# Hypermedia and Randomized Algorithms for Medical Expert Systems

R. Martin Chavez

Section on Medical Informatics, MSOB-X215
Stanford University School of Medicine
Stanford, California 94305

## Abstract

KNET is an environment for constructing probabilistic, knowledge-intensive systems within the axiomatic framework of decision theory. The KNET architecture defines a complete separation between the hypermedia user interface on the one hand, and the representation and management of expert opinion on the other. KNET offers a choice of algorithms for probabilistic inference. My coworkers and I have used KNET to build consultation systems for lymph-node pathology, bone-marrow transplantation therapy, clinical epidemiology, and alarm management in the intensive-care unit.

Most important, KNET contains a randomized approximation scheme (ras) for the difficult and almost certainly intractable problem of Bayesian inference. My algorithm can, in many circumstances, perform efficient approximate inference in large and richly interconnected models of medical diagnosis. In this article, I describe the architecture of KNET, construct a randomized algorithm for probabilistic inference, and analyze the algorithm's performance. Finally, I characterize my algorithm's empiric behavior and explore its potential for parallel speedups. From design to implementation, then, KNET demonstrates the crucial interaction between theoretical computer science and medical informatics.

## Introduction

Within the discipline of medical informatics, many researchers have studied methodologies for encoding the knowledge of expert clinicians as computational artifacts. KNET is a general-purpose environment for constructing probabilistic, knowledge-intensive systems based on belief networks and decision networks [15]. Such networks serve as graphical representations for decision models, in which the knowledge engineer must define clearly the alternatives, states, preferences, and relationships that constitute a decision basis. KNET differs from other tools for expert-system construction in that it combines a direct-manipulation visual interface with a normative, probabilistic scheme for the management of uncertain information and inference. The KNET architecture defines a complete separation between the hypermedia user interface on the one hand, and the representation and management of expert opinion on the other.

My coworkers and I have used KNET to build consultation systems for lymph-node pathology [10,4], bone-marrow transplantation therapy, clinical epidemiology [13], and alarm management in the intensive-care unit (ICU) [1]. KNET imposes few restrictions on the interface design. Indeed, we have rapidly prototyped several direct-manipulation interfaces that use graphics, buttons, menus, text, and icons to organize the display of static and inferred knowledge. The underlying normative representation of knowledge remains constant.

Early workers in the field of medical informatics observed that, in the absence of strong and often unrealistic simplifying assumptions, full Bayesian inference requires computation time exponential in the number of diagnostic variables [3]. With few exceptions, the early Bayesian information-processing systems never found widespread clinical acceptance. The difficulty of building Bayesian models, and the complexity of probabilistic inference with the algorithms and hardware available at the time, severely limited the usefulness of the approach.

Traditional rule-based expert systems, which appeared in the early 1970s, circumvented the computational intractability of exact probabilistic reasoning by proposing ad hoc schemes for managing uncertainty. Those schemes conflict with the classic theory of probability and draw erroneous conclusions in many circumstances. More recently, the emergence of powerful graphics, personal workstations, and inexpensive hardware demands a reconsideration of formalisms that seemed intractable 20 years ago.

Once again embracing probability theory as the epistemological groundwork of expert reasoning, I present a new algorithm, based on previous work in the Monte Carlo simulation of belief networks [16], that performs efficient approximate probabilistic inference. My algorithm, unlike the heuristic approaches to expert reasoning, specifies precise a priori bounds that describe the accuracy of its results as a function of computing time. The algorithm, which can perform probabilistic inference in very large KNET models, combines Monte Carlo simulation, area-estimation strategies, and convergence analysis for time-reversible Markov chains.

For arbitrarily complex diagnostic problems, my randomized approximation scheme currently offers the only computationally tractable methodology for probabilistic inference with a priori guarantees of convergence. In addition, the algorithm lends itself to full parallelization: For a given level of accuracy, $n$ parallel processors, each connected to a central node, can compute a probabilistic inference in $1/n$ of the time required by a single sequential machine. The ras is not a panacea, however; for inference problems with deterministic or nearly deterministic subgroups, the approximate scheme's performance may still degrade exponentially. Even so, the ras extends the applicability of Bayesian techniques to large knowledge-intensive models once deemed intractable.

KNET proves the feasibility of knowledge-intensive probabilistic systems that run on low-cost hardware and that fully develop the ergonomics of modern graphics displays. In this paper, I briefly describe the architecture of KNET, and emphasize the complete separation of *knowledge representation* from *inference* and *user interface*. I describe an algorithm for efficient approximate probabilistic inference, and sketch a proof of its computational complexity. (The details appear in [5].) I analyze the performance of my algorithm on DxNet, an 81-node belief network that partially captures an expert's knowledge of ventilation management in the ICU. Finally, I describe the application of my algorithm to QMR-DT, the decision-theoretic reformulation of QMR, a classic medical artificial-intelligence (AI) system for diagnosis in internal medicine [14].

