Spacetrack: Trading off Quality and Utilization in Oversubscribed Schedules

Andrew M. Sutton and **Adele Howe** and **L. Darrell Whitley**
Department of Computer Science
Colorado State University
Fort Collins, CO 80523
{sutton,howe,whitley}@cs.colostate.edu

Abstract

Many scheduling problems are posed as optimization problems where the goal is to find a feasible schedule that maximizes the utilization of some resource. In some domains it is also necessary to consider the quality of the resulting schedule. In most research these two quantities are independent. This paper introduces a real world problem in which radar tasks must be allocated to track objects in space. We explore the trade-off between off-line task resource utilization and a measure of task *quality* that correlates to whether tasks are actually successfully executed. We develop two general types of algorithms that differ in the way they reason about quality and explore the trade-off between high quality solutions and solutions with high resource utilization.

Introduction

Quality-based scheduling problems associate some measure of quality with each task in the schedule. In such problems, it may be desirable to trade-off the quality of scheduled tasks and the number of tasks that can be scheduled. For instance, Wang and Smith (2005) introduce an oversubscribed scheduling problem in which task durations can be condensed at the expense of a task quality measure to attain feasibility.

In this paper, we explore the trade-off between quality and utilization in static schedules created for an on-line scheduling problem. The *spacetrack scheduling problem* is to control resource allocation on a phased array radar to track objects in space; operators schedule tracking tasks to collect observations on near Earth and high orbit targets.

In this research, the *quality* of an observation is measured by a track's predicted signal to noise ratio (SNR): a function of the target object's location and orientation during the tracking task. The SNR approximately estimates the probability of target illumination. The *utilization* level of the off-line schedule measures how many tasks are feasibly accommodated given the resource limitations of the system.

There is a large degree of uncertainty regarding the position and visibility of an orbiting object. The quality of a task corresponds to the probability of detection. Therefore, *utilization* is a static measurement of the resource usage of the off-line schedule, while *quality* is a static surrogate metric for estimating the likelihood the tasks will have to be rescheduled later. We believe it is important to examine the interplay between quality and utilization in off-line schedules, and the performance of different algorithms on both measurements. This research is a first step toward designing a dynamic scheduling algorithm.

Problem Description

The Space Surveillance Network is an aggregation of 25 optical and radar tracking stations around the globe maintained by United States Space Command. A phased array radar is part of a population of devices used for sustaining a catalog of objects in orbit. Phased array radars are capable of steering the radar beam electronically using phase correlation between an array of geometrically positioned antennas. Such a configuration allows the radar to track many objects in virtual simultaneity. However, the number of objects that can be tracked at a given time is limited by the duty cycle of the radar.

Radar operators are given a list of prioritized targets to track during a 24 hour period and must compute a tracking schedule that efficiently allocates radar power to collect observations on each object. To scan multiple objects at once, phased array radars must interleave a number of pulses. The energy of a pulse is a function of its width in time; so an object with a higher range requires a longer pulse width to ensure illumination than a closer object. Since target range is time variant, the pulse width requirement for a track will change depending on where it occurs in the schedule. We are allowed to schedule several tracks at once provided that their interleaved pulses do not violate the feasibility of the schedule and the power limits of the system.

In addition to previously discussed factors, the SNR of a scheduled observation is a function of the target object's range and how it is situated with respect to the radar's field of view. Higher SNR levels correspond both to better expected observation quality and a greater probability that the target is illuminated by the radar.

Formally, let T be a set of tracking tasks that must execute using a renewable resource of finite power (i.e., capacity) P over a discrete time domain D. P models the duty cycle limits attributed to the radar system. Additionally, each task

$i \in T$ is associated with the following quantities:

$$\pi_i \in 1, 2, \ldots \quad (1)$$

$$w_i = \{(a_1, b_1), (a_2, b_2), \ldots, (a_{|w_i|}, b_{|w_i|})\} \quad (2)$$

$$d_i \in \mathbf{Z}^+ \quad (3)$$

$$o_i = \{\text{set of Keplerian orbital elements}\} \quad (4)$$

where (1) denotes the *priority* of task i, (2) is a set of visibility windows associated with the tracking task, and (3) is the duration in time units necessary to collect the required observations.

A set of Keplerian orbital elements o_i for each object (4) allows for the calculation of the object's position at any time $\tau \in D$. This allows us to compute the following functions:

$$r(o_i, \tau) \quad (5)$$

$$\theta(o_i, \tau) \quad (6)$$

$$snr(o_i, \tau) = \frac{rcs \cdot \cos^2 \theta(o_i, \tau) \cdot LG}{r(o_i, \tau)^4} \quad (7)$$

$$pw(o_i, \tau) = F(r(o_i, \tau)) \quad (8)$$

where (5) is the range of object o_i at time τ and (6) is the object's angle off the normal vector to the radar's face plane (known as the "boresight"). This is included to represent scan loss exhibited at high off-normal beam angles. These quantities are used to calculate the SNR (equation 7) associated with object o_i at time τ, and the pulse width (equation 8) required to illuminate it at its range. The rcs term is the radar cross section (scattering coefficient) and models the target object's profile with respect to the array. The LG term is *loop gain*: a constant defined by parameters of the specific radar system.

