Final Report for AFOSR #F49620-03-1-0233
Advancing Air Force Scheduling through Modeling Problem Topologies
April 1, 2003 to May 30, 2006

Adele Howe   L. Darrell Whitley
Computer Science Department
Colorado State University
Fort Collins, CO 80524
email: {howe,whitley}@cs.colostate.edu

August 3, 2006

Abstract

Because of the difficulties of obtaining data from real applications, researchers tend to develop their new algorithms on artificial problems and do not model what makes an algorithm successful. Developers then have little guidance on which algorithms are best for which applications. All of this makes it difficult for research results to transfer to deployment. Our project endeavored to ameliorate this situation by 1) modeling the topology of scheduling algorithms utilizing real problems of interest to the Air Force, 2) developing and evaluating new search algorithms by exploiting the modeled topology of real applications and 3) disseminating the problems, algorithms and results via a publically available repository.

Project accomplishments include new models of scheduling algorithm performance, new insights about algorithm performance and new algorithms for specific scheduling applications. We have developed dynamic search space models of Job Shop scheduling that identify key factors in the success of Tabu Search and Simulated Annealing. For oversubscribed scheduling problems (applications in which the number of tasks overwhelms the available resources), we have developed models of the plateaus that are caused by a combination of discretization factors. We have also explained the performance of local search and a hybrid heuristic/search algorithm called Squeaky Wheel Optimization. We also have collected evidence concerning the benefits of searching through infeasible space on oversubscribed problems.

Our models have motivated the development of new algorithms for three different scheduling applications. Each algorithm is simpler than previous state of the art algorithms and include only the features shown through modeling to influence performance. We have worked with several companies to transfer what we have learned to next generation systems for deployed applications.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 03-08-2006	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* 04/01/03-05/31/06

4. TITLE AND SUBTITLE
Advancing Air Force Scheduling through Modeling Problem Topologies

5a. CONTRACT NUMBER

5b. GRANT NUMBER
F49620-03-1-0233

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Adele Howe and L. Darrell Whitley

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Colorado State University
Fort Collins, CO 80526

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Department of the Air Force
Air Force Office of Scientific Research (AFOSR)
875 N. Randolph St.
Arlington, VA
22203-1768

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-SR-AR-TR-06-0470

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for Public Release – Distribution A

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Our project bridges a gap between research on artificial problems and requirements of scheduling applications by 1) modeling the topology of scheduling algorithms utilizing real problems of interest to the Air Force, 2) developing and evaluating new search algorithms by exploiting the modeled topology of real applications and 3) disseminating the problems, algorithms and results via a publically available repository. Project accomplishments include new models of scheduling algorithm performance, new insights about algorithm performance and new algorithms for specific scheduling applications. Our models characterize search in JSP and several satellite scheduling applications. Each algorithm is simpler than previous state of the art algorithms and include only the features shown through modeling to influence performance. We have worked with several companies to transfer what we have learned to next generation systems for deployed applications.

15. SUBJECT TERMS
optimization, scheduling

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Contents

1 Project Objectives

Current practice largely separates research and development of optimization systems. Researchers create new algorithms and test them on simplified problems using synthetic data. These often are not suited for use on more complex real world problems. Developers in turn may implement simple heuristic algorithms or use off-the-shelf packages; such methods may or may not work well. Off-the-shelf packages tend to be black boxes. On the other hand, simple methods that are easy to understand may not yield the best results.

Clearly, we need reduce the gap between research and development for scheduling applications. Thus, the primary goal of the project was to inform the design of optimizing search algorithms for new scheduling applications using models of algorithms and search spaces.

We sought to achieve the goal with these objectives:

1. Collect algorithms and problem data for a small set of Air Force related applications,
2. Model search space topologies and algorithm performance on different applications,
3. Develop new algorithms based on models of problem topology, and
4. Widely disseminate algorithms, data and models of problem topology.

An implied goal is that the new algorithms should be easier to understand because they more directly exploit problem topology. In fact, the new algorithms we have developed are simpler and easier to understand than current heuristic search methods.

A major barrier for researchers is obtaining access to practitioners and data for real applications. We were able to do so for several applications and have made available data, problem generators and our findings for the applications.

We also have developed new models of search spaces and algorithms. In several cases, we have confirmed conjectures from the literature concerning the difficult of different types of problems and explained why these difficulties arise.

Finally, we went one step further and showed how carefully constructed models can be used to strip away irrelevant algorithm features. These analyses have led to the development of new simple, very successful algorithms that are tailored to application specific features.

2 Accomplishments/New Findings

This section organizes the major results of our project in two categories: modeling search spaces and analyzing applications. The first presents conjectures about what factors influence algorithm performance and describes data collection and model construction to substantiate those conjectures. The second exploits the models to explain performance of algorithms in specific applications and tailoring new algorithms for them.

2.1 Introduction to the Scheduling Applications

To understand the significance and accomplishments of our research requires some knowledge of the scheduling applications being studied. We examined two types of problems: a classic application that has been well studied by researchers and a family of over-subscribed scheduling applications that are largely space-based and associated with deployed systems.

2.1.1 Job Shop (JSP)

JSP is an idealized form of manufacturing scheduling in which n jobs must be processed exactly once on each of m machines. Each job moves between machines in a set order specific to that job; each time a job is processed on a machine is called an *operation*. The scheduling does not allow concurrency (i.e., each machine processes only one job at a time) or pre-emption (i.e., once started on a machine a job cannot be interrupted). Solutions are evaluated based on their makespan: the time between the start of the first operation on any machine and the finishing of the last job on any machine. The problem is NP-Complete.

2.1.2 Over-subscribed Scheduling

In over-subscribed scheduling, problems may include more tasks than can be feasibly scheduled given the available resources. Objectives are often to maximize the number of tasks included or to maximize the utilization of the resources, while recognizing that some tasks may simply not be scheduled. This is a common characteristic of many scheduling applications.

We have examined four over-subscribed applications. Each application must be solved in practice by some form of semi-automated system. Each involves earth orbiting satellites and so share other interesting features. Each task has a time window for execution and usually has more than one alternative window. The periodicity and length of the time windows vary depending on the actual orbits. The resources have unit capacity.

Air Force Satellite Control Network (AFSCN) handles communication contacts between ground stations and satellites for the U.S.A. Air Force. The requests are scheduled on a set of 16 antennas at nine ground stations and can be grouped by whether they are contacting satellites at low or high altitude orbits. Low altitude contacts have short visibility windows and few alternatives. High altitude contacts may be scheduled at different ground stations and have more latitude in their contacts.

Human schedulers at Shriever Air Force Base construct the schedules about one week in advance of each working day. The process is partially automated, but requires considerable effort from the humans to repair the automated solutions and negotiate compromises in the requests so as to accommodate more requests. Our optimization used two different objectives: minimizing bumps and minimizing overlaps. Most previous optimization and the current system minimizes the number of tasks that cannot be fit in the schedule ("bumps"). Conversations with the human schedulers convinced us that instead the optimization should construct an infeasible schedule in which tasks overlapped by a small amount. The objective is to minimize the sum of amount of time two tasks are both scheduled on the same resource. By switching objectives, longer tasks such as maintenance would still be left in the schedule and the actual overlaps between jobs are minimized which often make the schedule much easier to repair; this means human schedulers will have a much better starting point for their negotiations.

We received seven days of actual AFSCN data circa 1993 from Dr. James Moore of AFIT and five days circa 2003 from Brian Bayless and William Szary of Schriever AFB. We based our studies on these problems.

ROADEF and EOS are applications that involve scheduling image capture on a single earth orbiting satellite for the European Space Agency and NASA, respectively. The ROADEF problems were constructed for a 2003 European competition of scheduling algorithms [5]. Images could be single image or a polygon (set of adjacent images) and could be mono or stereo. The satellite had a fixed camera and would shift to take images either forward or backward. The problems varied significantly in the number of acquisitions to be scheduled and the degree of oversubscription.

In EOS, the requests are scheduled on a set of one to three identical satellites that are spaced ten minutes apart in a sun-synchronous earth orbit at 800 km. The imaging sensor has a slew of 24 degrees off nadir. The satellite is limited in how many images it can take before it must download them at a tracking station located in Alaska. NASA would not provide us with actual data, but Al Globus from NASA did help us construct realistic problems like those used by NASA researchers. For each satellite, 2100 land based targets were selected and given image exposures that varied from 24 to 48 seconds. We used 10 test problems in our study.

SPACETRACK is a phased array radar system that is responsible for tracking everything from junk objects floating in space to satellites to space stations; it is also used for tracking objects in flight (e.g., missiles). There are approximately 12,000 objects that require tracking. As in satellite communication scheduling, tasks specify windows of visibility and alternative periodic observation opportunities. However, the total radar power is also a limiting factor.

