Selecting Strategies for Infinite-Horizon Dynamic LIMIDs

Marcel A.J. van Gerven
Institute for Computing and Information Sciences
Radboud University Nijmegen
Toernooiveld 1
6525 ED Nijmegen, The Netherlands

Francisco J. Díez
Department of Artificial Intelligence, UNED
Juan del Rosal 16
28040 Madrid, Spain

Abstract

In previous work we have introduced dynamic limited-memory influence diagrams (DLIMIDs) as an extension of LIMIDs aimed at representing infinite-horizon decision processes. If a DLIMID respects the first-order Markov assumption then it can be represented by 2TLIMIDS. Given that the treatment selection algorithm for LIMIDs, called single policy updating (SPU), can be infeasible even for small finite-horizon models, we propose two alternative algorithms for treatment selection with 2TLIMIDS. First, single rule updating (SRU) is a hill-climbing method inspired upon SPU which needs not iterate exhaustively over all possible policies at each decision node. Second, a simulated annealing algorithm can be used to avoid the local-maximum policies found by SPU and SRU.

1 Introduction

An important goal in artificial intelligence is to create systems that make optimal decisions in situations characterized by uncertainty. One can think for instance of a robot that navigates based on its sensor readings in order to achieve goal states, or of a medical decision-support system that chooses treatments based on patient status in order to maximize life-expectancy.

Limited-memory influence diagrams (LIMIDs) are a formalism for decision-making under uncertainty (Lauritzen and Nilsson, 2001). They generalize standard influence diagrams (Howard and Matheson, 1984) by relaxing the assumption that the whole observed history is taken into account when making a decision, and by dropping the requirement that a complete order is defined over decisions. This increases the size and variety of decision problems that can be handled, although possibly at the expense of finding only approximations to the optimal strategy. Often however, there is no predefined time at which the process stops (i.e. we have an *infinite-horizon decision process*) and in that case the LIMID would also become infinite in size. In previous work, we have introduced dynamic LIMIDs (DLIMIDs) and two-stage temporal LIMIDs (2TLIMIDs) as an extension of standard LIMIDs that allow for the representation of infinite-horizon decision processes (van Gerven et al., 2006). However, the problem of finding acceptable strategies for DLIMIDs has not yet been addressed. In this paper we discuss a number of techniques to approximate the optimal strategy for infinite-horizon dynamic LIMIDs. We demonstrate the performance of these algorithms on a non-trivial decision problem.

2 Preliminaries

2.1 Bayesian Networks

Bayesian networks (Pearl, 1988) provide for a compact factorization of a joint probability dis-

tribution over a set of random variables by exploiting the notion of *conditional independence*. One way to represent conditional independence is by means of an acyclic directed graph (ADG) G where vertices $V(G)$ correspond to random variables $\mathbf{X}$ and the absence of arcs from the set of arcs $A(G)$ represents conditional independence. Due to this one-to-one correspondence we will use vertices $v \in V(G)$ and random variables $X \in \mathbf{X}$ interchangeably. A *Bayesian network* (BN) is defined as $\mathcal{B} = (\mathbf{X}, G, P)$, such that the joint probability distribution P over $\mathbf{X}$ factorizes according to G:

$$P(\mathbf{X}) = \prod_{X \in \mathbf{X}} P(X \mid \pi_G(X))$$

where $\pi_G(X) = \{X' \mid (X', X) \in A(G)\}$ denotes the *parents* of X. We drop the subscript G when clear from context. We assume that (random) variables X can take values x from a set Ω_X and use $\mathbf{x}$ to denote an element in $\Omega_{\mathbf{X}} = \times_{X \in \mathbf{X}} \Omega_X$ for a set $\mathbf{X}$ of (random) variables.