## Methods and Procedures

The KNET project grew out of my interest in the discipline of medical informatics. Broadly defined, medical informatics addresses the problems of information storage, management, and utilization in health care. The complexity and heterogeneity of the domain pose specific challenges that lie beyond the purview of traditional computer science. Those challenges demand new solutions that integrate clinical practice, knowledge engineering, and decision support.



0195-4210/89/0000/0171$01.00 © 1989 SCAMC, Inc.

## Previous Work

KNET has its foundation in a decade of research on expert systems. In addition, KNET adheres to rigorous principles of modularity and abstraction in software engineering. More important, KNET unifies two parallel lines of research: automated reasoning and human-machine interfaces. I now place KNET in the context of previous work on knowledge engineering and interface design by briefly comparing and contrasting it with existing systems.

Designers of previous expert-system shells have attempted to accomodate all the requirements of an interface designer. EMYCIN, one of the earliest shells, offers one kind of interface: a linear list of questions, formulated by the computer and answered by the human user [3]. Most knowledge engineers find that they must carefully craft interlocking rules with multiple premise clauses in order to control the line of questioning. EMYCIN therefore enforces a tight coupling between the knowledge representation and the human-machine interface. KNET, on the other hand, consistently and clearly separates interface from representation and inference.

The Nexpert shell is a reasoning and representation system with a direct-manipulation graphical interface. It contains a rule-based inference engine, an object-oriented frame system, and an interface to the C programming language. Nexpert supports bidirectional rule-based reasoning, from evidence to hypotheses and from goals to acquired data. Nexpert, however, offers no calculus for computing with uncertain, incomplete, and contradictory evidence. Its inference engine reasons categorically. KNET, on the other hand, replaces Nexpert's tree of categorical rules with a normative belief network that consistently represents probabilistic information. KNET makes no distinction between forward and backward chaining; rather, the user offers evidence and KNET updates all hypotheses accordingly.

RACHEL, the first instance of an *intelligent decision system*, codifies a large portion of what is commonly known as the art of decision analysis [9]. RACHEL merges rule-based expert systems for capturing the expertise of professional decision analysts, with decision-theoretic domain models. Intelligent decision systems hold three distinct advantages over more traditional expert systems: normative power, ease of representation and use of uncertainty, and clarity in the acquisition of knowledge. Although KNET and RACHEL share the underlying, unifying technology of decision networks, they address distinct aspects of the knowledge-representation problem. Whereas RACHEL encodes the hypothetico-deductive processes of decision analysts in a rule-based framework, KNET supports the presentation graphics and tractable inference that ultimately make intelligent decision systems acceptable to the user.

Specifically, the knowledge kernel of KNET contains an ras that can, in many cases, perform approximate probabilistic inference on very large networks. By definition, an ras computes approximate answers that, with probability greater than $1 - \delta$, differ from the true answer by a relative error of no more than $\epsilon$; in addition, the ras requires computing time that is a polynomial in $1/\epsilon$, $1/\log\delta$, and the size of the input. RACHEL, on the other hand, rolls back the decision tree with a brute-force algorithm equivalent to calculation of the full joint distribution [17]. RACHEL, therefore, must use extensive deterministic senstivity analysis (as encoded by the rule-based component) to reduce the decision model to a small and manageable size by exploiting the particular characteristics of the current context.

## The KNET Architecture

To investigate belief networks as a knowledge representation for difficult problems in medical diagnosis, I have designed and implemented the KNET environment for building probabilistic, knowledge-intensive expert systems [4,13]. Such systems

- Manage large quantities of extensively cross-referenced information
- Emphasize clarity in acquiring, storing, and displaying expert knowledge

**The KNET Architecture**

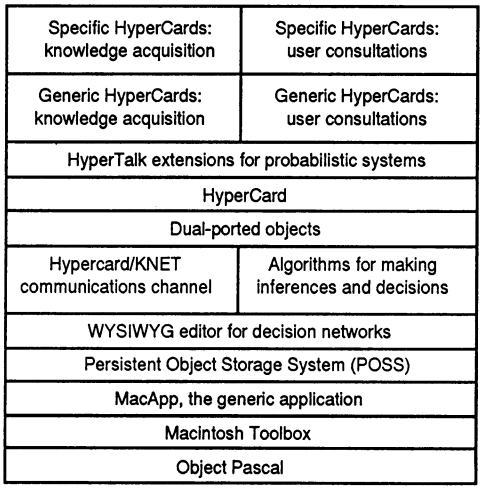



Figure 1: This figure defines the virtual machines that together constitute the KNET architecture. At each horizontal boundary, KNET defines a protocol for the flow of information across the boundary. Virtual machines communicate exclusively by passing messages across the interfaces with their immediate neighbors.

- Incorporate tools for building hypertext user interfaces
- Make normatively correct decisions and diagnoses in the face of uncertain, incomplete, and contradictory information
- Draw inferences from knowledge bases large enough to model significant, real-world medical domains, and do so in polynomial time on low-cost hardware

The challenges of medicine have guided this work toward its present form. The disciplines of AI, classical computer science, and decision analysis provide the theoretical framework and the arsenal of applied techniques for building knowledge-intensive systems that help to manage information in medical care.

