-up Goes to Rodrigo Carrasco at Columbia University

The 2012 ICS Best Student Paper Runner-up is Rodrigo Carrasco from Columbia University, for the paper "Resource Cost Aware Scheduling." His advisors are Garud Iyengar and Cliff Stein.

Readers are referred to the article "Highlights: Resource Cost Aware Scheduling" in this newsletter for further highlights and description.

2013 ICS Student Paper Award Committee: Laurent Michel (University of Connecticut), Cindy Phillips (Sandia), and Michael Fu, Chair (University of Maryland).

ICS Members in the News

Andrew Mason(a.mason@auckland.ac.nz), Ph.D., Associate Professor, Dept of Engineering Science, University of Auckland. Open Solver Updates: Many of you will have heard of, or used, OpenSolver, the Open Source optimizer for Excel. OpenSolver has, until now, supported solving linear and integer programmes using the COIN-OR CBC solver. We are pleased to announce that the latest version of OpenSolver now works with Gurobi, and also has non-linear capabilities thanks to the newly-added NOMAD engine. This experimental release can also translate a spreadsheet model into AMPL and solve it using one of the linear or non-linear COIN-OR solvers on NEOS. We'd welcome beta testers willing to try these new features. For more details and downloads, please see http://opensolver.org.

Erick Moreno-Centeno (e.moreno@tamu.edu), Ph.D., Assistant Professor at Texas A&M University, was awarded the 2014 Annual Award for Excellence in the Teaching of Operations Research (http://www.iienet2.org/details.aspx?id=882). This award is given by the Operations Research Division of IIE.

Manfred Wilhelm Padberg (born 10 October 1941 in Bottrop, Germany) died on May 12, 2014 after a long battle with cancer. His research centered on the study of linear and combinatorial optimization, with an emphasis on developing polyhedral theory that could aid in the solution of large, real-world optimization problems.

Manfred Padberg grew up in Zagreb, Croatia and Westphalia, Germany. He began his studies in 1961 at the Westphalian Wilhelms University in Münster, where he received his Diplom in mathematics in 1967. He then spent a year as an

Assistant Professor at the University of Mannheim. In 1968, he moved to the US under a Ford Foundation Fellowship to study operations research and industrial engineering at Carnegie Mellon University, where he received both his masters' degree and doctorate (1971) under the direction of Egon Balas. From 1971 to 1974, he was a research fellow at the International Institute of Management, Berlin, Germany. In 1974, he returned to the US as Associate Professor in operations research at Stern School of Business, New York University, becoming a Full Professor in 1978. He remained at NYU until becoming Professor Emeritus and moving to Paris in 2002. During his career in operations research, he was a guest scientist and/or visiting professor at the University of Bonn, at the IBM Thomas J. Watson Research Center in Yorktown Heights, INRIA in Rocquencourt, at the Ecole Polytechnique in Paris, the National Institute of Standards (NIST) in Maryland, the European Institute of Advanced Studies in Management (EIASM) in Brussels, the Center for Operations and Economics (CORE) in Louvain la Neuve, the Institute for Systems Analysis and Informatics (IASI) in Rome, and the State University of New York at Stony Brook. His fluency in Italian, French, English and German allowed him to lecture throughout the world. He spent his retirement in Paris and Marseille.

Over his lifetime, Manfred received all of the most significant prizes in the field of operations research, including:

- 1983: The Lanchester Prize of the Operation Research Society of America (ORSA).
- 1985: The George B. Dantzig Prize of the Mathematical Programming Society and the Society of Industrial and Applied Mathematicians (SIAM).
- 1989: The Alexander von Humboldt Senior US Scientist Research Award (Germany).
- 2000: The John von Neumann Theory Prize (INFORMS).
- 2002: INFORMS Fellow.

Manfred is best known for his work on applying polyhedral cuts to difficult optimization problems, often called "branch-and-cut" (a term he coined). He was most interested in graph-related problems and the algorithmic design of solution methodologies for such problems. He was always motivated by real-world applications and worked to solve previously unsolvable problems. During the latter part of his research, he worked on ideal matrices and almost-perfect graphs.

To best summarize his work, we quote the citation associated with his receiving the John von Neumann Theory Prize: *Since receiving his Ph.D. from Carnegie Mellon University in 1971, Manfred Padberg has made fundamental contributions to both the theoretical and computational side of integer programming and combinatorial optimization. His early work on facets of the vertex packing polytope and their liftings, and on vertex adjacency on the set partitioning polytopes, paved the way toward the wider us of polyhedral methods in solving integer programs. His characterization of perfect 0/1 matrices reinforced the already existing ties between graph theory and 0-1/programming. Padberg is the originator and main architect of the approach known as branch-and-cut. Concentrating on the traveling salesman problem as the main testbed, Padberg and Rinaldi successfully demonstrated that if cutting planes generated at various nodes of a search tree can be lifted so as to be valid everywhere, then interspersing them with branch and bound yields a procedure that vastly amplifies the power of either branch and bound or cutting planes themselves. This work had and continues to have a lasting influence. One of the basic discoveries of the 1980's in the realm of combinatorial optimization arrived at by three different groups of researchers in the wake of the advent of the ellipsoid method for convex programming, was the equivalence of optimization and separation: Padberg and M.R. Rao formed on these groups. Padberg's work combines theory with algorithm development and computational testing in the best tradition of Operations Research and the Management Sciences. In his joint work with Crowder and Johnson, as well as in subsequent work with others, Padberg set an example of how to formulate and handle efficiently very large scale practical 0/1 programs with important applications to industry and transportation.*

For more on his work, one should refer to the following texts (or to the more than 110 papers that he published):

- Martin Grötschel (editor) The Sharpest Cut: The impact of Manfred Padberg and his work, SIAM, 2004.
- Manfred Padberg and Dimitris Alevras Linear optimization and extensions, Springer, 3rd edition, 2001.
- Manfred Padberg and M. Rijal Location, Scheduling, design and integer Programming, Kluwer, 1996.

He is survived by his wife, Suzy Mouchet-Padberg, his brother Friedhelm Padberg, his sister Christa Padberg, his daughter Britta Padberg-Schmitt, his son Marc-Oliver Padberg, his stepson Hannibal Renberg and five grandchildren: Franziska, Franz-Josef, Mia, Maya and Mei.

Highlights: Segmenting a Geographic Region Optimally

John Gunnar Carlsson – University of Minnesota

I would like to take this opportunity to thank the 2013 ICS Prize Committee, which was chaired by Dorit Hochbaum and included Chris Beck and Daniel Bienstock. The prize was awarded for the three papers [14, 15, 13], which were co-authored with Erick Delage of HEC Montréal and Raghuveer Devulapalli of the University of Minnesota. I am overwhelmed by this honor and very grateful for this distinction.

The focus of these three papers is the problem of partitioning a geographic region into smaller sub-regions for allocating resources or distributing a workload among multiple agents. Dividing a territory into sub-regions is a natural problem that belongs to many different domains within the world of operations research, such as air traffic control, congressional districting, vehicle routing, facility location, urban planning, and supply chain management. Indeed, as exemplified in [12], effective division of geographic territory has been a fundamental societal problem since the times of antiquity:

> Homer, in describing the Phaiakian settlement in Scheria, speaks of a circuit wall for the city.... Implicit in the foundation of new colonies was the notion of equality among the members, exemplified in the division of their prime resource, the land. To achieve this, accurate measurement and equitable division were from the outset essential, even when gods or privileged men were to be honored with larger or better assignments.