2.2 LIMIDs

Although Bayesian networks are a natural framework for probabilistic knowledge representation and reasoning under uncertainty, they are not suited for decision-making under uncertainty. Influence-diagrams (Howard and Matheson, 1984) are graphical models that extend Bayesian networks to incorporate decision-making but are restricted to the representation of small decision problems. *Limited-memory influence diagrams* (LIMIDs) (Lauritzen and Nilsson, 2001) generalize standard influence-diagrams by relaxing the *no-forgetting* assumption, which states that, given a total ordering of decisions, information present when making decision D is also taken into account when making decision D', if D precedes D' in the ordering. A LIMID is defined as a tuple $\mathcal{L} = (\mathbf{C}, \mathbf{D}, \mathbf{U}, G, P)$ consisting of the following components. $\mathbf{C}$ represents a set of random variables, called *chance variables*, $\mathbf{D}$ represents a set of *decisions* available to the decision maker, where a decision $D \in \mathbf{D}$ is defined as a variable that can take on a value from a set of choices Ω_D, and $\mathbf{U}$ is a set of *utility functions*, which represent the utility of being in a certain state as defined by configurations of chance and decision variables. G is an acyclic directed graph (ADG) with vertices $V(G)$ corresponding to $\mathbf{C} \cup \mathbf{D} \cup \mathbf{U}$, where we use $\mathbf{V}$ to denote $\mathbf{C} \cup \mathbf{D}$. Again, due to this correspondence, we will use nodes in $V(G)$ and corresponding elements in $\mathbf{C} \cup \mathbf{D} \cup \mathbf{U}$ interchangeably. The meaning of an arc $(X, Y) \in A(G)$ is determined by the type of Y. If $Y \in \mathbf{C}$ then the conditional probability distribution associated with Y is conditioned by X. If $Y \in \mathbf{D}$ then X represents information that is present to the decision maker prior deciding upon Y. We call $\pi(Y)$ the *informational predecessors* of Y. The order in which decisions are made in a LIMID should be compatible with the partial order induced by the ADG and are based only on the parents $\pi(D)$ of a decision D. If $Y \in \mathbf{U}$ then X takes part in the specification of the utility function Y such that $Y \colon \Omega_{\pi(Y)} \to \mathbb{R}$. In this paper, it is assumed that utility nodes cannot have children and the joint utility function $\mathcal{U}$ is additively decomposable such that $\mathcal{U} = \sum_{U \in \mathbf{U}} U$. P specifies for each $\mathbf{d} \in \Omega_{\mathbf{D}}$ a distribution

$$P(\mathbf{C} : \mathbf{d}) = \prod_{C \in \mathbf{C}} P(C \mid \pi(C))$$

that represents the distribution over $\mathbf{C}$ when we externally set $\mathbf{D} = \mathbf{d}$ (Cowell et al., 1999). Hence, $\mathbf{C}$ is not conditioned on $\mathbf{D}$, but rather parameterized by $\mathbf{D}$, and if $\mathbf{D}$ is unbound then we write $P(\mathbf{C} : \mathbf{D})$.

A *stochastic policy* for decisions $D \in \mathbf{D}$ is defined as a distribution $P_D(D \mid \pi(D))$ that maps configurations of $\pi(D)$ to a distribution over alternatives for D. If P_D is degenerate then we say that the policy is deterministic. A *strategy* is a set of policies $\Delta = \{P_D \colon D \in \mathbf{D}\}$ which induces the following joint distribution over the variables in $\mathbf{V}$:

$$P_\Delta(\mathbf{V}) = P(\mathbf{C} : \mathbf{D}) \prod_{D \in \mathbf{D}} P_D(D \mid \pi(D)).$$

Using this distribution we can compute the expected utility of a strategy Δ as $\mathrm{E}_\Delta(\mathcal{U}) = \sum_{\mathbf{v}} P_\Delta(\mathbf{v}) \mathcal{U}(\mathbf{v})$. The aim of any rational decision maker is then to maximize the expected utility by finding the optimal strategy $\Delta^* \equiv \arg\max_\Delta \mathrm{E}_\Delta(\mathcal{U})$.

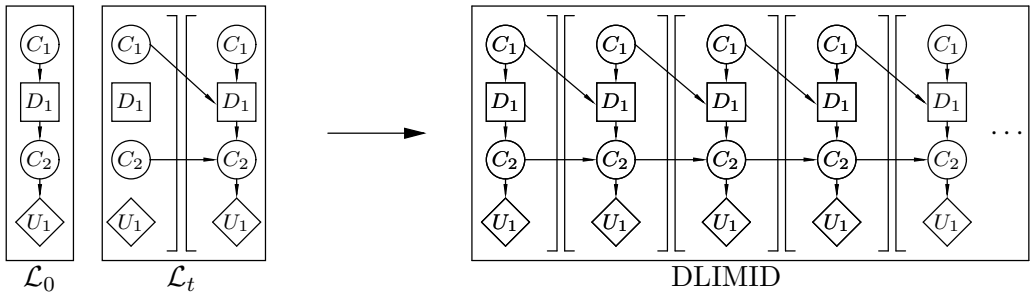

Figure 1: Chance nodes are shown as circles, decision nodes as squares and utility nodes as diamonds. The 2TLIMID (left) can be unrolled into a DLIMID (right), where large brackets denote the boundary between subsequent times. Note that due to the definition of a 2TLIMID, the informational predecessors of a decision can only occur in the same, or the preceding time-slice.

