

Optimization of Actions in Activation Timed Influence Nets

M. Faraz Rafi, Abbas K. Zaidi and Alexander H. Levis
 System Architectures Laboratory, ECE Dept., George Mason University, Fairfax, VA USA
 E-mail: {mrafi1, szaidi2, alevi}@gmu.edu

P. Papantoni-Kazakos
 EE Dept., University of Colorado Denver, Denver, CO USA
 E-mail: Titsa.Papantoni@cudenver.edu

Keywords: influence net, activation timed influence net, Bayesian net

Received: October 12, 2008

A sequential evolution of actions, in conjunction with the preconditions of their environment and their effects, are all depicted by Activation Timed Influence Nets. In this paper, we develop two algorithms for the optimal selections of such actions, given a set of preconditions. A special case for the two algorithms is also considered where the selection of actions is further constrained by the use of dependencies among them. The two algorithms are based on two different optimization criteria: one maximizes the probability of a given set of target effects, while the other maximizes the average worth of the effects' vector.

Povzetek: Predstavljena sta dva algoritma za optimizacijo akcij v časovno odvisnih mrežah.

1 Introduction

We consider the scenario¹ where a sequence of actions needs to be initialized towards the materializing of some desirable effects. As depicted in Figure 1, each action is supported by a set of preconditions and gives rise to a set of effects; the latter become then the preconditions of the following action(s) which, in turn, gives rise to another set of effects. Such sequential evolution of actions is termed Activation Timed Influence Nets (ATINs), where the action performers may be humans. ATINs are an extension of an earlier formalism called Timed Influence Nets (TINs) [6-12, 20-27, 30, 31] that integrate the notions of time and uncertainty in a network model. The TINs are comprised of nodes that represent propositions (i.e., pre-and post-conditions of potential actions as well as assertions of events which may indirectly describe such actions), connected via causal links that represent relationships between the nodes, without any explicit representation of actions. TINs have been experimentally used in the area of Effects Based Operations (EBOs) for evaluating alternate courses of actions and their effectiveness to mission objectives in a variety of domains, e.g., war games [20-22, 25], and coalition peace operations [24, 27], to name a few. A number of analytical tools [6-12, 23, 24, 27, 30] have also been developed over the years for TIN models to help an analyst update conditions/assertions, represented as nodes in a TIN, to map a TIN model to a Time Sliced Bayesian Network for incorporating feedback evidence, to determine best set of pre-conditions for both timed and un-timed versions of Influence Nets, and to assess temporal aspects of the in-

fluences between nodes. A recent work [31] on TINs, underlying constructs and the computational algorithms, provides a comprehensive analytical underpinning of the modeling and analysis approach.

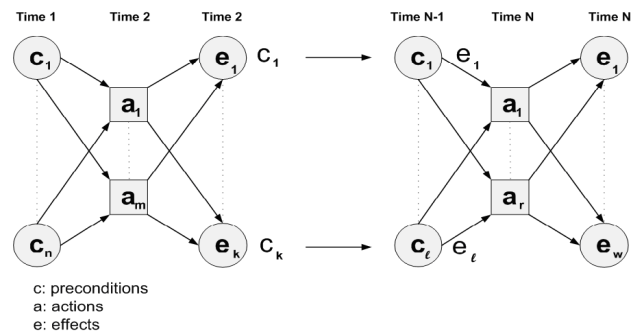


Figure 1: Network Representation of an Activation Timed Influence Net (ATIN)

In contrast to their predecessors (i.e., TINs), ATINs explicitly incorporate as nodes the mechanisms and/or actions that are responsible for changes in the state of a domain; other nodes represent preconditions and effects of actions. A set of preconditions may support a number of different actions, each of which may lead to the same effects, with different probabilities and different costs/awards, however. The objective is to select an optimal set of actions, where optimality is determined via a pre-selected performance criterion. In this paper, we present two algorithms which attain such an objective. We note that an effort to develop an action selection algorithm is also presented in [1].

The organization of the paper is as follows: In Section 2, we present the core formalization of the problem,

¹ This work was supported by the Air Force Office of Scientific Research (AFOSR) under Grants FA9550-05-1-0106 and FA9550-05-1-0388.

including two different optimization criteria. In Section 3, we derive the two algorithms which address the latter criteria. In Section 4, we express the extensions of the two algorithms to the network propagation scenario. In Section 5, we include numerical evaluations while in Section 6, we draw some conclusions.

1.1 Related Work

ATINs include action planning. In the domain of action planning, classical planners assume that the effects of an action are known with certainty and generate a set of actions that will achieve the desired goals [19]. Some planners do monitor for errors as actions are executed, but no action adaptations are incorporated [29]. Other planners assign probabilities to the effects of actions [2, 13, 14, 16, 28], but provide no mechanisms for reacting to changes in the environment. Reactive planners [5, 15, 17, 18] are designed to select and execute actions in response to the current state of the world, but, with a few exceptions [3], [4], they do not use probabilistic information to determine the likelihood of success of the actions. In [1], probabilistic information is used, in an effort to deal with environmental uncertainties, but no optimal action selection strategies are considered and/or proposed.

The ATIN formalism in this paper is similar to an earlier work by Sugato Baghci et al [1] on planning under uncertainty. The similarity, however, stops with the graph representation of preconditions, actions and their effects. Similar parallels can also be drawn with other graph-based planning approaches, e.g. GraphPlan (<http://www.cs.cmu.edu/~avrim/graphplan.html>). The approach in this paper represents a new formalism and is based on well established statistical results.

2 Problem formalization – core

In this section, we consider a modular core problem. We initially isolate a single action with its supporting preconditions and its resulting effects, as depicted in Fig. 2.

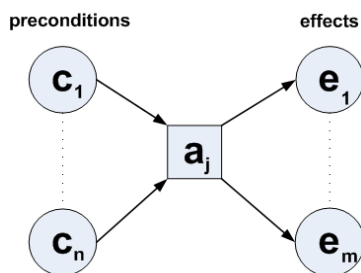


Figure 2: A Single Action ATIN

$X_1^n = [X_1, \dots, X_n]^T$ The status random vector of the preconditions, where $X_i = 1$, if precondition c_i is present and $X_i = 0$ if precondition c_i is absent. x_1^n denotes binary vector value realizations of X_1^n .

$Y_1^m = [Y_1, \dots, Y_m]^T$ The status random vector of the effects, where $Y_i = 1$, if effect e_i is present and $Y_i = 0$ if effect e_i is absent. y_1^m denotes binary vector value realizations of Y_1^m .

$p_j(x_1^n)$ The probability of success for action a_j , given that the value of the precondition status vector is x_1^n ;
 $P(\text{success for action } a_j | x_1^n)$

$q_j(y_1^m)$ The probability that the value of the effects' status vector is y_1^m , given that the action a_j is taken;
 $P(y_1^m | a_j \text{ taken})$

$q_0(y_1^m)$ The probability that the value of the effects' status vector is y_1^m , given that no action is taken;
 $P(y_1^m | \text{no action taken})$

$U_j(y_1^m)$ The utility of the value y_1^m of the effects' status vector, when action a_j is taken.

$U_0(y_1^m)$ The utility of the value y_1^m of the effects' status vector, when no action is taken.

We note that the utility function $U_j(y_1^m)$ measures the net worth of the effects' vector value y_1^m when action a_j is taken; thus, $U_j(y_1^m)$ is computed as the worth of y_1^m minus the cost of deployment for action a_j .