A solution to the problem specifies a subset of tasks $T' \subset T$ to be placed in the schedule and a mapping $s : T' \rightarrow D$ that assigns start times to tasks. A *feasible* solution is defined as follows. Let M_τ be the set of all tasks $i \in T'$ scheduled to be concurrently executing at time τ: this implies $s(i) \leq \tau \leq s(i) + d_i$ for all $\tau \in D$. Then,

$$\forall i \in T', \exists (a_j, b_j) \in w_i, a_j \leq s(i) \leq s(i) + d_i \leq b_j \quad (9)$$

$$\forall \tau \in D, \textstyle\sum_{i \in M_\tau} pw(o_i, \tau) \leq P \quad (10)$$

$$\forall \tau \in D, \forall i \in M_\tau, snr(o_i, \tau) \geq 11dB \quad (11)$$

Constraint (9) asserts the requirement that the task must execute during a visibility window, and (10) states that the pulse width needed by a set of concurrently executing tracks must not exceed the radar system's duty cycle limits P at any time. Lastly, (11) models the requirement that the SNR (in decibels) of a tracking task must be above an 11 dB threshold to obtain any return.

Algorithms

As a target moves through the visibility cone, its range and angle changes to produce a mound shaped signal to noise curve during the window of visibility. To maximize the SNR of tasks, a promising approach is to assign tracks to occur directly on the peak of this curve, when the SNR is highest during the visibility window. Empirically, using a maximal SNR "on-peak" scheduling strategy does not necessarily correspond to an optimal solution; if many peaks occur in concert, assigning tracks to peaks may produce duty cycle violations. Moving tracks off of their peaks or scheduling them to occur in a visibility window of lower average SNR might resolve resource contention. The general idea is to balance power usage to obtain a higher utilization without a great sacrifice in overall SNR of tasks.

We define two classes of algorithms that differ in the way they explore the solution space. We refer to the class of methods that only schedule tracks to occur on their SNR peaks as *on-peak* methods. These algorithms must choose a set of peaks in such a way to achieve high utilization. Since *on-peak* methods are constrained to a significantly small subset of the solution space, it may be possible to attain higher utilization if we allow tasks to be scheduled off of their SNR peaks. In this case, latent contention may be eliminated by expanding the set of allowable solutions.

This motivates the second class of algorithms which we shall call *off-peak*. These methods search the set of feasible start times: a significantly larger superset of the region allowed to *on-peak* algorithms. Because of this, *off-peak* methods are not as confined and can explore more feasible schedules, but solutions with high utilization may not be as densely distributed in this space.

In each algorithm class we apply three approaches: a greedy constructive framework, local search, and a genetic algorithm.

	on-peak	*off-peak*
Greedy	**bestWindow**	**priorityOffPeak**
Local	**onPeakLocal**	**offPeakLocal**
GA	**onPeakGA**	**offPeakGA**

For the *on-peak* methods, the **bestWindow** constructive approach operates by placing all time windows in a list sorted by descending peak SNR. While the list is not empty, it chooses the first window and attempts to schedule the corresponding task. If the task does not fit into the schedule in the corresponding time window, it deletes the window from the list; if the task has no windows remaining, it is effectively discarded from the schedule. Otherwise, it schedules the task and deletes the remaining time windows on the list associated with that task [1].

Both the **onPeakLocal** and the **onPeakGA** employ a schedule builder that takes an ordering of tasks and inserts each of them into their first available on-peak time in the schedule. The **onPeakLocal** method searches for task orderings by exploring neighborhoods induced by the shift operator while the **onPeakGA** evolves a population of insertion orderings.

The shift operator defines a neighborhood of a current task permutation p (an insertion ordering) as all $(|T| - 1)^2$ pairs (x, y) in p subject to $y \neq x - 1$. A neighbor of p parameterized by the position pair (x, y) is the permutation p' obtained by *shifting* the task at position x into position y leaving the relative order of all other tasks unchanged. Likewise, the *onPeakGA* maintains a population of task permutations

[1] We thank Ross M. McConnell for the idea behind the **bestWindow** heuristic.

and employs a schedule builder to translate each population member into a fitness value.