2.3 Dynamic LIMIDs and 2TLIMIDs

Although LIMIDs can often represent finite-horizon decision processes more compactly than standard influence diagrams, they cannot represent infinite-horizon decision processes. Recently, we introduced *dynamic* LIMIDs (DLIMIDs) as an explicit representation of decision processes (van Gerven et al., 2006). To represent time, we use $\mathbf{T} \subseteq \mathbb{N}$ to denote a set of time points, which we normally assume to be an interval $\{u \mid t \leq u \leq t', \{t, u, t'\} \subset \mathbb{N}\}$, also written as $t : t'$. We assume that chance variables, decision variables and utility functions are indexed by a superscript $t \in \mathbf{T}$, and use $\mathbf{C}^{\mathbf{T}}$, $\mathbf{D}^{\mathbf{T}}$ and $\mathbf{U}^{\mathbf{T}}$ to denote all chance variables, decision variables and utility functions at times $t \in \mathbf{T}$, where we abbreviate $\mathbf{C}^{\mathbf{T}} \cup \mathbf{D}^{\mathbf{T}}$ with $\mathbf{V}^{\mathbf{T}}$.

A DLIMID is simply defined as a LIMID $(\mathbf{C}^{\mathbf{T}}, \mathbf{D}^{\mathbf{T}}, \mathbf{U}^{\mathbf{T}}, G, P)$ such that for all pairs of variables $X^t, Y^u \in \mathbf{V}^{\mathbf{T}} \cup \mathbf{U}^{\mathbf{T}}$ it holds that if $t < u$ then Y^u cannot precede X^t in the ordering induced by G. If $\mathbf{T} = 0 : N$, where $N \in \mathbb{N}$ is the (possibly infinite) *horizon*, then we suppress $\mathbf{T}$ altogether, and we suppress indices for individual chance variables, decision variables and utility functions when clear from context. If a DLIMID respects the Markov assumption that the future is independent of the past, given the present, then it can be compactly represented by a 2TLIMID (see Fig. 1), which is a pair $\mathcal{T} = (\mathcal{L}_0, \mathcal{L}_t)$ with *prior model* $\mathcal{L}_0 = (\mathbf{C}^0, \mathbf{D}^0, \mathbf{U}^0, G^0, P^0)$ and *transition model* $\mathcal{L}_t = (\mathbf{C}^{t-1:t}, \mathbf{D}^{t-1:t}, \mathbf{U}^t, G, P)$ such that for all $V^{t-1} \in \mathbf{V}^{t-1:t}$ in the transition model it holds that $\pi_{G^t}(V^{t-1}) = \varnothing$. The transition model is not yet bound to any specific t, but if bound to some $t \in 1 : N$, then it is used to represent $P(\mathbf{C}^t : \mathbf{D}^{t-1:t})$ and utility functions $U \in \mathbf{U}^t$, where both G and P do not depend on t. The prior model is used to represent the initial distribution $P^0(\mathbf{C}^0 : \mathbf{D}^0)$ and utility functions $U \in \mathbf{U}^0$. The *interface* of the transition model is the set $\mathbf{I}^t \subseteq \mathbf{V}^{t-1}$ such that $(V_i^{t-1}, V_j^t) \in A(G) \Leftrightarrow V_i^{t-1} \in \mathbf{I}^t$. Given a horizon N, we may *unroll* a 2TLIMID for n *time-slices* in order to obtain a DLIMID with the following joint distribution:

$$P(\mathbf{C}, \mathbf{D}) = P^0(\mathbf{C}^0 : \mathbf{D}^0) \prod_{t=1}^{N} P(\mathbf{C}^t : \mathbf{D}^{t-1:t}).$$

Let $\Delta^t = \{P_D^t(D \mid \pi_G(D)) \mid D \in \mathbf{D}^t\}$ denote the strategy for a time-slice t and let the whole strategy be $\Delta = \Delta^0 \cup \cdots \cup \Delta^N$. Given Δ^0, $\mathcal{L}_0$ defines a distribution over the variables in $\mathbf{V}^0$:

$$P_{\Delta^0}(\mathbf{V}^0) = P^0(\mathbf{C}^0 : \mathbf{D}^0) \prod_{D \in \mathbf{D}^0} P_D(D \mid \pi_{G^0}(D))$$

and given a strategy Δ^t with $t > 0$, $\mathcal{L}_t$ defines the following distribution over variables in $\mathbf{V}^t$:

$$P_{\Delta^t}(\mathbf{V}^t \mid \mathbf{I}^t) = P(\mathbf{C}^t : \mathbf{D}^u) \prod_{D \in \mathbf{D}^t} P_D(D \mid \pi_G(D))$$

with $u = t - 1 : t$. Combining both equations, given a horizon N and strategy Δ, a 2TLIMID induces a distribution over variables in $\mathbf{V}$:

$$P_\Delta(\mathbf{V}) = P_{\Delta^0}(\mathbf{V}^0) \prod_{t=1}^{N} P_{\Delta^t}(\mathbf{V}^t \mid \mathbf{I}^t). \quad (1)$$

