Proceedings of the 2000 Winter Simulation Conference
J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds.

COST/BENEFIT ANALYSIS OF INTERVAL JUMPING IN POWER-CONTROL SIMULATION

David M. Nicol

Department of Computer Science
Dartmouth College
Hanover, NH 03755, U.S.A.

L. Felipe Perrone

Department of Computer Science
College of William and Mary
Williamsburg, VA 23185, U.S.A.

ABSTRACT

Computation of power control calculations is one of the most time-consuming aspects of simulating wireless communication systems. These calculations are critical to understanding how a wireless network will perform, and so cannot be conveniently ignored. Power-control calculations implement solutions to discretized differential equations, and so are essentially time-stepped. In a previous paper (Perrone and Nicol, (1998)) we proposed a technique for *interval jumping*, that allows for substantially many time-steps to be jumped over, thereby reducing the amount of computation needed to achieve the same state as would straightforward time-stepping. The technique involves identification of a region of simulation time during which no channel assignments change due to limits on transmitter power, and a "jump" over that region. In this paper we examine the cost/benefit tradeoffs between policies which seek to minimize the work done to identify a jump interval, and the cost of computing those policies. We find that a tiered dynamic programming approach yields policies that very nearly minimize the searching overhead, while enjoying substantively lower computation costs than does the policy which strictly minimizes the searching overhead.

1 INTRODUCTION

Modern wireless communication systems adaptively compute the power level each transmitter uses. Since wireless communication is broadcast, the stronger the power, the farther the signal reaches, *everywhere*. However, the stronger the power, the larger is the potential for a given signal to interfere with other concurrent wireless communications. Adaptive systems therefore seek to adjust each transmitter's power to use what is needed to communicate effectively, and no more. Good treatments of these and other problems in wireless communication are found in Balston and Macario (1993), Stüber (1996), Webb (1998), and Asha (1994).

The power control laws we have studied were introduced and studied by Foschini and Miljanic (1993,1994,1995) and Foschini et al (1994a,1994b) in the context of schemes where each connection is assigned a small band of frequencies—called a channel—within which its communication occurs). Intuitively, each transmitter-receiver pair seeks to maintain a target quality of signal, measured in terms of the *signal-to-noise* ratio (or SNR) at the receiver. "Noise" is really signal degradation, and derives from a variety of sources (internal (thermal) noise, shadow-fading, distance, external interference). One we are most concerned with is external interference from other signals being broadcast concurrently (co-channel interference), either using this same frequency band (some distance away), or "spill-over" from adjacent frequency bands (adjacent-channel interference). The transmitter changes its power level every δ units of time, during which the SNR can be measured at the receiver. At the very beginning of a time-step, the transmitter learns the SNR observed during the last time-step, and adjusts its power to achieve the desired SNR, assuming that the signal degradation in the current time-step is equal to that in the last time-step. However, a transmitter has a maximum power level. If a sequence of a few time-steps go by in which the target SNR cannot be achieved owing to this power limitation, the wireless system seeks an alternative channel in the hope that the interference level will be lower there.

From this description it is clear that power control is a distributive adaptive dance, with each transmitter using the recent past as a guess of the near future in order to set its power level. Such networks work because the system is engineered so that power levels do not change rapidly. Noise due to interference changes in part because participants are mobile, so that those moving away from a tower require increasing levels of power. A radical change in the interference observed by some communications occurs when some channel allocation changes, a user either just begins to use a new channel, or a user ceases to use an existing channel. We call this a *critical event* in the simulation. Critical events arise either from the discrete portion

of the simulation (e.g., call arrivals/departures governed by stochastic sampling), or may be triggered by the continuous (time-stepped) portion of the simulation. The latter situation occurs when the power assignment is such that for a sufficiently long period of time, some communication pair cannot achieve their target SNR, and change the channel assignment, accordingly. Critical events caused by some transmitter reaching maximum power cannot easily be predicted, whereas critical events caused by scheduled call arrival/departures can be predicted, with pre-sampling of random number streams.