The amount of power required and the probability of observing and tracking an object is affected by the object's range and the object's position and orientation. An object is said to be "on-peak" when the probability of detecting the object is greatest; currently all scheduling is done "on-peak." However, restricting observations to only those times that are "on-peak" can limit the total number of observations possible. The goal in this case to maximize the total number of successful tracks in a given time period (or to minimize the time needed to track a given set of objects). Also, some objects have higher priorities than others, and some objects must be tracked more frequently than others.

The ITT Sensor group based in Colorado Springs holds a contract to develop the next SpaceTrack system (of which the scheduler is one part). ITT has asked us to work with them and has provided real orbit and tracking data on 6000 objects; ITT is also interested in using any promising algorithms that we develop.

The optimization work on the SpaceTrack problem is mostly future research. Under the current grant, we have built a highly detailed simulator and software environment to model orbits and to construct schedules using permutation based prior queues (which implicitly correspond to schedules) as well as real-valued representations that explicitly represent the schedule start-times. Preliminary results using simple greedy scheduling methods, local search and genetic algorithms strongly suggest that off-peak scheduling methods can be used to improve on the current state of the art.

2.2 Search Space Topologies

The performance of heuristic search algorithms is dictated in large part by the topology of the search space. For example, it has been known for some time that local search algorithms are vulnerable to plateaus (neighborhoods with no gradient information) and hidden ridges (better solutions that are missed by the discretization of a neighborhood on continuous space).

Much of the research on search space topologies has been done for SAT and other general combinatorial optimization problems. We have focused on scheduling. In particular, our project has examined 1) what topological features can be found in scheduling applications, 2) how to detect their presence, 3) how their presence influences algorithm performance and 4) how the search space models can motivate the design of new algorithms. These questions were analyzed for a set of applications. This section discusses our findings on the search spaces. The next section covers what we have learned about specific applications of interest to the Air Force. We first cover the influence of plateaus, then cover random walk models for both JSP and AFSCN scheduling.

2.2.1 Capturing the Influence of Plateaus

We studied the factors that influence algorithm performance for the AFSCN oversubscribed scheduling application [2]. After testing a wide range of algorithms, we identified a set of fairly simple algorithms that perform best relative to several constraint directed and heuristic techniques. The algorithms in our set are: local search, a genetic algorithm (GA) and Squeaky Wheel Optimization (SWO). These algorithms are very different from each other in the way they traverse the search space. From our studies, we identified the problem characteristics that favor each of the algorithms in the set, such as the redundancy of the search space, the size of the search neighborhood and the proclivity of plateaus.

We find that the redundancy of the search space is the key factor in algorithm performance. Redundancy occurs for two reasons: different schedules can have the same evaluation, and different representations can map to the same schedule. For the first, the objective function was quite coarse: counting the number of tasks that could not be scheduled or number of minutes that tasks overlapped when all tasks are placed in a (non-feasible) schedule. Because the evaluations are coarse grain, many very different solutions may have the same evaluation. For the second reason, all of the competitive algorithms searched a permutation space that ranked relative task priorities. The permutation was converted to a schedule. Consequently, many permutations mapped to the same schedule because there was no contention for resources at specific times between the tasks that differed. This redundancy produces plateaus in the search space that must be traversed during search.

We formulated several hypotheses about why particular algorithms excel. First, plateaus appear to be a dominant feature of this search space due to the discrete nature of both objective functions and the n to 1 mapping from the permutation representation to the schedules. Each of the algorithms handle plateaus differently. Local search randomly walks on the plateaus until it finds exits to lower plateaus: the higher percentage of the space occupied by plateaus, the more random wandering is likely for local search. We found that indeed 80% of the steps taken by local search are on plateaus. We show that the n to 1 mapping from the permutation to the

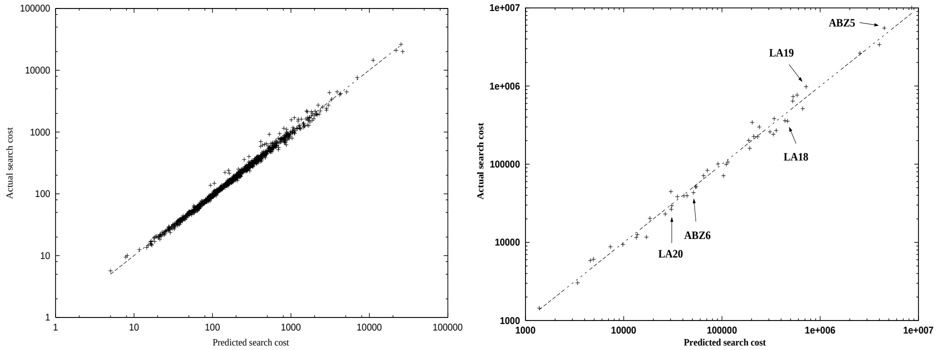

Figure 1: Scatter-plots of the observed versus predicted mean cost ($\overline{c}$) to locate an optimal solution under $TS_{Taillard}$ for 6×6 (left figure) and 10×10 (right figure) random JSPs; the least-squares fit lines are sup er-imposed.

schedule space is the main factor in algorithm performance for local search. Additionally, we show that the ordering of the neighbors matters a great deal for expediting plateau traversal for local search.

Next, we hypothesized that the size of the search space and the proclivity of plateaus favor algorithms that take long, directed leaps across the search space. By leveraging this similarity between the GA and SWO, we were able to design experiments to demonstrate that their power is obtained from taking multiple steps and therefore instituting large changes to the schedules; this is in contrast to taking single steps, which translate into minor, incremental changes to the schedule. Initializing the GA and SWO with a greedy solution is also a factor in algorithm performance, but is not as crucial as making long leaps.

We should also note that other researchers continue to apply exact MIP and LP methods to the AFOSR problem. To allow these methods to terminate in reasonable time the problem must be broken into smaller subpieces, but this decomposition also results in inferior solutions. A recent INFORMS conference presentation [1] reported LP results that were not only inferior to those produced using the GA or SWO, the methods were also 1000 times slower.

2.2.2 Random Walk Models

In an earlier project, we had developed static and quasi-dynamic models of JSP. From those, we found that the relative difficulty of JSPs is primarily due to the size of the search space. Also, the variability of problem difficulty for JSPs is too complex to be captured by simple summary statistics or static models. The most notable failing of the static and quasi-dynamic models was they were least accurate on the most difficult problems. We were able to overcome those failings by constructing a *dynamic* model of the search space [16] for one of the most successful algorithms for JSP, Taillard's Tabu Search.

We found that the dynamics of the search space could be modeled with surprisingly high fidelity (r^2=.99 on log-log plots of predicted and actual cost) as a simple Markov chain (see figure 1)[17]. The states were represented by the distance from the optimum and whether the search was moving toward or away from the optimum. These results suggest that Taillard's

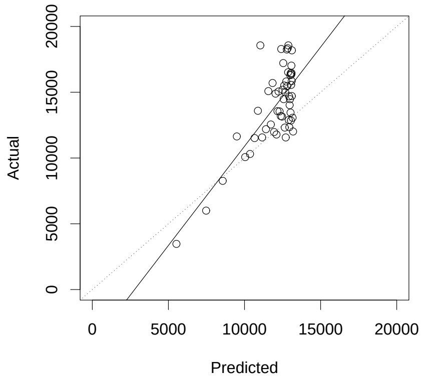

Figure 2: The correlation plot for Mar07, the largest AFSCN instance. The dashed line indicates perfect correlation. This plot reveals evidence of compression in the model for states far away from the optimum.

Tabu Search algorithm can be viewed as a one-dimensional random walk with a bias toward solutions that are roughly equi-distant from the nearest optimal solution and solutions that are maximally distant.

Although many researchers have examined search spaces and identified key features such as plateaus, we are one of the few who have constructed dynamic models of the search space. The paper introducing this model for JSP won the Best Paper award at the First Multi-disciplinary Conference on Scheduling in August 2003.

We applied the dynamic model methodology to the AFSCN and EOS applications. We and others [8] had found local search to be extremely effective in solving both applications and wondered whether similar forces were at work with this application, i.e., that the algorithms are performing a random walk. In this case, we did not know the optimal solutions to the real problems and so could not define states in the same way. Because we knew that plateaus were common in the space (see section 2.2.1 for details), a state was defined by the evaluation of a schedule; thus, all solutions with the same value were aggregated.

We found the Markov Models to be somewhat less accurate for these problems than for JSP, but still quite predictive. Models were constructed for each of the problems (12 for AFSCN and 6 for EOS). For the simpler AFSCN data (the 7 days of circa 1993 AFIT data) the Markov Model produced r^2 values that varied from 0.04 to 0.855 (mean of 0.64). For the EOS problems, the Markov Model produced r^2 values that varied from 0.61 to 0.84.

For the more difficult and recent AFSCN data, we used an *Extend Markov Model* that also includes information about how long the search has been stuck on a plateau; this model produced r^2 values that varied from 0.85 to 0.95, which is actually quite good. Figure 2 shows one of the typical correlations. Interestingly, the random walk model was most accurate for the more difficult problems in the set.