Let $\mathcal{U}^0(\mathbf{V}^0) = \sum_{U \in \mathbf{U}^0} U(\pi_{G^0}(U))$ denote the joint utility for time-slice 0 and let $\mathcal{U}^t(\mathbf{V}^{t\text{-}1:t}) = \sum_{U \in \mathbf{U}^t} U(\pi_G(U))$ denote the joint utility for time-slice $t > 0$. We redefine the joint utility function for a dynamic LIMID as

$$\mathcal{U}(\mathbf{V}) = \mathcal{U}^0(\mathbf{V}^0) + \sum_{t=1}^{N} \gamma^t \mathcal{U}^t(\mathbf{V}^{t\text{-}1:t})$$

where γ with $0 \leq \gamma < 1$ is a *discount factor*, representing the notion that early rewards are worth more than rewards earned at a later time.

In this way, we can use DLIMIDs to represent infinite-horizon Markov decision processes.

2.4 Memory variables

Figure 1 makes clear that the informational predecessors of a decision variable D^t can only occur in time-slices t or $t-1$ (viz. Eq. 1). Observations made earlier in time will not be taken into account and as a result, states that are qualitatively different can appear the same to the decision maker, which leads to suboptimal policies. This phenomenon is known as *perceptual aliasing* (Whitehead and Ballard, 1991). We try to avoid this problem by introducing *memory variables* that represent a summary of the observed history. With each observable variable $V \in \mathbf{V}$, we associate a memory variable $M \in \mathbf{C}$, such that the parents of a memory variable are given by $\pi(M^0) = \{V^0\}$ and $\pi(M^t) = \{V^t, M^{t-1}\}$ for $t \in \{1, \ldots, N\}$ and all children of M^t, with $t \in \{0, \ldots, N\}$, are decision variables $D \in \mathbf{D}^t$.

Figure 2 visualizes the concept of a memory variable, and is used as the running example for this paper. It depicts a 2TLIMID for the treatment of patients that may or may not have a *disease* D. The disease can be identified by a *finding* F, which is the result of a *laboratory test* L, having an associated cost that is captured by the utility function U_2. The *memory* concerning findings is represented by the memory variable M, and based on this memory, we decide whether or not to perform *treatment* T, which has an associated cost, captured by the utility function U_3. Memory concerning *past* findings will be used to decide whether or not to perform the laboratory test. If the patient has the disease then this decreases the chances of patient

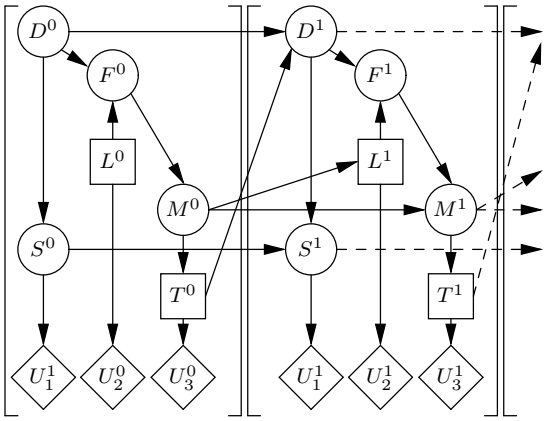

Figure 2: A DLIMID for patient treatment as specified by a 2TLIMID.

survival S. Patient survival has an associated utility U_1. An initial strategy, as for instance suggested by a physician, might be to *always treat* and *never test*.

There are various ways to define Ω_M and the distributions $P(M^0 \mid V^0)$ and $P(M^t \mid V^t, M^{t-1})$ for a memory variable M. The optimal way to define our memory variables is problem dependent, and we assume that this definition is based on the available domain knowledge. For our running example, we choose $\Omega_M = \{a, n, p\} \times \{a, n, p\} \times \{a, n, p\}$, where a stands for the *absence* of a finding, n for a *negative* finding, and p for a *positive* finding, which is evidence for the presence of the disease. $M^t = (z, y, x)$ then denotes the current finding x, the finding in the previous time-slice y, and the finding two time-slices ago z. If the new finding is $F^{t+1} = f$ then $M^{t+1} = (y, x, f)$, and since we have not yet observed any findings at $t = 0$, the initial memory is (a, a, a) if we did not test and (a, a, n) or (a, a, p) if the test was performed.

3 Improving Strategies in Infinite-Horizon DLIMIDs

Although DLIMIDs constructed from 2TLIMIDs can represent infinite-horizon Markov decision processes, to date, the problem of strategy improvement using 2TLIMIDs has not been addressed. In this section, we explore techniques for finding strategies with high expected utility.