In our earlier work we noticed that Foschini's distributed power control law is a fixed point computation in a high dimensional vector space, in the following way. We can describe the assignment of all T transmitter powers at the i^{th} time step as a vector P_i, whose j^{th} component gives the transmitted power of the j^{th} transmitter in the system. We formally describe the collection of all contributions to signal degradation other than external interference (e.g., distance) as a vector of constants α. Then we formally describe the power update law as F_α, a function from $\mathcal{R}^T \rightarrow \mathcal{R}^T$, so that $P_{i+1} = F_\alpha(P_i)$. The fact that F_α for normal definitions of α is a fixed-point function means that for any two initial starting states S and S', repeated application of F_α pushes towards the same result $\mathcal{P}$. That is, if P_i is the i^{th} power vector computed starting with $P_0 = S$ and P'_i is the i^{th} power vector computed starting with $P_0 = S'$, then as i grows large then P_i and P'_i approach $\mathcal{P}$. It turns out that we can exploit this powerful property of the power update law to accelerate a simulation, by completely skipping over some time-steps.

The technique we describe is useful when the power levels used are not interesting in and of themselves, except to detect when critical events caused by power levels occur. For instance, our technique will not help a simulation that tracks power consumption in battery operated devices, because there the actual power used must be computed.

To see how the fixed-point property helps us, consider an interval $[a, b]$ of simulation time, discretized into $n+1$ points. We denote these points by defining $x_k = a + k(b-a)/n$, for $k = 0, 1, \ldots, n$. As time passes in the simulation, the environmental characteristics that contribute to signal degradation evolve; for each time x_i we let α_i denote the characteristics affecting the power control law at x_i, *except* for interference due to power levels. We assume that these α_i's can be computed in advance, if needed, as a function of predictable (pre-sampled) mobility. Now for any time-step x_k, consider two separate ways of computing the power vector P_k used at that time-step. Both methods start with the same known power vector P_0, at time-step x_0. The first method computes $P_1 = F_{\alpha_0}(P_0)$, $P_2 = F_{\alpha_1}(P_1)$, and so on until computing $P_k = F_{\alpha_{k-1}}(P_{k-1})$. The second method doesn't time-step at all, it iteratively applies $F_{\alpha_{k-1}}$ to compute the fixed point, i.e., $P'_1 = F_{\alpha_{k-1}}(P_0)$, $P'_2 = F_{\alpha_{k-1}}(P'_1)$, and so on until successively computed power vectors are numerically very much alike. Because we compute the power vector without time-stepping, we call this method an *interval jump*. Assuming that both methods compute nearly the same result, the latter one will be more efficient than the former one if it requires fewer than k iterations to achieve convergence. In practice convergence is achieved in a small number of iterations, say, 5 to 10. Our earlier work showed the very significant performance gains that are possible using interval jumping.

If interval $[a, b]$ is chosen so that the state of the power vector at time a is known, and time b is the earliest next time of a known call arrival/departure, then we hope to be able to compute P_n efficiently with an interval jump. However, we must be concerned with the possibility that the power vector induces a critical event somewhere within $[a, b]$, for we cannot jump over a critical event. If interval jumping is to work, we must efficiently determine if a critical event occurs in $[a, b]$, and if so, where.

To address this problem we exploit one further property of the power control law—if a source of signal degradation increases the level of degradation, (e.g., the mobile moves farther away from the base station) then the transmitter correspondingly increases the power level. What this means in practice is that even if degradation vector α_k is costly to compute, if we can easily compute a degradation vector α'_k that dominates α_k (in the sense that every component of noise is at least as bad as in α_k), then every component of $F_{\alpha'_k}(P'_i)$ will be at least as large $F_{\alpha_k}(P_i)$, as we iterate towards their respective fixed point values. What *this* means is that if none of the power vectors encountered computing towards the fixed point of $F_{\alpha'_k}$ have any component that exceeds the allowed power limit, than none of the power vectors encountered computing towards the fixed point of F_{α_k} do either.