Let us now assume mutually exclusive actions, which are supported by the same preconditions, to lead to the same set of effects (as shown in Fig. 3). Let $\{a_j\}_{1 \leq j \leq k}$ be this set of actions and let X_1^n and Y_1^m denote the common status random vectors of preconditions versus effects, respectively. Let the utility functions for each action in the set $\{a_j\}_{1 \leq j \leq k}$ be nonnegative; let also $U_0(y_1^m)$ be nonnegative.

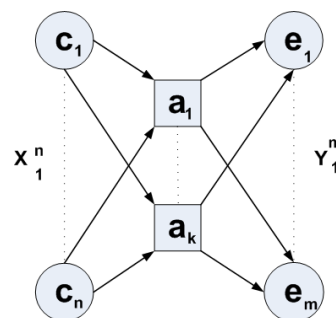


Figure 3: A Single Level ATIN

We now state multiple versions of the core problem, based on two different optimization criteria. Problem 3a and 3b are the constrained versions of the first two problems.

Problem 1 (Optimal Path Problem)

Given a preconditions vector value x_1^n , given an effects vector value y_1^m , find the maximum probability action that connects them. That is, find the action that maximizes the conditional probability $P(y_1^m | x_1^n)$.

Problem 2 (Average Utility Maximization)

Given a preconditions vector value x_1^n , find the action or actions that maximize the effects' average utility.

Problem 3a (Optimal Path Problem with Constrained Actions)

Given a preconditions vector value x_1^n , given an effects vector value y_1^m , and an action dependency matrix, find the maximum probability action that connects them. That is, find the action that maximizes the conditional probability $P(y_1^m | x_1^n)$. In this case, only those action combinations are considered that are allowed by the constraints in the dependency matrix.

Problem 3b (Average Utility Maximization with Constrained Actions)

Given a preconditions vector value x_1^n , find the action or actions that maximize the effects' average utility. As in Problem 3a, only those action combinations are considered that are allowed by the constraints in the dependency matrix.

Action Dependency Matrix (ADM)

An action dependency matrix is a tool which defines dependency among actions in the network. It reduces the number of combinations of actions by considering only those allowed by the dependency matrix. It also reduces significantly the amount of calculations required to obtain the optimal path. The value of the variable a_{ij} reflects the existence or absence of dependency between actions a_i and a_j , where a_{ij} equals 1; for positive dependency and equals 0; for negative dependency, and $1 \leq i, j \leq n$, where 'n' represents the total number of actions in the network. The elements of an ADM are determined as follows:

$$a_{ij} = \begin{cases} 1 & ; \text{ if action } a_i \text{ is selected for execution} \\ & \text{ in level } l, \text{ then } a_j \text{ refers to the action} \\ & \text{ that has to be or has been selected} \\ & \text{ for execution in level } k, \text{ where } l \neq k \\ 0 & ; \text{ if action } a_i \text{ is selected for execution} \\ & \text{ in level } l, \text{ then } a_j \text{ refers to the action} \\ & \text{ that must not be selected for execution} \\ & \text{ in level } k, \text{ where } l \neq k \end{cases}$$

where, level l in an ATIN corresponds to a set of preconditions ($C_1, C_2 \dots C_n$) followed by a set of actions ($a_1, a_2 \dots a_k$) and a set of effects ($e_1, e_2 \dots e_m$) (as shown in Fig. 3). The effects of this level then serve as the preconditions for the next level $l+1$ and so on.

3 Solutions to the core problems

We present the solutions to the two core problems posed in Section 2 in the form of a theorem, whose proof is in the Appendix.

Theorem 1

a. Given x_1^n , given y_1^m , and given a set of actions $\{a_j\}_{1 \leq j \leq k}$, the conditional probability $P(y_1^m | x_1^n)$ is maximized as follows:

by action a_{j^*} ; if

$$q_{j^*}(y_1^m) p_{j^*}(x_1^n) = \max_{1 \leq j \leq k} q_j(y_1^m) p_j(x_1^n) > q_0(y_1^m) \quad (1)$$

where then $\max P(y_1^m | x_1^n) = q_{j^*}(y_1^m) p_{j^*}(x_1^n)$

by no action; if

$$q_0(y_1^m) > \max_{1 \leq j \leq k} q_j(y_1^m) p_j(x_1^n) \quad (2)$$

where then $\max P(y_1^m | x_1^n) = q_0(y_1^m)$

If more than one action satisfy the maximum in (1), then one of these actions may be selected randomly.

b. Given x_1^n , given a set of actions $\{a_j\}_{1 \leq j \leq k}$, and given utility functions $\{U_j(y_1^m)\}_{1 \leq j \leq k}$ and $U_0(y_1^m)$, the average utility

$$\bar{U}(x_1^n) = \sum_{1 \leq j \leq k} \sum_{y_1^m} P(a_j \text{ taken} | x_1^n, y_1^m) \cdot U_j(y_1^m) + \sum_{y_1^m} P(\text{no action taken} | x_1^n) \cdot U_0(y_1^m)$$

is maximized as follows:

by action a_{j^*} ; if

$$A_{j^*}(x_1^n) \stackrel{\Delta}{=} p_{j^*}(x_1^n) \sum_{y_1^m} q_{j^*}(y_1^m) U_{j^*}(y_1^m) = \max_{1 \leq j \leq k} p_j(x_1^n) \sum_{y_1^m} q_j(y_1^m) U_j(y_1^m) > \sum_{y_1^m} q_0(y_1^m) U_0(y_1^m) \quad (3)$$

by no action; if

$$\sum_{y_1^m} q_0(y_1^m) U_0(y_1^m) > \max_{1 \leq j \leq k} p_j(x_1^n) \sum_{y_1^m} q_j(y_1^m) U_j(y_1^m) \quad (4)$$

$A_{j^*}(x_1^n)$ in (3) is the award assigned to action a_{j^*} ; it is also the worth assigned to the precondition vector value x_1^n by the action a_{j^*} .

If more than one action attain the maximum award $A_{j^*}(x_1^n)$ in (3), one of them is selected randomly.

4 Solutions of the network propagation problem

In this section, we generalize the core problem solutions expressed in Theorem 1, Section 3, to the sequence of actions depicted by the ATIN in Fig. 1.

Problem 1 (The Optimal Path Problem)

In the ATIN in Fig. 1, we fix the preconditions vector value $x_1^n(1)$, at time 1, and the effects' vector value $y_1^m(N)$, at time N. We then search for the sequence of actions that maximizes the probability $P(y_1^m(N)|x_1^n(1))$. The solution to this problem follows a dynamic programming approach where $x_1^n(l) = y_1^m(l-1); 2 \leq l \leq N$, in our notation. The proof of the step evolution is included in the Appendix.

Step 1

For each $y_1^m(1) = x_1^n(2)$ value, find

$$r(y_1^m(1)) \stackrel{\Delta}{=} \max [q_0(y_1^m(1)), \max_j q_j(y_1^m(1)) p_j(x_1^n(1))]$$

and the action index $j^*(y_1^m(1))$ that attains $r(y_1^m(1))$.

Step l

The values $r(y_1^m(l-1)) \stackrel{\Delta}{=} \max P(y_1^m(l-1)|x_1^n(l))$, for each $y_1^m(l-1)$ value, are in memory, as well as the actions that attain them. At step l , the values $r(y_1^m(l)) \stackrel{\Delta}{=} \max_{y_1^m(l-1)} r(y_1^m(l-1)) \times \max [q_0(y_1^m(l)), \max_j q_j(y_1^m(l)) p_j(y_1^m(l-1))]$

are maintained, as well as the sequence of actions leading to them.