3.1 Single Policy Updating

One way to improve strategies in standard LIMIDs is to use an iterative procedure called *single policy updating* or SPU for short (Lauritzen and Nilsson, 2001). Let $\Delta_0 = \{p_1, \ldots, p_n\}$ be an ordered set representing the initial strategy where p_j, $1 \leq j \leq n$ stands for a (randomly initialized) policy P_{D_j} for a decision D_j. We say p_j is the *local maximum policy* for a strategy Δ at decision D_j if $\mathrm{E}_\Delta(\mathcal{U})$ cannot be improved by changing p_j. Single policy updating proceeds by iterating over all decision variables (called a cycle) to find local maximum policies, and to reiterate until no further improvement in expected utility can be achieved. SPU converges in a finite number of cycles to a *local maximum strategy* Δ where each $p_j \in \Delta$ is a local maximum policy. Note that this local maximum strategy is not necessarily the global maximum strategy Δ^*.

To find local maximum policies in standard LIMIDs, Lauritzen and Nilsson (2001) use a message passing algorithm, optimized for single policy updating. In this paper, we resort to standard inference algorithms for finding strategies for (infinite-horizon) DLIMIDs. We make use of the fact that given Δ, a LIMID $\mathcal{L} = (\mathbf{C}, \mathbf{D}, \mathbf{U}, G, P)$ may be converted into a Bayesian network $\mathcal{B} = (\mathbf{X}, G', P')$. Since Δ induces a distribution over variables in $\mathbf{V}$ (viz. Eq. 1), we can use Δ to convert decision variables $D \in \mathbf{D}$ to random variables $X \in \mathbf{X}$ with parents $\pi_G(D)$ such that $P(X \mid \pi_{G'}(X)) = P_D(D \mid \pi_G(D))$. Additionally, utility functions $U \in \mathbf{U}$ may be converted into random variables X by means of Cooper's transformation (Cooper, 1988), which allows us to compute $\mathrm{E}_\Delta(\mathcal{U})$. We use $B(\mathcal{L}, \Delta)$ to denote this conversion of a LIMID into a Bayesian network.

Single policy updating cannot be applied directly to an infinite-horizon DLIMID since computing $\mathrm{E}_\Delta(\mathcal{U})$ would need an infinite number of steps. In order to approximate the expected utility given Δ, we assume that the DLIMID can be represented as a 2TLIMID $\mathcal{T} = (\mathcal{L}_0, \mathcal{L}_t)$ and Δ can be expressed as a pair (Δ^0, Δ^t), such that Δ^0 is the strategy at $t = 0$ and Δ^t is a *stationary* strategy at $t \in 1 : \infty$. Note that the optimal strategy is deterministic and stationary for infinite-horizon and fully observable Markov decision processes (Ross, 1983). In the partially observable case, we can only expect to find approximations to the optimal strategy by using memory variables that represent part of the observational history (Meuleau et al., 1999).

SPU($\mathcal{T}$,Δ_0,ϵ):
  $\Delta = \Delta_0$, $euMax = \mathrm{E}^\epsilon_{\Delta_0}(\mathcal{U})$.
  repeat
    $euMaxOld = euMax$
    for $j = 1$ to n **do**
      for all policies p'_j for Δ at D_j **do**
        $\Delta' = p'_j * \Delta$
        if $\mathrm{E}^\epsilon_{\Delta'}(\mathcal{U}) > euMax + \epsilon$ **then**
          $\Delta = \Delta'$ **and** $euMax = \mathrm{E}^\epsilon_{\Delta'}(\mathcal{U})$
        end if
      end for
    end for
  until $euMax = euMaxOld$
  return Δ

Figure 3: Single policy updating for 2TLIMIDs.

We proceed by converting $(\mathcal{L}_0, \mathcal{L}_t)$ into $(\mathcal{B}_0, \mathcal{B}_t)$ with $\mathcal{B}_0 = B(\mathcal{L}_0, \Delta^0)$ and $\mathcal{B}_t = B(\mathcal{L}_t, \Delta^t)$, where $(\mathcal{B}_0, \mathcal{B}_t)$ is known as a *two-stage temporal Bayes net* (Dean and Kanazawa, 1989). We use inference algorithms that operate on $(\mathcal{B}_0, \mathcal{B}_t)$ in order to compute an approximation of the expected utility. In our work, we have used the *interface algorithm* (Murphy, 2002), for which it holds that the space and time taken to compute each $P(\mathbf{X}^t \mid \mathbf{X}^{t-1})$ does not depend on the number of time-slices. The approximation $\mathrm{E}^\epsilon_\Delta(\mathcal{U})$ is made using a finite number of time-slices k, where k is such that $\gamma^k < \epsilon$ with $\epsilon > 0$. The discount factor γ ensures that $\lim_{t\to\infty} \mathrm{E}_\Delta(\mathcal{U}) = 0$. Let $\Delta_0 = \Delta^0 \cup \Delta^t$ be the initial strategy with $\Delta^0 = \{p_1, \ldots, p_m\}$ and $\Delta^t = \{p_{m+1}, \ldots, p_n\}$, where m is the number of decision variables in $\mathcal{L}_0$ and $n-m$ is the number of decision variables in $\mathcal{L}_t$. Following (Lauritzen and Nilsson, 2001), we define $p'_j * \Delta$ as the strategy obtained by replacing p_j with p'_j in Δ. SPU based on a 2TLIMID $\mathcal{T}$ with initial strategy Δ_0 is then defined by the algorithm in Fig. 3.