2.2.3 Infeasible Space

Searches that include both feasible and infeasible solutions have proved to be efficient algorithms for solving some scheduling problems. Researchers conjecture that these algorithms yield two primary benefits: 1) they tend to focus on solutions close to the boundary between feasible and infeasible solutions, where active constraints are likely to yield optimal values, and 2) moves that cross infeasible regions may uncover short-cuts in a search space. Published empirical studies confirm the value of searching along the feasible-infeasible boundary, but until now there has been little direct evidence that infeasible search yields short-cuts.

We present empirical results in two oversubscribed scheduling domains for which boundary region search in infeasible space appears to offer advantages over search in strictly feasible space: ROADEF and EOS. In both domains the best solutions are likely to contain tasks with active constraints. Cordeau and Laporte implemented a strategic oscillation tabu search algorithm ($Tabu_{CL}$) that performed exceptionally well [4]. We use their Tabu Search algorithm as the basis for our work.

Our results confirm that infeasible search finds shortcuts that may improve search efficiency more than boundary region search alone. However, our results also reveal that inefficient infeasible paths which we call *detours* may degrade search performance, potentially offsetting the efficiency shortcuts may offer.

To assess the impact of search through infeasible space, we defined a set of path types that classify search space traversals. We define a *path segment* as any sequence of schedules explored during a search. A *feasible* segment is one in which each schedule is feasible. An *infeasible* segment is one where each schedule in the segment contains at least one constraint violation except for the start and end schedules, which are feasible. This restriction on the start and end schedules helps us divide infeasible search into distinct infeasible segments.

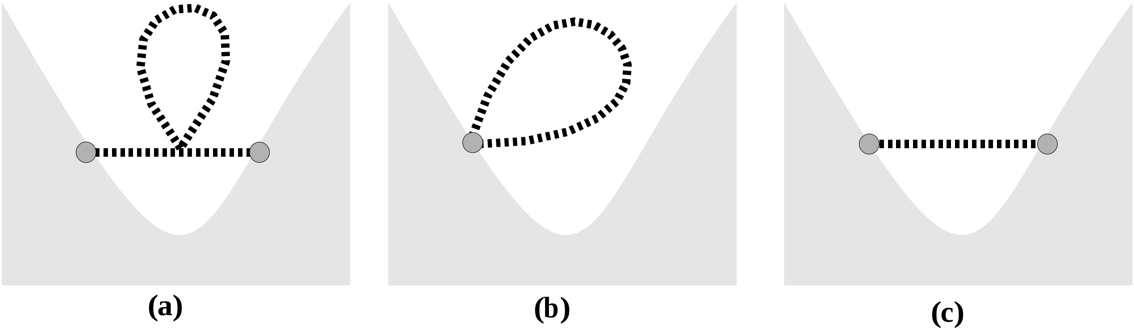

Figure 3: Three infeasible segment types: (a) a *detour* encounters the same state twice before returning to feasible space, (b) a *cycle* begins and ends at the same state, and (c) a short-cut cannot be duplicated in the same number of feasible moves.

Specific to infeasible search, we consider three phenomena: cycles, detours and short-cuts (see Figure 3). When a segment's start and finish state are identical, then the segment is a *cycle*. In addition to simple cycles, if there is a cycle within an infeasible segment from one infeasible state to another, then we identify the segment as a *detour*. The final infeasible segment we consider in this study is the *short-cut*. We define a short-cut as an infeasible segment whose length is shorter than any feasible segment between two feasible states.

We found evidence of short-cuts as well as traversals on the boundary region. For many problems, allowing search to traverse infeasible search led to more efficient algorithms. However, the benefits of infeasible traversal must be balanced against the costs. Experimental results show that infeasible paths tend to yield a high proportion of cycles. For the ROADEF problem, cycles comprise 65% of attempted infeasible paths, and for EOS, cycles represent 24%.

2.3 New Solutions for Applications

Our methodology is to study an application, identify the prominent features of its search space, model their influence on design decisions in state of the art algorithms and then design a simplified algorithm that includes only the most efficacious components. We have followed this procedure for each of the applications that we have studied and produced much simpler algorithms that are easier to understand and relate to the applications.

2.3.1 New Algorithm for JSP

JSP has received considerable attention over the years. The state of the art algorithms solve large problems in short order. Yet they are fairly complex and little is known about why they perform well.

The state-of-the-art JSP algorithm is Nowicki and Smutnicki's *i*-TSAB tabu search algorithm; the algorithm [13] is an extension of their earlier landmark TSAB algorithm [11]. In addition to the short-term memory mechanism found in all implementations of tabu search, *i*-TSAB is characterized by the following algorithmic components: (1) the highly restrictive *N5* move operator, [1] (2) re-intensification of search around previously encountered high-quality solutions, and (3) diversification of search via path relinking between high-quality solutions. Despite its effectiveness, almost nothing is known about how both these and various secondary components interact to achieve state-of-the-art performance, or even if all of the components are necessary. Thus, a key research goal is to determine which components of *i*-TSAB are integral to its remarkable performance, and the degree to which they share the responsibility.

We demonstrate that the "core" tabu search metaheuristic does not provide any distinct advantage over other metaheuristics for the JSP based on the *N5* operator. In particular, both iterated local search and Monte Carlo sampling provide competitive performance. Next, we show that intensification and diversification can significantly improve the performance of both tabu search and the other metaheuristics we consider, such that the resulting performance is consistently competitive with *i*-TSAB. Finally, we analyze the relative impact of intensification and diversification on the performance of tabu search algorithms for the JSP. Here, we find that although application of either mechanism in isolation can lead to performance improvements, their mutual effect is multiplicative: both components are required to achieve state-of-the-art performance.

---

[1] Our notation for move operators is taken from [3].

In JSP, attractor basins of local optima are surprisingly weak; Watson [16] shows that the local optima of random JSP instances can be escaped with high probability simply by accepting one or two dis-improving moves and re-initiating greedy descent. This result provides a straightforward operational definition for an N_l operator, which in turn forms the basis for a new *N5*-based ILS metaheuristic for the JSP. The resulting metaheuristic is denoted IJAR, for *I*terated *J*ump *A*nd *R*edescend.

In a typical iteration, IJAR accepts a random sequence of at most k monotonically dis-improving neighbors from the current solution s, where k is a user-specified parameter. Randomized next-descent is then used to transform the resulting solution s' into a local optimum s'', which then serves as the s in the next iteration. If search consistently locates new local optima, IJAR continues to perform nominal iterations. However, in practice IJAR occasionally encounters local optima for which the ascent-descent mechanism either cannot guarantee escape or does so with only very low probability. Consequently, IJAR is equipped with mechanisms to both detect these situations and initiate an appropriate recovery.

To detect search stagnation, IJAR analyzes the makespan of solutions generated OTCI (Optima Trap Check Interval) iterations apart. Let s and s^- respectively denote the current solution and the solution observed OTCI iterations prior. If $C_{max}(s) = C_{max}(s^-)$, search is likely either permanently trapped in a local optimum or has a very low probability of escape. In this case, IJAR enters an optimum escape mode or OEM. While in OEM, random walks – with respect to the *N5* operator – of monotonically increasingly length are substituted for the usual ascent-descent process. A fully detailed description of IJAR, including pseudo-code, is provided in [16].

The third novel *N5*-based metaheuristic we introduce is based on Monte Carlo or Metropolis sampling, which serves as the basis of the well-known simulated annealing metaheuristic [10]; run temperatures (see below) are fixed in Monte Carlo sampling, while they are variable in simulated annealing. We refer to this algorithm as Random Monte Carlo (RMC).

Discounting a few exceptional instances, IJAR obtains results similar to a standard Simple Tabu Search (STS) implementation; STS is slightly better, but the difference is minimal. Similarly, RMC consistently *outperforms* STS, but the difference is typically minimal. Overall, the results indicate that tabu search is *not* inherently superior to other metaheuristics for the JSP: straightforward implementations of iterated local search and Monte Carlo sampling provide quantitatively similar performance for the same computational effort.

A key part of i-TSAB is an *intensification-diversification mechanism*; while it includes other features as well, our research indicates that only the *N5* operators and the *intensification-diversification mechanism* are critical. Thus we added the same *intensification-diversification mechanism* to STS, RMC and IJAR, and denote them by i-STS, i-RMC and i-IJAR.

We contrast the performance of algorithms in the follow manner. Let $C^*_{max}(X)$ denote either the optimal makespan of instance X, if known, or the largest known lower bound on the optimal makespan. Up-to-date values of $C^*_{max}(X)$ can be obtained from Taillard's web site [14]; we use values reported as of January 1, 2004.

Let $P(X)$ denote the quality of solutions obtained by a given algorithm on instance X; $P(X)$ represents either the best or mean makespan observed over 10 trials, depending on the context. Algorithm performance on instance X is quantified as the percent excess RE(X)= $\frac{P(X)-C^*_{max}(X)}{C^*_{max}(X)} *$

Table 1: Performance of augmented metaheuristics on Taillard's benchmark problems.