Scientifically speaking, one of the major difficulties that problems of this type pose is their intrinsically interdisciplinary nature; in order to determine an optimal partition of a territory, one must combine tools from a variety of disciplines, such as mathematical optimization, computational geometry, geometric probability theory, and geospatial analysis. More concretely, complications arise when we are forced to reconcile the usual *allocation objectives* (such as minimizing workloads within sub-regions or ensuring equitable provision of a resource distributed in the region) with *geometric shape constraints* (such as requirements that all sub-regions be contiguous or convex).

The current approach to problems of this type is to first discretize the region into pixels and then solve a large (combinatorial) integer program, e.g. one having a binary variable x_{ij} for each region i and pixel j. This approach suffers from several drawbacks: first, large-scale combinatorial programs are often computationally intractable for large problem instances. Second, it may be difficult to impose geometric shape conditions (requiring connected sub-regions, for example) within a combinatorial framework. Third, some problem instances possess certain properties that allow us to obtain a solution rapidly by exploiting their structure (which can be lost when discretizing the problem). A further advantage is that by exploiting a particular problem structure, we are often able to determine what attributes of the problem affect the outcome most significantly. For these reasons, our approach is to instead formulate the problems geometrically and then use a (fast) geometric (instead of slow combinatorial) algorithm to solve them.

The three awarded papers are all devoted to the same family of problems, but the tools used therein are all considerably different from one another: the paper [13] uses divide-and-conquer recursion, the paper [14] uses distributionally robust optimization, and [15] uses infinite-dimensional linear programming. The following sections briefly describe these three solutions.

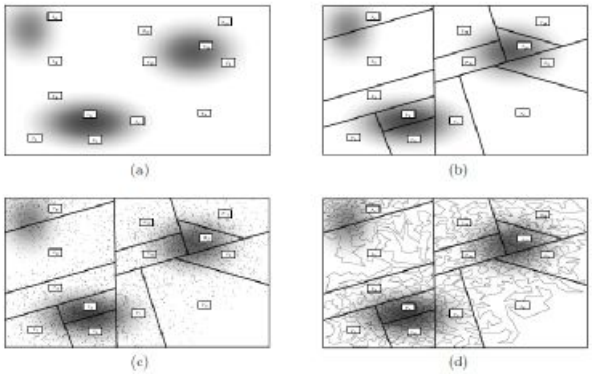


Figure 1: We begin with a set of $n = 13$ vehicle "depots" p_i with fixed locations and a probability density $f(\cdot)$ defined on the rectangular service region R (1a), which we then partition into n pieces (1b). This partition should be constructed so that, when a large collection of points is sampled independently from $f(\cdot)$ (1c), the n TSP tours of all the points in each sub-region plus the depot point are balanced (1d).

1 "Dividing a territory among several vehicles", *INFORMS Journal on Computing 24.4* (2012)

This paper is concerned with the problem of dividing a geographic region into pieces in order to distribute the workloads of a fleet of vehicles that originate at a collection of depots. Specifically, we are given a *simply connected* polygonal region R (i.e. a connected region with no holes) that contains a collection of n depot points $P = \{p_1, \ldots, p_n\}$, representing the starting locations of a fleet of vehicles. The vehicles must visit clients whose exact locations are unknown, but are assumed to be independent and identically distributed (i.i.d.) samples from a known probability density $f(\cdot)$. Our goal is to partition R into n disjoint sub-regions, with one vehicle assigned to each sub-region, so that the workloads in all sub-regions are asymptotically equal when a large number of samples is drawn. For each sub-region R_i, we will solve a travelling salesman problem in which the point set consists of a depot point plus all points in R_i. See Figure 1.

One of the main difficulties in this division problem lies in estimating the workload in a sub-region. Specifically, if N points are sampled independently from $f(\cdot)$, we let $\mathrm{TSP}(R_i; N)$ denote the length of a TSP tour of the sampled points that lie in R_i. Applying a standard coupling argument to a well-known result of geometric probability, the *BHH Theorem* [10], we can show that that $\mathrm{TSP}(R_i; N)$ obeys a law of large numbers. Specifically, with probability one, it turns out that as $N \to \infty$,

$$\frac{\mathrm{TSP}(R_i; N)}{\sqrt{N}} \to \beta \iint_{R_i} \sqrt{f(x)}\, dA \qquad (1)$$

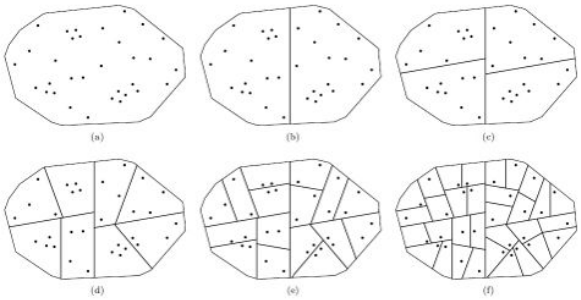


Figure 2: Recursive partitioning for the case where $f(\cdot)$ is the uniform distribution (so that we want all sub-regions to have equal area) and there are $n = 32$ depot points.

where β is a universal constant known to satisfy $0.6250 \leq \beta \leq 0.9204$ and "dA" denotes the usual area integral. This tells us that the quantity $(\beta \iint_{R_i} \sqrt{f(x)}\, dA)\sqrt{N}$ estimates $\mathrm{TSP}(R_i; N)$ within a term of $o(\sqrt{N})$.

Based on the preceding paragraph, it is clear that we desire a partition of R such that $\iint_{R_i} \sqrt{f(x)}\, dA$ is equal for all $i \in \{1, \ldots, n\}$ (this guarantees that all workloads are balanced within $o(\sqrt{N})$ as $N \to \infty$). This is very easy to achieve, in the absence of other criteria; for example, a partition might consist exclusively of vertical lines, with each vertical strip cutting off $\iint_{\text{strip}} \sqrt{f(x)}\, dA = 1/n \iint_R \sqrt{f(x)}\, dA$. For this reason, we impose additional constraints on our algorithm that should, in principle, give a better solution. A natural constraint to impose is that each sub-region R_i should contain the depot point that we have assigned to it. This still leaves us with considerable freedom because we have not yet imposed any constraints on the sub-regions. For example, one would expect that sub-regions ought to be connected. A further property that might be desired is that for any two points $u, v \in R_i$, the shortest path between u and v be contained in R_i. When the input region is convex, this constraint is equivalent to requiring that each sub-region R_i also be convex. When R is not convex, this property is called *relative convexity*: each sub-region R_i must be convex "relative" to the input region R. Thus, we seek a partition of R into pieces that satisfies three constraints:

- The asymptotic workloads $\iint_{R_i} \sqrt{f(x)}\, dA$ must be equal for all i,
- Each sub-region R_i contains exactly one depot point p_i, and
- All sub-regions must be relatively convex.

We find this partition using a recursive algorithm that divides R into successively smaller pieces, as shown in Figure 2. This algorithm is based on the famous topological *ham sandwich theorem*, which is now more than 75 years old [11].