The action of starting with state P_0 and computing the fixed point of $F_{\alpha'_k}$ is called a *probe*. During every iteration of a probe we update every power level and compute anew every SNR at every receiver. With T transmitters, each capable of interfering with each other, $O(T)$ computation is needed to compute the interference at a single receiver; each iteration of probe costs $O(T^2)$ time. Each iteration we check each new power level for violation of its maximal level. In the the case that any power level achieves this level, the probe ends immediately and is said to *fail*. If otherwise the probe *succeeds*, we know that we can interval jump from a to x_k safely, because no power-law induced critical event can occur in interval $[a, x_k]$.

The problem we consider in this paper is now accessible: given interval $[a, b]$, to devise an effective strategy for scheduling probes so as to find the latest point x_k for which the probe fails. We can then interval jump from a to x_{k-1}. Note that a failed probe at time x_k does not imply the existence of a critical event at that time, only the possibility

of one. Our overall strategy is to follow the following sequence:

1. If the correct power vector is known at time a, find the earliest next scheduled critical event (or a lower bound on it), say at time b,
2. Apply probes in $[a, b]$ to determine the earliest point x_k, for which the probe fails;
3. Compute the actual degradation vector α_{k-1}, and compute the fixed point of $F_{\alpha_{k-1}}$ as the power vector at time x_{k-1};
4. Time-step through x_k and up to K time-steps beyond that searching for a power-law induced critical event. If one occurs, simulate it, and return to step 1. If one does not occur in K steps, return to step 1. In both cases the new value of a is the new time associated with the last known power vector.

Before considering the problem of effective probe scheduling we should note that we have skimmed over an important issue, the construction of fictitious systems that give rise to artificially poor degradation vectors α'_k. There are tradeoffs here left to explore. A very tight but computationally expensive method may increase the reach of a probe into $[a, b]$, but cost more than the extra probes required by a looser degradation vector. Consideration of these trade-offs is beyond the scope of this paper.

2 PROBE SCHEDULING

The high cost of applying a probe motivates us to find a jump interval, while minimizing the computation expended on probing. In effect, we need a probe scheduling policy. The probe scheduling problem we consider is formulated as follows. We are given interval $[a, b]$, discretized as described before into $n+1$ points $\{x_i\}$. The smallest k for which the probe fails at x_k is modeled as a random variable S; if no probe fails over $[a, b]$ we define $S = \infty$. The distribution of S is allowed to be general. The execution cost of probing x_k is defined to be a function ϕ whose argument is the number of time-steps between x_k and the largest known time-step which has already passed the probe (and whose state serves as the starting state for the probe at x_k). The dependence on this difference supports the possibility that further temporal separation between x_k and the point of the last known power-vector may require more iterations from the fixed-point algorithm. The probe scheduling problem is to find a schedule of probes that minimizes the expected execution cost of performing the search.

It is not difficult to see that this problem can be approached using dynamic programming. The *state* of the search is described by a pair (i, j), where x_i is the largest time-step known to have passed the probe (and x_0 by definition is known to pass the probe), and x_j is the smallest time-step beyond which we will not consider probing (with $x_n = b$ initially). Thus the initial state of the search is $(0, n)$. From any search state (i, j) (with $i < j$) a probe point is selected, say at x_k, and the probe is applied. If it fails the new state is (i, k); if it succeeds the new state is (k, j) (although the special cases of $i = j+1$ or $k = j$ are slightly different). The probability of the probe failing at k from state (i, j) is denoted q^k_{ij}—this can be computed from first principles given the distribution function of S and need not be explained here. The execution cost of probing at k from state (i, j) is denoted $\phi(i, k)$. Cost function $C^k(i, j)$ gives the minimal cost of the optimal schedule, starting in state (i, j) and given a probe at k; $C(i, j) = \min_k\{C^k(i, j)\}$ is the minimal cost of the optimal schedule starting in state (i, j). The structure of $C^k(i, j)$ is given by