Problem	Best	*i*-RMC		*i*-IJAR		*i*-STS		*i*-TSAB
Group	Known	b-MRE	m-MRE	b-MRE	m-MRE	b-MRE	m-MRE	
ta01-10	0.0	0.05	0.12	0.04	0.09	0.04	0.12	0.11
ta11-20	2.33	2.48	2.74	2.61	2.80	2.58	2.79	2.81
ta21-30	5.45	5.60	5.91	5.62	5.85	5.58	5.79	5.68
ta31-40	0.52	0.71	0.95	0.83	1.01	0.67	0.83	0.78
ta41-50	4.12	4.57	4.90	4.58	4.95	4.35	4.77	4.70

100 of $P(X)$ relative to $C^*_{max}(X)$. We then respectively define b-RE(X) and m-RE(X) as the value of RE(X) computed using the best and mean makespans observed over the 10 trials. Finally, for each problem group, we let b-MRE and m-MRE respectively denote the mean b-RE(X) and m-RE(X) observed across the member instances X. Results comparing *i*-RMC, *i*-IJAR, *i*-STS and *i*-TSAB are given in table 1.

In terms of m-MRE, *i*-STS only slightly under-performs *i*-TSAB; the largest absolute difference is a minimal 0.11. Further, the difference is only 0.07 on the notoriously difficult 30×20 problem group. In terms of b-MRE, *i*-STS consistently outperforms *i*-TSAB, with the largest difference of 0.35 observed on the most difficult problem group. Overall, the performance of *i*-TSAB is bracketed by the b-MRE and m-MRE obtained by both *i*-RMC and *i*-STS (and by *i*-IJAR except on one problem set), and generally the performance of *i*-TSAB is closer to the mean than the best. Unfortunately, the deterministic nature of *i*-TSAB prevents us from establishing a well-defined "mean" performance metric, such that we are unable to assess whether the observed differences are statistically significant.

The primary goal of this research was to explicitly identify those components of Nowicki and Smutnicki's *i*-TSAB tabu search algorithm that enable it to achieve state-of-the-art performance levels, and to quantify the overall contribution of each such component. We show that the core metaheuristic, and in particular tabu search, is *not* integral to the performance of *i*-TSAB. Rather, *i*-TSAB achieves state-of-the-art performance levels through the use of both the *N5* move operator and a balanced combination of intensification and diversification. Viewed from another standpoint, we have shown that state-of-the-art algorithms for the JSP are, at a very fundamental level, quite simple.

2.4 Oversubscribed Scheduling Applications

We developed our algorithm design methodology by working on the well studied JSP. However, we have applied it to the oversubscribed satellite scheduling applications as well.

2.4.1 AFSCN

In striving to improve the solution to AFSCN, we analyzed three factors: objective function, neighborhood for local search and mechanisms traversing the plateaus quickly. The objective function used in prior research and development of deployed systems was to minimize the number of conflicts (number of tasks that could not be fit). We constructed an objective function that minimizes overlaps instead, based on the fact that human schedulers have some flexibility in modifying request durations or setup times. The professional schedulers from

Schriever agreed that this objective function would be more helpful for them than minimizing the number of conflicts. Our empirical results are promising. The new objective function yields less brittle schedules that are more likely to be modifiable by the human schedulers.

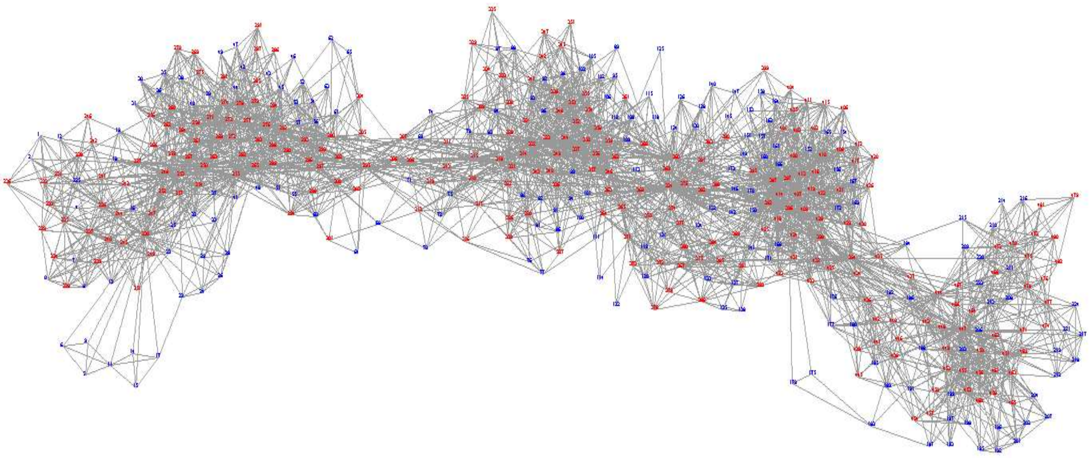

Figure 4: A force-directed layout of the largest connected component in G for Mar07. The problem contains a single connected component that spans most (92%) of the problem; the remaining tasks have zero degree. High-altitude tasks (red when viewed in color) are the most connected tasks and are usually in the center. Low-altitude tasks (blue when viewed in color) are less connected and tend to be along the outside of the graph.

Neighborhood Bias Generally, carefully crafted neighborhoods lead to performance improvements in hill climbing search. We developed an alternative shift neighborhood for AFSCN that was designed to exploit task interactions known to impact schedule quality. In particular, we constructed an interaction graph for problems that captured pairwise task contention for the same resource (two tasks have a link between them if their time windows overlap on the same resource). We then restricted the shift neighborhood for any task to include only those tasks known to cause an interaction. Figure 4 shows an interaction graph for March 7 (one of the more difficult problems).

We found that the restricted neighborhood actually performed slightly worse than a next descent randomly ordered full neighborhood. We conjecture that this is because interactions may be n-way (see Section 2.2.1 for an explanation) and the random walk across the plateaus is not informed by knowing the interactions.

New Algorithm: Attenuated Leap Local Search (ALLS) The empirical data and analyses suggest that the key to competitive performance on the AFSCN application is moving as quickly as possible across the plateaus. Two of the competitive algorithms, Genitor and SWO, perform multiple moves. A simpler Random Local Search (RLS) algorithm actually finds more best known solutions in 8000 evaluations, even though it does not perform multiple moves. RLS selects a random job pair for its shift move operator (and is therefore a form of stochastic hill climbing). RLS does however, perform a significant number of "neutral moves" between solutions with the same evaluation. Given this, we conjecture that a version of local search that performs multiple moves before evaluating the result may be even better suited to this application. The intuition behind this conjecture is that the search should sample at greater distances (i.e., longer than a single move) to more quickly find exits from plateaus.

We modified the RLS move operator as follows: choose a number of pairs of positions and apply shifting to these pairs, one after another, without building the schedule after each shift; build the schedule only after shifting a designated number of pairs. In our first version, we tried a static number of shifts (10 turned out to be the best value); however, it performed no better and sometimes worse than the original move operator. We next conjectured that as search progresses to better solutions, the number of shifts should also decrease because the probability of finding detrimental moves (rather than improving) increases significantly as well. The better the solution, the fewer exits are expected and the harder they are to find.

We implemented a multiple move hill-climber with a variable move count operator: given a decay rate, we start by shifting ten requests, then nine, eight etc. We chose to decrement the number of shifts for every 800 evaluations; we call this version of hill-climbing *Attenuated Leap Local Search* (ALLS).

ALLS performs remarkably well. As shown in Table 2, it finds best known values for all problems using conflicts and all but two of the problems using overlaps (as does RLS). Additionally, it finds better best values than all the algorithms in our set for the two problems with non-best solutions. In fact, a single tailed, two sample t-test comparing ALLS to RLS shows that ALLS finds statistically significantly better solutions ($p < 0.023$) on both conflicts and overlaps for the five more recent (and more oversubscribed) days of data.

Under a much more restrictive performance comparison, ALLS still outperforms RLS, SWO and *Genitor* for most of the pair-wise tests. Both when minimizing conflicts and when minimizing overlaps, ALLS significantly outperforms all other algorithms on R1. When minimizing conflicts, ALLS outperforms for all but five of the twelve pair-wise tests on the other four days (for which the difference was not significant). The exceptions are: R2, R3, R4, and R5 for *Genitor* and R4 for RLS. When minimizing overlaps, ALLS significantly outperforms *Genitor* for R2, RLS for R3, *Genitor* for R4 and SWO for R5; the rest of the pair-wise comparisons were not statistically significant at $p < 0.005$. It is clear that ALLS is at least as good as the best algorithms and outperforms them on most days of data.

ALLS also finds improving solutions faster than both *Genitor* and RLS (see Figures 5 and 6 for R4 on both conflicts and overlaps). ALLS achieves such good performance by combining the power of finding good solutions fast using multiple moves in the beginning of the search with the accuracy of locating the best solutions using one-move shifting at the end of the search.