The KNET architecture integrates HyperCard, a hypertext authoring tool, with a suite of knowledge-engineering modules written in Object Pascal. The design of KNET places a premium on the precise separation of user interface from knowledge-engineering kernel. KNET requires that the designer represent his domain as a diagnosis problem (with belief networks) or as a decision problem (with decision networks). KNET provides a WYSIWYG (What You See Is What You Get) module for drawing networks on a high-resolution bitmapped display. Those networks, in turn, precisely and intuitively capture the causal-associational relationships of Bayesian diagnostic models.

Figure 1 defines the KNET architecture precisely. Each layer in the figure represents a software-engineering abstraction known as a virtual machine. At each interface between a virtual machine at level $i$ and a machine at level $i + 1$, KNET defines a communications protocol for the bidirectional flow of information across the boundary. Virtual machines never inspect the hidden variables or call the hidden methods of machines at lower levels in the hierarchy; rather, they access the services of inferior machines by passing messages across the boundary. The virtual-machine abstraction facilitates the design, development, and maintenance of large software systems.

The entire KNET environment runs on low-cost, general-purpose hardware. Using HyperTalk, HyperCard's object-oriented authoring language, knowledge engineers and even relatively naive users can incorporate sound, synthesized speech, videodisc images, and animation in their knowledge-intensive applications. My experience, corroborated by my coworkers [4], indicates that we can prototype, debug, and refine different hypertext presentations of the knowledge captured by a

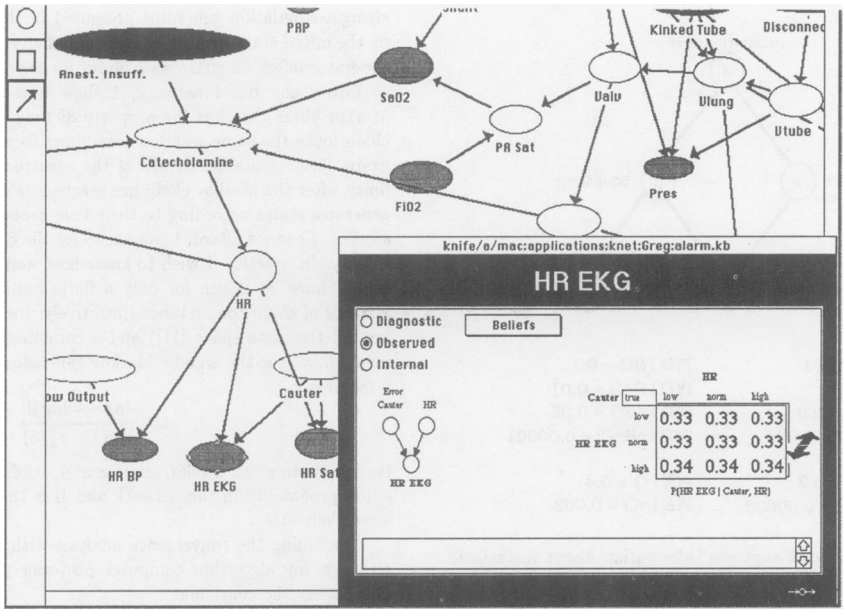



Figure 2: This figure contains a screen bitmap from a knowledge-engineering session with KNET on the ALARM network for ventilation management in the intensive-care unit. By selecting the node corresponding to **HR-EKG** (heart rate as measured by the EKG), the knowledge engineer selects a HyperCard for defining the conditional probability of **HR-EKG** contingent on **HR-EKG**'s parent nodes, **HR** (true heart rate) and **Cauterization**. In particular: The probability of a high heart rate as measured by the EKG, given a true heart in the normal range and the presence of cauterization, is 0.34.

probabilistic model in a single session. Other workers in the field have observed that powerful and understandable user interfaces can absorb as much as two-thirds of a project's design and implementation cycle [2]. KNET alters the software-engineering balance: With HyperCard as the front end, development of an appropriate user interface expends fewer human resources than were heretofore required.

The KNET kernel does not change across applications. But the packaging and presentation of a specific knowledge-intensive system can vary according to the requirements of the target audience. The KNET environment offers a number of possible user interfaces that the knowledge engineer can tailor to the problem at hand. The prefabricated interfaces encompass a wide range of paradigms for human-machine interaction. Whether the designer chooses to tailor an existing user interface or to build a new one from scratch, KNET defines a single consistent protocol for interacting with the knowledge kernel.

Figure 2 shows a fragment of the ALARM belief network for ventilation management in the ICU [1]. The figure also displays a HyperCard window corresponding to the chance node **HR-EKG**, which represents the patient's heart rate as determined by an electrocardiogram (EKG) monitor.

The KNET architecture specifies extensions to HyperTalk for updating beliefs, recording observations, storing consultations, calculating optimal decisions, and computing the expected value of perfect information. From the perspective of an interface designer, KNET's facilities for managing probabilistic information appear as transparent extensions of HyperCard's object-oriented universe. The interface designer uses those tools to construct a visual representation, which the user manipulates directly, of the underlying knowledge base.