2 "Robust partitioning for stochastic multivehicle routing", *Operations Research* 61.3 (2013)

As in the preceding section, this paper is concerned with the problem of dividing a geographic region into pieces in order to distribute the workloads of a fleet of vehicles that originate at a collection of depots. However, we now have an ambiguous distribution setting because the demand density function $f(\cdot)$ is not known; rather, we only have access to first and second moment information in the form of a center of mass $\mu \in R$ and a covariance matrix $\Sigma \succ \mathbf{0}$. Our objective is to find a partition $R_1, \ldots, R_n$ such that the *worst-case workload* (taken over all possible distributions with the given first and second moments) is as small as possible. The worst-case workload in a particular region, R_i, can be computed using the following infinite-dimensional optimization problem

$$\begin{aligned} \underset{f(\cdot)\geq 0}{\text{maximize}} \iint_{R_i} \sqrt{f(x)}\, dA \qquad & s.t. \\ \iint_R f(x)\, dA &= 1 \\ \iint_R x f(x)\, dA &= \mu \\ \iint_R x x^T f(x)\, dA &\leq \Sigma + \mu\mu^T . \end{aligned}$$

By using Lagrangian duality, it can be shown that the distribution $f^*(\cdot)$ defined on R that maximizes the workload for sub-region R_i must take the form

$$f^*(x) = \frac{1}{4\left(r + \mathbf{x}^T\mathbf{q} + \mathbf{x}^T\mathbf{Q}\mathbf{x}\right)^2}\mathcal{I}(\mathbf{x} \in R_i) + s\delta(\mathbf{x} - \bar{\mathbf{x}}) ,$$

where $\mathcal{I}(\cdot)$ denotes the indicator function and $\delta(\cdot)$ denotes the Dirac delta function. By combining this result with a Sperner's lemma-type argument, we give a fast algorithm for finding a "robust partition" that makes better use of first and second moment data as they become available.

3 "Dividing a territory among several facilities", *INFORMS Journal on Computing* 25.4 (2012)

This paper considers the problem of dividing a geographic region into pieces in order to distribute the workloads of a collection of *facilities* located within that region. Specifically, given a geographic region R, a probability density $f(\cdot)$ defined on R, and a collection of facilities $p_1, \ldots, p_n$ contained in R, we measure the workload of facility p_i in servicing sub-region R_i to take the form $\iint_{R_i} \alpha_i \|x - p_i\|^k\, dA$ for given α_i and k, i.e., proportional to the integral of a monomial function of the distance

between a point in R_i and its associated facility p_i. In order to balance the workloads of these facilities, our partitioning problem can be expressed as

$$\begin{aligned} \underset{R_1,\ldots,R_n}{\text{minimize}} \max_i \left\{ \iint_{R_i} \alpha_i \|x - p_i\|^k f(x)\, dA \right\} \qquad & s.t. \\ \bigcup_i R_i &= R \\ R_i \cap R_j &= \emptyset \quad \forall i \neq j\,. \end{aligned}$$

We can convert the above formulation into an infinite-dimensional integer program which happens to have an integrality gap of unity, and whose dual program is the finite-dimensional problem

$$\begin{aligned} \underset{\lambda}{\text{maximize}} \iint_R f(x) \min_i \{\lambda_i \alpha_i \|x - p_i\|^k\}\, dA \qquad & s.t. \\ \sum_{i=1}^{n} \lambda_i &= 1 \\ \lambda_i &\geq 0 \quad \forall i\,. \end{aligned}$$

By studying the complementary slackness conditions of these primal-dual pairs, we conclude that the optimal solution to our original problem consists of sub-regions whose boundary components are those curves that satisfy

$$\frac{\|x - p_i\|}{\|x - p_j\|} = \text{constant}\,,$$

which are actually *circular arcs*, or more precisely, arcs belonging to *circles of Apollonius*. We can also consider the related problem of minimizing the *aggregate workload* subject to mass constraints, given by

$$\begin{aligned} \underset{R_1,\ldots,R_n}{\text{minimize}} \sum_{i=1}^{n} \iint_{R_i} \|x - p_i\|\, f(x)\, dA \qquad & s.t. \\ \iint_{R_i} f(x)\, dA &= 1/n \quad \forall i \\ \bigcup_i R_i &= R \\ R_i \cap R_j &= \emptyset \quad \forall i \neq j\,, \end{aligned}$$

which we conclude (based on a similar analysis) must have an optimal solution whose boundary curves are *hyperbolic* arcs.

Highlights: Sequential Bayes-Optimal Policies for Multiple Comparisons with a Known Standard

Jing Xie and Peter I. Frazier–Cornell University

Multiple Comparisons with a Known Standard (MCS) is a fundamental problem from simulation in which we allocate simulation effort across a number of simulated systems, so as to best determine whether or not each system's true expected performance exceeds a known threshold. It arises most frequently when determining which proposed systems meet a performance requirement, e.g., a service-level agreement for a call center, or a mandated maximum risk probability in an air traffic control system. It also arises when checking feasibility as part of a larger optimization via simulation problem, and outside of simulation, when crowdsourcing classification tasks to human workers on sites like Amazon's Mechanical Turk.

In the MCS problem, if our ability to simulate is unlimited, then we can simulate each system a very large number of times, get very accurate estimates of system performance, and classify systems as above-threshold or below-threshold with high accuracy. However, if simulations are time-consuming, as they often are when simulating complex systems, we cannot afford to do this for all systems, and we must instead choose intelligently the number of simulation samples to take from each system.

To help us make such intelligent choices, we can behave adaptively, using the first few samples to get a rough idea of each system's performance, and then adjusting later sampling effort: allocating fewer samples to systems whose performance is far from the threshold and thus easy to classify as above or below; more samples to hard-to-classify systems close to the threshold; and in some cases abandoning with no additional samples those impossible-to-identify systems with performance extremely close to the threshold.

The contribution of [1] is a Bayes-optimal strategy for making these adaptive sampling decisions. [1] formalizes the MCS problem as a Bayesian sequential decision-making problem, by placing a Bayesian prior distribution over the true performance of each system, and then seeking to do well with respect to the average-case performance under this probability distribution. Sampling is constrained either by a price paid for each sample taken (appropriate when using cloud computing services, which charge per CPU hour), or by a simulation budget, which is assumed to be random and geometrically distributed for tractability.

As a sequential decision-making problem, the optimal strategy is characterized as the solution to a stochastic dynamic program, but solving this dynamic program directly is impossible for even moderately-sized problems, because the curse of dimensionality causes the state space to grow exponentially in the number of systems. The key insight of [1] is to show that this dynamic program can be decomposed across systems, and the optimal strategy then computed by solving a number of much smaller dynamic programs. The computation for this new method grows linearly in the number of systems, rather than exponentially, allowing it to be used in practice. This decomposition technique also links the MCS problem to multi-armed

bandits, as the decomposition used in the random geometric horizon setting is the same as the one used in the multi-armed bandit problem.

[1] Jing Xie, Peter I. Frazier, "Sequential Bayes-Optimal Policies for Multiple Comparisons with a Known Standard," Operations Research, vol. 61, no. 5, pp 1174Ű1189, 2013.

Highlights: Resource Cost Aware Scheduling

Carrasco, Iyengar, and Stein–Columbia University and Universidad Adolfo Ibáñez

1 Introduction

Managing non-renewable resource consumption is fast emerging as a problem of critical importance. There is always a trade-off between resource consumption and performance: more resource consumption typically results in better performance. This trade-off also arises in many scheduling problems, where resource management decisions must be combined with the scheduling decisions to optimize a global objective.