We developed and tested four hypotheses explaining the performance of three competitive algorithms for a real scheduling application. We found that all of the hypotheses held to varying degrees. Based on the evidence, we designed a new algorithm that combined what appeared to be critical elements of the best performing algorithms and produced an algorithm that performed better than the original ones. Our results suggest that multiple moves are a useful algorithm feature to obtain good performance results for AFSCN scheduling. Alternatively, it is possible that in fact only one move during each iteration would be enough to obtain good performance, but it is difficult to identify which request to move.

	Minimizing Conflicts			Minimizing Overlaps		
Day	Min	Mean	Stdev	Min	Mean	Stdev
A1	**8**	8.2	0.4	**104**	107.1	1.24
A2	**4**	4.0	0.0	**13**	13.0	0.0
A3	**3**	3.0	0.0	**28**	28.33	1.3
A4	**2**	2.03	0.18	**9**	9.13	0.73
A5	**4**	4.1	0.3	**30**	30.23	0.43
A6	**6**	6.0	0.0	**45**	45.0	0.0
A7	**6**	6.0	0.0	**46**	46.0	0.0
R1	**42**	42.63	0.72	785	817.83	27.07
R2	**29**	29.1	0.3	490	510.37	19.14
R3	**17**	17.5	0.57	**250**	273.33	43.68
R4	**28**	28.07	0.25	**725**	740.07	19.56
R5	**12**	12.0	0.0	**146**	146.03	0.19

Table 2: Statistics for the results obtained in 30 runs of *ALLS*, with 8,000 evaluations per run. The best and mean values as well as the standard deviations are shown. Bold indicates best known values. Days A1 to A7 are seven days of data circa 1993. Days R1 or R5 are five days of data circa 2003.

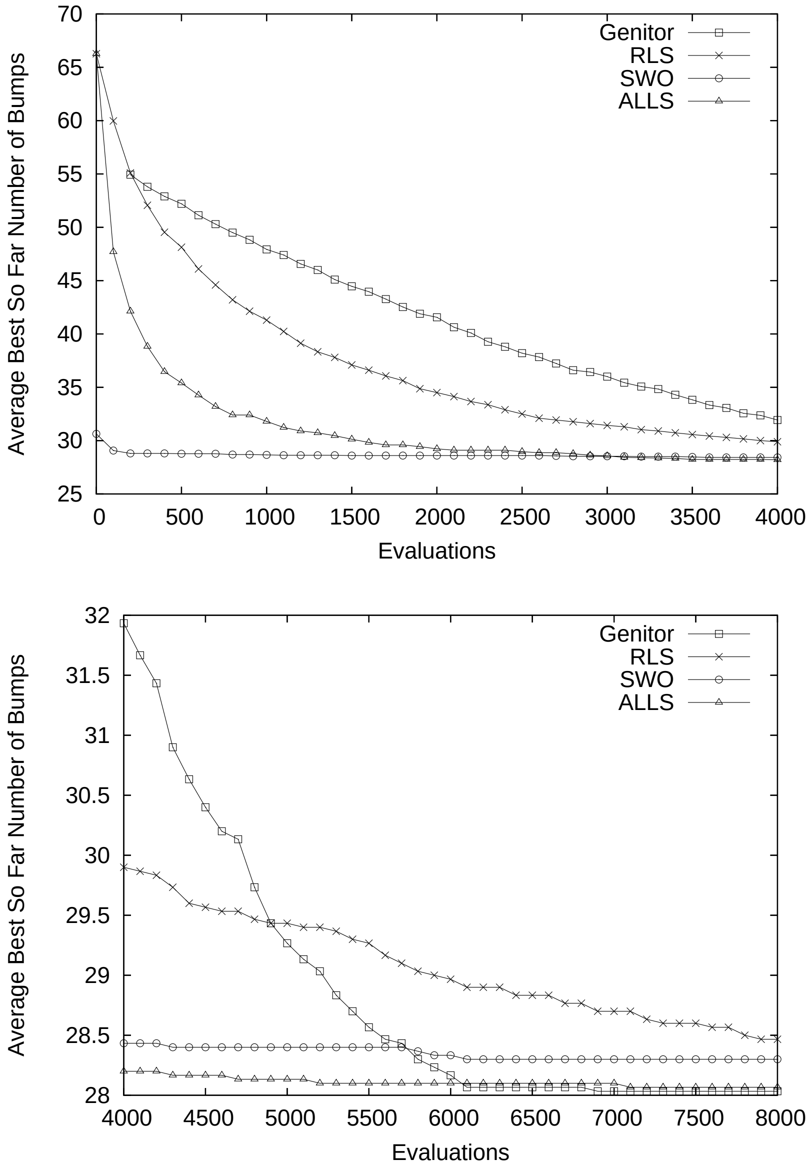

Figure 5: Evolutions of the average best value obtained by *Genitor*, RLS, SWO and *ALLS* during 8000 evaluations, over 30 runs. The improvement over the first 4000 evaluations is shown in the top figure. The last 4000 evaluations are depicted in the bottom figure; note that the scale is different on the y-axis. The graphs were obtained for $R4$; best solution value is 28.

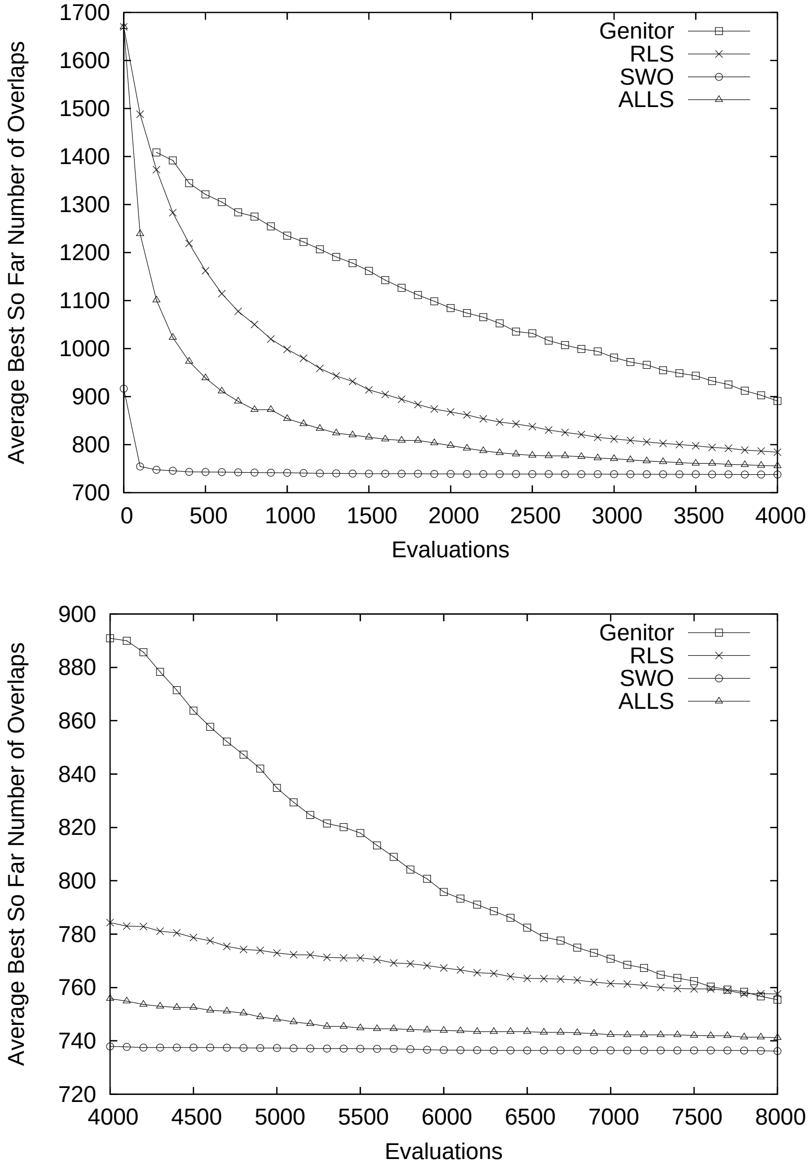

Figure 6: Evolutions of the average best value obtained by *Genitor*, RLS, SWO and *ALLS* during 8000 evaluations, over 30 runs. The improvement over the first 4000 evaluations is shown in the top figure. The last 4000 evaluations are depicted in the bottom figure; note that the scale is different on the y-axis.The graphs were obtained for $R4$; best solution value is 725.

Future research in this direction will examine heuristics such as combining HBSS and SWO to decide which request to move forward, as well as heuristics to find where to move the request to guarantee a change in the schedule. Also as future research, we will be testing other oversubscribed scheduling applications to determine to what extent our analyses and results generalize: do they exhibit the same characteristics and are they amenable to the same kind of solution?