3.2 Single Rule Updating

An obstacle for the use of SPU for strategy improvement is the fact that if the state-space $\Omega_{\pi(D)}$ for informational predecessors $\pi(D)$ of a decision variable D becomes large, then it becomes impossible in practice to iterate over all possible policies for D. The number of policies that needs to be evaluated at each decision variable D grows as k^{m^r}, with $k = |\Omega_D|$, assuming that $|\Omega_{V_j}| = m$ for all $V_j \in \pi(D)$, and r is the number of informational predecessors of D. For example, looking back for two time-slices within our model leads to 2 policies for L^0 and 2^{27} policies for T^0, L^1 and T^1, which is computationally infeasible, even for this small example.

For this reason, we introduce a hill-climbing search called *single rule updating* (SRU) that is inspired upon single policy updating. A deterministic policy can be viewed as a mapping $p_j \colon \Omega_{\pi(D_j^t)} \rightarrow \Omega_{D_j^t}$, describing for each configuration $\mathbf{x} \in \Omega_{\pi(D_j^t)}$ an action $a \in \Omega_{D_j^t}$. We call $(\mathbf{x}, a) \in p_j$ a *decision rule*. Instead of exhaustively searching over all possible policies for each decision variable, we try to increase the expected utility by local changes to the decision rules within the policy. I.e., at each step we change one decision-rule within the policy, accepting the change when the expected utility increases. We use $(\mathbf{x}, a') * p_j$ to denote the replacement of $(\mathbf{x}, a)$ by $(\mathbf{x}, a')$ in p_j. Similar to SPU, we keep iterating until there is no further increase in the expected utility (Fig. 4).

SRU decreases the number of policies that need to be evaluated in each *local* cycle for a decision variable to km^r, where notation is as before. For our example, we only need to evaluate 2 policies for L^0 and 54 policies for T^0, L^1 and T^1 in each local cycle, albeit at the expense of replacing the exhaustive search by a hill-climbing strategy, increasing the risk of ending up in local maxima, and having to run local cycles until convergence.

3.3 Simulated Annealing

SPU and SRU both find local maximum strategies, which may not be the optimal strategy Δ^*. To see this, consider the proposed strategy for our running example (Fig. 2) to never test and always treat. Suppose this is our initial strategy Δ_0 for either the SPU or SRU algorithm. Trying to improve the policy for the laboratory test L we find that performing the test will only decrease the expected utility since the test has no informational value (we always treat) but does have an associated cost. Conversely, trying to improve the policy for treatment we find that the test is never performed and therefore it is more safe to always treat. Hence, SPU and SRU will stop after one cycle, returning the proposed strategy as the local optimal strategy.

SRU($\mathcal{T}$,Δ_0,ϵ):
  $\Delta = \Delta_0$, $euMax = \mathrm{E}^{\epsilon}_{\Delta_0}(\mathcal{U})$
  repeat
    $euMaxOld = euMax$
    for $j = 1$ to n **do**
      repeat
        $euMaxLocal = euMax$
        for all configurations $\mathbf{x} \in \Omega_{\pi(D_j)}$ **do**
          for all actions $a' \in \Omega_{D_j}$ **do**
            $p'_j = (\mathbf{x}, a') * p_j$
            $\Delta' = p'_j * \Delta$
            if $\mathrm{E}^{\epsilon}_{\Delta'}(\mathcal{U}) > euMax + \epsilon$ **then**
              $\Delta = \Delta'$ **and** $euMax = \mathrm{E}^{\epsilon}_{\Delta'}(\mathcal{U})$
            end if
          end for
        end for
      until $euMax = euMaxLocal$
    end for
  until $euMax = euMaxOld$
  return Δ

Figure 4: Single rule updating for 2TLIMIDs.