The complexity of this problem is polynomial with respect to the number of links. Assume that a given ATIN model has 'N' number of levels and each level has 'k' links, then the complexity is given as $O(N \times k)$.

Problem 2 (The Average Utility Maximization)

In the ATIN in Fig. 1, we fix the value of the precondition vector at time 1, denoted $x_1^n(1)$. For each value $y_1^w(N)$ of the effects vector at time N, we assign worth functions $U(y_1^w(N))$. For each action $a_j(l)$, at time l , we assign a deployment cost $c_j(l)$. The utility of the effects' vector value $y_1^w(N)$, when action $a_j(N)$ is taken, is then equal to $U_j(y_1^w(N)) \stackrel{\Delta}{=} U(y_1^w(N)) - c_j(N)$, while the utility of the same value, when no action is taken, equals $U_0(y_1^w(N)) \stackrel{\Delta}{=} U(y_1^w(N))$. We are seeking the sequence of actions which lead to the maximization of the average utility. The evolving algorithm, from part (b) of Theorem 1, back propagates as follows. The proof is in the Appendix.

Step 1

Compute the action awards (including that to no action), with notation of Figure 1, as follows: $0 \leq j \leq r$;

$$A_j(x_1^l(N-1)) \stackrel{\Delta}{=} p_j(x_1^l(N-1)) \sum_{y_1^w(N)} q_j(y_1^w(N)) U_j(y_1^w(N))$$

with $p_0(x_1^l(N-1)) \stackrel{\Delta}{=} 1$

Select $A_{j^*(x_1^l(N-1))}(x_1^l(N-1)) \stackrel{\Delta}{=} \max_{0 \leq j \leq r} A_j(x_1^l(N-1))$;

for each $x_1^l(N-1)$ value.

Take action $a_{j^*(x_1^l(N-1))}(N)$ for preconditions vector value $x_1^l(N-1)$ and simultaneously assign worth $A_{j^*(x_1^l(N-1))}(x_1^l(N-1))$ to $x_1^l(N-1)$. That is, assign: $U(x_1^l(N-1)) = A_{j^*(x_1^l(N-1))}(x_1^l(N-1))$ (5)

Step 2

Back propagate to the preconditions at N-2, as in Step 1, starting with the worth assignments in (5), and subsequent utilizations

$$U_j(x_1^l(N-1)) = \max [A_{j^*(x_1^l(N-1))}(x_1^l(N-1)) - c_j(N-1), 0]$$

Step n

As in Steps 1 and 2 (for subsequent levels) the above described algorithm generates the optimal sequence of actions for given initial preconditions $x_1^n(1)$. The optimal such preconditions can be also found via maximization of the utility $U_j(x_1^k(2))$, with respect to $x_1^n(1)$.

The complexity of this problem is also polynomial with respect to the number of links.

Problems 3a, 3b (Optimization with Constrained Actions)

Problems 3a and 3b impose dependency constraints on the actions in the ATIN network. As explained in Section 2, an ADM defines the dependency of one action on every other one, where positive dependency is depicted by 1 and negative dependency is depicted by 0. The dependency constraints are taken into account, when, at a certain level, an optimal action is finalized. At any given level, only positively related actions are considered in the calculations.

As described in Step 1 of Problem 1 (see Section 4), for the first level, $r(y_1^m(1))$ is calculated the same way for constrained actions also. But for the rest of the levels, it is calculated in a different manner.

Consider,

$$r(y_1^m(l)) \stackrel{\Delta}{=} \max_{y_1^m(l-1)} r(y_1^m(l-1)) \times \max [q_0(y_1^m(l)), \max_j q_j(y_1^m(l)) p_j(y_1^m(l-1))]$$

The parameter $\max_{y_1^m(l-1)} r(y_1^m(l-1))$ corresponds to an ac-

tion selected for execution in level $l-1$. Its dependent actions can be known from the ADM. In this way, those combinations of actions which are not allowed by the ADM are eliminated from the calculation of $r(y_1^m(l))$, hence eliminating all links to and from the actions exhibiting negative dependencies. As a result of which it yields a network with lesser number of links and eases the determination of optimal sequence of actions.

5 Numerical evaluations

In this section, we focus on numerical scenarios. We first state the experimental setup. We then, evaluate and discuss a specific experimental scenario. We only state the experimental setups for Problems 1 and 2, since those of Problems 3a and 3b are straight forward modifications of the former.

5.1 Experimental Setups

Experimental Setup for Problem 1

Assign the probabilities

$\{q_j(x_1^k(l))\}$ and $\{p_j(x_1^k(l))\}$ as in problem 2.

Given these probabilities:

a. Compute first:

$$r(y_1^m(1)) \stackrel{\Delta}{=} \max_{y_1^m(1)} [q_0(y_1^m(1)), \max_j q_j(y_1^m(1)) p_j(x_1^n(1))]$$

and the action $j^*(y_1^m(1))$ that attains $r(y_1^m(1))$.

b. For each $l: 2 \leq l \leq N$, maintain in memory the values

$$r(y_1^m(l-1)) \stackrel{\Delta}{=} \max P(y_1^m(l-1) | x_1^n(l)), \text{ for each } y_1^m(l-1) \text{ value, and the actions that attain them. Then, compute and maintain the values:}$$

$$r(y_1^m(l)) \stackrel{\Delta}{=} \max_{y_1^m(l-1)} r(y_1^m(l-1)) \times \max_j [q_0(y_1^m(l)), \max_j q_j(y_1^m(l)) p_j(y_1^m(l-1))]$$

Also, maintain the actions that attain the values $r(y_1^m(l))$.

Experimental Setup for Problem 2

Considering the network in Fig. 1, assign:

a. Worth function $U(y_1^w(N))$ for all $y_1^w(N)$ values of the effects' status vector, at level N .

b. Probabilities $q_j(x_1^k(l)) \stackrel{\Delta}{=} P(x_1^k(l) \text{ occurring} | \text{action } j \text{ at step } l-1) \text{ at all levels, } 2 \text{ to } N,$

where $q_0(x_1^k(l)) \stackrel{\Delta}{=} P(x_1^k(l) \text{ occurring} | \text{no action } j \text{ at step } l-1) \text{ at all levels, } 2 \text{ to } N,$

c. Probabilities $p_j(x_1^k(l)) \stackrel{\Delta}{=} P(\text{action } j \text{ succeeds} | x_1^k(l) \text{ preconditions}) \text{ at all levels, from } 1 \text{ to } N-1,$

where $p_0(x_1^k(l)) \stackrel{\Delta}{=} 1; \forall l$

d. Implementation/deployment costs $c_j(l)$ for all actions, at all levels 2 to N .

Given the above assignments,

a. Compute first,

$$A_j(x_1^l(N-1)) \stackrel{\Delta}{=} p_j(x_1^l(N-1)) \sum_{y_1^w(N)} q_j(y_1^w(N)) U_j(y_1^w(N))$$

where,

$$p_0(x_1^l(N-1)) \stackrel{\Delta}{=} 1;$$

$$U_j(y_1^w(N)) = \max [U(y_1^w(N)) - c_j(N), 0]$$

$$A_{j^*(x_1^l(N-1))}(x_1^l(N-1)) \stackrel{\Delta}{=} \max_{0 \leq j \leq r} A_j(x_1^l(N-1)) ;$$

for all $x_1^l(N-1)$ values.

b. Take action $a_{j^*(x_1^l(N-1))}$ for each precondition vector value $x_1^l(N-1)$.