$$C^k(i,j) = \begin{cases} 0, & \text{for } i = j \\ \phi(i,k), & \text{for } k = j,\ j-i = 1 \\ \phi(i,k) + q^k_{ij}\, C(i,k-1), & \text{for } k = j,\ j-i > 1 \\ \phi(i,k) + q^k_{ij}\, C(i,k) + (1-q^k_{ij})\, C(k,j), & \text{for } k < j,\ j-i > 1 \end{cases}$$

The first case says that when $i = j$ we need not probe since (i, i) is a terminal state. The second case says that from state $(i, i+1)$ we have only to probe at x_{i+1} and then enter a terminal state. The third case says that from state (i, k), probing at k either succeeds and puts us in terminal state (k, k), or fails, leaving us with interval $(i, k-1)$ left to search. The last case considers the two possibilities that occur when (i, j) has more that two points, and an interior point is probed.

Brute force solution of this equation requires $O(n^3)$ time and $O(n^2)$ space, recalling that n is the number of equidistant points covering $[a, b]$. Usual techniques for reducing this cost rely on special properties of the functions involved, such as ϕ and q^k_{ij}. In our application we have no particular hope for nice structure for these functions and are therefore relegated to the straightforward solution. This is the root cause of the problem we consider in this paper.

Power control updates occur at a time-scale of milliseconds, whereas call arrival/departures occur at a time-scale of minutes. For small sized update increments then, the number of time-steps to the next scheduled critical event (n) may be large. This fact makes solution of the DP equation above entirely impractical. Another difficulty is the estimation of the probability distribution of S. We are presently researching the latter problem and will not discuss it further except to say that the experiments we describe here consider a variety of forms that distribution might take.

The scheduling strategy we propose is based on the observation that reducing n by half reduces the computational work of solving the DP equation by a factor of eight. By solving a series of small DP problems, we can find the same earliest critical point that the optimal schedule finds, with a slightly larger probe execution cost, but with a radically reduced schedule computation cost—so much so that computing the schedule dynamically is feasible. Our goal is to be able to compute and use a probe scheduling policy on-the-fly, for every interval we seek to jump. To do this the computation cost of finding the policy must be small.

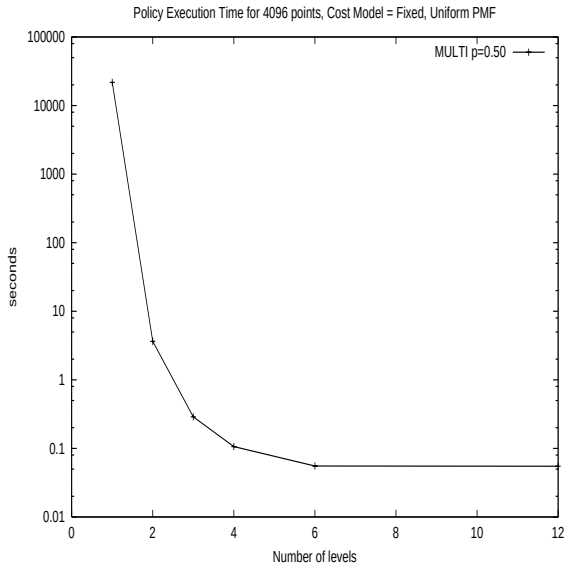
For simplicity we work with powers-of-two numbers of points and assume the original problem has $n = 2^L + 1$ points, and let d be any integer divisor of L. Rather than discretize $[a, b]$ with 2^L+1 points, δ apart, discretize it more coarsely with $2^d + 1$ points, $2^{L-d}\delta$ apart. We construct and solve the smaller DP problem, and use the optimal probe schedule to identify the earliest point on the coarse grid that fails the probe, say at j_0. On the fine grid let x_S be the earliest point where the probe fails; we have then that $i_0 < x_S \leq j_0$, where i_0 is the the point that immediately precedes j_0 in the coarse grid. Solution and application of the coarse-grained problem yields for us an interval that contains x_S. We then discretize *this* interval with $2^d + 1$ points that are $2^{L-2d}\delta$ apart, and repeat the process, finding an interval $[i_1, j_1]$ that contains x_S. Continuing in this fashion we solve L/d smaller DP problems in such a way that we are guaranteed to find the same point x_S that the optimal search would have done. We trade one expensive DP computation on 2^L+1 points for L/d DP computations, each on $2^d + 1$ points.