In order to improve upon the strategies found by SRU, we resort to *simulated annealing* (SA), which is a heuristic search method that tries to avoid getting trapped into local maximum solutions that are found by hill-climbing techniques such as SRU (Kirkpatrick et al., 1983). SA chooses candidate solutions by looking at neighbors of the current solution as defined by a *neighborhood function*. Local maxima are avoided by sometimes accepting worse solutions according to an *acceptance function*.

SA($\mathcal{T}$,Δ_0,ϵ,τ_0, T):
  $\Delta = \Delta_0$, $t = 0$, $eu = \mathrm{E}^{\epsilon}_{\Delta}(\mathcal{U})$
  repeat
    select a random decision variable D_j
    select a random decision rule $(\mathbf{x}, a) \in p_j$
    select a random action $a' \in \Omega_{D_j}$, $a' \neq a$
    $p'_j = (\mathbf{x}, a') * p_j$
    $\Delta' = p'_j * \Delta$
    $eu' = \mathrm{E}^{\epsilon}_{\Delta'}(\mathcal{U})$
    if $\theta \leq P(a(\Delta') = \text{yes} \mid eu + \epsilon, eu', t)$ **then**
      $\Delta = \Delta'$ **and** $eu = eu'$
    end if
    $t = t + 1$
  until $T(t) < \tau_0$
  return SRU($\mathcal{T}, \Delta, \epsilon$)

Figure 5: Simulated annealing for 2TLIMIDs.

In this paper, we have chosen the acceptance function $P(a(\Delta') = \text{yes} \mid eu, eu', t)$ equal to 1 if $eu' > eu$ and equal to $\exp(\frac{eu' - eu}{T(t)})$ otherwise, where $a(\Delta')$ stands for the acceptance of the proposed strategy Δ', $eu' = \mathrm{E}^{\epsilon}_{\Delta'}(\mathcal{U})$, $eu = \mathrm{E}^{\epsilon}_{\Delta}(\mathcal{U})$ for the current strategy Δ, and T is an *annealing schedule* that is defined as $T(t+1) = \alpha \cdot T(t)$ where $T(0) = \beta$ with $\alpha < 1$ and $\beta > 0$.

With respect to strategy finding in DLIMIDs, we propose the simulated annealing scheme as shown in Fig. 5, where θ is repeatedly chosen uniformly at random between 0 and 1. Note that after the annealing phase we apply SRU in order to greedily find a local maximum solution.

4 Experimental Results

We have compared the solutions found by SRU and SA to our running example based on twenty randomly chosen initial strategies. Note that SPU was not feasible due to the large number of policies per decision variable. We have chosen a discounting factor $\gamma = 0.95$ and a stopping criterion $\epsilon = 0.01$. After some initial experiments we have chosen $\alpha = 0.995$, $\beta = 0.5$ and $tMin = 3.33 \cdot 10^{-3}$ for the SA parameters. In order to reduce computational load, we assume that the parameters for $P(T^0 \mid M^0)$ and $P(T^t \mid M^t)$ are tied such that we need only estimate three different policies for $P(L^0)$, $P(L^t \mid M^{t-1})$ and $P(T^0 \mid M^0) = P(T^t \mid M^t)$.

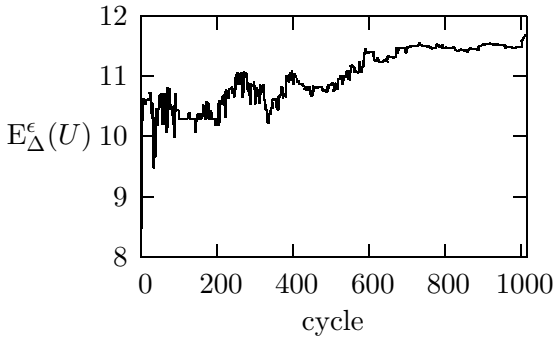

Figure 6: Change in $\mathrm{E}^{\epsilon}_{\Delta}(U)$ for one run of SA. The sudden increase at the end of the run is caused by the subsequent application of SRU.

For the twenty experiments, SRU found strategies with an average expected utility of $\mathrm{E}^{\epsilon}_{\Delta}(\mathcal{U}) = 11.23$ with $\sigma = 0.43$. This required the evaluation of 720 different strategies on average. SA found strategies with an average expected utility of $\mathrm{E}^{\epsilon}_{\Delta}(\mathcal{U}) = 11.51$ with $\sigma = 0.14$. This required the evaluation of 1546 different strategies on average. In 16 out of 20 experiments, SA found strategies with higher expected utility than SRU. Furthermore, due to the random behavior of the algorithm it avoids the local maximum policies found by SRU. For instance, SRU finds a strategy with expected utility 10.29 three out of twenty times, which is equal to the expected utility of the proposed strategy to always treat and never test. The best strategy was found by simulated annealing and has an expected utility of 11.67. The subsequent values of $\mathrm{E}^{\epsilon}_{\Delta}(\mathcal{U})$, found during that experiment, are shown in Fig. 6.