Recently, scheduling problems in which one has to balance scheduling performance (using metrics such as completion time, tardiness, or flow time) with CPU speed, and therefore the energy consumed, have been extensively studied. However, the problem of balancing resource consumption with scheduling performance was proposed much earlier. Vickson [32] observed that in many practical settings, the processing time of a job depends on the amount of resources (e.g. catalizer, workforce size, energy, etc.) utilized, and the relationship between resource utilization and processing time depends on each job's characteristics. Other examples of scheduling problems with resource dependent job processing time include repair and maintenance processes [19]; ingot preheating processes in steel mills [24, 33]; many workforce intensive operations; VLSI circuit design [25]; and more recently processing tasks in a CPU, where the job processing times depends on CPU speed, the available RAM, bus speed, as well as other system resources.

The literature on resource dependent job processing time problems has mainly focused on two models. In the first model the processing time p_i of job i as function of resource consumption level u_i is piece-wise linear function of the form $p_i(u_i) = \min\{\underline{p}_i, b_i - a_i u_i\}$, where a_i, b_i are job parameters and $\underline{p}_i$ is the smallest possible processing time. More recently, the processing time as a function of the resource consumption level is assumed to be of the form $p_i(u_i) = (\rho_i/u_i)^k$ for some $\rho > 0$ and $k > 0$. A survey the many different approaches to these problems can be found in [30].

Energy aware scheduling (EAS) of computing tasks is an important example of resource aware scheduling problems, and has received much attention recently. CPUs account for 50-60% of a typical computer's energy consumption [1]; consequently, CPU energy management is especially important for laptops and other mobile devices. It is clear that when scheduling computing tasks, it is important to take both the relevant scheduling quality of service (QoS) metrics such as makespan, weighted completion time or weighted flow time, and the energy consumption into account. Modern CPUs can run at multiple speeds; the lower the speed, the less energy used, and the relationship is device-dependent but typically superlinear. Thus, the energy consumed can be controlled by *speed scaling*.

In the EAS literature the power P consumed by the CPU is a polynomial function of speed s of the form $P(s) = s^\beta$ for some constant $\beta \in [2, 3]$. Recent work uses a more general power function with minimum regularity conditions, like non-negativity, but in all the cases the power function is *not* job-dependent since the jobs are homogeneous [4, 6]. Our approach allows job-dependent power functions, and thus can be applied to a more general class of problems outside the specific setting where only speed is the controllable resource. Furthermore, most energy aware algorithms assume cost functions that are closely related to energy consumption; however in practice, the actual energy cost is not simply a function of energy consumption, it is a complicated function of discounts, pricing, time of consumption, storage costs (in the case of mobile devices), etc. That observation motivated our consideration of a more general class of cost functions that are only restricted to be non-negative. We are not aware of any other work that allows such general costs.

There are three main settings for energy aware scheduling problems: optimizing a QoS metric with an energy budget [28, 29], minimizing energy subject to a QoS constraint [5, 7, 8, 34], or optimizing some combination of a scheduling objective and energy consumption [2, 4, 6, 9]. Our work is in the third setting. Implicit in the last criterion is the assumption that both the resource used and time can be (implicitly) converted into a common unit, such as dollars. The prior work on speed scaling algorithms assumes that the energy cost is *only* a function of the speed. We allow for the cost to be dependent on *all* the resources being utilized. For example, in the context of scheduling computational tasks, we can allow for the cost to depend on the CPU speed, the available RAM, and bus speed and size, among others.

In this paper we consider the commonly studied scheduling metric, *weighted completion time*. This metric has not received attention in the resource or energy cost aware scheduling literature, even though it has applications in several different areas such as software compilers, instruction scheduling in VLIW processors, MapReduce-like systems, manufacturing processes, and maintenance procedures among others [16, 17, 18, 27]. In all these applications there are related resources that can be used to control the speed at which jobs are processed, which should be taken into account. Furthermore, in many settings

jobs have precedence constraints as well, something that has not been dealt with in the current literature, and by allowing general precedence constraints we could extend our results to other metrics such as makespan.

Minimizing weighted completion time is well studied in the combinatorial scheduling literature. Phillips et al. [26] and Hall et al. [22, 23] introduced the concept of α-*points* that has lead to small constant factor approximation algorithms for many scheduling problems [31]. In the α-point approach, the scheduling problem is formulated as an integer program in terms of decision variable x_{it} that is 1 if job i completes at time t. The α-point of each job is defined as the earliest time at which an α fraction of the job has completed in the linear relaxation. The jobs are ordered in the order of their α-points and run in non-pre-emptive fashion. We extend the α-point technique by defining α-*speeds* that are achieved by time-sharing between resource operating points.

2 Results Highlighs

We make several contributions to the problem of scheduling with non-renewable resources:

• We introduce a model that extends the previous cost models (linear, convex, and other energy models) by allowing a more general relation between job processing time (or equivalent processing speed) and resource consumption.

• We further generalize the problem by allowing arbitrary precedence constraints and release dates.

• We give approximation algorithms for minimizing an objective that is a combination of a scheduling metric (weighted completion time) and resource consumption cost.

• We introduce the concept of α-speeds, which extend the α-points technique to problems with multiple speeds.

• We show that these algorithms have small constant approximation ratios and also demonstrate the effectiveness of the algorithms via experiential results, as well as test its performance with other metrics.

2.1 Cost Model

We consider a more general model of resource cost than has previously been used. Our setting captures both of the currently used models by considering an arbitrary non-negative speed function $\mathcal{S}(\mathbf{\Psi}^{(i)})$, where $\mathbf{\Psi}^{(i)} \in \mathbf{\Psi} = \{\mathbf{\Psi}^{(1)}, \ldots, \mathbf{\Psi}^{(q)}\}$ denotes one of the q allowable operating points of the resources. This is a typical situation in computers and server clusters where only a discrete number of configurations is available: bus speed, available RAM, CPU speed, etc. We also generalize the resource cost, which is generally linear in the literature, by considering an arbitrary non-negative job-dependent resource cost function $\mathcal{R}_i(\mathbf{\Psi}^{(i)})$, which gives the additional flexibility of allocating different levels of resources to different jobs.

2.2 Main Result

Our paper contains results for two related scheduling problems, we state here the most general result:

Theorem 2.1. *Given n jobs with precedence constraints and release dates and a general non-negative resource cost function, there is an $O(1)$-approximation algorithm for the problem of non-preemptively minimizing a weighted sum of the completion time and resource cost.*

The constants in the $O(1)$ are modest. Given some $\epsilon > 0$, the algorithm has a $(4 + \epsilon)$-approximation ratio when only precedence constraints exist, and $(3 + 2\sqrt{2} + \epsilon)$-approximation ratio when release dates are added.