2.4.2 Other Satellite Applications

With ROADEF and EOS, the best solution was one that also searched infeasible space. However, we found that many of the forays into infeasible space were wasted effort due to cycles and detours. We hypothesized that a second infeasible tabu list could force a search to explore a more diverse selection of infeasible paths than the original search, thereby eliminating many cycles and increasing the number of shortcuts that are explored. Our experimental results indicate that when we add an infeasible tabu list, we mitigated cycles and detours in infeasible paths and increased the proportion of shortcuts found during a search. As we predicted, the proportion of cycles declined dramatically in both domains while the proportion of detours declined slightly. These enhancements yielded good results for the EOS problems. However, the performance varied on the ROADEF problems. The infeasible tabu list nearly doubled the proportion of shortcuts for the ROADEF problems, yet we saw little improvement in the results. Thus although we have confirmed that infeasible search can find shortcuts, shortcuts do not always improve search efficiency for every problem domain.

Our results in AFSCN demonstrate that a key algorithm characteristic appears to be multiple moves. In fact, this observation might hold for other oversubscribed scheduling problems as well. Globus et al. [8] found that when solving EOS using hill-climbing, moving only one request at a time was inefficient. They developed a “temperature dependent” hill-climbing move operator, for which the number of requests to move is chosen by random (from between 1 and 15) but biased such that a large number of requests are moved early in the search while later only a few requests are moved. The temperature dependent hill-climbing operator proved to work better than simply choosing a random number of requests to move. As in our domain, a permutation representation and a greedy deterministic schedule builder are used. We conjecture that their schedule builder also results in multiple permutations being mapped to the same schedule, and therefore that plateaus are present in the EOS search space as well.

The fact that moving more than one request improved the results suggests that our conjecture could also hold for EOS scheduling: multiple moves might speed up plateau traversal for this domain as well. We found additional evidence to support this when we analyzed actual runs of our Tabu Search algorithm for this problem: the same moves were rising to the top during subsequent traversals of the search space. It appeared that search could be expedited by simply taking the top n moves before evaluating the schedule. However, when we implemented this strategy in our infeasible search framework, it worsened performance, even when only two moves at a time were taken.

The most promising future avenue for algorithm development is to consider where a shortcut is likely to find an improving path. Our on-going research is aimed at using machine learning methods to predict when a shortcut is likely to be productive; if we can predict when a shortcut is likely to find an improved solution, the amount of time spent exploring the infeasible space can be greatly reduced.

3 Executive Summary

3.1 Personnel

During the grant period, the following personnel were supported at the indicated level:

PIs:		
Adele Howe	4.5	months
L. Darrell Whitley	1	months
Research Assistants:		
Laura Barbulescu	15.5	full-time months (5 full, 21 half-time)
Mark Roberts	16	full-time months (4.75 full-time, 22.5 half-time)
Mark Rogers	14.11	full-time months (5.11 full-time, 18 half-time)
Andrew Sutton	7.36	full-time months (2.86 full-time, 9 half-time)
Jean-Paul Watson	2.9	full-time months (2.5 full-time, 0.86 half-time)

3.2 Publications

Journals

- L. Barbulescu, A.E. Howe, L. Darrell Whitley, M. Roberts. “Algorithm Performance Factors for an Oversubscribed Scheduling Application”, to appear in *Journal of Artificial Intelligence Research*.

- L. Barbulescu, A.E. Howe and L.D. Whitley. 2006. “AFSCN Scheduling: How the Problem and Solution Have Evolved”, in *Mathematical and Computer Modelling*, Vol. 43, No. 9-10, pp. 1023-1037, May.

- J.P. Watson, A.E. Howe and L.D. Whitley. 2005. “Deconstructing Nowicki and Smutnicki’s *i*-TSAB Tabu Search Algorithm for the Job-Shop Scheduling Problem”, in *Computers and Operations Research*, Vol. 33, No. 9, pp. 2623-2644, September.

- J.P. Watson, L.D. Whitley, and A.E. Howe. “Linking Search Space Structure, Run-Time Dynamics and Problem Difficulty: A Step Toward Demystifying Tabu Search”, in *Journal of Artificial Intelligence Research*, Volume 24, pp. 221-261, August 2005.

- J. Rowe, D. Whitley, L. Barbulescu and J.P. Watson. 2004. “Properties of Gray and Binary Representations. Evolutionary Computation”, 12(1):47-76.

- L. Barbulescu, J.P. Watson, D.L. Whitley, A.E. Howe. 2004. “Scheduling Space-Ground Communications for the Air Force Satellite Control Network”, Vol. 7, Issue 1, pp. 7-34, January.

Conferences

- A. Sutton, D. Whitley, M. Lunacke and A. Howe. 2006. "PSO and Multi-Funnel Landscapes: How Cooperations Might Limit Exploration", *Genetic and Evolutionary Computation Conference*, July, pp. 75-81. (Nominated for Best Paper Award).
- M. Rogers, A.E. Howe, and L.D. Whitley. 2006. "Looking for Shortcuts: Infeasible Search Analysis for Oversubscribed Scheduling Problems", *Proceedings of the Sixteenth International Conference on Automated Planning and Scheduling*, June, pp. 314-323.
- A. Sutton, L.D. Whitley and A.E. Howe. 2006. "Spacetrack: Trading off Quality and Utilization in Oversubscribed Schedules", in *Proceedings of the Sixteenth International Conference on Automated Planning and Scheduling, short paper*, June, pp. 430-433.
- M. Roberts, L.D. Whitley, A.E. Howe and L. Barbulescu. 2005. "Random Walks and Neighborhood Bias in Oversubscribed Scheduling". in *Proceedings of Second Multidisciplinary International Conference on Scheduling (MISTA 2005)*, NY, NY, July.
- M. Roberts, A. Howe and L.D. Whitley. 2005. "Modeling Local Search: A First Step Toward Understanding Hill-Climbing in Oversubscribed Scheduling", extended abstract in *Poster Proceedings of International Conference on Automated Planning and Scheduling (ICAPS 2005)*, Monterey, CA, June.
- L. Barbulescu, L.D. Whitley and A.E. Howe. 2004. "Leap Before You Look: An Effective Strategy for Oversubscribed Scheduling", in *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, San Jose, CA, July.
- L. Barbulescu, A.E. Howe, L.D. Whitley and M. Roberts. 2004. "Trading Places: How to Schedule More in a Multiple Resource Over-Subscribed Scheduling Problem", In *Proceedings of 14th International Conference on Automated Planning and Scheduling*, Whistler, British Columbia, June.
- J.P. Watson, A.E. Howe and L.D. Whitley. 2003. "An Analysis of Iterated Local Search for Job-Shop Scheduling", In *Fifth Metaheuristics International Conference (MIC 2003)*, Japan, September.
- J.P. Watson, L.D. Whitley, and A.E. Howe. 2003. "A Dynamic Model of Tabu Search for the Job-Shop Scheduling Problem", In the *First Multidisciplinary International Conference on Scheduling (MISTA 2003)*, Nottingham, England. August

3.3 Graduate Theses

During this project, two of the graduate research assistants completed their Ph.D.s and one completed his M.S. as follows.

"Empirical Modeling and Analysis of Local Search Algorithms for the Job-Shop Scheduling Problem" Ph.D. Thesis by Jean-Paul Watson, Summer 2003. Abstract:

> Local search algorithms are among the most effective approaches for locating high quality solutions to a wide range of combinatorial optimization problems. However, our theoretical understanding of these algorithms is very limited, leading to significant problems for both researchers and practitioners. Specifically, the lack of a theory of local search impedes the development of more effective algorithms, prevents practitioners from identifying the algorithm most appropriate for a given problem, and allows widespread conjecture and misinformation regarding the benefits and/or drawbacks of particular algorithms. This thesis represents a significant step toward a theory of local search. Using empirical methods, we develop theoretical models of the behavior of four well-known local search algorithms: a random walk, tabu search, iterated local search and simulated annealing. The analysis proceeds in the context of the well-known job-shop scheduling problem, one of the most difficult NP-hard problems encountered in practice. The large volume of prior research on the job-shop[scheduling problem provides a diverse range of available algorithms and problem instances, in addition to numerous empirical observations regarding local search algorithm behavior; the latter are used to validate our behavioral models.
>
> We show that all four local search algorithms can be modeled with high fidelity using straightforward variations of a generalized one-dimensional Markov chain. The states in these models represent sets of solutions a given fixed distance from the nearest optimal solution. The transition probabilities in all of the models are remarkably similar, in that search is consistently biased toward solutions that are roughly equidistant from the nearest optimal solution and solutions that are maximally distant from the nearest optimal solution. Surprisingly, the qualitative form of the transition probabilities is simply due to the structure of the representation used to encode solutions: the binary hypercube. The models account for between 96% and 99% of the variability in the cost required to locate both optimal and sub-optimal solutions to a wide range of problem instances, and provide explanations for numerous phenomena related to problem difficulty for local search in the job-shop scheduling problem. In the course of our analysis, we also disprove many conjectures regarding the behavior and benefits of particular algorithms.
>
> Our research indicates that despite their effectiveness, local search algorithms for the job-shop scheduling problem exhibit surprisingly simple run-time dynamics. Further, we observe minimal differences between the dynamical behavior of different algorithms. As expected given similar run-time dynamics, although contrary to numerous reports appearing in the literature, we also show that the performance of different algorithms is largely indistinguishable. Ultimately, our behavioral models serve to unify and provide explanations for a large body of observations regarding problem difficulty for local search in the job-shop scheduling problem, and identify new research areas of the development of more effective local search algorithms.