The **priorityOffPeak** constructive heuristic simply inserts tasks in priority order (ties are broken arbitrarily) to their first feasible position in the schedule. The **offPeakLocal** search first greedily places all tracks on their first available peaks and performs a neighborhood search by perturbing each start time by a Gaussian random variable ϵ. If a start time is pushed out of a window, it is moved to the next window. Similarly, the **offPeakGA** evolves a population of feasible time vectors. Each vector corresponds to a set of task start times in a schedule. The genetic algorithm employs mutation and recombination to drive a search for more promising schedule times. Note that the *off-peak* methods are priority biased: if a subset of tasks violate duty cycle constraints, those with lowest priority are bumped.

Experiment

Based on data obtained from the SENSOR Eglin SLEP program at ITT Industries, we constructed a model of the scheduling problem and generated 15 synthetic problems. To control the degree of resource constraint, we divided the problems into three sets of five with differing task cardinalities but equivalent power limits. The sets contained 1000, 2000, and 3000 tasks. For each task, we synthesized a corresponding target object by computing orbital elements through randomized generate and test. We calculated the range and boresight angle for a given object and time by using a Simple General Perturbations (SGP) orbit propagator (Hoots & Roehrich 1980) to compute position vectors from the generated elements. From these quantities we computed the SNR and pulse width using equations (7) and (8). We chose required durations for tracking tasks uniformly from 30-50 seconds. Each problem had a 24 hour time domain.

We assigned task priorities by drawing one of five categories from a distribution that models real data from the Space Control Center. To weight significance, we assigned a bump penalty of 50, 40, 30, 20, and 10 to each priority, from high to low respectively. We selected these values to induce a penalty proportional to the priority values used at Eglin, but also to produce penalties that lie in the same interval as the SNR values.

We used a loop gain of 150 dB which is incorporated into the SNR calculations (equation 7) as the LG term. We raise the required pulse width for a track by a power of 2 for every 1000 km of range. We imposed a $P = 256\ \mu$s pulse width limit to simulate duty cycle bounds and artifacts introduced by radar resolution.

To collect data that characterize the relationship between solution quality and schedule utilization, we ran each algorithm on each set and recorded the total SNR and penalty of tasks bumped. The searches optimize an objective function defined as the ratio of the total solution quality to priority weighted penalty of discarded tasks (inverse utilization):

$$\frac{\sum_{i \in T'} snr(o_i, \tau)}{\sum_{b \in T-T'} penalty(\pi_b)}, \text{ for } \tau = s(i), \ldots, s(i) + d_i$$

where $penalty(1)$ = 50, $penalty(2)$ = 40, ...,$penalty(5)$ = 10. We ran these searches for 20000 evaluations each and averaged their results over 10 runs.

The **onPeakLocal** search implemented a next descent strategy in permutation space over the neighborhood defined by the shift operator. The **offPeakLocal** search employed a Gaussian step size with a standard deviation of 10 seconds.

The **onPeakGA** used an order-based permutation crossover (Syswerda 1991). The **offPeakGA** employed HUX recombination and Gaussian mutation with a standard deviation of 10 seconds. The difference in crossover operator is due to the differences in representations (permutation gene representation versus integer gene). The Gaussian mutation was used in the **offPeakGA** to perturb task start positions in the same way the **offPeakLocal** does. Both genetic algorithms maintained a population size of 100 members and utilized the Genitor steady-state framework originated by Whitley (1989).

The quality and utilization results from the experiments are shown in Figure 1.

Discussion

The spacetrack scheduling problem poses some interesting problems to AI scheduling methods. The difficult quality function serves to confound the topology of the problem space and solutions with high quality and utilization measures are elusive. Good solutions will be found along a Pareto front. Specifically, there is an inherent trade-off between high SNR values and high utilization measures.

We conjecture that high-quality schedules, those whose tracking tasks must lie on or near SNR peaks, tend to have more rigidity than schedules that allow more diffusion of tasks in time. The pliability in lower quality schedules may serve to mitigate resource competition.

The **priorityOffPeak** heuristic consistently found the solutions that maximized utilization compared to other methods. Since its performance eclipses every *on-peak* method, there is evidence that allowing the scheduler to consider off peak insertion does admit higher utilization by allowing greater task dispersion in the schedule to smooth out contention peaks.

However, the relatively poor performance of the *off-peak* local and genetic algorithms raised a number of questions. The solution space explored by the *off-peak* GA and local search is significantly larger and less constrained than the *on-peak* search space, and promising solutions will tend to be much more sparsely distributed. This may cause *off-peak* search algorithms to spend most of their time in uninteresting regions of the problem space. On the other hand, the *on-peak* methods, though confined to a subset of accessible schedules, are able to move much more quickly through the space and find an acceptable interposition of quality and utilization.