2.3 Our Methodology

We extend the interval-indexed IP proposed by Hall et al. [23] to handle resource costs and speed scaling, and then design a new α-point based rounding algorithm to obtain the resulting schedules. In doing so we introduce the new concept of α-*speeds*. We assume that we have a discrete set of q allowable resource operating points $\mathbf{\Psi} = \{\mathbf{\Psi}^{(1)}, \ldots, \mathbf{\Psi}^{(q)}\}$, and that the speed at which the job is precessed is a general non-negative function of the resource operating point. In our interval-and-operating-point-indexed IP, a variable x_{ijt} is 1 if job i runs at resource operating point $\mathbf{\Psi}^{(j)}$ and completes in interval t. We can then extend the standard interval-indexed integer programming formulation to take the extra dimensions of resource consumption and speed into account Once we have solved its linear program relaxation (LPi), we need to determine *both an α-point and α-speed*. The key insight is that by "summarizing" each dimension appropriately, we are able to make the correct choice for the other dimension. At a high level, we first choose the α-point by "collapsing" all pieces of a job that completes in the LPi in interval t (these pieces have different speeds), being especially careful with the last interval, where we may have to choose only some of the speeds. We then use *only* the pieces of the job that complete before the α-point to choose the speed, where the speed is chosen by collapsing the time dimension and then interpreting the result as a probability mass function (pmf), where the probability that the job is run at speed $\mathcal{S}(\mathbf{\Psi}^{(j)})$ depends on the total amount of processing done at that operating point. We then define the concept of α-speeds, which is related to the expected value under this pmf, and run the job at this speed. We combine this new rounding method with extensions of the more traditional methods for dealing with precedence constraints and release dates to obtain our algorithm.

3 Problem Formulation

We are given a single machine that requires p different resources to run. The machine has q different resource operating points

$\boldsymbol{\Psi}^{(i)} \in \boldsymbol{\Psi} = \{\boldsymbol{\Psi}^{(1)}, \ldots, \boldsymbol{\Psi}^{(q)}\}$, where $\boldsymbol{\Psi}^{(i)} = \begin{bmatrix} \Psi_1^{(i)} & \ldots & \Psi_p^{(i)} \end{bmatrix}$ is described by a vector of p values, one for each resource. We are also given a function $\mathcal{S} : \mathbb{R}^p \to \mathbb{R}_+$ which maps each operating point $\boldsymbol{\Psi}^{(j)}$ to a speed $\sigma_j = \mathcal{S}(\boldsymbol{\Psi}^{(j)})$, and a function $\mathcal{R}_i(\boldsymbol{\psi}^{(i)})$, with $\mathcal{R} : \mathbb{R}^p \to \mathbb{R}_+$, which denotes the cost of running job i at the resource operating point $\boldsymbol{\psi}^{(i)}$. Additionally, we are given n jobs, where job i has a processing requirement of ρ_i machine cycles, a release time r_i, and an associated positive weight w_i. We may also be given precedence constraints among the jobs and we do not allow preemption.

A *schedule* defines, for each job, a time interval during which it runs, and for each time in that interval, a resource operating point from the allowable set. As in previous work, we can make some observations that simplify the structure of a schedule. By time sharing between different operating points the machine can run at any point within the convex hull of $\boldsymbol{\Psi}$. We thus extend the domain of the speed function and the cost function to include points ψ in the convex hull of $\boldsymbol{\Psi}$ in the natural way: for $\boldsymbol{\psi}^{(i)}$ such that $\boldsymbol{\psi}^{(i)} = \sum_{j=1}^{q} \lambda_j \boldsymbol{\Psi}^{(j)}$, with $\sum_j \lambda_j = 1$ and $\lambda_j \in [0, 1]$, then if $\psi = \sum_j \delta_j \Psi^{(j)}$ then $\mathcal{S}(\psi) = \sum_j \delta_j \mathcal{S}(\Psi^{(j)})$ and $\mathcal{R}_i(\boldsymbol{\psi}^{(i)}) = \sum_{j=1}^{q} \lambda_j \mathcal{R}_i(\boldsymbol{\Psi}^{(j)})$. By extending our domain in this way, we can assume that each job runs at one resource operating point, and one speed. We can further assume that a point with lower speed also has lower cost, for otherwise we could achieve that point by running at a higher speed and then idling, thereby achieving an even better cost. Throughout the paper, we will use capital Ψ to denote the input set of operating points and lowercase ψ to denote points in the convex hull.

We can define a schedule precisely as follows. Let $\boldsymbol{\psi}^{(i)}$ denote the operating point at which job i runs, thus $s_i = \mathcal{S}(\boldsymbol{\psi}^{(i)})$ denotes the speed at which job i runs in the machine, and $p_i = \frac{\rho_i}{\mathcal{S}(\boldsymbol{\psi}^{(i)})}$, its processing time. Let C_i denote the completion time of job i, and let $\Pi = \{\pi(1), \ldots, \pi(n)\}$ denote the order in which the jobs are processed, i.e. $\pi(k) = i$ implies that job i is the k-th job to be processed. Then $C_{\pi(i)} = \max\{r_{\pi(i)}, C_{\pi(i-1)}\} + \frac{\rho_{\pi(i)}}{s_{\pi(i)}}$ is the completion time of the i-th job to be processed, with $C_{\pi(0)} = 0$.

The objective is to compute a feasible schedule consisting of an order Π, possibly subject to precedence and/or release date constraints, and the vector of resource requirements $\boldsymbol{\psi} = \begin{bmatrix} \boldsymbol{\psi}^{(1)} & \ldots & \boldsymbol{\psi}^{(n)} \end{bmatrix}$ minimizes the total cost,

$$f(\Pi, \boldsymbol{\psi}) = \sum_{i=1}^{n} \left[\mathcal{R}_i(\boldsymbol{\psi}^{(i)}) + w_{\pi(i)} C_{\pi(i)} \right] \ . \tag{2}$$

For convenience we will use an extended version of the notation of Graham et al. [20] to refer to our different resource cost aware scheduling problems, i.e. $1|r_i, prec| \sum \mathcal{R}_i(\boldsymbol{\psi}^{(i)}) + w_i C_i$, will refer to the problem setting with 1 machine, with r_i release dates, precedence constraints, and the weighted completion time as the scheduling performance metric. We assume, w.l.o.g., that the resource operating points are ordered by speed (slowest first), and use $\sigma_i = \mathcal{S}(\boldsymbol{\Psi}^{(i)})$ to denote the i^{th} slowest speed.

To model this problem we modify and extend the interval-indexed formulation proposed by Hall et al. [23] to accommodate speeds and resource cost. The interval-indexed formulation divides the time horizon into geometrically increasing intervals, and the completion time of each job is assigned to one of these intervals. Since the completion times are not associated to a specific time, the completion times are not precisely known but are lower bounded. By controlling the growth of each interval one can obtain a sufficiently tight bound.

The problem formulation is as follows. We divide the time horizon into the following geometrically increasing intervals: $[\kappa, \kappa]$, $(\kappa, (1+\epsilon)\kappa]$, $((1+\epsilon)\kappa, (1+\epsilon)^2\kappa]$, ..., where $\epsilon > 0$ is an arbitrary small constant, and $\kappa = \frac{\rho_{\min}}{\sigma_{\max}}$ denotes the smallest interval size that will hold at least one whole job. We define interval $I_t = (\tau_{t-1}, \tau_t]$, with $\tau_0 = \kappa$ and $\tau_t = \kappa(1+\epsilon)^{t-1}$. The interval index ranges over $\{1, \ldots, T\}$, with $T = \min\{\lceil t \rceil : \kappa(1+\epsilon)^{t-1} \geq \max_{i=1}^{n} r_i + \sum_{i=1}^{n} \frac{\rho_i}{\sigma_1}\}$; and thus, we have a polynomial number of indices t.