Jean-Paul Watson is now a staff scientist at Sandia National Laboratories.

“Analysis of Algorithm Performance for an Oversubscribed Scheduling Problem”

Ph.D. Thesis by Laura Barbulescu, Fall 2005. Abstract:

We analyze the factors that influence algorithm performance for an oversubscribed application, scheduling for the Air Force Satellite Control Network (AFSCN). AFSCN scheduling assigns access requests to specific time slots on antennas at ground stations. The application is oversubscribed: not all tasks can be accommodated given the available resources. As a special class of scheduling problems, oversubscribed problems present additional challenges. While, in general, solutions to scheduling problems specify the start times and resources assigned to tasks, in oversubscribed scheduling the maximal subset of tasks that can be scheduled with the available resources also needs to be identified.

We implemented various algorithms for AFSCN scheduling. Some algorithms, such as a domain-specific repair-based algorithm or constraint-based scheduling heuristics, failed to identify good solutions. We have found a set of fairly simple algorithms that perform well on the AFSCN scheduling domain, for both real and synthetically generated problems. The algorithms in the set are: hill-climbing, a genetic algorithm (GA) and Squeaky Wheel Optimization (SWO). All the algorithms in the set are designed to traverse the same search space: solutions are represented as permutations of tasks; a greedy schedule builder converts the permutation into a schedule by assigning a start time and resources to the requests in the order in which they appear in the permutation. However, these algorithms vary in the *way* they traverse the search space. This research identifies performance factors that make each of the algorithms a good fit for AFSCN scheduling.

The AFSCN scheduling search space is dominated by plateaus, due to both the discrete nature of the objective function and to the fact that the schedule builder converts multiple permutations into identical schedules. Each algorithm handles plateaus differently. Hill-climbing randomly walks on the plateaus until it finds exits to lower plateaus: the higher the percentage of the space occupied by plateaus, the more random wandering is likely for hill-climbing. We found the ordering of the neighbors to be the main performance factor in expediting plateau traversal for hill-climbing. The GA and SWO both traverse the plateaus quickly, by making multiple changes to the solutions. The long, directed leaps across the search space are the main performance factor for the GA and SWO.

We also investigated whether initializing the search closer to the best solutions is the key to performance. We found that such initialization helps but is not by itself enough to explain algorithm performance results.

The main contributions of this research work are: 1) We performed the first coupled formal and empirical analysis of the AFSCN scheduling problem. 2) We designed techniques for analyzing algorithm performance, which could transfer to other applications. 3) We identified algorithm performance factors, which are likely to hold on other similar problems. 4) We designed a new best performing algorithm, by combining the features we found to have most influence on performance.

Laura Barbulescu is now a researcher at Carnegie-Mellon University.

"Modeling Choices in Two Oversubscribed Scheduling Problems" M.S. Dissertation by Mark Roberts, Fall 2005. Abstract:

> A critical component of a search algorithm is the decision it makes about subsequent states in the search space: what are the possibilities and how should they be prioritized? This thesis examines the decisions in two categories of algorithms for two oversubscribed scheduling problems: scheduling satellite communications for the Air Force Satellite Control Network (AFSCN) and scheduling image requests on a set of Earth Observing Satellites (EOS).
>
> The first category of algorithm is heuristic search, for example, constructive and repair-based search. For constructive search, we examine three ordering heuristics: sum-height, job flexibility, and size of the alternative list. For repair-based search, we implement the "TaskSwapping" framework. We compare these approaches to previously published results for a next-descent hill-climber and find that constructive and repair-based search performances are dominated by the hill-climber. We conjecture that this poor performance results from the myopic decisions of constructive and repair-based search.
>
> The second category of algorithm is hill climbing. Previous research showed that the search topology of AFSCN is dominated by plateaus and that algorithms that do well in AFSCN take multiple, simultaneous moves. Our next-descent hill-climber chooses from the shift neighborhood, a very large neighborhood that is $O(n^2)$, where n is the number of jobs to schedule. We examine one domain-justified way to reduce this neighborhood size and find that randomized moves in a large neighborhood still performs best for local search in both AFSCN and EOS applications.
>
> Because of the large plateaus and the success of randomized choices in the neighborhood, we hypothesized that hill-climbing search could be modeled as a random walk. We model the simple hill-climber as a Markov process and validate the models for AFSCN and EOS. Our results show that the models are reasonably accurate for EOS and for many problems in AFSCN.

Mark Roberts is currently in the Ph.D. program at Colorado State University.

3.4 Interactions/Transitions

3.4.1 Presentations at Meetings

J.P. Watson, "Modeling Local Search Algorithm Performance in Scheduling", AFOSR PI meeting, Estes Park, Colorado, May 27, 2003.

L. Barbulescu, "Oversubscribed Scheduling: A Case Study of Air Force Ground Station Scheduling", AFOSR PI meeting, Estes Park, Colorado, May 27, 2003.

L.D. Whitley, Keynote Address: "No Free Lunch and Beyond". at the First Mexican Conference on Evolutionary Computation, Guanajuato, Mexico, May 29, 2003.

L.D. Whitley, Tutorial: "Evaluating Evolutionary Algorithms", Genetic and Evolutionary Computation Conference, July 13, 2003.

L.D. Whitley, Tutorial: "Genetic Algorithms in Search and AI", International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 10, 2003.

L.D. Whitley, Tutorial: "No Free Lunch and Complexity", at First Multidisciplinary International Conference on Scheduling", Nottingham, England, August 12, 2003.

J.P. Watson, "A Dynamic Model of Tabu Search for the Job-Shop Scheduling Problem, at First Multidisciplinary International Conference on Scheduling", Nottingham, England, August 14, 2003.

J.P. Watson, "An Analysis of Iterated Local Search for Job-Shop Scheduling" at Fifth Metaheuristics International Conference (MIC 2003), August 28, 2003.

J.P. Watson, "Modeling the Dynamic Behavior of Metaheuristics: A Case Study in Job-Shop Scheduling" at CORS/SCRO-INFORMS International Meeting. Banff, Canada. May 2004.

J.P. Watson, "Reconsidering the Effectiveness of Simulated Annealing for Job-Shop Scheduling. Optimization Days. Montreal, Canada. May 2004.

A. Howe, "Trading Places: How to Schedule More in a Multiple Resource Over-Subscribed Scheduling Problem", at 14th International Conference on Automated Planning and Scheduling, Whistler, CA, June 4, 2004.

A. Howe, "Leap Before You Look: An Effective Strategy for Oversubscribed Scheduling" at Nineteenth National Conference on Artificial Intelligence, San Jose, CA, July 27, 2004.

A. Howe and D. Whitley, "Modeling Search Algorithm Performance in Two Scheduling Applications: AFSCN and JSP", AFOSR PI meeting, Pasadena, California, August 10, 2004.

L.D. Whitley, "Invited Tutorial: Evaluating Evolutionary Algorithms", Parallel Processing from Natural Conference. Birmingham, UK, September 19, 2004.

L. Barbulescu, "Identifying Key Factors in Solutions to Air Force Satellite Control Network Scheduling", oral presentation at INFORMS Conference, Denver, CO, October 26, 2004.

L. Barbulescu, "Why `<fill in your favorite algorithm>` is the `<best|worst>` for an Oversubscribed Scheduling Problem", seminar at Carnegie-Mellon University, Pittsburgh, PA, February 17, 2005.

L. Barbulescu, "Why `<fill in your favorite algorithm>` is the `<best|worst>` for an Oversubscribed Scheduling Problem", seminar at University of Colorado at Denver, March 1, 2005.

L.D. Whitley, "Invited Tutorial: An Introduction to Evolutionary Algorithms", Constraint Programing, Artificial Intelligence and Operations Research Conference. Prague, CZ, May 30, 2005.

L. Barbulescu, "Identifying Algorithm Performance Factors for an Oversubscribed Scheduling Problem", poster presentation at ICAPS 2005 Doctoral Consortium, Monterey, CA, June 8, 2005.

M. Roberts, "Modeling Local Search: A First Step Toward Understanding Hill-Climbing in Oversubscribed Scheduling", poster presentation at ICAPS 2005, Monterey, CA, June 8, 2005.

L.D. Whitley, "Invited Tutorial: No Free Lunch", Genetic and Evolutionary Computation Conference. Washington, DC. June 25, 2005.

L.D. Whitley, "Invited Tutorial: Practical Guidelines for Using Evolutionary Algorithms", IEEE Congress on Evolutionary Computation. Edinburgh, UK. September 2-5, 2005.

M. Roberts, "Random Walks and Neighborhood Bias in Oversubscribed Scheduling", oral presentation at MISTA 2005, July 19, 2005.

A. Howe and L.D. Whitley, "Advancing Air Force Scheduling through Modeling Problem Topologies", AFOSR PI meeting, St. Louis, MO, August 22, 2005.