Assign worth $A_{j^*(x_1^l(N-1))}(x_1^l(N-1))$ to $x_1^l(N-1)$, as

$$U(x_1^l(N-1)) = A_{j^*(x_1^l(N-1))}(x_1^l(N-1))$$

Repeat steps (a) and (b) for level $N-1$ and back propagate to level $N-2$. Continue back propagation to level 1.

5.2 A Specific Experimental Scenario

In this section, we illustrate the use of Activation Timed Influence Nets with the help of an example ATIN, and present the results of the algorithms included in this paper, when applied to this ATIN. The model used in this section was derived from a Timed Influence Net presented in Wagenhals et al., in 2001 [27] (which was developed with the help of a team of subject matter experts) to address the internal political instabilities in Indonesia in the context of East Timor. For purposes of results illustration, we have selected a part of this network, as shown in Fig. 4.

Example ATIN:

The model provides detailed information about the religious, ethnic, governmental and non-governmental organizations of Indonesia. In this section, the propositions and actions referred are given in *italic* text. According to the model, rebel militia formed by a minority group poses the main concern which has captured a large number of people under its secured territory. Amongst these people in the community, some are against the rebels and considered to be at risk, in case the negotiations with the local government didn't work. For this example,

consider the initial conditions when *the rebels are getting local support*, the *community is in unrest* and the *local administration is losing control*. Based on the data provided, only one action can be executed from a possible set of actions at a given time i.e. either of the *Indonesian press* or *provincial authority* or the *minister of interior would declare resolve to keep peace*. Depending upon this selected action and the data provided for the effects, only a specific set of events can result. For instance, *rebels may or may not start thinking that they are getting publicity*, *GOI (original anti-government of Indonesia) war may or may not expand*, *GOI chances of intervention and international attention may increase or decrease*. Similarly, this specific set of events forms the set of possible pre-conditions for a later time. Depending upon which conditions actually become true, second action can be selected for execution from another set of actions, i.e. *Security Council* and *General Assembly* may or may not pass resolutions or *UN may or may not declare resolve to keep peace*. Depending upon this action and the data provided for the effects, *coalition may or may not form*, *rebels may or may not contemplate talks*, *GOI support may increase or decrease or may not increase at all*, or *GOI may or may not allow coalition into territories*. Ultimately, the *coalition may authorize use of force* which might compel *rebels to negotiate* and the *humanitarian assistance (HA) may start preparing for the worst case*. Depending upon which conditions meet, the *coalition may declare resolve to keep peace* or *may declare war on rebels*. This may affect the *chances of military confrontation*, *rebels' popularity* and *chances of negotiated settlement* which represents the final effects in the network.

Table 1 lists some of the parameters (and their values) required by the network in Fig. 4. The parameters in the table are listed by their abbreviated labels also in addition to the phrases shown inside the network nodes in the figure. For the sake of brevity, we do not list all the values.

Solutions to Problems:

Solution to Problem 1 (Optimal Path Problem):

Consider the example scenario described earlier, we need

to identify an optimal path (i.e., the sequence of actions) resulting into the final effect when, *military confrontation chances are reduced*, while *rebels start losing local support* and *negotiation chances start increasing*. This set of effects (post-conditions) leads to the following output state in the ATIN model:

- *Reduction in the chances of military confrontation* (i.e. $Y_{12} = 0$)
- *Decrease in local support and popularity for Rebels* (i.e. $Y_{13} = 1$)
- *Increase in chances of negotiated settlement* (i.e. $Y_{14} = 1$).

The above defined conditions lead to a post-condition vector $[0, 1, 1]^T$ at level 4, i.e. $y_{12}^{14}(4)$.

After fixing the post-condition vector, we define the initial preconditions, when *rebels have been getting local support*, the *community has been in unrest* and the *local administration has started losing control*. This set of pre conditions given by $x_1^3(1)$ results into a vector value of $[1, 1, 1]^T$, where

- $X_1 = 1$; represents the condition *Rebels are getting Local Support*
- $X_2 = 1$; represents the condition *There is unrest in the Community*
- $X_3 = 1$; represents the condition *Local Administration is losing Local Control*.

We want to find out the sequence of actions which achieves the desired effects $y_{12}^{14}(4)$ given the initial preconditions $x_1^3(1)$. Technically, we want to identify the sequence of actions which maximizes the probability $P(y_{12}^{14}(4) | x_1^3(1))$. Applying the optimal path algorithm (see Section 4) results that if the *provincial authority* and *UN declare resolve to keep peace* and *coalition does not take any action*, instead it *declares resolve to keep peace*, then the desired effects will be achieved which will result into *less chances of military confrontation*, *reduction in local support for rebels* and *more chances of a negotiated settlement*.

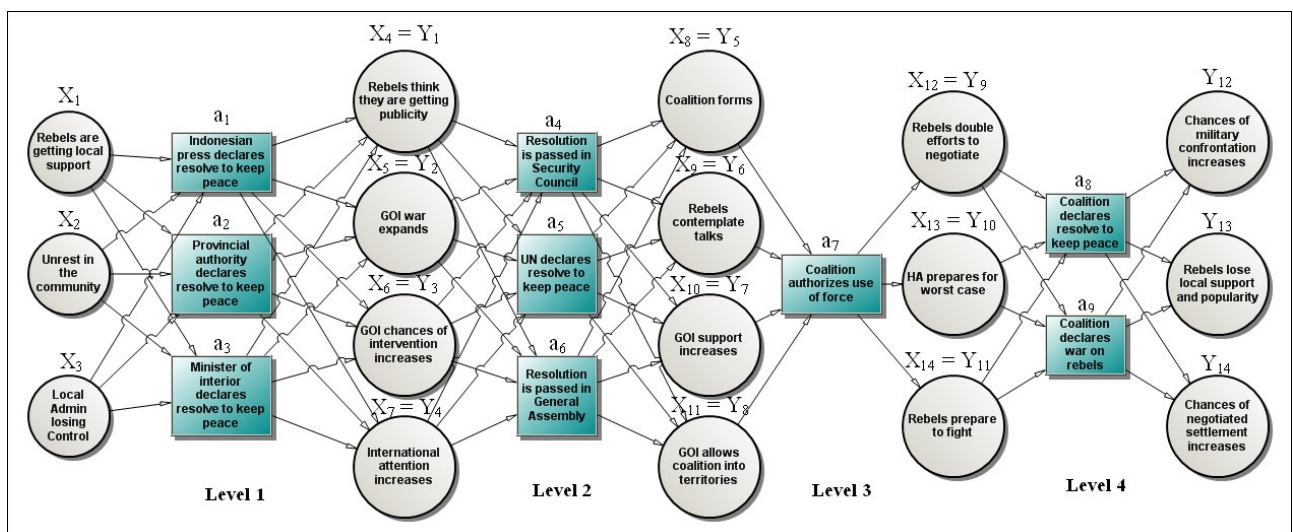


Figure 4: Example ATIN.