Let x_{ijt} equals 1 if job i runs at o.p. $\boldsymbol{\Psi}^{(j)}$ and completes in time interval I_t, and 0 otherwise. By using the lower bounds τ_{t-1} of each time interval I_t, a lower bound to (2) is written as,

$$\min_{\mathbf{x}} \sum_{i=1}^{n} \sum_{j=1}^{q} \sum_{t=1}^{T} \left(\mathcal{R}_i(\boldsymbol{\Psi}^{(j)}) + w_i \tau_{t-1} \right) x_{ijt} \ . \tag{3}$$

The following are the constraints required:

$$\sum_{j=1}^{q} \sum_{t=1}^{T} x_{ijt} = 1, \forall i \ , \tag{4}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{q} \sum_{u=1}^{t} \frac{\rho_i}{\sigma_j} x_{iju} \leq \tau_t, \forall t \ , \tag{5}$$

$$x_{ijt} = 0, \ \text{ if } \tau_t < r_i + \frac{\rho_i}{\sigma_j}, \forall i, j, t \ , \tag{6}$$

$$x_{ijt} \in \{0, 1\}, \forall i, j, t \ , \tag{7}$$

$$\sum_{j=1}^{q} \sum_{u=1}^{t} x_{i_1 ju} \geq \sum_{j=1}^{q} \sum_{u=1}^{t} x_{i_2 ju}, \forall t, i_1 \prec i_2 \ . \tag{8}$$

It is important to note that this integer program only provides a lower bound for (2); in fact its optimal solution may not be schedulable.

4 Approximation Algorithm

We now describe our proposed approximation algorithm, called Schedule by α-intervals and α-speeds (SAIAS), detailed in Algorithm 1.

Let $\bar{x}_{ijt}$ denote the optimal solution of the linear relaxation of the integer program (3)-(8). In steps 1 and 2 of the algorithm, we divide the time into geometrically increasing intervals and compute the set of possible speeds $\mathbf{S}$. Next, in 3 we compute the optimal solution $\bar{\mathbf{x}}$ and in step 4, given

Algorithm 1 Schedule by α-intervals and α-speeds for Resource Costs (SAIAS)

Inputs: set of jobs, $\alpha \in (0,1)$, $\epsilon > 0$, set of resource operating points $\mathbf{\Psi}$, speed function S, and resource function R.
1 Divide time into increasing time intervals $I_t = (\tau_{t-1}, \tau_t]$, with $\tau_t = \kappa(1+\epsilon)^{t-1}$.
2 Compute the set of possible speeds $\mathbf{S} = \{\sigma_1, \ldots, \sigma_q\}$.
3 Compute an optimal solution $\bar{\mathbf{x}}$ to the linear relaxation (3)-(8).
4 Compute the α-intervals $\mathbf{I}^\alpha$ and the sets J_t.
5 Compute an order Π^α that has sets J_t ordered in non-decreasing values of t and the jobs within each set in a manner consistent with the precedence constraints.
6 Compute the α-speeds $\mathbf{s}^\alpha$ via (10).
7 Set the i-th job to start at time $\max\{r_{\pi(i)}, C^\alpha_{\pi(i-1)}\}$, where $C^\alpha_{\pi(i-1)}$ is the completion time of the previous job using the rounded α-speeds, and $C^\alpha_{\pi(0)} = 0$.
8 **return** speeds $\mathbf{s}^\alpha$, order Π^α, and completion times $\bar{\mathbf{C}}^\alpha$.

$0 \le \alpha \le 1$, we compute the α-interval I_i^α of job i, defined as $I_i^\alpha = \min\left\{t : \sum_{j=1}^q \sum_{u=1}^t \bar{x}_{iju} \ge \alpha\right\}$.

Since several jobs may finish in the same interval, let J_t denote the set of jobs that finish in interval I_t, $J_t = \{i : I_i^\alpha = t\}$, and we use these sets to determine the order Π^α as described in step 5. Next, in step 6, we compute the α-speeds as follows. Since $\sum_{j=1}^q \sum_{u=1}^{I_i^\alpha} \bar{x}_{iju} \ge \alpha$, we define auxiliary variable $\{\tilde{x}_{ijt}\} = \bar{x}_{ijt}$ when $t < I_i^\alpha$, $\max\left\{\min\left\{\bar{x}_{ijI_i^\alpha}, \alpha - \sum_{l=1}^{j-1} \bar{x}_{ilI_i^\alpha} - \beta_i\right\}, 0\right\}$ when $t = I_i^\alpha$ and 0 otherwise, where $\beta_i = \sum_{j=1}^q \sum_{u=1}^{I_i^\alpha - 1} \bar{x}_{iju} < \alpha$. Note that for this auxiliary variable, we have that $\sum_{j=1}^q \sum_{u=1}^{I_i^\alpha} \tilde{x}_{iju} = \alpha$. This is a key step that allows us to truncate the fractional solution so that for every job i, the sum of $\tilde{x}_{ijt}$ up to time interval I_i^α for each speed j can be interpreted as a probability mass function. We define this probability mass function (pmf) $\boldsymbol{\mu}^i = (\mu_1^i, \ldots, \mu_q^i)$ on the set of speeds $\mathbf{S} = \{\sigma_1, \ldots, \sigma_q\}$ as

$$\mu_j^i = \frac{1}{\alpha} \sum_{u=1}^{I_i^\alpha} \tilde{x}_{iju} \quad . \tag{9}$$

Let $\hat{s}_i$ define a random variable distributed according to the pmf $\boldsymbol{\mu}^i$, i.e. $\mu_j^i = \mathbb{P}(\hat{s}_i = \sigma_j)$. Then, the α-speed of job i, s_i^α, is defined as follows:

$$\frac{1}{s_i^\alpha} = \mathbb{E}\left[\frac{1}{\hat{s}_i}\right] = \sum_{j=1}^q \frac{\mu_j^i}{\sigma_j} \quad . \tag{10}$$

We define the α-speeds using the reciprocal of the speeds since the completion times are proportional to the reciprocals. Note that (9) defines the fraction of the machine cycles requirement ρ_i that must be processed at each operating point $\mathbf{\Psi}^{(j)}$ to achieve the α-speed s_i^α.

Finally, in steps 7 and 8 we compute the completion times given the calculated speeds and return the set of speeds $\mathbf{s}^\alpha$, the order Π^α and the completion times $\mathbf{C}^\alpha$.

To analyze the performance guarantee of our algorithm, we first prove that the output of the SAIAS algorithm is indeed feasible, which is done by analyzing the constraints of the IP.

Lemma 4.1. *Suppose $i_1 \prec i_2$. Then (8) implies that $I_{i_1}^\alpha \le I_{i_2}^\alpha$.*

Since the SAIAS algorithm schedules jobs by first ordering the sets J_t in increasing order of t, and then orders the jobs within each set in a way that is consistent with the precedence constraints, Lemma 4.1 implies that the SAIAS algorithm preserves the precedence constraints, and, therefore, the output of the algorithm is feasible.

Finally, we can compute performance guarantees for our algorithm. The key ideas behind our proofs is that we can bound the resource costs using Jensen's Inquality, thanks to the convexity requirements in our functions, and the completion times can be bounded by using the α-speeds.

Theorem 4.2. *The SAIAS algorithm with $\alpha = \frac{1}{2}$ is a $(4+\epsilon)$-approximation algorithm for the $1|prec|\sum \mathcal{R}_i(\psi^{(i)}) + w_i C_i$ problem, with a general non-negative $\mathcal{R}_i(\psi)$ resource cost function.*

Release dates makes the problem somewhat harder since they can introduce idle times between jobs, but we can obtain the following result for that setting.