The found strategy can be represented by a *policy graph* (Meuleau et al., 1999); a finite state controller that depicts state transitions, where states represent actions and transitions are induced by observations. Figure 7 depicts the policy graph for the best found strategy. Starting at the left arrow, each time slice constitutes a test decision (circle) and a treatment decision (double circle). Shaded circles stand for positive decisions (i.e., $L^t = yes$ and $T^t = yes$) and clear circles stand for negative decisions (i.e., $L^t = no$ and $T^t = no$). If a test has two outgo-

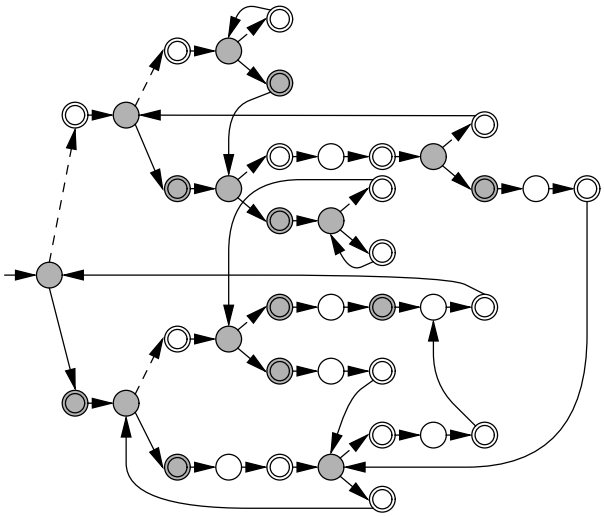

Figure 7: Policy graph for the best strategy found by simulated annealing.

ing arcs, then these stand for a negative finding (dashed arc) and positive finding (solid arc) respectively. Most of the time, the policy graph associates a negative test result with no treatment and a positive test result with treatment.

5 Conclusion

In this paper we have demonstrated that reasonable strategies can be found for infinite-horizon DLIMIDs by means of SRU. Although computationally more expensive, SA considerably improves the found strategies by avoiding local maxima. Both SRU and SA do not suffer from the intractability of SPU when the number of informational predecessors increases. The approach does require that good strategies can be found using a limited amount of memory, since otherwise, found strategies will fail to approximate the optimal strategy. This requirement should hold especially between time-slices, since the state-space of memory variables can become prohibitively large when a large part of the observed history is required for optimal decision-making. Although this restricts the types of decision problems that can be managed, DLIMIDs, constructed from a 2TLIMID, allow the representation of large, or even infinite-horizon decision problems, something which standard influence diagrams cannot manage in principle. Hence, 2TLIMIDs are particularly useful in the case of problems that cannot be properly approximated by a short number of time slices.

References

G.F. Cooper. 1988. A method for using belief networks as influence diagrams. In *Proceedings of the 4th Workshop on Uncertainty in AI*, pages 55–63, University of Minnesota, Minneapolis.

R. Cowell, A. P. Dawid, S. L. Lauritzen, and D. Spiegelhalter. 1999. *Probabilistic Networks and Expert Systems*. Springer.

T. Dean and K. Kanazawa. 1989. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(3):142–150.

R.A. Howard and J.E. Matheson. 1984. Influence diagrams. In R.A. Howard and J.E. Matheson, editors, *Readings in the Principles and Applications of Decision Analysis*. Strategic Decisions Group, Menlo Park, CA.

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

S.L. Lauritzen and D. Nilsson. 2001. Representing and solving decision problems with limited information. *Management Science*, 47(9):1235–1251.

N. Meuleau, K.-E. Kim, L.P. Kaelbling, and A.R. Cassandra. 1999. Solving POMDPs by searching the space of finite policies. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pages 417–426, Stockholm, Sweden.

K.P. Murphy. 2002. *Dynamic Bayesian Networks*. Ph.D. thesis, UC Berkely.

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 2 edition.

S. Ross. 1983. *Introduction to Stochastic Dynamic Programming*. Academic Press, New York.

M.A.J. van Gerven, F.J. Díez, B.G. Taal, and P.J.F. Lucas. 2006. Prognosis of high-grade carcinoid tumor patients using dynamic limited-memory influence diagrams. In *Intelligent Data Analysis in bioMedicine and Pharmacology (IDAMAP) 2006*. Accepted for publication.

S.D. Whitehead and D.H. Ballard. 1991. Learning to perceive and act by trial and error. *Machine Learning*, 7:45–83.