It is worth noting that this method actually describes a family of solution strategies, depending on d. At one extreme—when $d = L$—there is only one level and so the solution method is simply computation of the optimal strategy. At the other extreme—when $d = 1$—each level has only three points (two endpoints and a middle), which leads to a policy that is close to (but not identical to) binary search. Between these extremes we have a spectrum of policies whose execution costs increase with increasing d, as does performance measured in terms of mean probing cost needed to find x_S. The remainder of the paper considers this tradeoff, looking at various models of critical point placement probability and different probe costs.

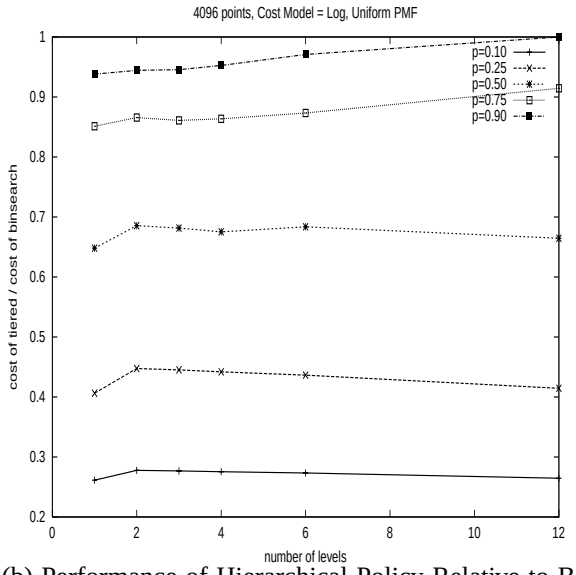
3 EXPERIMENTS

We now report on experiments conducted to evaluate the feasibility of using a tiered search-strategy solution approach dynamically. Our experiments consider two different models of probe cost $\phi(i, j)$: constant ($\phi(i, j) = c$ for all $i < j$), and logarithmic ($\phi(i, j) = \alpha \log(j - i + 1)$ for all $i < j$)). The logarithmic model is included to allow for qualitative variations due to larger probe costs when computing a fixed point starting with degradation conditions that are significantly unlike those experienced at the last known fixed-point (at x_i). We also consider two families of stochastic models for placement of x_S. One family is the conditional uniform—with probability p we have $a < S \leq b$, and conditioned on this event the distribution over $[a, b]$ is uniform. The second family is defined in terms of an linearly increasing hazard rate function over $[a, b]$. The slope of the hazard rate function is the parameter; a slope may be steep enough to ensure that $a < S \leq b$, or may be shallow enough to allow $S > b$. We will characterize different members of this family by the probability of $a < S \leq b$—mapping this characterization to the appropriate hazard rate function is a straightforward technical exercise.

We first look at the cost of computing different probe scheduling policies. The experiment was run on an SGI Origin 2000 with a 180MHz clock. This experiment measured the cost of a priori calculation of the hierarchical policy, which is even more costly than a dynamic version that would calculate the policy only over those intervals that the search process shows must be considered. Figure 1(a) plots the policy computation execution time as a function of the number of levels (L/d), on a problem with $n = 4097$ points on the fine grid. A conditional uniform distribution with $p = 0.5$ models S. We see a huge difference, between a six hour running time for the optimal policy calculation and a 50 millisecond running time for the policy that discretizes each interval into three points. The computational savings here are enormous. The question is how good the quickly computed policy is relative to optimal, and relative to the easy and intuitive binary search policy. Figure 1(b) plots the ratio of the mean tiered policy probe cost to that of binary search (applied to the fine grid). Several lines are plotted, for various probabilities of finding a critical event in the interval. The plots shown assume that the probe cost increases logarithmically in the jump distance and that the critical event distribution is conditional uniform. We have studied variations on these assumptions and all the results are quantitatively similar to those shown. Two basic messages come out of this graph. One is that relative performance is insensitive to the number of levels employed, and the other is that relative performance depends very much on the likelihood p of finding a critical event in the interval. The basic difference between binary search and the tiered policy is that when p is low, the tiered policy has knowledge of this and can encourage probes at the right end of the interval. A successful probe near b in interval $[a, b]$ quickly leads to a terminal condition, in fewer steps than required by the binary search. However, if a critical event does appear in $[a, b]$, then the tiered policy will have to do much of the same honing in on it that the binary search does, and so the costs become more similar. The difference works to our advantage, for wireless systems are engineered to minimize



(a) Execution Cost of Computing Tiered Search Policy



(b) Performance of Hierarchical Policy Relative to Binary Search

Figure 1: Policy Calculation Cost, and Performance Relative to Binary Search

unscheduled channel re-assignments, which are precisely the critical events we are concerned about encountering in $[a, b]$. In the common case then, the added effort of managing the tiered policy offers substantial reduction in probe costs over binary search.

We next compare the performance of tiered policy schedules to optimum. The graphs in Figure 2 work through the four combinations of probe cost and critical event placement distribution possible under our experimental assumptions. Each line is plotted as a function of the probability that the interval contains a critical event; the dependent variable is the ratio of the mean probe cost of the tiered policy to the mean probe cost of the optimal policy. Two interesting things emerge from all of these graphs. First and foremost, it is evident that the performance degradation due to using a tiered policy rather than optimal is almost always less than 10% in these experiments, usually much less. Given the significant performance advantage over binary search in the common case, this observation demonstrates the utility of the proposed approach. The second interesting feature of this data is its shape. When the probability of seeing a critical event is low, the policies with more levels (hence fewer points involved in a policy calculation) perform better than policies with fewer levels (with the exception of course of the 1 level policy, which is optimal). We have noted the reason for this behavior already—on a coarse grid there is strong encouragement to probe the rightmost point, because if it passes the probe we immediately enter a terminal state. The policies with more levels are more likely to enter a terminal state before descending all the way to the bottom level. However, if the probability of encountering a critical event in the interval is large, then the policies that have more points in a subinterval are more efficient at honing in the point. Between these two extremes the relative performance of these policies "cross over" at various points.

As a final note on this data, we see that the variations in our assumptions about probe cost and distribution of the critical point placement have only a second order effect on relative performance. This is encouraging, because we are looking only at models of these facets anyway. Relative insensitivity between our assumed models suggests that whatever the real behavior might be, it too will be relatively insensitive in these same ways (provided of course that it is not terribly different from our assumptions).

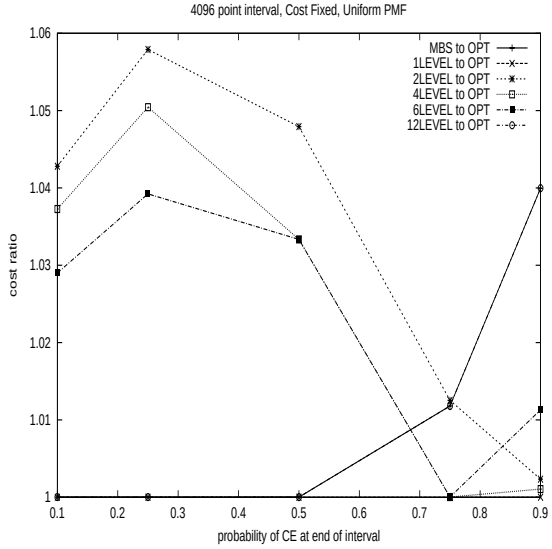
4 FUTURE WORK

The study reported here is theoretical, in the sense that it is concerned entirely with models of a wireless communication network behavior. We are working to incorporate interval jumping and probe scheduling into a wireless simulator and evaluate these policies in that more real context.

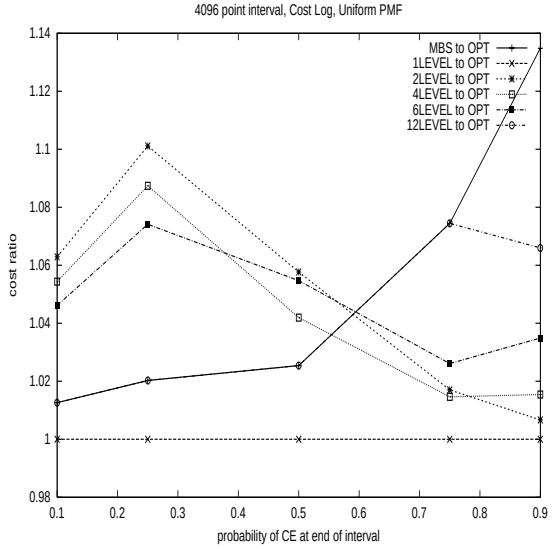
A significant piece of the problem is to determine how to model the probability distribution of the critical event placement. Our results show that the relative utility and performance of the tiered policy is sensitive to the probability of finding an critical event in the interval. It will likely be necessary that in a deployed system we be able to estimate this probability from run-time observations.

5 CONCLUSIONS

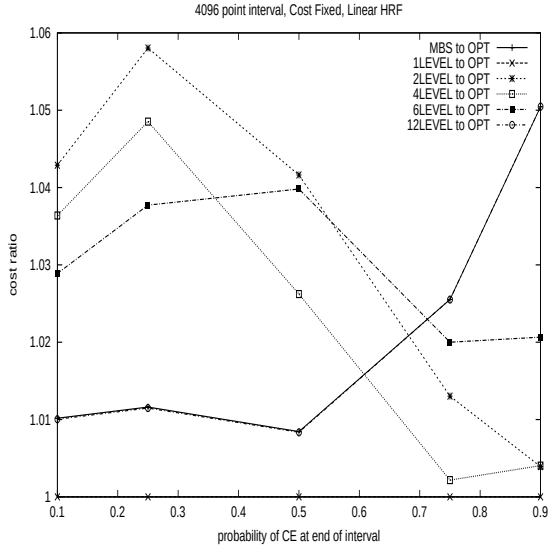
Interval jumping is a technique we've developed to accelerate the simulation of a class of frequency-multiplexed wireless



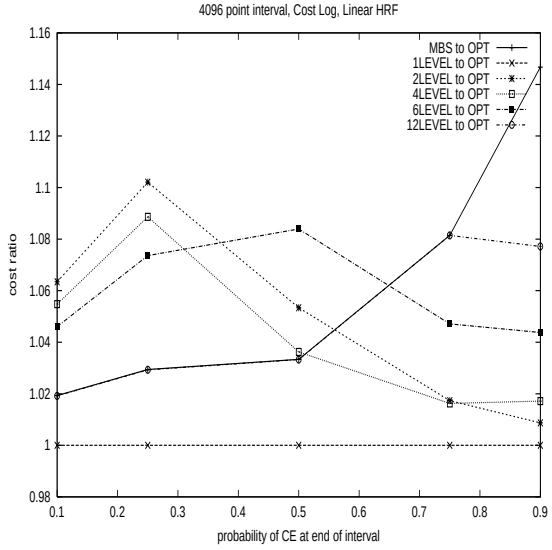
(a) Constant Probe Cost, Condition Uniform Critical Event Distribution



(b) Logarithmic Probe Cost, Condition Uniform Critical Event Distribution



(c) Constant Probe Cost, Increasing Hazard Rate Critical Event Distribution



(d) Logarithmic Probe Cost, Increasing Hazard Rate Critical Event Distribution

Figure 2: Performance Relative to Optimal

communication systems. A critical problem is to compute a schedule of probes that reveal how large an interval the simulation may jump. This paper proposes and studies a family of policies that solve a sequence of optimization problems. We observe that the computational cost of this method is not large, that the schedules it produces are in common circumstances significantly better than the naive binary search technique, but are not significantly worse than the optimal schedule. Collectively this suggests that the method is valuable, provided that in practice we can construct stochastic models of model behavior that are similar to those assumed by our policy.

ACKNOWLEDGMENTS

This research was supported in part by NSF grants ANI-98-08964, EIA-98-02068 and DARPA contract N66001-96-C-8530.

REFERENCES

Asha, Mehrotra. 1994. *Cellular radio*. Norwood, Massachussets: Artech House.

Balston, D.M. and R.C.V. Macario. 1993. *Cellular radio systems*. Norwood, Massachussets: Artech House.

Foschini, Gerard J. and Zoran Miljanic. Distributed autonomous wireless channel assignment algorithm with power control. *Internal AT&T Bell Laboratories Document*.

Foschini, Gerard J. and Zoran Miljanic. 1993. A simple distributed autonomous power control algorithms and its convergence. *IEEE Transactions on Vehicular Technology*, 40(4):641–646.

Foschini, Gerard J. and Zoran Miljanic. 1994. Channel cost of mobility. *IEEE Transactions on Vehicular Technology*, 42(4):414–424.

Foschini, Gerard J. and Zoran Miljanic. 1995. Distributed autonomous wireless channel assignment algorithm with power control. *IEEE Transactions on Vehicular Technology*, 44(3):420–429.

Cimini, Leonard J., Jr., Gerard J. Foschini, Chih-Lin I, and Zoran Miljanic. 1994. Call blocking performance of distributed algorithms for dynamic channel allocation in microcells. *IEEE Transactions on Communications*, 42(8):2600–2607.

Cimini, Leonard J., Jr., Gerard J. Foschini, and Larry A. Shepp. 1994. Single-channel user-capacity calculations for self-organizing cellular systems. *IEEE Transactions on Communications*, 42(12):3137–3143.

Perrone, L. Felipe and David M. Nicol. 1998. Rapid simulations of wireless networks. In *Proceedings of the 12th Workshop on Parallel and Distributed Simulation (PADS '98)*, 170–177.

Stüber, Gordon L. 1996. *Principles of mobile communication*. Norwell, Massachusetts: Kluwer Academic Publishers.

Webb, William. 1998. *Understanding cellular radio*. Norwood, Massachussets: Artech House, Inc.

AUTHOR BIOGRAPHIES

DAVID M. NICOL is Professor and Chair of the Department of Computer Science at Dartmouth College. He received the B.A. in mathematics from Carleton College in 1979, and the Ph.D. in computer science from the University of Virginia in 1985. He has since then contributed to the field of simulation, particularly in the area of parallel discrete-event simulation. He is Editor-in-Chief of the ACM Transactions on Modeling and Computer Simulation, and is co-author (with Banks, Carson, and Nelson) of the text *Discrete Event System Simulation, 3rd Edition*. His current research interests are focused on simulation of large-scale networks, and network security.

L. FELIPE PERRONE received a bachelor's degree in electrical engineering from the Federal University of Rio de Janeiro in 1989, and the M.Sc. degree in systems engineering from the same institution in 1992. He joined the Ph.D. program at the College of William and Mary in 1992. His dissertation is on methods of accelerating wireless communication network simulations.