Theorem 4.3. *The SAIAS algorithm with $\alpha = \sqrt{2} - 1$ is a $(3 + 2\sqrt{2}+\epsilon)$-approximation algorithm for the $1|r_i, prec|\sum \mathcal{R}_i(\psi^{(i)}) + w_i C_i$ problem, with a general non-negative $\mathcal{R}_i(\psi)$ resource cost function.*

We also further study the problem when the resource is just the energy consumption, i.e. $\mathcal{R}_i(\psi^{(i)}) = v_i \rho_i s_i^{\beta-1}$, where $v_i > 0$ and $\beta > 2$ are constants. In this setting we can improve our solution by recalculating the optimal resource operating point once the order is defined by the SAIAS algorithm. The following result establishes that we can compute the optimal resource operating point for each job, for any given order.

Lemma 4.4. *Given the schedule order Π^α, the optimal speed at which to run job i is given by*

$$s_i^* = \sqrt[\beta]{\frac{\sum_{j=i}^n \sum_{k=1}^i \lambda_{jk}^*}{(\beta-1)v_i}}, \ \forall i \in \{1, \ldots, n\} \ , \tag{11}$$

where λ_{jk}^* is the optimal solution of the following optimization

Problem	Instances	Size (n)	Average Ratio	99.5%	Worst Ratio
Offline	20,000	7	1.055	1.231	1.420
	20,000	100	1.135[1]	1.218	1.273
	20,000	500	1.133[1]	1.157	1.184
	3,000	1,000	1.136[1]	1.150	1.155
Heuristic Improvement	20,000	7	0.991	1.000	1.000
	20,000	100	0.991	0.994	0.995
Online (no *prec*)	20,000	7	1.141	1.600	2.456
	20,000	100	1.397[1]	1.496	1.627

Table 1: Experimental Results Summary for Total Weighted Completion Time.

problem:

$$\begin{array}{lll} \max\limits_{\lambda} & \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{i}\lambda_{ij}r_j + \sum\limits_{i=1}^{n}\mathcal{B}\rho_i v_i^{\frac{1}{\beta}}\left(\sum\limits_{j=i}^{n}\sum\limits_{k=1}^{i}\lambda_{jk}\right)^{b} & , \quad (12) \\ s.t.: & \sum_{j=1}^{i}\lambda_{ij} = w_i, \quad \forall i, & \\ & \lambda_{ij} \geq 0, \quad \forall j \in \{1,\ldots,i\}, \forall i, & \end{array}$$

with $b = \frac{\beta-1}{\beta}$, *and* $\mathcal{B} \equiv \frac{\beta}{(\beta-1)^b}$.

This result follows from a careful analysis using the optimality conditions of the problem. Note from (11) that the optimal speed of the i-th job only depends of the dual variables of the completion time constraints of future jobs, and not past ones.

Corollary 4.5. *If* $r_i = 0$, $\forall i$, *then the optimal speed of job* i *is given by* $s_i^* = \sqrt[\beta]{\frac{\sum_{j=i}^{n} w_j}{(\beta-1)v_i}}$.

This result is an extension of the speed rule used in most of the energy aware scheduling literature for the flow time metric [3, 4]. Furthermore, using Lemma 4 and Corollary 4.5 one can design an algorithm that computes the optimal speeds for a given order Π in $O(n)$ time, when there are no release dates, and in $O(n^2)$ time, when there are release dates.

When no precedence constraints and release dates exist, there are two versions of this problem that can be optimally solved in polynomial time: when all weights w_i are equal and when all jobs have the same size and energy cost function:

Theorem 4.6. *If* $w_i = w$, $\forall i$ *or* $\rho_i v_i^{\frac{1}{\beta}} = \xi$, $\forall i$ *then the order* Π *is optimal if*

$$\frac{w_{\pi(i)}}{\rho_{\pi(i)} v_{\pi(i)}^{\frac{1}{\beta}}} \geq \frac{w_{\pi(i+1)}}{\rho_{\pi(i+1)} v_{\pi(i+1)}^{\frac{1}{\beta}}}, \ \forall i \in \{1,\ldots,n-1\} \ .$$

This theorem is an extension of Smith's Rule for the energy setting, and its proof is based on an interchange argument, making sure to account for the changes in energy consumption when jobs are interchanged.

5 Experimental Results

There are very few examples of performance analysis in the resource aware scheduling literature. A notable exception is [3]. We also present experimental results for other settings not covered in our theoretical results, such as online scheduling and when the total weighted flow time is used as the scheduling metric.

We restricted our simulations to the speed-scaling energy-aware scheduling case, that is, we have q different operating points, where $\Psi^{(j)} = \sigma_j$ and $\mathcal{S}(\sigma_j) = \sigma_j$. Furthermore, we use the standard polynomial relationship between speed and energy used in most of the literature as the resource cost function, i.e. $\mathcal{R}_i(s_i) = v_i\rho_i s_i^{\beta-1}$, where $v_i > 0$ is a job parameter, $\rho_i \in \mathbb{N}_+$ is the job size, and $\beta = 3$. For each analysis we simulated a large number of randomly generated instances, which are available in the author's webpage.

For small instances we compared the output of our algorithm with the integer solution of the interval-and-speed-indexed formulation (IPi), its linear relaxation (LPi), and the integer and relaxed solutions of a time-and-speed-indexed formulation for this problem (IPt and LPt respectively). Although we do not explicitly give these formulations, we use them in the experiments to help us understand whether the error comes from the rounding in the algorithm or the interval relaxation in the LP. All simulations were done in Matlab, using Gurobi [21] to solve the IP and LP relaxations of each instance.

Our experimental results show that the SAIAS algorithm, in practice performs very close to optimal, with average approximation ratios below 1.14. Furthermore we also show that these results remain similar even when the size of the instances grow several orders of magnitude. It is important to note that when

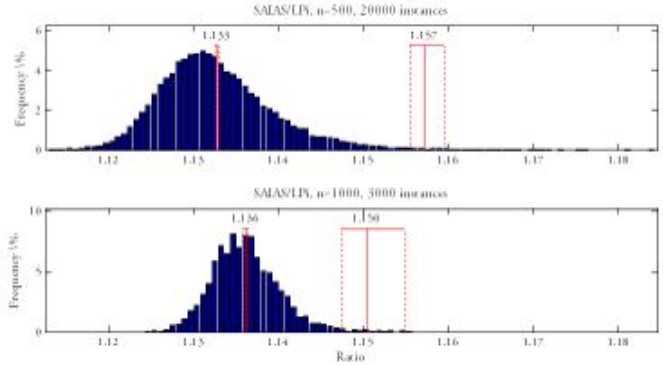


Figure 3: SAIAS/LPi Ratios with $n = 500$ and $n = 1,000$.

analysing large instances, since the IPt formulation is too large to be solved in a reasonable time, we compared the algorithm's output with the LPi solution, and thus the real approximation ratio is likely to be even better. The results also show that a modification to the algorithm, where we compute the optimal speeds given the order computed by the SAIAS algorithm, further reduces the approximation ratios. This improvement can also be used in the online setting. Table 1 shows a summary of all the results when the weighted completion time metric is used.

We also modified the SAIAS algorithm to handle the case when the total weighted flow time is used as scheduling metric, showing very good results as well, with an average approximation ratio of 2.59.

For each instance size we characterize the distribution of the approximation or competitive ratios via histograms. We believe that displaying the entire distribution is important since it gives a better understanding of how the algorithm performs. In the histogram we highlight the average value for all simulations and the 99.5% quantile. For both these measures we also display the 99.99% confidence intervals, shown as doted lines around the corresponding value.

As an example, Figure 3 shows the results for instances of $n = 500$ and $n = 1000$ jobs. Although we are comparing the algorithm's output to the LPi solution, the approximation ratio remains small. The average, worst, and 99.5% percentile can be found in Table 1. The fact that the approximation ratio is not much bigger is important since for $n = 7$ the ratio between the LPi and the IPt solution was in average 0.83, hence much of the error shown in Figure 3 could be attributed to the relaxation interval relaxation.

Acknowledgments

This research was partially supported by NSF grants CCF-0728733 and CCF-0915681; NSF grant DMS-1016571, ONR grant N000140310514, and DOE grants DEFG02-08ER25856 and DE-AR0000235; and Fulbright/Conicyt Chile.

References

[1] Albers, S. Algorithms for Energy Saving. In *Efficient Algorithms*, S. Albers, H. Alt, and S. Näher, Eds., vol. 5760 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 173–186.

[2] Albers, S., and Fujiwara, H. Energy-efficient algorithms for flow time minimization. *ACM Transactions on Algorithms 3*, 4 (Nov. 2007), 49–es.

[3] Andrew, L. L., Lin, M., and Wierman, A. Optimality, fairness, and robustness in speed scaling designs. *ACM Sigmetrics* (2010).

[4] Andrew, L. L., Wierman, A., and Tang, A. Optimal speed scaling under arbitrary power functions. *ACM SIGMETRICS Performance Evaluation Review 37*, 2 (Oct. 2009), 39.

[5] Bansal, N., Bunde, D., Chan, H. L., and Pruhs, K. R. Average rate speed scaling. In *Proceedings of the 8th Latin American conference on Theoretical informatics* (Dec. 2008), Springer-Verlag, pp. 240–251.

[6] Bansal, N., Chan, H. L., and Pruhs, K. R. Speed scaling with an arbitrary power function. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (2009), Society for Industrial and Applied Mathematics, pp. 693–701.

[7] Bansal, N., Kimbrel, T., and Pruhs, K. R. Dynamic speed scaling to manage energy and temperature. *Energy* (2004).

[8] Bansal, N., Kimbrel, T., and Pruhs, K. R. Speed scaling to manage energy and temperature. *Journal of the ACM (JACM) 54*, 1 (Mar. 2007), 3.

[9] Bansal, N., Pruhs, K. R., and Stein, C. Speed scaling for weighted flow time. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (2007), vol. pages, Society for Industrial and Applied Mathematics, p. 813.

[10] Beardwood, J., Halton, J., and Hammersley, J. The shortest path through many points. *Proceedings of the Cambridge Philosophical Society 55* (1959), 299–327.

[11] Beyer, W. A., and Zardecki, A. The early history of the ham sandwich theorem. *American Mathematical Monthly* (2004), 58–61.

[12] Boyd, T. D., and Jameson, M. H. Urban and rural land division in ancient greece. *Hesperia: The Journal of the American School of Classical Studies at Athens 50*, 4 (1981), pp. 327–342.

[13] Carlsson, J. Dividing a territory among several vehicles. *INFORMS Journal on Computing 24*, 4 (2012), 565 – 577.

[14] Carlsson, J. G., and Delage, E. Robust partitioning for stochastic multivehicle routing. *Operations Research 61*, 3 (2013), 727–744.

[15] Carlsson, J. G., and Devulapalli, R. Dividing a territory among several facilities. *INFORMS Journal on Computing 25*, 4 (2012), 730–742.

[16] Chang, H., Kodialam, M., Kompella, R. R., Lakshman, T. V., Lee, M., and Mukherjee, S. Scheduling in mapreduce-like systems for fast completion time. *2011 Proceedings IEEE INFOCOM* (Apr. 2011), 3074–3082.

[17] Chekuri, C., and Khanna, S. 11. Approximation Algorithms for Minimizing Average Weighted Completion Time. In *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. 2004, pp. 1–30.

[18] Chekuri, C., Motwani, R., Natarajan, B., and Stein, C. Approximation Techniques for Average Completion Time Scheduling. *SIAM Journal on Computing 31*, 1 (2001), 146.

[19] Duffuaa, S., Duffuaa, S. O., Raouf, A., and Campbell, J. D. *Planning and control of maintenance systems: modeling and analysis*. John Wiley & Sons Inc, 1999.

[20] Graham, R., Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. G. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Discrete optimization 5* (1979), 287–326.

[21] Gurobi Optimization, I. Gurobi Optimizer Reference Manual, 2012.

[22] Hall, L. A., Schulz, A. S., Shmoys, D. B., and Wein, J. Scheduling to Minimize Average Completion Time: Off-Line and On-Line Approximation Algorithms. *Mathematics of Operations Research 22*, 3 (Aug. 1997), 513–544.

[23] Hall, L. A., Shmoys, D. B., and Wein, J. Scheduling to minimize average completion time: off-line and on-line algorithms. In *Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms* (Philadelphia, PA, USA, Aug. 1996), SODA '96, Society for Industrial and Applied Mathematics, pp. 142–151.

[24] Janiak, A. Single machine scheduling problem with a common deadline and resource dependent release dates. *European Journal of Operational Research 53* (1991), 317–325.

[25] Monma, C. L., Schrijver, A., Todd, M. J., and Wei, V. K. Convex resource allocation problems on directed acyclic graphs: duality, complexity, special cases, and extensions. *Mathematics of Operations 15*, 4 (1990), 736–748.

[26] Phillips, C. A., Stein, C., and Wein, J. Minimizing average completion time in the presence of release dates. *Mathematical Programming 82*, 1-2 (June 1998), 199–223.

[27] Pinedo, M. *Scheduling: Theory, Algorithms, and Systems*, 3rd ed. Springer New York, New York, NY, 2008.

[28] Pruhs, K. R., Stee, R., and Uthaisombut, P. Speed Scaling of Tasks with Precedence Constraints. *Theory of Computing Systems 43*, 1 (Oct. 2007), 67–80.

[29] Pruhs, K. R., Uthaisombut, P., and Woeginger, G. J. Getting the best response for your erg. *ACM Transactions on Algorithms 4*, 3 (June 2008), 1–17.

[30] Shabtay, D., and Steiner, G. A survey of scheduling with controllable processing times. *Discrete Applied Mathematics 155*, 13 (Aug. 2007), 1643–1666.

[31] Skutella, M. List Scheduling in Order of α-Points on a Single Machine. In *Efficient Approximation and Online Algorithms*, E. Bampis, K. Jansen, and C. Kenyon, Eds., vol. 3484 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 250–291.

[32] Vickson, R. G. Choosing the job sequence and processing times to minimize total processing plus flow cost on a single machine. *Operations Research 28*, 5 (1980).

[33] Williams, T. J. *Analysis and design of hierarchical control systems: with special reference to steel plant operations*, vol. 3. Elsevier, 1985.

[34] Yao, F., Demers, A., and Shenker, S. A scheduling model for reduced CPU energy. In *Proceedings of IEEE 36th Annual Foundations of Computer Science* (1995), IEEE Comput. Soc. Press, pp. 374–382.

Acknowledgments

The Editor would like to thank all contributors who helped to make this newsletter available.