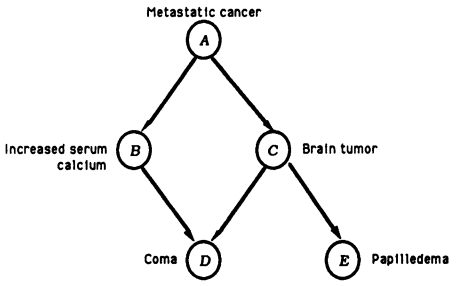
## Computation on Belief Networks

Belief networks provide high-level graphical representations of expert opinion. More precisely, a belief network is a directed acyclic graph wherein nodes represent domain variables and edges represent causal or associational relationships [15]. For each orphan node (a domain variable that lacks incoming arcs or preconditioning influences), the network designer must specify a prior-probability distribution. For all other nodes, the expert must assess a probability mass function (pmf) conditioned on the node's parents. In general, the size of a node's pmf increases exponentially with the number of incoming arcs. PIBNET (Probabilistic Inference in Belief NETworks) is the problem of calculating posterior probabilities for the outcomes of a domain variable, given evidence about other variables in the network. In the ALARM network, for instance, I wish to calculate the probability of a pulmonary embolus given physiologic evidence about the patient's ventilation status.

Every belief network represents a joint-probability space. Belief networks also capture intuitive notions of conditional independence. Those explicitly represented independence assumptions can greatly reduce the complexity of recovering the complete joint pmf over all the variables in the network. For modeling problems that specify singly connected belief networks (otherwise known as causal polytrees), the exact methods compute a probability assignment for all nodes in time proportional to the diameter of the network on parallel machines, and in $O(n)$ time on sequential machines, where $n$ denotes the number of nodes. In the worst case, however, PIBNET seems to require exponential time. More specifically, all known exact algorithms for inference on multiply connected networks operate in exponential time with respect to the size of the network [18].

Figure 3 contains a simple belief network from the domain of medicine. The probability that a patient has a *primary brain tumor* (that is, event $C$) given that he has *metastatic cancer* (that is, $A$) is 20 percent, in this example. The absence of an arc from $B$ to $C$ implies the conditional independence of *increased total serum calcium* and *brain tumor* given *metastatic cancer*; that is, $P(BC|A) = P(B|A) \cdot P(C|A)$ or $P(B|CA) = P(B|A)$. Given a network topology, the appropriate conditional probabilities, and a set of evidence, PIBNET computes the posterior marginal probabilities for each of the uninstantiated nodes.

Let $\mathcal{P}$ denote the class of problems that can be solved in polynomial time on a Turing machine. $\mathcal{NP}$ is the class of problems that can be solved nondeterministically in polynomial time on a Turing machine that can compute along multiple paths simultaneously. Equivalently, $\mathcal{NP}$ contains those problems for which a Turing machine can guess a solution and then verify the solution's correctness in deterministic




| | |
|---|---|
| $P(A) = 0.001$ | $P(D \mid BC) = 0.1$ |
| | $P(D \mid B\neg C) = 0.01$ |
| $P(B \mid A) = 0.3$ | $P(D \mid \neg BC) = 0.05$ |
| $P(B \mid \neg A) = 0.001$ | $P(D \mid \neg B\neg C) = 0.00001$ |
| $P(C \mid A) = 0.2$ | $P(E \mid C) = 0.4$ |
| $P(C \mid \neg A) = 0.00005$ | $P(E \mid \neg C) = 0.002$ |

Figure 3: This belief network captures information about metastatic carcinoma and serum calcium. It has been greatly simplified for purposes of illustration, and does not convey a realistic representation of the causal-associational relationships.

polynomial time.

Let $A$ be a problem that belongs to a class $\mathcal{C}$ of problems. I say that $A$ is *complete* for $\mathcal{C}$ if there exists a construction that reduces every instance of every problem in $\mathcal{C}$ to an instance of $A$ using space that is logarithmic in the size of the instance. In some sense, then, $A$ captures the maximal complexity of the class $\mathcal{C}$; $A$ is at least as difficult to decide, for arbitrary problem instances, as is any other problem in the class. If every problem in $\mathcal{C}$ is reducible to $A$, but $A$ is not known to belong to $\mathcal{C}$, we say that $A$ is *hard* for $\mathcal{C}$.

The hardest problems in $\mathcal{NP}$, known as $\mathcal{NP}$-complete, probably do not admit polynomial-time deterministic algorithms [7]. If any of the $\mathcal{NP}$-complete problems admits a polynomial-time deterministic algorithm, then $\mathcal{P} = \mathcal{NP}$, and all problems in $\mathcal{NP}$ can be solved in polynomial time. The vast majority of theoreticians believe, however, that $\mathcal{NP}$ properly contains $\mathcal{P}$, and that polynomial-time algorithms for $\mathcal{NP}$-complete problems do not exist. $\mathcal{NP}$-hard problems are at least as difficult to answer as are the $\mathcal{NP}$-complete problems, if not more so.

The probabilistic inference problem for belief networks is hard for $\mathcal{NP}$ [6]. The classification of PIBNET as $\mathcal{NP}$-hard has prompted a shift in focus away from deterministic algorithms and toward approximate methods, heuristics, and analyses of average-case behavior.

I now offer a complexity-theoretic treatment of approximate probabilistic inference. I use methods drawn from the analysis of ergodic Markov chains and randomized complexity theory to build an algorithm that efficiently approximates the solutions of inference problems for many belief networks to within arbitrary precision. I slightly alter the standard straight-simulation scheme [16] so as to render an analysis of its computational characteristics. I give the full derivation in [5], and present only the salient results here.

Suppose that I wish to compute all posterior probabilities in the network to within absolute error $\alpha$. Suppose, in addition, that I am willing to tolerate a small probability $\delta$ that the algorithm fails to converge within the $\alpha$ bound. The detailed argument [5], based on Chebyshev's inequality and the scheme of Karp and Luby [12], reveals that

$$N \geq 1/(4\delta\alpha^2)$$

guarantees the convergence criteria, where $N$ is the total number of trials.

I have predicated my analysis on the existence of a trial generator that accurately produces states of the network according to their true probabilities, contingent upon the available evidence. The original straight-simulation generator presented in the literature [16] depends on the initial state (that is, it lacks ergodicity). Moreover, the standard generator offers no guarantees about its convergence properties.

Given any belief network, I show how to construct an ergodic Markov chain that has the property of *time reversibility*. That is, the chain looks the same whether time flows forward or backward. (Once again, [5] presents the details of the construction.) In the limit of infinity, after the Markov chain has reached its stationary distribution, it generates states according to their true probabilities, for Karp—Luby scoring. I cannot afford, however, to let the chain reach equilibrium at infinity. In practice, I wish to know how well the chain has converged after I have let it run for only a finite number $t$ of time steps. An analysis of chain conductance (intuitively, the chain's tendency to flow around the state space [11]) and a combinatoric path-counting argument show that the ergodic Markov generator for belief-network states $s$ requires

$$t \geq \frac{\log\gamma + \log\Pi}{\log(1 - p_0^2/8)}$$

time steps to reach a relative error of $\gamma$, where $p_0$ is the smallest transition probability in the network and $\Pi$ is the joint probability of the least likely state.

Combining the convergence analysis with the Karp—Luby scoring strategy, my algorithm computes posterior probability estimators $\tilde{Y}$ that satisfy the constraint

$$\frac{\Pr[e|\xi]}{(1+\gamma)} - \alpha \leq \tilde{Y} \leq (1+\gamma)\Pr[e|\xi] + \alpha$$

with probability greater than $1 - \delta$. To do so, the algorithm must perform

$$\left\lceil \frac{4(1+\gamma)^3}{3\alpha^2} \right\rceil \cdot (12\lceil -\log\delta \rceil + 1) \cdot \frac{\log\gamma + \log\Pi}{\log(1 - p_0^2/8)}$$

cycles, where each cycle corresponds to a transition of the underlying Markov chain. I can then use those posterior probability estimators to rank the leading diagnoses. My algorithm, therefore, efficiently computes approximate inferences without sacrificing the elegant normative framework of probability theory.

## Results

A close inspection of the bound on $t$ reveals that the value of $p_0$ dominates the performance of the algorithm. For $p_0$ on the order of 0.1, the Markov chain reaches the stationary state quite rapidly. For very small $p_0$, however, the bound $t$ approaches infinity. Figure 4 graphically illustrates the number of transitions needed to guarantee various levels $\gamma$ of rapid mixing for the Markov chain. My detailed analysis of the randomized approximation scheme sets an agenda for further investigation: Focus not on more sophisticated scoring schemes, but rather on state generators that converge rapidly to their stationary distributions.

Note that small probabilities $p_0$ do not entail the failure of the approximation scheme; they merely suggest that the analytic bounds cannot guarantee efficient computation. The analysis of complexity clarifies the underlying computational properties of the algorithm, but it says little about the method's performance on examples drawn from the real world.

I have studied the performance of my algorithm in great detail on the 81-node DxNet model (Figure 5). Figure 6 illustrates the most crucial insight of this empiric study. With 100 trials and 5000 Markov transitions per trial, my algorithm achieves an average error of 0.035 after 76 seconds of computation on a Sun-4 processor. Taking more than 100 samples hardly decreases the error; taking more transitions per trial, however, greatly improves accuracy at the expense of increasing computing time. The convergence depends not so much on the number of tabulated trials, but rather on the quality of those trials. In other words, if I had an ideal trial generator, I could expect very rapid convergence; inasmuch as the raw Markov chain reaches the stationary

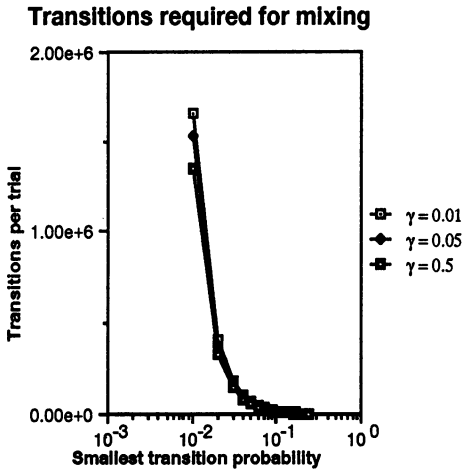



Figure 4: This graph illustrates the crucial relationship between $p_0$, the smallest transition probability in the Markov chain derived from the network, the number of transitions $t$ required to guarantee that the $t$-step transition probability differs from the stationary distribution by a relative error of $\gamma$ or less. Notice that $t$ depends on $\gamma$ only slightly. I expect the ras to perform best on belief networks with conditional probabilities bounded away from 0 and 1.

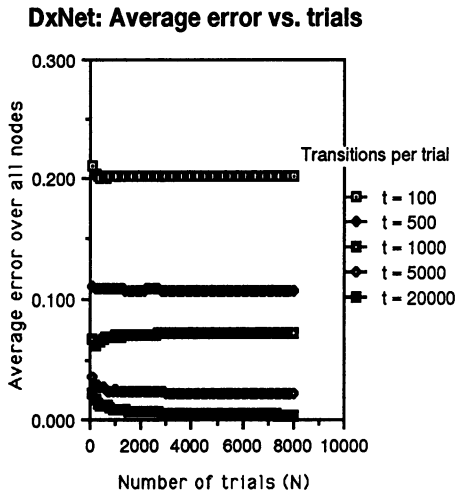



Figure 6: This graph plots the average error over all nodes against the number of trials, for different values of $t$, the number of transitions per trial. Observe that $t$, and therefore the rate of mixing for the Markov chain, almost completely determine the convergence of the algorithm. After a few hundred trials, more computation may serve to guarantee the failure probability $\delta$ on the absolute error, but otherwise appears to be useless for DxNet.

distribution only after many thousands of transitions, however, trial generation in DxNet poses the greatest difficulty.

The shift of focus away from obtaining more samples and toward improving the accuracy of the sampling stream follows directly from my detailed analysis of the algortihm. The complexity calculation yields insight into the deep structure of approximate probabilistic inference. Thus, my analytic framework for analyzing inference, although useful as an end in itself, has important pragmatic consequences for the design and delivery of efficient probabilistic systems.

For chains with large transition probabilities, I expect rapid convergence. For other networks, there is yet hope: My algorithm, in contrast to the exact methods, requires time linear in the number of nodes and outcomes, in the worst case. The known deterministic algorithms all require exponential time in the worst case [18]. The theoretical analysis of my algorithm, however, indicates that the latter remains insensitive to network topology in the worst case, and degrades only as the smallest transition probability approaches 0.

Perhaps more important, the ras lends itself to full parallelization. As low-cost parallel processors become more widely available, I expect performance that compares favorably with the deterministic approaches on networks of moderate size. For large and complex networks, the ras offers the only polynomial-time approach.

To study the parallelization of the ras, I computed the prior-probability distribution for every node in DxNet to within a maximum absolute error of 0.25 on an Intel iPSC 32-node Hypercube. Although the Sun-4 far outdistances the performance of individual nodes in the cube, the availability of multiple processors renders my ras particularly efficient. As Figure 7 demonstrates, I can almost exactly halve the running time by doubling the number of processors. The performance enhancement continues as I add processors. After 16 processors, however, the initial loading time (which depends on the bandwidth of the data path that connects the host to the nodes) dominates the computation.

Having analyzed performance data for the serial Sun-4 processor and for the Hypercube, I make the following prediction: An array with 256 processors, each equivalent in floating-point operations per second (FLOPs) to a Sun-4, can compute probabilities for DxNet to within a maximum error of 0.05 in 4 seconds. Inasmuch as DxNet contains many probabilities close to 0 and therefore has poor conductance, I present DxNet as a realistic model for networks encountered in practice. At some crossover point in the near future, therefore, I expect the ras to deliver performance that compares favorably with the deterministic approaches on networks of moderate size.

My empirical and theoretical results together suggest that the amount of computation required to guarantee a certain absolute error for probabilistic inference depends critically on the smallest transition probability in the network, and on little else. In testing the ras on networks whose sizes differ, but whose transition probabilities are nearly identical, I expect to observe nearly constant CPU times for achieving a given average or maximum absolute error.

Trivial modifications of the ras will support the calculation of approximate probabilistic inferences in QMR-DT [8], our laboratory's decision-theoretic reformulation of Internist/QMR [14], an expert system (developed at the University of Pittsburgh) with almost 600 diseases and over 4000 manifestations. The fastest known deterministic algorithm for inference over the two-level QMR-DT belief network requires time exponential in the number of findings, and cannot currently handle more than about 20 findings. Extrapolating from the results of this paper, which show that the smallest transition probability dominates convergence of the Markov chain to the stationary distribution, I expect that inference in QMR-DT will require about the same time as does inference in DxNet, which also has small transition probabilities. On a 256-node multiprocessor, where each node has the computing power of a Sun-4, I expect to rank the leading diagnoses in less than 4 seconds. More important, the running time of the ras can only improve as the number of physical findings increases.

## Conclusions

KNET demonstrates the feasibility of normative techniques for the management of uncertain information in medical expert systems. Perhaps more important, KNET's existence shows that probabilistic expert systems can exploit the full expressive power of hypermedia for designing interfaces to knowledge-intensive medical applications. KNET supports backward chaining, forward chaining, visual explanation, and

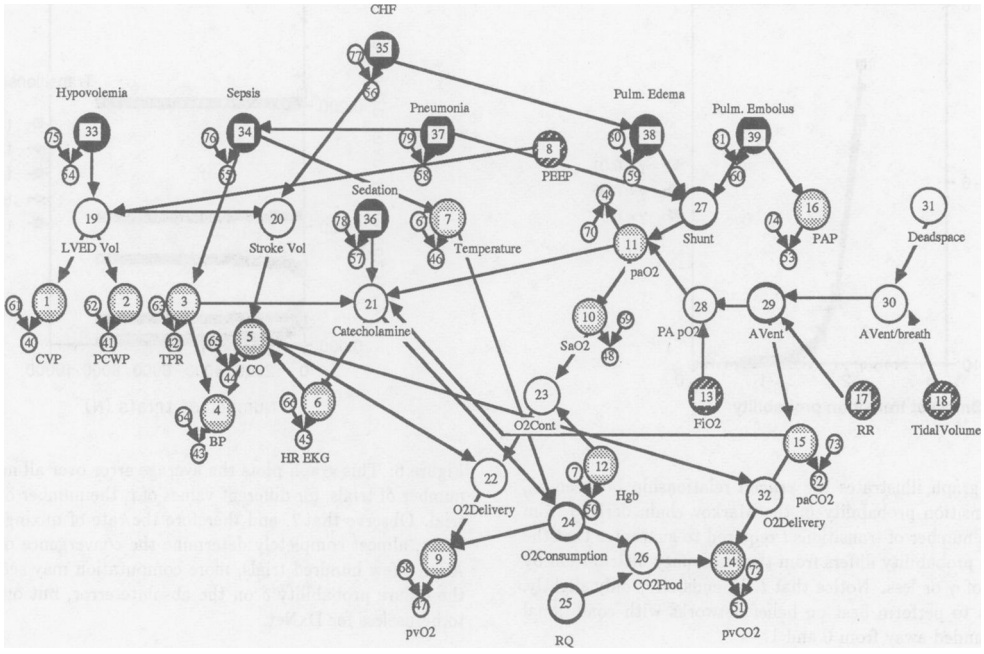



Figure 5: This 81-node network captures an expert anesthesiologist's knowledge of the ICU domain. Solid nodes represent diagnostic outcomes; shaded nodes represent physiologic parameters for which estimates from ICU sensors are available; transparent nodes signify intermediate pathophysiologic concepts; and the small nodes represent error and noise in the sensor data. CO: cardiac output, CVP: central venous pressure, LVED Volume: left ventricular end-diastolic volume, LV Failure: left ventricular failure, MV: minute ventilation, PA Sat: pulmonary artery oxygen saturation, PAP: pulmonary artery pressure, PCWP: pulmonary capillary wedge pressure, Pres: breathing pressure, RR: respiratory rate, TPR: total peripheral resistance, TV: tidal volume.

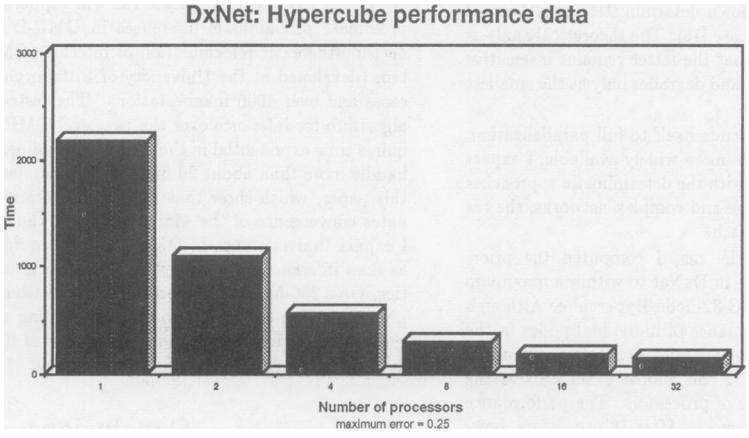



Figure 7: This graph shows the advantage of parallel computation for the PIBNET ras. I calculate the prior-probability distribution of DxNet on a 32-node Intel iPSC Hypercube.

decision making within a single framework for the representation and computation of probabilistic knowledge.

KNET incorporates a novel randomized algorithm that performs approximate probabilistic inference efficiently—that is, in polynomial time. Unlike ad hoc and heuristic schemes for managing uncertainty, my algorithm does not sacrifice the axiomatic foundation of probability theory. Unlike all known exact algorithms for probabilistic inference, the ras requires time linear in the problem size, for arbitrary belief networks, with transition probabilities held constant. And in contrast to other schemes for stochastic simulation, my algorithm states precise a priori criteria for its accuracy.

My theoretical analysis shows that the generation of good trials for the Monte Carlo estimation, and not the generation of many trials, dominates the convergence of the randomized algorithm. I corroborate that insight with a detailed study of DxNet, an actual belief network of significant size. In addition, an implementation of my algorithm on a 32-node Hypercube demonstrates full parallelization. As parallel machines become increasingly widespread, my algorithm will provide a tractable and efficient mechanism for probabilistic inference in domains such as internal medicine. My work, therefore, demonstrates the fruitful application of precise theoretical techniques to the design and implementation of a new environment for building large, probabilistic, knowledge-intensive systems in medicine.

**Acknowledgments.** Gregory F. Cooper guided me throughout my work on KNET and on randomized approximation schemes for probabilistic inference. Harold P. Lehmann, Thierry Barsalou, H. J. Suermondt, and Ingo A. Beinlich made essential contributions to the development of KNET. Edward H. Shortliffe, Ross D. Shachter, Bruce G. Buchanan, Harry R. Lewis, and Les Valiant provided important feedback. Evi Nemeth and Tyler Stevens made the Hypercube at the University of Colorado available. Randy Miller's work inspired the QMR-DT project. Lyn Dupré edited the manuscript.

This work has been supported by grant IRI-8703710 from the National Science Foundation, grant P-25514-EL from the U.S. Army Research Office, Medical Scientist Training Program grant GM07365 from the National Institutes of Health, and grant LM-07033 from the National Library of Medicine. Computer facilities were provided by the SUMEX-AIM resource under grant RR-00785 from the National Institutes of Health.

## References

[1] I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. Technical Report KSL-88-84, Medical Computer Science Group, Knowledge Systems Laboratory, Stanford University, Stanford, CA, December 1988.

[2] D.G. Bobrow, S. Mittal, and M.J. Stefik. Expert systems: Perils and promise. *Communications of the ACM*, 29(9):880–894, 1986.

[3] B. G. Buchanan and E. H. Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, MA, 1984.

[4] R. M. Chavez and G. F. Cooper. KNET: Integrating hypermedia and Bayesian modeling. In *Proceedings of the Fourth Workshop on Uncertainty in Artificial Intelligence*, pages 49–54, University of Minnesota, Minneapolis, MN, August 1988. American Association for Artificial Intelligence.

[5] R.M. Chavez. *Hypermedia and randomized algorithms for probabilistic expert systems.* PhD thesis proposal, Knowledge Systems Laboratory, Stanford University, Stanford, CA, January 1989.

[6] G. F. Cooper. Probabilistic inference using belief networks is NP-hard. Technical Report KSL-87-27, Medical Computer Science Group, Knowledge Systems Laboratory, Stanford University, Stanford, CA, May 1987.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York, 1979.

[8] M. Henrion. Towards efficient probabilistic diagnosis in multiply connected belief networks. In *Influence diagrams for decision analysis, inference, and prediction.* John Wiley and Sons, New York, NY, 1988. In preparation.

[9] S. Holtzman. *Intelligent Decision Systems.* Addison-Wesley, Reading, MA, 1988.

[10] E.J. Horvitz, D. E. Heckerman, B. N. Nathwani, and L. M. Fagan. Diagnostic strategies in the hypothesis-directed pathfinder system. In *First Conference on Artificial Intelligence Applications*, pages 630–636. IEEE Computer Society, 1984.

[11] M. Jerrum and A. Sinclair. Conductance and the rapid mixing property for Markov chains: The approximation of the permanent resolved. In *Proceedings of the Twentieth ACM Symposium on Theory of Computing*, pages 235–244, 1988.

[12] R. M. Karp and M. Luby. Monte-Carlo algorithms for enumeration and reliability problems. In *Proceedings of the Twenty-fourth IEEE Symposium on Foundations of Computer Science*, 1983.

[13] H. P. Lehmann. Knowledge acquisition for probabilistic expert systems. In R.A. Greenes, editor, *Proceedings of the Twelfth Symposium on Computer Applications in Medical Care*, pages 73–77, Washington, D.C., November 1988. IEEE Computer Society Press. To be reprinted in Computer Methods and Programs in Biomedicine.

[14] R. A. Miller, M. A. McNeil, S. M. Challinor, F. E. Masarie, and J. D. Myers. The Internist-1/Quick Medical Rference project — status report. *The Western Journal of Medicine*, 6(145):816–822, 1986.

[15] J. Pearl. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29:241–288, 1986.

[16] J. Pearl. Addendum: Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 33:131, 1987.

[17] R. D. Shachter. Evaluating influence diagrams. *Operations Research*, 34:871–882, 1986.

[18] H. J. Suermondt and G. F. Cooper. Updating probabilities in multiply connected belief networks. In *Proceedings of the Fourth Workshop on Uncertainty in Artificial Intelligence*, pages 335–343, University of Minnesota, Minneapolis, MN, August 1988. American Association for Artificial Intelligence.