A. Howe, "Modeling Search for Scheduling", invited talk at Ninth International Symposium on Artificial Intelligence and Mathematics, Ft. Lauderdale, FL, January 4, 2006.

L.D. Whitley, "Space Based Scheduling Applications", invited talk, Genetic and Evolutionary Computation Conference. Seattle, July 8-12, 2006.

L.D. Whitley, "Invited Tutorial: No Free Lunch", Genetic and Evolutionary Computation Conference. Seattle, July 8-12, 2005.

M. Rogers, "Looking for Shortcuts: Infeasible Search Analysis for Oversubscribed Scheduling Problems", International Conference on Automated Planning and Scheduling, June 8, 2006.

A. Sutton, "PSO and Multi-Funnel Landscapes", oral presentation, Genetic and Evolutionary Computation Conference. Seattle, July 8-12, 2006.

3.4.2 Consultative Functions at Laboratories and Agencies

Schriever AFB, AFSCN Schedulers meeting at Colorado State University in April 2003 to discuss our formulation of the AFSCN scheduling problem. Attendees were Brian Bayless and five of the schedulers.

DARPA, Coordinators Program Adele Howe reviewed proposals during August and September 2004 for Tom Wagner and attended PI Meetings February 22-24, 2005 and May 22-24, 2006.

Sandia National Laboratory Darrell Whitley visited Jean-Paul Watson in April 2004 to discuss models of stochastic hill climbing optimization methods. Sandia also provide summer support for a Ph.D. students in summer 2005 and summer 2006. Jean-Paul Watson visited Fort Collins in June 2006.

Air Mobility Command discussed airlift scheduling with Dave Merrill on August 23, 2005 and corresponded with Major David Van Veldhuizen in Fall 2005 about obtaining data.

3.4.3 Transitions

Analytical Graphics and OrbitLogic email and telephone exchanges over summer/fall 2003 in which we tested their solutions to AFSCN on our data sets and gave them feedback.

KinetX met Jonathan Murray on July 2, 2004 to discuss AFSCN analysis and demonstrate software and interface for the application.

ITT met members of the team developing the new SPACETRACK system for Eglin AFB five times between July 2004 and present. We described our analyses and algorithm development for AFSCN and are now working on SPACETRACK application.

Northrup Grumman Mission Systems discussed over-subscribed military scheduling applications in series of email and telephone conversations with Bill Courtney between May and October 2005.

3.5 Inventions or Patents

None.

3.6 Honors/Awards

Darrell Whitley was elected "Senior Fellow" of the International Society for Genetic and Evolutionary Computation. (1 of 10). July 2003.

"A Dynamic Model of Tabu Search for the Job-Shop Scheduling Problem" by J.P. Watson, L.D. Whitley, and A.E. Howe garnered the Best Paper award at the First Multidisciplinary International Conference on Scheduling in August 2003.

"Trading Places: How to Schedule More in a Multiple Resource Over-Subscribed Scheduling Problem" by L. Barbulescu, A.E. Howe, L.D. Whitley and M. Roberts was runner-up for best student paper award at the Int'l Conference on Automated Planning and Scheduling in May 2004.

A. Data Collected

AFSCN Data: We obtained new data directly from Schriever for 5 days from circa 2003. We also obtained an existing data set from AFIT for the AFSCN scheduling problem; these are smaller problems that date from circa.

SpaceTrack Data: We obtained tracking data, including orbits, radar cross section and request type, on approximately 6000 objects (from junk to satellites) for which radar tracking is necessary. The data was made available by ITT.

EOS Data: We worked with NASA to create a test set of EOS Data using a standard set of tools; it is our understand that this is also how NASA has been testing some of their own scheduling methods.

ROADEF Data: This data was obtained directly from the ROADEF competition.

JSP Data: We did most of our experiments with standard JSP benchmark, but we have also developed our own problem generator.

B. Software

We implemented state-of-the-art scheduling algorithms based on their descriptions in the literature; the re-implementations are in C++ to be run under the Linux operating system. The algorithms for JSP are:

Taillard's Tabu Search with N1 Move Operator [7] is a local search algorithm that uses a dynamic tabu tenure (a variable length list of recently visited states that should be avoided). The N1 operator is designed for the JSP and guarantees that a path exists from an arbitrary solution to some optimal solution.

TSAB Tabu Search with N5 Move Operator is a straightforward implementation of tabu search in conjunction with periodic re-intensification around previously encountered high quality solutions. N5 is generally a better move operator than N1 for the JSP, but it does not provide the path guarantee. Our version is a modified version of the original described in [12].

Simulated Annealing with N1 Move Operator is a stochastic, next descent local search algorithm in which the likelihood of accepting the improving move increases as search progresses. Our version is based on [15].

Simulated Annealing with N5 Move Operator is a variant of the original algorithm that uses the N5 move operator.

IJAR Iterated Jump And Redescend, algorithm developed based on JSP models. Several variants were also developed.

The algorithms for AFSCN are:

Random Sampling creating a random permutation for a schedule.

Local Search using the shift and swap operators over permutations.

Three Satellite Interchange heuristic for moving tasks around in a schedule that interchanges three tasks [9]. This was developed for an earlier solution to the AFSCN problem.

GreedyDP extension of a one machine heuristic based on [6].

Split Heuristic heuristic developed in house for favoring low altitude satellites before high altitude satellites.

Squeaky Wheel Optimization Our own version of a patented algorithm developed by CIRL that combines greedy initialization with multiple moves based on credit assignment.

ALLS The local search algorithm that makes multiple moves with the number diminishing over time.

We were able to obtain public domain versions of the code for the following algorithms:

Genitor A genetic algorithm developed by Darrell Whitley's lab.

The algorithms for ROADEF and EOS are:

TabuCL re-implementation of Cordeau and Laporte's Tabu search algorithm for satellite image scheduling

TabuBR variant on TabuCL that only searches in the boundary region.

We will make our software available on request.

Web Site Our project web site is available at `http://www.cs.colostate.edu/sched/`. From that site, you can access publications, problem instances and generators from the project.

References

[1] Andy Armacost, Tom DeLaCruz, and Lee J. Lehmkuhl. Af satellite control network scheduling using binary integer programming. presentation at INFORMS Annual Meeting, October 26 2004.

[2] Laura Barbulescu. *Analysis of Algorithm Performance for an Oversubscribed Scheduling Problem*. PhD thesis, Colorado State University, Fall 2005.

[3] J. Błażewicz, W. Domschke, and E. Pesch. The job shop scheduling problem: Conventional and new solution techniques. *European Journal of Operational Research*, 93(1):1–33, 1996.

[4] J. F. Cordeau and G. Laporte. Maximizing the value of an earth observation satellite orbit. Technical Report CRT-2003-27, Centre de recherche sur les transports, July 2003.

[5] Van-Dat Cung. ROADEF 2003: Results of the final stage (base X) of the challenge. http://www.prism.uvsq.fr/ vdc/ROADEF/CHALLENGES/2003/results030203_final.html.

[6] S. Dauzère-Pérès. Minimizing late jobs in the general one machine scheduling problem. *European Journal of Operational Research*, pages 131–142, 1995.

[7] Éric D. Taillard. Parallel taboo search technique for the jobshop scheduling problem. Technical Report ORWP 89/11, DMA, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 1989.

[8] Al Globus, James Crawford, Jason Lohn, and Anna Pryor. A comparison of techniques for scheduling earth observing satellites. In *Proceedings of the Sixteenth Innovative Applications of Artificial Intelligence Conference*, San Jose, CA, July 2004.

[9] T.D. Gooley. Automating the Satellite Range Scheduling Process. In *Masters Thesis*. Air Force Institute of Technology, 1993.

[10] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[11] E. Nowicki and C. Smutnicki. A fast taboo search algorithm for the job shop problem. *Management Science*, 42(6):797–813, 1996.

[12] E. Nowicki and C. Smutnicki. A fast tabu search algorithm for the permutation flow-shop problem. *European Journal of Operations Research*, 91:160–175, 1996.

[13] E. Nowicki and C. Smutnicki. New algorithm for the job shop problem. Technical report, Institute of Engineering Cybernetics, Wroclaw University of Technology, Wroclaw, Poland, 2003.

[14] É.D. Taillard. http://ina.eivd.ch/collaborateurs/etd/default.htm. January, 2004.

[15] P.J.M van Laarhoven, E.H.L. Aarts, and J.K. Lenstra. Job shop scheduling by simulated annealing. *Operations Research*, 40(1):113–125, 1992.

[16] Jean Paul Watson. *Empirical Modeling and Analysis of Local Search Algorithms for the Job-Shop Scheduling Problem.* PhD thesis, Colorado State University, Summer 2003.

[17] J.P. Watson, L.D. Whitley, and A.E. Howe. A dynamic model of tabu search for the job-shop scheduling problem. In *First Multidisciplinary International Conference on Scheduling (MISTA 2003)*, Nottingham, England, August 2003.