Table 1: Parameter values in the Example ATIN

Level 1							
Y_1^4	Action a_j	$q_j(y_1^4)$	$q_0(y_1^4)$	X_1^3	$p_j(x_1^3)$	$p_j(x_1^3) * q_j(y_1^4)$	$r(y_1^4)$
[1,0,1,1] ^T	Indonesian press declares resolve to keep peace (a1)	68.00%	6.45%	[1,1,1] ^T	80.00%	54.40%	63.64%
	Provincial Authority declares resolve to keep peace (a2)	74.00%			86.00%	63.64%	
	Minister of interior declares resolve to keep peace (a3)	56.00%			20.00%	11.20%	
Level 2							
Y_5^8	Action a_j	$q_j(y_5^8)$	$q_0(y_5^8)$	X_4^7	$p_j(x_4^7)$	$p_j(x_4^7) * q_j(y_5^8)$	$r(y_5^8)$
[1,1,1,1] ^T	Resolution is passed in Security council (a4)	14.00%	0.95%	[0,0,0,0] ^T	16.00%	2.24%	34.02%
				
				[1,0,1,1] ^T	91.00%	12.74%	
			
	[1,1,1,1] ^T	18.00%		2.52%			
	UN declares resolve to keep peace (a5)	42.00%		[0,0,0,0] ^T	66.00%	27.72%	
				
				[1,0,1,1] ^T	81.00%	34.02%	
			
	[1,1,1,1] ^T	13.41%		5.63%			
	Resolution is passed in General Assembly (a6)	39.00%		[0,0,0,0] ^T	43.05%	16.79%	
				
[1,0,1,1] ^T			48.20%	18.80%			
...					
[1,1,1,1] ^T	24.87%	9.70%					
Level 3							
Y_9^{11}	Action a_j	$q_j(y_9^{11})$	$q_0(y_9^{11})$	X_8^{11}	$p_j(x_8^{11})$	$p_j(x_8^{11}) * q_j(y_9^{11})$	$r(y_9^{11})$
[1,0,0] ^T	Coalition Authorizes the Use of Force (a7)	43.00%	59.00% (No Action)	[0,0,0,0] ^T	0.00%	0.00%	59.00%
				
				[1,0,1,1] ^T	64.00%	27.52%	
				
[1,1,1,1] ^T	18.00%	7.74%					
Level 4							
Y_{12}^{14}	Action a_j	$q_j(y_{12}^{14})$	$q_0(y_{12}^{14})$	X_{12}^{14}	$p_j(x_{12}^{14})$	$p_j(x_{12}^{14}) * q_j(y_{12}^{14})$	$r(y_{12}^{14})$
[0,1,1] ^T	Coalition declares resolve to keep peace (a8)	21.00%	1.50%	[0,0,0] ^T	16.00%	3.36%	19.11%
				
				[1,0,0] ^T	91.00%	19.11%	
			
	[1,1,1] ^T	38.00%		7.98%			
	Coalition declares war on rebels (a9)	17.00%		[0,0,0] ^T	67.00%	11.39%	
...					
[1,0,0] ^T			18.48%	3.14%			
...					
[1,1,1] ^T	30.88%	5.25%					

The details of this result are given in Table 1. It only contains the values that correspond to the selected actions at their respective levels, while a complete set of probabilities has been used to calculate the actual final sequence. The optimal actions, their corresponding state vectors and the probabilities are underlined in the table. The Optimal Path algorithm is of dynamic programming nature, so it requires two traversals to finalize the sequence of actions. During the forward traversal, $r(y_1^m)$ is calculated for each level for all possible post-condition combinations. At the last level, the post-condition vector $y_{12}^{14}(4)$ is fixed to be the desired effect of the network which is $[0, 1, 1]^T$ as determined earlier. The best action associated with this post-condition vector is identified along with its pre-condition vector $x_{12}^{14}(4)$. Using this pre-condition vector (which is the post-condition vector of the second last level), the network is traversed in reverse direction identifying actions and their corresponding preconditions, from last to the first level. The action at the first level is identified by fixing the pre-condition to the value determined earlier, i.e. $x_1^3(1)$ which is $[1, 1, 1]^T$. Completing both forward and reverse traversals gives the optimal actions which achieve the desired effects when the initial causes are given.

Solution to Problem 2 (Average Utility Maximization):

Consider a scenario where we need to identify the sequence of actions which maximizes the effects' average utility (at level 4) for the same input pre-condition as it was used in the solution of Problem 1, i.e. $[1, 1, 1]^T$. Assume, that the deployment costs for actions a_8 and a_9 are 25 and 30 units, respectively. The worth of each effect in the last level (i.e. level 4) is given by the worth function values $U(y_{12}^{14}(4))$ given in Table 2 and 3. Each effect also has a net utility which is determined by subtracting the deployment cost of the action from the worth of the effect. This net utility $U_j(y_{12}^{14}(4))$ (when action a_j is taken) and the action awards are given in Tables 2 and 3. The action award is calculated for each action corresponding to all of its pre-conditions. Similarly, these calculations are performed for the rest of the actions in ATIN model (after costs are assigned to every action in the model), but for the sake of brevity only the results for actions a_8 and a_9 are shown in Tables 2 and 3, respectively.

As described in Section 4, the action award is calculated for all actions in each level. For instance, starting from the last level, the action awards are calculated for actions a_8 and a_9 . The selected action is the one which maximizes the average utility and its action index 'j' is recorded. As each action award is calculated, it is also assigned as the worth function to the previous level effects vector. The latter worth function is used to calculate the utilities at the previous level, and calculations are repeated similarly. This procedure is back traversed from last to first levels. Table 4 summarizes the action awards of those actions which maximize the effects' average utility at their respective levels.

Table 2: Utility Functions and Action awards for Action a_8

Level 4 – Action a_8						
$X_{12}^{14}(4)$	$p_8(x_{12}^{14}(4))$	$Y_{12}^{14}(4)$	$q_8(y_{12}^{14}(4))$	$U(y_{12}^{14}(4))$	$U_8(y_{12}^{14}(4))$	$A_8(x_{12}^{14}(4))$
$[0,0,0]^T$	16.00%	$[0,0,0]^T$	37.00%	40	15	11.11
$[0,0,1]^T$	24.00%	$[0,0,1]^T$	65.00%	30	5	16.66
$[0,1,0]^T$	75.00%	$[0,1,0]^T$	53.00%	60	35	52.07
$[0,1,1]^T$	85.00%	$[0,1,1]^T$	21.00%	79	54	59.02
$[1,0,0]^T$	91.00%	$[1,0,0]^T$	19.00%	41	16	11.11
$[1,0,1]^T$	72.00%	$[1,0,1]^T$	43.00%	65	40	49.99
$[1,1,0]^T$	16.00%	$[1,1,0]^T$	29.00%	37	12	63.18
$[1,1,1]^T$	38.00%	$[1,1,1]^T$	27.00%	51	26	26.38

Table 3: Utility Functions and Action awards for Action a_9 .

Level 4 – Action a_9						
$X_{12}^{14}(4)$	$p_9(x_{12}^{14}(4))$	$Y_{12}^{14}(4)$	$q_9(y_{12}^{14}(4))$	$U(y_{12}^{14}(4))$	$U_9(y_{12}^{14}(4))$	$A_9(x_{12}^{14}(4))$
$[0,0,0]^T$	67.00%	$[0,0,0]^T$	41.00%	40	10	48.25
$[0,0,1]^T$	97.15%	$[0,0,1]^T$	26.00%	<u>30</u>	<u>0</u>	<u>69.96</u>
$[0,1,0]^T$	58.29%	$[0,1,0]^T$	71.00%	60	30	41.97
$[0,1,1]^T$	13.00%	$[0,1,1]^T$	17.00%	79	49	9.36
$[1,0,0]^T$	18.48%	$[1,0,0]^T$	26.00%	41	11	13.31
$[1,0,1]^T$	39.28%	$[1,0,1]^T$	54.00%	65	35	28.29
$[1,1,0]^T$	38.67%	$[1,1,0]^T$	62.00%	37	7	27.85
$[1,1,1]^T$	30.88%	$[1,1,1]^T$	58.00%	51	21	22.24

From Table 4 it can be seen that the sequence of actions that maximizes the effects' average utility, obtained as a result of applying the algorithm is given by: a_1 (i.e. *Indonesian press declares resolve to keep peace*), a_6 (i.e. *Resolution is passed in General Assembly*), a_7 (i.e. *Coalition authorizes use of Force*), a_9 (i.e., *Coalition declares war on rebels*). The underlined entries in Table 3 correspond to the worth, utility function and action award of action a_9 .

Solution to Problem 3a, 3b (Constrained Actions):

The dependencies among the actions in the example ATIN model are defined in the action dependency matrix given in Figure 5.

Most of the dependencies given in the matrix are quite evident. For instance, *the peace resolution declaration by UN (a5)* ensures that either of *Indonesian press, provincial authority or minister of interior must also have declared the resolution to keep peace* (either of a_1 or a_2 or a_3 must have been executed in the past) which would represent the opinion of the locals in general. Similarly, *resolution passed by the Security Council or General Assembly (a4 or a6)* makes sure that *whether or not the coalition will have to authorize the use of force (a7)*, considering the resolution is in support of use of force. This infers that if *the coalition authorizes the use of force*, it will *declare war on Rebels* otherwise, it will *declare resolve to keep peace*. All of these dependencies can be observed from the ADM (as shown in Fig. 5).

Consider a_{25} in ADM, (as shown in Fig. 5) which

corresponds to a positive dependency between *peace declaration by the provincial authority* (a_2) and *peace declaration by UN* (a_5). The ADM suggests that there exist negative dependencies between action a_2 and actions a_4 , a_6 , a_7 and a_9 which means that if *Provincial authority declares peace resolution, Security Council and General Assembly won't pass resolution and the Coalition will not authorize the use of force and hence will declare resolve to keep peace*. This knowledge of dependencies from the ADM certainly reduces an extensive amount of effort in calculating the optimal path. While calculating the optimal path, during the forward traversal, only those paths are considered which satisfy the constraints defined in ADM yielding less number of combinations to consider for calculation and making it easy to back traverse and identify the optimal actions.

The same applies to the solution of the second problem of identifying sequence of actions maximizing the effects' average utility under constraints. The action awards are calculated for those actions only which satisfy constraints defined in ADM, and hence reducing the effort of calculating action awards and assignment of worth function at each level.

Table 4: Action Awards.

Level 1	Level 2	Level 3	Level 4
$A_1(x_1^3(1))$	$A_6(x_4^7(2))$	$A_7(x_8^{11}(3))$	$A_9(x_{12}^{14}(4))$
151.02	85.18	77.92	69.96

6 Conclusion

This paper presented an extension of a Timed Influence Net, termed ATIN (Activation Timed Influence Net). An ATIN utilizes a set of preconditions required for the undertaking of an action and produces a set of effects. These effects become then the preconditions for the next level of action(s), resulting in a sequential evolution of actions. Some other probabilistic planning techniques were also discussed. The paper identified several pre-

selected performance criteria regarding ATINs (i.e., optimal path and average utility maximization with and without constrained actions) and recommended algorithms for their satisfaction. A tool called ADM (Action Dependency Matrix) was introduced, which induces dependencies among the actions. It is represented with the help of a $m \times m$ matrix, where ' m ' represents the total number of actions in the network.

The implementation of the suggested algorithms was illustrated with the help of a real world example. The example demonstrated a politically unstable situation in Indonesia. Sets of actions preceded by preconditions and followed by sets of effects were demonstrated in the form of an ATIN Model (see Figure 4). The experiment was formulated based on a previous Timed Influence Network model for the same scenario. The experimental procedure was applied to the network with a set of probability data. Solutions of both problems were discussed in depth. The optimal path problem required the knowledge of an initial set of causes (preconditions) and the final set of effects (postconditions). With the help of the algorithm, an optimal sequence of actions was identified which maximized the conditional probability of achieving the desired effects, when the initial conditions were given. For the sake of brevity, only significant parts of the probability data used were shown in Table 1. For the same scenario, the second algorithm yielded a sequence of actions, which maximized the effects' average utility. The solution for both problems was comprehended in detail. The experiment was repeated with constrained actions considering only dependent actions as defined in the Action Dependency Matrix (see Figure 5) which produced similar results and required lesser effort to calculate than without ADM.

References

[1] Bagchi S., Biswas G., and Kawamura K., (Nov, 2000), "Task Planning under Uncertainty using a Spreading Activation Network", IEEE Transactions

	a1 Indonesian Press declares resolve to keep peace	a2 Provincial Authority Press declares resolve to keep peace	a3 Minister of Interior Press declares resolve to keep peace	a4 Resolution is passed in Security council	a5 UN Declares resolve to keep peace	a6 Resolution is passed in General Assembly	a7 Coalition authorizes use of Force	a8 Coalition declares resolve to keep peace	a9 Coalition declares war on Rebels
a1 Indonesian Press declares resolve to keep peace	1	0	0	0	1	0	0	1	0
a2 Provincial Authority Press declares resolve to keep peace	0	1	0	0	1	0	0	1	0
a3 Minister of Interior Press declares resolve to keep peace	0	0	1	0	1	0	0	1	0
a4 Resolution is passed in Security council	0	0	0	1	0	0	1	0	1
a5 UN Declares resolve to keep peace	1	1	1	0	1	0	0	1	0
a6 Resolution is passed in General Assembly	0	0	0	0	0	1	1	0	1
a7 Coalition authorizes use of Force	0	0	0	1	0	1	1	0	1
a8 Coalition declares resolve to keep peace	1	1	1	0	1	0	0	1	0
a9 Coalition declares war on Rebels	0	0	0	1	0	1	1	0	1

Figure 5: Action Dependency Matrix

- on Systems, Man and Cybernetics, Vol. 30, No. 6, pp 639-650.
- [2] Dean T. and Wellman (1991) M.P., Planning and Control. San Mateo, CA: Morgan Kaufmann.
 - [3] Dean T., Kaelbling L.P., Kirman J., and Nicholson A., (1993), "Planning with deadlines in stochastic domains," in Proc. 11th Nat. Conf. Artificial Intelligence (AAAI-93). Chicago, IL: AAAI Press, pp. 574–579.
 - [4] Drummond M. and Bresina J., (1990), "Anytime synthetic projection: Maximizing the probability of goal satisfaction," in Proc. 8th Nat. Conf. Artificial Intelligence (AAAI-90). Chicago, IL: AAAI Press, pp.138–144.
 - [5] Firby R.J., (1987), "An investigation into reactive planning in complex domains,"in Proc. 6th Nat. Conf. Artificial Intelligence (AAAI-87). San Mateos, CA: Morgan Kaufmann, vol. 1, pp. 202–206.
 - [6] Haider S., (October, 2003), "On Computing Marginal Probability Intervals in Probabilistic Inference Networks," in Proceedings of IEEE-SMC Conference.
 - [7] Haider S. (August, 2003), "A Hybrid Approach for Learning Parameters of Probabilistic Networks from Incomplete Databases," Design and Application of Hybrid Intelligent Systems, Proceedings of the Third International Conference on Hybrid Intelligent Systems (HIS'03), IOS Press, Amsterdam, The Netherlands, ISBN 158603-394.
 - [8] Haider S., Zaidi A. K., and Levis A. H. (Nov. 2004), "A Heuristic Approach for Best Set of Actions Determination in Influence Nets," in Proc. IEEE International Conference on Information Reuse and Integration, Las Vegas.
 - [9] Haider S. and Levis A. H. (2004), "An Approximation Technique for Belief Revision in Timed Influence Nets," in Proceedings of Command and Control Research and Technology Symposium.
 - [10] Haider S. and Zaidi A. K. (2004), "Transforming Timed Influence Nets into Time Sliced Bayesian Networks," in Proceedings of Command and Control Research and Technology Symposium.
 - [11] Haider S. and Zaidi A. K. (2005), "On Temporal Analysis of Timed Influence Nets using Point Graphs," in the Proc. of the 18th International FLAIRS Conference, FL.
 - [12] Haider S. and Levis A. H. (June 2005), "Dynamic Influence Nets: An Extension of Timed Influence Nets for Modeling Dynamic Uncertain Situations," in Proc. 10th International Command and Control Research and Technology Symposium, Washington DC.
 - [13] Kushmerick N., Hanks S., and Weld D., (1995), "An algorithm for probabilistic planning," Univ. of Washington, Seattle, Artificial Intelligence Journal, pp. 76(1-2):239-86
 - [14] Likhachev M. and Stentz A., (2006), "PPCP: Efficient Probabilistic Planning with Clear References in Partially Known Environments", Twenty-First AAAI Conference on Artificial Intelligence, Boston, Massachusetts, The AAAI Press/Menlo Park, California, pp 860-868.
 - [15] Maes P., (1990), "Situated agents can have goals," Robot. Auton. Syst., vol. 6, pp. 49-70.
 - [16] Mausam and Weld D.S., "Probabilistic Temporal Planning with Uncertain Durations", Twenty-First AAAI Conference on Artificial Intelligence, Boston, Massachusetts, The AAAI Press/Menlo Park, California, pp 880-888.
 - [17] Mitchell T.M., (1990), "Becoming increasingly reactive," in Proc. 8th Nat.Conf. Artificial Intelligence (AAAI-90). Chicago, IL: AAAI Press/The MIT Press, pp. 1051–1058.
 - [18] Nikolova E. and Karger D.R., "Route Planning under Uncertainty: The Canadian Traveler Problem", Twenty-Third AAAI Conference on Artificial Intelligence, Chicago, Illinois, The AAAI Press/Menlo Park, California, pp 969-975.
 - [19] Tate A., Hendler J., and Drummond M., (1990), "A review of AI planning techniques,"in Readings in Planning, J. Allen, J. Hendler, and A. Tate,Eds. San Mateo, CA: Morgan Kaufmann, pp. 26–49.
 - [20] Wagenhals L. W. and Levis A. H. (June 2002), "Modeling Support of Effects-Based Operations in War Games," in Proc. 2002 Command and Control Research and Technology Symposium, Monterey, CA.
 - [21] Wagenhals L.W., Levis A. H., and McCrabb M. (June 2003), "Effects Based Operations: a Historical Perspective for a Way Ahead," in Proc. 8th Int'l Command and Control Research and Technology Symposium, National Defense University, Washington, DC.
 - [22] Wagenhals L. W. and Wentz L. K., (June 2003), "New Effects-Based Operations Models in War Games," in Proceedings of the 2003 International Command and Control Research and Technology Symposium, National Defense University, Washington, DC.
 - [23] Wagenhals L. W. (2000), "Course of Action Development and Evaluation Using Discrete Event System Models of Influence Nets", PhD Dissertation, George Mason University.
 - [24] Wagenhals L. W., Levis A. H. (2000), "Course of Action Development and Evaluation," in Proceedings of the 2000 Command and Control Research and Technology Symposium.
 - [25] Wagenhals L. W. and Levis A. H., "Modeling Effects Based Operations in Support of War Games," in Proc. Of the 15th International Symposium on Aerospace / Defense Sensing, Internal Society for Optical Engineering, Proceedings of SPIE, Vol. # 4367, 2001.
 - [26] Wagenhals L. W., Shin I., and Levis A. H., (1998) "Creating Executable Models of Influence Nets with Coloured Petri Nets," Int. J. STTT, Springer-Verlag, Vol. 1998, No. 2.
 - [27] Wagenhals L. W., Reid T. J., Smillie R. J., & Levis A. H. (June 2001), "Course of Action Analysis for Coalition Operations," Proceedings of 6th Interna-

- tional Command and Control Research and Technology Symposium, Annapolis, Maryland.
- [28] Wellman M. P., (1990), “Formulation of tradeoffs in planning under uncertainty”. San Mateo, CA: Morgan Kaufmann.
- [29] Wilkins D.E., (1988), “Practical Planning: Extending the Classical AI Planning Paradigm”. San Mateo, CA: Morgan Kaufmann.
- [30] Zaidi A. K., Wagenhals L W., and Haider S. (2005), “Assessment of Effects Based Operations Using Temporal Logic,” in Proc. of the 10th International Command and Control Research and Technology Symposium.
- [31] Zaidi A. K., Mansoor F., and Papantoni-Kazakos P. (October 2007), “Modeling with Influence Networks Using Influence Constants: A New Approach,” in Proceedings of IEEE-SMC Conference.

Appendix

Proof of Theorem 1

In the derivations below, the following considerations are incorporated:

1. Effects are fully dictated by the actions taken; thus, when probabilities are conditioned on actions and preconditions, the conditioning on preconditions drops.
2. By probability of action success, we mean the probability that the action may succeed, given the preconditions. The final action is selected among those that have positive probability of success. The probability of action taken, given that the action may succeed is the criterion that dictates the final action selection.

$$(a) \quad P(y_1^m | x_1^n) = \sum_{1 \leq j \leq k} P(y_1^m, a_j \text{ taken} | x_1^n) + P(y_1^m, \text{no action taken} | x_1^n) \quad (1.1)$$

where

$$\begin{aligned} P(y_1^m, a_j \text{ taken} | x_1^n) &= \\ &= P(y_1^m | a_j \text{ taken}, x_1^n) P(a_j \text{ taken} | x_1^n) = \\ &= P(y_1^m | a_j \text{ taken}) P(a_j \text{ taken} | x_1^n) = \\ &= P(a_j \text{ taken} | x_1^n) q_j(y_1^m) \\ &= [P(a_j \text{ taken}, \text{succ for action } a_j | x_1^n) + \\ &+ P(a_j \text{ taken}, \text{no succ for action } a_j | x_1^n)] \times \\ &\times q_j(y_1^m) \\ &= [P(a_j \text{ taken} | \text{succ for action } a_j, x_1^n) \times \\ &\times P(\text{succ for action } a_j | x_1^n) + \\ &+ P(a_j \text{ taken} | \text{no succ for action } a_j, x_1^n) \times \\ &\times P(\text{no succ for action } a_j | x_1^n)] q_j(y_1^m) \end{aligned}$$

$$\begin{aligned} &= [P(a_j \text{ taken} | \text{succ for action } a_j) \times \\ &\times P(\text{succ for action } a_j | x_1^n) + \\ &+ P(a_j \text{ taken} | \text{no succ for action } a_j) \times \\ &\times P(\text{no succ for action } a_j | x_1^n)] q_j(y_1^m) \\ &= P(a_j \text{ taken} | \text{succ for action } a_j) \times \\ &\times p_j(x_1^n) q_j(y_1^m) \\ &\quad \{\text{Using } P(a_j \text{ taken} | \text{no succ for action } a_j) = 0\} \end{aligned}$$

Equating in (1.1)

$$\begin{aligned} P(y_1^m | x_1^n) &= \\ &= \sum_{1 \leq j \leq k} P(a_j \text{ taken} | \text{succ for action } a_j) \times \\ &\times p_j(x_1^n) q_j(y_1^m) + P(y_1^m, \text{no action taken} | x_1^n) \quad (1.2) \end{aligned}$$

where

$$\begin{aligned} P(y_1^m, \text{no action taken} | x_1^n) &= \\ &= P(y_1^m | \text{no action taken}, x_1^n) P(\text{no action taken} | x_1^n) \\ &= P(y_1^m | \text{no action taken}) P(\text{no action taken} | x_1^n) \\ &= P(\text{no action taken} | x_1^n) q_0(y_1^m) \\ &= P(\text{no action taken}, \text{no action succ} | x_1^n) q_0(y_1^m) \\ &= P(\text{no action taken} | \text{no action succ}) q_0(y_1^m) \\ &\quad \{\text{Using } P(\text{no action succ} | x_1^n) = 1\} \end{aligned}$$

Equating in (1.2)

$$\begin{aligned} P(y_1^m | x_1^n) &= \\ &= \sum_{1 \leq j \leq k} P(a_j \text{ taken} | \text{succ for action } a_j) p_j(x_1^n) q_j(y_1^m) + \\ &+ P(\text{no action taken} | \text{no action succ}) q_0(y_1^m) \end{aligned}$$

$\Rightarrow \max P(y_1^m | x_1^n)$ attained
 if $P(a_{j^*} \text{ taken} | \text{succ for action } a_{j^*}) = 1$;
 for $p_{j^*}(x_1^n)q_{j^*}(y_1^m) = \max_{1 \leq j \leq k} p_j(x_1^n)q_j(y_1^m) > q_0(y_1^m)$
 otherwise, $\max P(y_1^m | x_1^n)$ attained
 if $P(\text{no action taken} | \text{no action succ}) = 1$;
 for $\max_{1 \leq j \leq k} p_j(x_1^n)q_j(y_1^m) < q_0(y_1^m)$

(b)

$$\begin{aligned}
 \bar{U}(x_1^n) &= \sum_{1 \leq j \leq k} \sum_{y_1^m} P(a_j \text{ taken}, y_1^m | x_1^n) U_j(y_1^m) + \\
 &+ \sum_{y_1^m} P(\text{no action taken}, y_1^m | x_1^n) U_0(y_1^m) \\
 &= \sum_{1 \leq j \leq k} \sum_{y_1^m} P(y_1^m | a_j \text{ taken}) P(a_j \text{ taken} | x_1^n) U_j(y_1^m) + \\
 &+ \sum_{y_1^m} P(y_1^m | \text{no action taken}) P(\text{no action taken} | x_1^n) \\
 &\times U_0(y_1^m) \\
 &= \sum_{1 \leq j \leq k} P(a_j \text{ taken} | \text{succ for action } a_j) p_j(x_1^n) \times \\
 &\times \sum_{y_1^m} q_j(y_1^m) U_j(y_1^m) + P(\text{no action taken} | \text{no action succ}) \times \\
 &\times \sum_{y_1^m} q_0(y_1^m) U_0(y_1^m)
 \end{aligned}$$

$\Rightarrow \max \bar{U}(x_1^n)$ attained for :
 $P(a_{j^*} \text{ taken} | \text{succ for action } a_{j^*}) = 1$;
 if $p_{j^*}(x_1^n) \sum_{y_1^m} q_{j^*}(y_1^m) U_{j^*}(y_1^m) =$
 $= \max_{1 \leq j \leq k} p_j(x_1^n) \sum_{y_1^m} q_j(y_1^m) U_j(y_1^m) > \sum_{y_1^m} q_0(y_1^m) U_0(y_1^m)$
 $P(\text{no action taken} | \text{no action succ}) = 1$;
 if $\sum_{y_1^m} q_0(y_1^m) U_0(y_1^m) > \max_{1 \leq j \leq k} p_j(x_1^n) \sum_{y_1^m} q_j(y_1^m) U_j(y_1^m)$

Proof of the Network Propagation – Problem 1

Using the notation in Section 4, Problem 1, and via the Theorem of Total Probability and the Bayes Rule, we obtain:

$$\begin{aligned}
 r(y_1^m(N)) &= \max_{\text{sequence of actions}} \Delta P(y_1^m(N) | x_1^n(1)) = \\
 &= \max_{\text{sequence of actions}} \sum_{y_1^m(N-1)} P(y_1^m(N), y_1^m(N-1) | x_1^n(1)) = \\
 &= \max_{\text{sequence of actions}} \sum_{y_1^m(N-1)} P(y_1^m(N) | y_1^m(N-1)) \times \\
 &\times P(y_1^m(N-1) | x_1^n(1)) \leq \max_{y_1^m(N-1)} \left[\max_{\text{action}} P(y_1^m(N) |
 \end{aligned}$$

$$\begin{aligned}
 &| y_1^m(N-1) \max_{\text{sequence of actions}} P(y_1^m(N-1) | x_1^n(1)) \Big] \\
 &= \max_{y_1^m(N-1)} \\
 &\left[\left\{ \max_{\text{action}} P(y_1^m(N) | y_1^m(N-1)) \right\} r(y_1^m(N-1)) \right]
 \end{aligned}$$

where, via Theorem 1 we have:

$$\begin{aligned}
 \max_{\text{action}} P(y_1^m(N) | y_1^m(N-1)) &= \\
 &= \max \left[\max_j p_j(y_1^m(N-1)) q_j(y_1^m(N)), q_0(y_1^m(N)) \right]
 \end{aligned}$$

Thus, via substitution principle, we obtain:

$$\begin{aligned}
 r(y_1^m(N)) &\leq \max_{y_1^m(N-1)} \left[r(y_1^m(N-1)) \times \right. \\
 &\times \max \left\{ \max_j p_j(y_1^m(N-1)) \times q_j(y_1^m(N)), q_0(y_1^m(N)) \right\} \Big]
 \end{aligned}$$

with equality iff the $y_1^m(N-1)$ value that attains $r(y_1^m(N-1))$ is selected. The above proves the general step in the network propagation of Problem 1.

Proof of the Network Propagation – Problem 2

Using the notation in Section 4, Problem 2, and via the use of the Theorem of Total Probability and the Bayes Rule, we obtain:

$$\begin{aligned}
 \max_{\text{sequence of actions } y_1^w(N)} \sum U(y_1^w(N)) P(y_1^w(N) | x_1^n(1)) &= \\
 &= \max_{\text{sequence of actions } y_1^w(N)} \sum U(y_1^w(N)) \sum_{x_1^l(N-1)} P(y_1^w(N), \\
 &x_1^l(N-1) | x_1^n(1)) = \\
 &= \max_{\text{sequence of actions } y_1^w(N)} \sum U(y_1^w(N)) \sum_{x_1^l(N-1)} P(y_1^w(N) | \\
 &| x_1^l(N-1)) P(x_1^l(N-1) | x_1^n(1)) = \\
 &= \max_{\text{sequence of actions } x_1^l(N-1)} \sum P(x_1^l(N-1) | x_1^n(1)) \times \\
 &\times \sum_{y_1^w(N)} U(y_1^w(N)) P(y_1^w(N) | x_1^l(N-1)) = \\
 &= \max_{\substack{a_j(x_1^l(k)); \\ k=1, \dots, N-2}} \sum_{x_1^l(N-1)} P(x_1^l(N-1) | x_1^n(1)) \times \\
 &\times A_{j^*(x_1^l(N-1))} x_1^l(N-1)
 \end{aligned}$$

The latter expression proves the back propagation property and the steps in the algorithm.