It should also be noted that the representations used by the *on-peak* methods and the *off-peak* methods are very different. The *on-peak* representations and search operators have been shown to be effective in other related scheduling problems, and are more customized to fit known properties of the

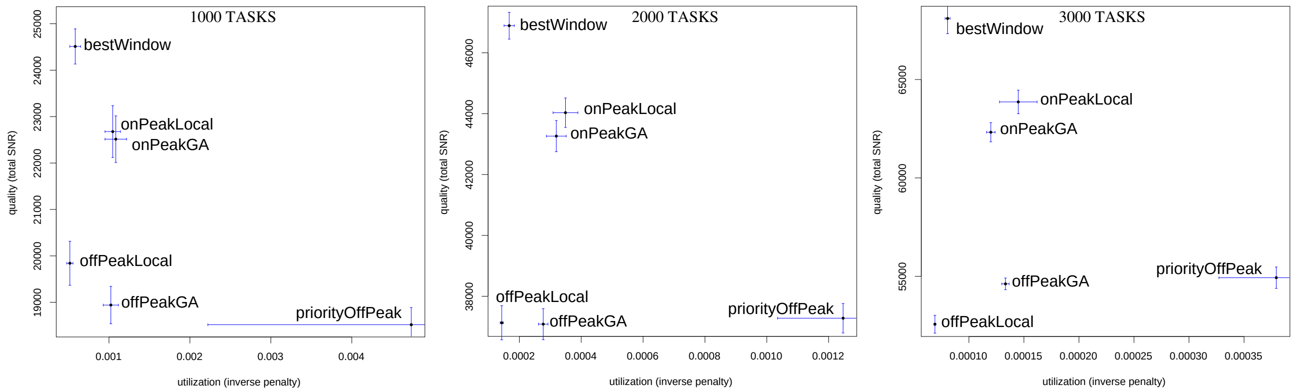

Figure 1: Solution quality vs. utilization for solutions found by algorithms on each problem set. Error bars denote standard deviation between problems.

search space. The *off-peak* methods however treat the problem as a black box parameter optimization problem, and thus use less problem specific information.

To maximize quality, **bestWindow** is consistently the top performer at the cost of reduced utilization. However, at high SNR values, tracking tasks are more likely to be successful; so under some circumstances, such a trade-off may be desirable. Conversely, if high utilization offsets the degradation in quality (e.g., there are more total successful observations under a high utilization strategy), simple *off-peak* heuristics such as **priorityOffPeak** could be sufficient. One potential problem is that this method does not pay enough attention to high priority tasks.

The crucial issue with the **priorityOffPeak** method is that such a decline in quality may in reality correspond to a potentially poor utilization if SNR levels become low enough to cause tracks to be missed. For example, in the case of the 1000 task set, if the quality degradation causes over 6% of the tasks to be missed, it will attain a true utilization that is no better than the solution discovered by the **bestWindow** heuristic.

A good trade-off between the two measures is found by the *on-peak* iterative search algorithms. These algorithms are inherently *quality* biased since they search the set of solutions that have all of their tasks scheduled on the highest possible SNR of their assigned time window. In our experiment, the local search and the GA were able to exploit information about which peaks yielded better utilization by moving around contention. Although both performed similarly, the **onPeakLocal** search dominates the **onPeakGA** when the number of tasks to be scheduled is 2000 or 3000. Furthermore, the *on-peak* searches are more mobile and explore a wider range of potential solutions compared to the other methods.

Conclusions

In this paper we have presented a new scheduling problem in which a balance of resource utilization and quality in a static schedule will affect the final on-line utilization of a dynamic schedule. We have explored the relationship between static utilization and quality as a first step toward designing a dynamic scheduler. We have found an empirical trade-off between the rigidity of high quality solutions and the plasticity of high utilization solutions.

Acknowledgments

This research was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-03-1-0233. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. L. Darrell Whitley was supported by the National Science Foundation under Grant No. IIS-0117209. Adele Howe was also supported by the National Science Foundation under Grant No. IIS-0138690. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We wish to thank Patrick O'Kane and Richard Frank at ITT industries for their generosity and assistance regarding the Eglin radar model.

References

Hoots, F. R., and Roehrich, R. L. 1980. Spacetrack report no. 3: Models for propagation of NORAD element sets. Technical report, Peterson Air Force Base.

Syswerda, G. 1991. Schedule optimization using genetic algorithms. In Davis, L., ed., *Handbook of genetic algorithms*. New York, NY: Van Nostrand Reinhold. 332–349.

Wang, X., and Smith, S. 2005. Retaining flexibility to maximize quality: When the scheduler has the right to decide activity durations. In *Proceedings of the 15th International Conference on Automated Planning and Scheduling, ICAPS'05*.

Whitley, D. 1989. The GENITOR algorithm and selection pressure. In *Third International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufman.