The 10th International Command and Control Research and Technology Symposium

Title: **On Finding Effective Courses of Action in a Complex Situation Using Evolutionary Algorithms**[*]

Authors: **Sajjad Haider and Alexander H. Levis**

Point of Contact: **Sajjad Haider**

Organization: System Architectures Laboratory, George Mason University.

Address: System Architectures Laboratory
MSN 4B5
George Mason University
Fairfax, VA 22030

Author Information:

Mr. Sajjad Haider
System Architectures Laboratory.
MS 4B5
George Mason University
Fairfax, VA 22030-4444
703-993-1725 (v)
703-993-1706 (f)
shaider1@gmu.edu

Dr. Alexander Levis
System Architectures Laboratory
MS 1G5
George Mason University
Fairfax, VA 22030-4444
703-993-1619 (v)
703-993-1601 (f)
alevis@gmu.edu

Student Paper

[*] This research was sponsored by Office of Naval Research under Grant No. N00014-03-1-0033 and the Air Force Research Laboratory Information Directorate under Grant No. F30602-01-C-0065.

On Finding Effective Courses of Action in a Complex Situation Using Evolutionary Algorithms*

Sajjad Haider and Alexander H. Levis
System Architectures Laboratory, MS 4B5
George Mason University
Fairfax, VA 22030
<shaider1>,<alevis>@gmu.edu

Abstract

This paper proposes an Evolutionary Algorithm (EA) based approach for finding effective courses of action (COAs) in a complex uncertain situation. The complex situation is modeled using a probabilistic modeling and reasoning framework, referred to as Timed Influence Nets (TINs). The TIN-based framework helps a system modeler in connecting a set of actionable events and a set of desired effects through chains of cause and effect relationships. Once a TIN is built, the optimization task confronted by the modeler is to identify a course of action that would increase the likelihood of achieving the desired effects over a pre-specified time interval. The paper uses Evolutionary Algorithms to accomplish this task. The proposed approach generates multiple COAs that are close enough in terms of achieving the desired effect. The purpose of generating multiple COAs is to give several alternatives to a decision maker. Moreover, the alternate COAs could be generalized based on the relationships that exist among the actions and their execution timings. While determining an effective course of action in a given situation, a system modeler has to consider several temporal/causal constraints that are present in a problem domain. The paper also proposes a constraint specification language that would help a system modeler in specifying these constraints.

1. Introduction

Any organization's effort, whether in the military or business domains, is guided by a set of objectives/desired effects that it plans to achieve in an uncertain complex environment. A decision maker, working in that organization, is typically confronted by the task of finding a supposedly optimal strategy to achieve these effects. The pre-requisite of this task is the modeling of cause-effect/relevance relationships among the variables that exist in the environment. The last two decades have seen an emergence of the use of probabilistic reasoning framework as a modeling tool for capturing such relationships. Commonly referred to as Bayesian Networks (BN) [Charniak, 1991; Neapolitan, 2003; Pearl, 1987], the framework uses a graph-theoretic representation to explicitly show the dependencies among variables in a problem domain. Despite their ability to represent uncertain domain in a compact and easy to read manner, the knowledge acquisition process and inference mechanism of BNs is intractable. [Cooper, 1990; Dagum and Luby, 1993] Influence Nets (INs), an instance of the Bayesian framework, were proposed [Chang et al., 1994; Rosen and Smith, 1996] a decade ago to overcome the

* This research was sponsored by Office of Naval Research under Grant No. N00014-03-1-0033 and the Air Force Research Laboratory Information Directorate under Grant No. F30602-01-C-0065.

intractability issues present in BNs. They employ an approximation inference algorithm and a non-probabilistic knowledge acquisition interface that does not require an exponential number of parameters.

Both Bayesian Networks and Influence Nets are designed to capture *static* interdependencies among variables in a system. However, a situation where the impact of a variable takes some time to reach the affected variable(s) cannot be modeled by either of the two approaches. In the last several years, efforts have been made to integrate the notion of time and uncertainty. [Boyen and Koller, 1998; Hanks et al., 1995; Kjaerulff, 1992; Santos and Young, 1999] Wagenhals et al. [Wagenhals et al. 1998] have added a special set of temporal constructs to the basic formalism of Influence Nets. The Influence Nets with these additional temporal constructs are called Timed Influence Nets (TINs) [Haider and Levis, 2004; Haider and Zaidi, 2004]. TINs have been experimentally used in the area of Effects Based Operations (EBOs) for evaluating alternative courses of action and their effectiveness in achieving mission objectives. [Wagenhals and Levis, 2000; Wagenhals and Levis, 2001; Wagenhals et al., 2003].

Once a dynamic uncertain situation is modeled using TIN, a decision maker is interested in the identification of a set(s) of actions and their time of execution that would maximize the likelihood of achieving the desired effect. The task is sometimes informally referred to as the *best* course of action (COA) determination. This paper applies Evolutionary Algorithms (EA) to accomplish the task of effective COA determination. It is important to note that any approach, which attempts to automate the process of identification of an effective COA, has to consider several temporal and causal constraints that exist among actionable events. Currently, TINs do not have a mechanism to specify these constraints. The paper proposes a constraint specification language (CSL) that aids a system modeler in specifying these temporal and causal constraints. The proposed technique takes these constraints into consideration while identifying an effective strategy in a given situation.

The optimization problem described above belongs to the category of *Mixed Integer Nonlinearly Constrained Optimization*. The nonlinearities are encoded in the form of conditional probabilities, while integral constraints result from the binary nature of actionable events. Besides providing the *optimal* or *near optimal* courses of actions, the proposed approach also provides a scheme to generalize the alternate effective courses of actions. The problems presented in this paper are relatively new and not much work has been reported in the literature that attempts to solve them in an automated fashion. A few exceptions are the work presented by Wagenhals [Wagenhals, 2000] and Haiying et al. [Haiying et al., 2004]. The current practice among the community is to use hit and trial methods to identify an effective course of action.

The rest of the paper is organized as follows. Sections 2 and 3 provide a brief overview of Timed Influence Nets and Evolutionary Algorithms, respectively. Section 4 describes the issues that need to be considered while finding an effective course of action. The section also explains the proposed constraint specification language and the specifics of the EA used in this paper. The results produced by the proposed approach, when applied on a real model, are explained in Section 5. Finally, Section 6 concludes the paper and provides future research directions.

2. Timed Influence Nets [Haider et al., 2005]

The modeling of the causal relationships in TINs is accomplished by creating a series of cause and effect relationships between some desired effects and the set of actions that might impact their occurrence in the form of an acyclic graph. The actionable events in a TIN are drawn as root nodes (nodes without incoming edges). A desired effect, or an objective in which a decision

maker is interested, is modeled as a leaf node (node without outgoing edges). Typically, the root nodes are drawn as rectangles while the non-root nodes are drawn as rounded rectangles. Figure 1 shows a partially specified TIN. Nodes B and E represent the actionable events (root nodes) while node C represents the objective node (leaf node). The directed edge with an arrowhead between two nodes shows the parent node promoting the chances of a child node being true, while the roundhead edge shows the parent node inhibiting the chances of a child node being true. The inscription associated with each arc shows the time delay it takes for a parent node to influence a child node. For instance, event B, in Figure 1, influences the occurrence of event A after 5 time units.

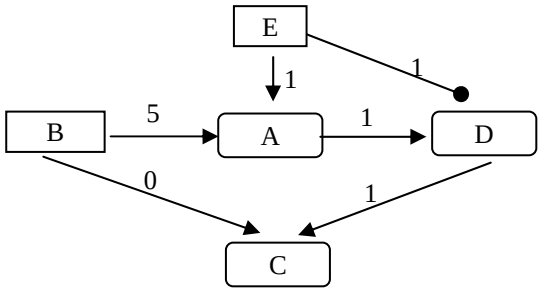


Figure 1: An Example Timed Influence Net (TIN)

The purpose of building a TIN is to evaluate and compare the performance of alternative courses of action. The impact of a selected course of action on the desired effect is analyzed with the help of a *probability profile*. Consider the TIN shown in Figure 1. Suppose the following *input scenario* is decided: actions B and E are taken at times 1 and 7, respectively. Because of the propagation delay associated with each arc, the influences of these actions impact event C over a period of time. As a result, the probability of C changes at different time instants. A probability profile draws these probabilities against the corresponding time line. The probability profile of event C is shown in Figure 2.

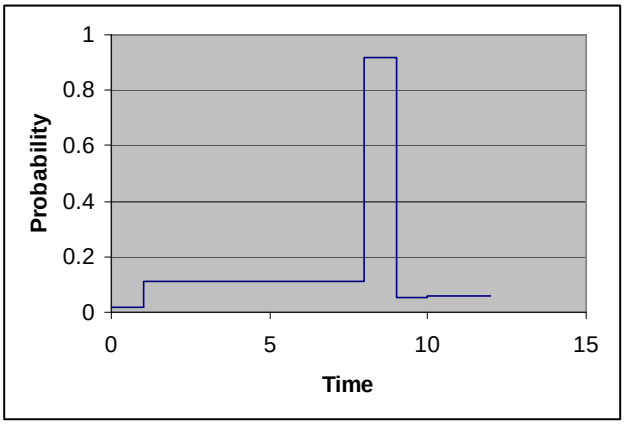


Figure 2: Probability Profile for Node C

The following items characterize a TIN:

1. A set of random variables that makes up the nodes of a TIN. All the variables in the TIN have binary states.
2. A set of directed links that connect pairs of nodes.
3. Each link has associated with it a pair of parameters that shows the causal strength of the link (usually denoted as g and h values).
4. Each non-root node has an associated baseline probability, while a prior probability is associated with each root node.
5. Each link has a corresponding delay d (where $d \geq 0$) that represents the communication delay.
6. Each node has a corresponding delay e (where $e \geq 0$) that represents the information processing delay.
7. A pair (p, t) for each root node, where p is a list of real numbers representing probability values. For each probability value, a corresponding time interval is defined in t. In general, (p, t) is defined as

   $([p_1, p_2, \ldots, p_n], [[t_{11}, t_{12}], [t_{21}, t_{22}], \ldots, [t_{n1}, t_{n2}]])$,

   where $t_{i1} < t_{i2}$ and $t_{ij} > 0 \; \forall$ i = 1, 2, …., n and j = 1, 2

The last item in the above list is referred to as an input scenario, or sometimes (informally) as a course of action. Formally, a TIN is described by the following definition.

Definition 1 Timed Influence Net (TIN)
A TIN is a tuple (**V, E, C, B, D_E, D_V, A**) where
V: set of Nodes,
E: set of Edges,
C represents causal strengths:
E → { (**h**, **g**) such that -1 < **h**, **g** < 1 },
B represents the Baseline / Prior probability: **V** → [0,1],
$\mathbf{D_E}$ represents Delays on Edges: **E** → **N**,
$\mathbf{D_V}$ represents Delays on Nodes: **V** → **N**, and
A (input scenario) represents the probabilities associated with the sequence of actions and the time associated with them.
A: **R** → $\{([p_1, p_2, \ldots, p_n],[[t_{11},t_{12}], [t_{21},t_{22}], \ldots,[t_{n1},t_{n2}]])$
such that $p_i = [0, 1]$, t_{ij} → **Z** and $t_{i1} \leq t_{i2}, \; \forall$ i = 1, 2, …., n and j = 1, 2 where **R** ⊂ **V** }

3. Evolutionary Algorithm

Evolutionary Algorithms (EAs) can be interpreted as a parallel adaptive search procedure. They have been applied to a wide variety of application areas including optimization, search, learning, automated programming, adaptation, etc. They are modeled after the organic evolutionary processes found in nature. The current EAs have their roots in three distinct efforts that were initiated in parallel almost four decade ago: Evolution Strategies [Schwefel, 1975;Schwefel, 1995] Evolutionary Programming [Fogel et al., 1966], and Genetic Algorithms [DeJong, 1975;Goldberg, 1989].

An EA consists of a *population* of *individual solutions* that are selected and modified in order to discover overall better solutions in the search space. An individual in the population is referred to as *genome* or *phenome*. One (or more) solution(s) is (are) modified to produce new *offspring* in the population. Each solution is evaluated using a *fitness* value that represents the

quality of the solution in the context of a given problem. The major steps involved in a canonical EA are shown in Table 1 [DeJong, 2005 (To Appear)]. There are three major design issues, namely, representation, selection, and variation operators, that need to be considered while designing an EA for a specific problem. These design issues are briefly explained in the following sub-sections.

Table 1: A Canonical Evolutionary Algorithm

Randomly generate the initial population Do until a stopping criterion is met Select parent(s) using a *selection procedure*. Create new offspring(s) by applying the *variation operators* on the parents. Compute the fitness of the new offspring(s). Select member(s) of the population to die using a selection procedure.

3.1 Representation

Representation is among the most critical design issues while developing an EA. In an EA, there are two possible types of representations. When a representation is stated in its natural state it is said to be a *phenotype*. For example, a real valued optimization problem would be represented as a set of real valued coordinates. On the other hand, it may be necessary to map a phenotype representation into another structure to make it easier for the algorithm to modify and exchange information. In many cases the phenotype can be mapped into a representation that resembles, at a very high level, a sequence of local structures or building blocks. In this case, the representation is referred to as a *genotype*.

3.2 Selection

Selection in evolutionary algorithms is the process of choosing which individuals reproduce offspring and which individuals survive to the next generation. When selection is used to choose which individuals reproduce, the process is referred to as pre-selection (parent(s) selection). When it is used to select the individuals that survive to the next generation it is called post-selection (survival selection). Selection can further be categorized as deterministic or probabilistic. Deterministic selection tends to behave more like greedy hill-climbing algorithms and exploits the nearest areas with promising solutions. Probabilistic selection schemes are more exploratory and search the landscape.

An important decision required when deciding on what type of selection schemes to use is whether to emphasize *exploration* or *exploitation*. Schemes based on exploration, are said to have a low selection pressure, while schemes based on exploitation, are said to have greater selection pressure. It can be said that the selection pressure is a vague measure of how often more fit individuals are selected to reproduce and/or live to the next generation.

EAs are not necessarily good at finding optimal solutions. For complex problems, the global optimal solution may be very difficult to locate for any algorithm. This leads to a trade-off between exploring the landscape for areas that appear to hold good solutions and exploiting the good area found. This can be a very difficult balancing act. Too much exploration may result in not finding the local optima in the regions explored, while too much exploitation can lead to

behavior similar to a greedy hill climbing algorithm. The key is to find a selection pressure that balances exploration and exploitation.

Selection schemes can be further categorized into generational or steady-state schemes. A selection scheme is generational when the entire current population is replaced by its offspring to create the next generation; while, a scheme is referred to as steady-state when a selected few offspring replace a few members of the current generation to form the next generation. Some of the popular selection schemes are given below:

Fitness Proportional: In this scheme, individuals are selected based on their fitness in proportion to the other individuals in the population.

Rank Selection: In rank selection, individuals in the population are first ranked by fitness and then selected for reproduction based on a probability proportional to rank.

Binary Tournament: In this scheme, two individuals are randomly selected from the population and compared. The one with the highest fitness is selected for reproduction. Then another two individuals are randomly selected and the best fit is kept as the mate to the first parent.

Truncation: This scheme is used during post-selection. Populations of parents and offspring are merged together and the top k fittest individuals survive to the next generation.

Uniform Stochastic: In this scheme, individuals are selected with uniform probabilities.

3.3 Variation Operators

Mutation: Mutation is a genetic operator by which small modifications are made to a single genotype or phenotype of a selected individual. For real-valued phenotype approaches, the mutation operator is usually of the form of a small Gaussian change to the phenotype. For binary coded genotypes, a mutation is a random bit flip.

Crossover: Crossover is where two or more individuals exchange information to create one or more new individuals. In its simplest form, a single random locus is selected as a slice point and the segments are exchanged between two individuals at the locus. Figure 3 shows an example of a one-point crossover. In contrast to the one-point crossover, a multi-point crossover occurs when more than one locus is selected and information is exchanged in segments between the locus points.

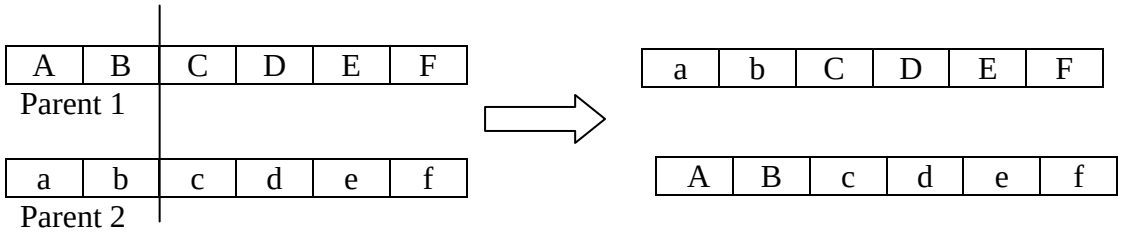


Figure 3: One-Point Crossover

4. ECAD-EA Methodology

The previous two sections described the main tools i.e., TIN and EA, used in this paper to identify an effective course of action in a dynamic uncertain situation. As discussed earlier, the

term course of action in the context of this paper means the identification of a set of actions to be selected in a plan, their sequence, and their time of execution that maximizes/minimizes a certain metric (or set of metrics) related to the desired effect(s). The proposed approach that accomplishes this task is termed ECAD-EA (Effective Courses of Action Determination Using Evolutionary Algorithms). In addition to a completely specified TIN, the approach requires a system modeler to specify several other items before it determines an effective course of action. Some of these items are given below followed by their explanations, while Figure 4 shows the inputs and outputs of the ECAD-EA methodology in the form of a block diagram.

(a) Windows of Opportunity and Observation
(a) Identification of a Robust Metric
(b) Single or Multiple Desired Effects
(c) Constraints among Actionable Events

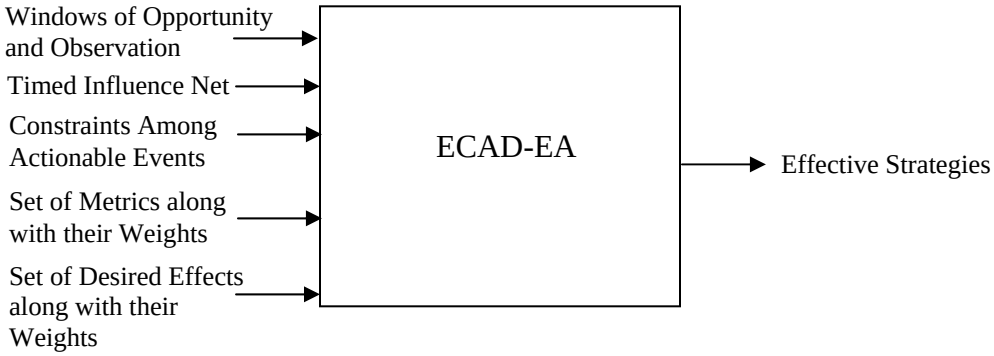


Figure 4: Block Diagram of ECAD-EA

4.1 Windows of Opportunity and Observation

The proposed ECAD-EA methodology requires a system modeler to specify two parameters that aid in creating a boundary around the solution space. The first parameter is the duration in which all the actions must take place. This duration is referred to as either *window of opportunity* or *action window*. It should be mentioned that the window of opportunity refers to the time interval during which the whole plan should be executed. If some actions have more strict temporal requirements for their execution, then these requirements can be specified with the help of the constraint specification language discussed in subsection 4.5. The second parameter is the time period in which a decision maker is interested in getting the desired results. This duration is referred to as *observation window*. The two terms are explained further towards the end of this section with the help of an example.

4.2 Identification of a Robust Metric

To judge the fitness of a COA, a set of metrics is needed that can be used as Measures of Performance (MOPs). All the metrics discussed in this paper evaluate a COA in terms of the desired effect's probability profile produced by that particular COA. The fitness of a COA can be measured using either a simple or a composite metrics. In case of a composite metric, weights should be assigned to each of the individual metric. Weights assigned to a metric may vary from situation to situation and are upon the discretion of the system modeler. A selected metric (or set of metrics) guides the EA through probability profiles' space while the EA searches for an

effective COA. There are several features of a probability profile that can be considered during its evaluation. A preliminary list of features has been identified that could constitute potential metrics. The list includes the following and is explained with the help of the probability profile of Figure 5.

1) Maximum Probability Achieved: This metric specifies the highest probability achieved by the probability profile of a desired effect during the window of observation. For instance, in the profiles of Figure 5 the highest probability produced by COA #1 is 0.9 in contrast to 0.8 produced by COA #2.
2) Time to Reach the Maximum Probability: This metric is related to the first one and it specifies the time at which the maximum probability is achieved. For instance, COA #1 in Figure 5 has produced a better probability value (0.9) but the value is achieved at time 8. COA #2, on the other hand reaches the probability of 0.8 at time 6. If Metric 1 is considered alone then COA #1 would be considered better, but if a decision maker is interested in getting the probability of desired effect above a certain threshold, then COA #2 might be considered better if it satisfies the threshold requirement.
3) Probabilities at Specific Points/Intervals: A decision maker might be interested in profiles that have probabilities above (below) a certain threshold or highest at specific time instances or intervals. For instance, in Figure 5, a decision maker might be interested in those profiles which have higher probabilities during the time interval 4 to 8. COA # 1, in this case, is superior to COA #2. On the other hand, if the interval under consideration is between times 8 and 11, then COA #2 would be considered better. The time instances/intervals could be a list of values, i.e., the decision maker requires the probability to be above (below) certain thresholds at time 3, 9, and 13.
4) Area Under the Curve (AUC): This metric represents the area under a probability profile. The computation of AUC for probability profiles of Figure 5 is shown in Table 2. Based on the metric, it can be said that COA #2 is better than COA #1.

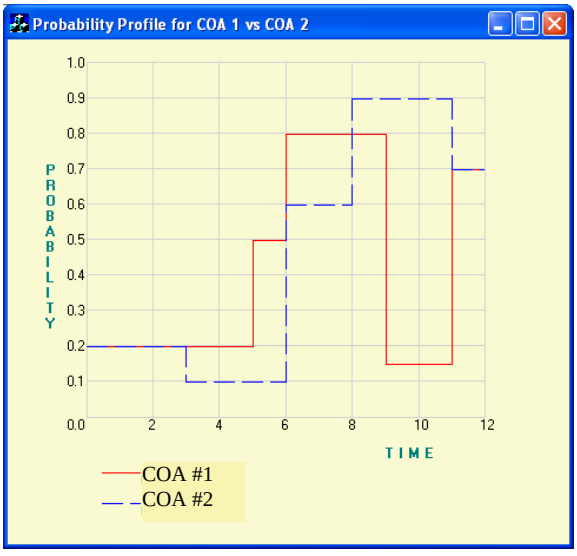


Figure 5: Two Competing Courses of Actions

Once a set of metrics is specified, along with their corresponding weights, the fitness of a particular COA is measured as a weighted sum of the selected metrics. In other words,

$$\text{fitness(COA x)} = w_1 m_1 + w_2 m_2 + \ldots\ldots\ldots\ldots w_n m_n \quad (1)$$

where w_1 is the weight corresponding to the metric m_1. In such cases when multiple metrics are considered, each metric is normalized on the scale of 0 to 1.

Table2: Computation of AUC Metric for COAs in Figure 3.16(a)

COA # 1				COA # 2			
Time at which Change Occurs	Duration of the Interval **d**	Probability Of the Target Node **t**	Area **d x t**	Time at which Change Occurs	Duration of the Interval **d**	Probability Of the Target Node **t**	Area **d x t**
5	5	0.2	1	3	3	0.2	0.6
6	1	0.5	0.5	6	3	0.1	0.3
9	3	0.8	2.4	8	2	0.6	1.2
11	2	0.15	0.3	11	3	0.9	2.7
Sum			**4.2**				**4.8**

4.3 Single/Multiple Desired Effects (or Objectives)

A given problem could have either single desired effect or multiple (at times conflicting) desired effects. In case of multiple desired effects, a system modeler needs to weight the importance of each desired effect. The objective function then becomes a weighted average of certain functions of each desired effect's probability profile. Each profile can be evaluated using a different set of metrics and their corresponding weights. For instance, if there are two desired effects d1 and d2, then a decision maker might be interested in using Metric 4 (AUC) for d1 while using Metric 1 and 2 for d2. Equation 1 can thus be modified as

$$\begin{aligned}\text{fitness(COA x)} = {} & w_1 (w_{11} m_1 + w_{12} m_2 + \ldots\ldots\ldots\ldots + w_{1n} m_n) \\ & + w_2 (w_{21} m_1 + w_{22} m_2 + \ldots\ldots\ldots\ldots + w_{2n} m_n) \\ & : \\ & + w_k (w_{k1} m_1 + w_{k2} m_2 + \ldots\ldots\ldots\ldots + w_{kn} m_n)\end{aligned} \quad (2)$$

when there are k number of desired effects and each effect can be evaluated using n number of metrics. w_i represents the weight assigned to the desired effect i, while w_{ij} represents the weights assigned to metric j for the same effect.

4.4 Constraints among Actionable Events: Several temporal and logical constraints might exist among the actionable events that are not directly modeled using TIN. The presence of these constraints results in the need of a constraint specification language that helps a system modeler in modeling temporal/logical constraints. The paper proposes one such language that assists a modeler in specifying these types of constraints. The consistency of these constraints can be checked by propositional and temporal logic based systems depending upon the nature of the given constraints. Table 3 describes the constructs of the proposed constraint specification language (CSL). Constraints 1 and 2 are for the cases where it is required that some actions must occur in a specific state. Constraint 3 requires that A1 and A2 should have the same state (whether "True" or "False"). Constraint 4 specifies the time 't' at which action A must take place

while Constraint 5 shows that action A1 must happen before action A2. If the value of *v* is zero, then A1 should happen earlier than A2 but if *v* is non-zero (positive) then A1 must happen at least *v* time units before A2. Constraint 6 specifies that two actions A1 and A2 should happen at the same time.

Table 3: Constructs of the Constraint Specification Language

1. True	A			// Factual Constraint
2. False	A			// Factual Constraint
3. Same_State	A1	A2		// Causal Constraint
4. At_Time	A	t		// Temporal or Factual Constraint
5. Before	A1	A2	*v*	// Temporal or Causal Constraint
6. Equal_Time	A1	A2		// Temporal or Conceptual Constraint

4.5 Specifics of the Evolutionary Algorithm

Once all the inputs to the ECAD-EA methodology are specified, the proposed approach searches the probability profile solution space to determine effective strategies. Consider the TIN shown in Figure 6. There are two actionable events, A and B, and one desired effect H. Suppose a decision maker is interested in selecting a course of action that maximizes the probability of H over a given time interval. The window of opportunity is set as an interval between time instances 1 and 10 while the observation window is set between time instances 1 and 20. Thus, the solution space consists of 2^2 x 10^2 = 400 probability profiles. Figure 7 shows the solution space when both actions A and B are true given that the fitness of a profile is measured using AUC metric. As it can be seen from the figure, the global maximum lies in the region when action B is taken earlier than action A.

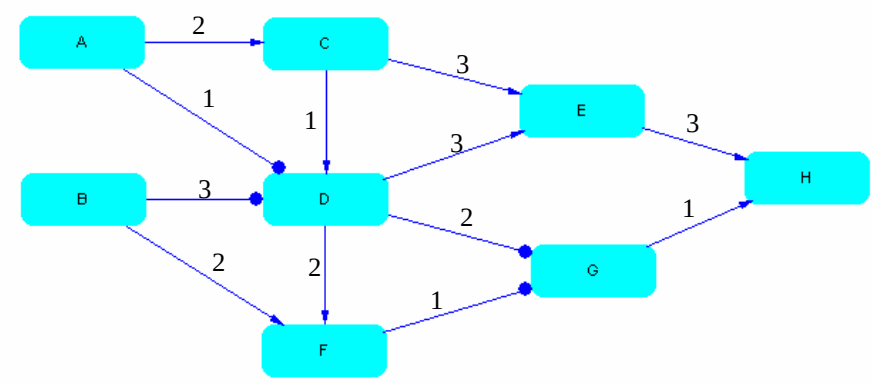


Figure 6: A Sample Timed Influence Nets

The proposed methodology explores the solution space of Figure 7 using an Evolutionary Algorithm. As described in Section 3, the main characteristics of an EA are the representation of individuals in the population, the selection mechanism, and the variation operators. The population in the EA of ECAD-EA consists of candidate courses of action that maximize a metric. Each individual in the population consists of actions and the times at which the actions are executed. Due to the fact that all the variables in a TIN have binary states, a bit representation

is used for the actionable events. An integer value representation is used for the time at which an action is executed. Thus an individual (phenome) in the population of solutions has the following structure

<Action 1, time (Action 1), Action 2, time (Action 2), …..., Action n, time (Action n)>

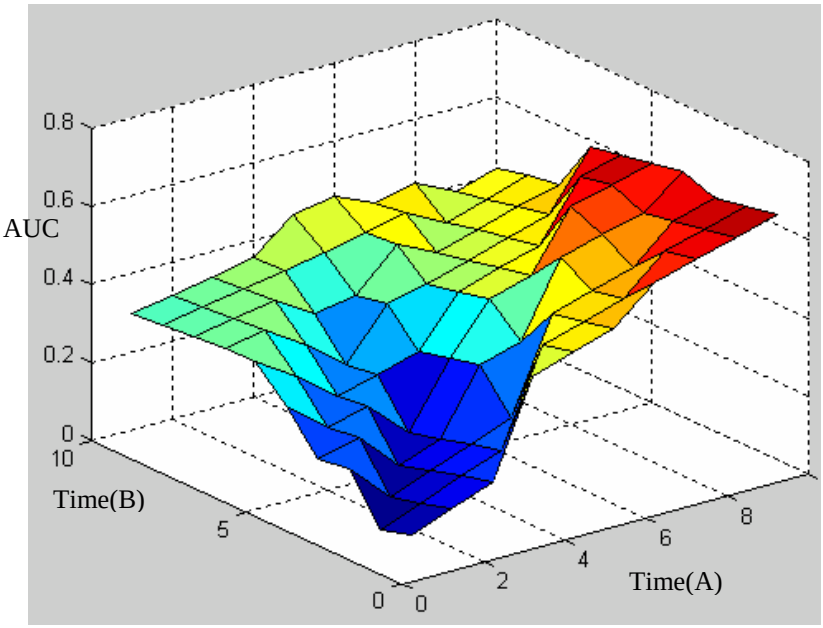


Figure 7: The Solution Space of the TIN Shown in Figure 6

For the example under consideration, the AUC metric is used to evaluate the fitness of a probability profile. The parent selection scheme is uniform stochastic, while the survival selection scheme is binary tournament. The last major ingredient of the EA is the variation operators. Since all the actions have binary states, a bit flip mutation operator is used for them, while a delta x mutation operator is used for the times at which actions are executed. The size of the population is kept at 10, while the EA is run till 20 generations. The details of the EA are summarized in Table 4.

Table 4: Parameters of the Evolutionary Algorithm

Representation: Phenotype
Parent Selection: Uniform Stochastic
Survival Selection: Binary Tournament
Mutation: Bitflip / Delta x
Mutation Rate: 0.2
Crossover: one point
Population Size: 10
Number of Generations: 20

The EA of Table 4 is applied on the TIN of Figure 6 along with the constraints regarding windows of opportunity and observation to find courses of action that maximize the AUC metric

for event H's probability profile. Several solutions with very close fitness values are produced. Out of those solutions, the four top ones are shown in Table 5. The solution shown in the first row is read as follows: "in order to maximize AUC, action A is made True at time 9, while action B is made True at time 5". The other rows in the table can be read in a similar manner. The results agree with the intuition developed earlier by looking at the solution space of Figure 7. All the solutions are centered on the region where the global maximum lies. The solutions also conformed to the temporal relationship exhibited by Figure 7. In all the solutions, action B is executed before action A. Thus, the advantage of having multiple solutions, having very close fitness, is that the solutions can be generalized to produce a plan that describes time at a qualitative level or at a more general quantitative level. A rather detailed discussion on this issue is presented in Section 4.6.

Table 5: Four Best Solutions for TIN of Figure 6

A	time(A)	B	time(B)
True	9	True	5
True	9	True	6
True	8	True	5
True	9	True	4

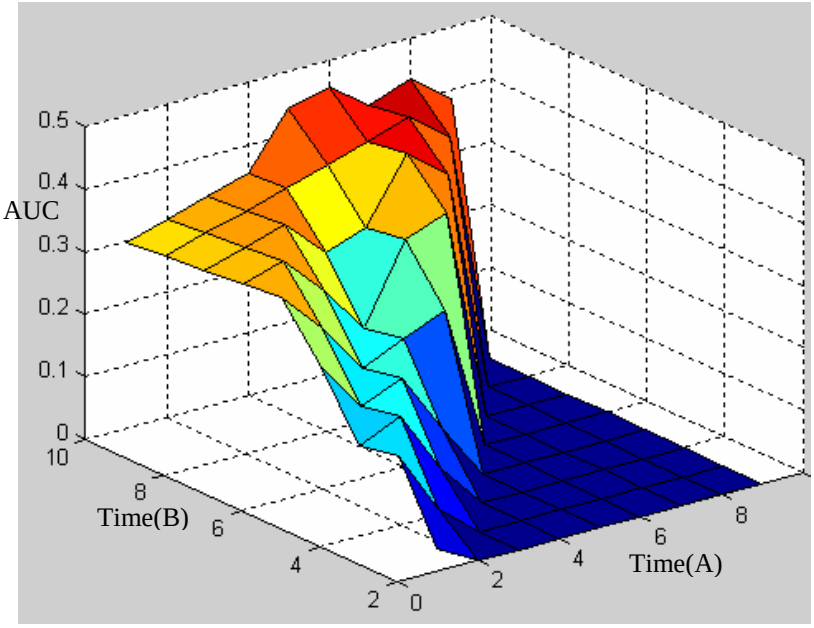


Figure 8: The Constrained Solution Space of the Model Shown in Figure 6

The solutions of Table 5 are produced without considering any user-defined constraints. Suppose it is required that a feasible solution must satisfy the constraint that A must be executed before B. Using the constructs of the proposed CSL, the constraint can be written as:

Before A B 0

The reduced solution space (based on AUC metric) is shown in Figure 8. When the EA of Table 4 is run on this modified solution space, it generates courses of action that maximize the AUC metric for event H's probability profile under the given constraint. The four top courses of action

are shown in Table 6. Like the previous case, the solutions produced by the EA are in the region that is close to the global maximum of Figure 8.

Table 6: Four Best Solutions for the TIN of Figure 6 with Constrained Solution Space

A	time(A)	B	time(B)
True	8	True	10
True	7	True	9
True	6	True	9
True	4	True	9

4.6 Generalization of Temporal Relationships

The purpose of generating alternate courses of action is to determine any pattern that exists among the alternate courses of action. If there exists a pattern among the solutions that can be generalized, then this information can be very helpful to a decision maker as it gives more flexibility to him while he plans a course of action. This section presents a scheme for generalizing the temporal relationships that exist among the actions in the solutions, produced by the EA, having very close fitness value. Consider the three courses of actions shown in Table 7. There are four actionable events, namely, A, B, C, and D. Actions in a TIN are instantaneous. Thus, only two kinds of temporal relationships can exist between actionable events, namely, 'Before' and 'Equal'; a point in time can be either 'Before' another point in time or it can be 'Equal' to some other point. For a detailed discussion of possible temporal relationships between points and interval, refer to [Zaidi, 1999;Zaidi and Levis, 2001].

Table 7: 3 Alternate Courses of Actions

A	time(A)	B	time(B)	C	time(C)	D	time(D)
T	1	F	14	T	8	T	5
T	8	F	14	T	12	T	13
T	4	F	5	T	13	F	3

The temporal relationship that exists among the actionable events can be represented in the form of a grid, as shown in Table 8. In each cell, there are 3 entries that correspond to the 3 solutions presented in Table 7. For instance, in all 3 solutions time(A) is always less than time(B). Thus, the relationship between A and B in all cases is "Before" as shown in the cell corresponding to the intersection of A and B (1^{st} Row and 2^{nd} Column). Similarly, the relationship between A and C is "Before" in all three cases as time(A) is less than time(C) in all solutions. D, on the other hand, is "Before" A in the last solution while A is "Before" D in the first two solutions. Other entries in the table can be read in a similar fashion. From the table, it can be seen that there are some cells where corresponding actions have a similar type of temporal relationship in all the solutions. Thus, the temporal relationship between those actions can be generalized. For instance, the following generalization can be depicted from Table 8.

A Before B

A Before C

D Before B

Table 8: Temporal Relationships among Actionable Variables

	A	B	C	D
A		Before Before Before	Before Before Before	Before Before --
B			-- -- Before	
C		Before Before --		-- Before --
D	-- -- Before	Before Before Before	Before -- Before	

The above information aids a decision maker during the planning stage of a mission. It simply states that any plan that is developed to accomplish a particular objective should satisfy the above mentioned temporal constraints. Figure 9 shows an instance of possible relationships among the four actionable events on a single time line. The generalization of temporal relationships can be extended further. For instance, in the temporal relationship of Table 7, whenever D is in state True, it happens after A. On the other hand, when D is in state False, it happens before A. Thus, the temporal relationship between two actions can be generalized using If-Then type of rules. For instance, the following rules can describe the temporal relationship that exists between A and D.

If D is True Then
A is Before D
If D is False Then
D is Before A

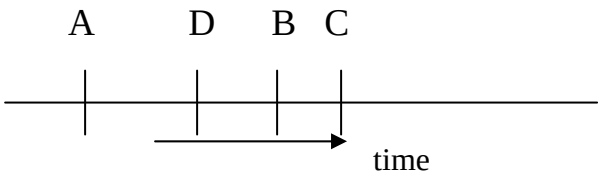


Figure 9: Actionable Events on a Single Time Line

5. Application of the ECAD-EA Methodology

This section applies the ECAD-EA methodology on a real model. The TIN model, shown in Figure 10, is developed to capture the events and the associated uncertainties in combating insider threat[1]. Only the structure of the TIN is shown in the figure. The quantitative strengths and the time delays associated with each arc are not shown to keep things simple. The reader, however, should not have any difficulty in understanding the results presented in this section. The purpose of building the insider threat model is twofold: to identify and analyze the actions that could be used by an insider to become a security risk for an organization and to analyze the actions taken by the organization to prevent any potential damage caused by the insider. In other words, a system modeler is interested in identifying an effective strategy that maximizes the likelihood of achieving the desired effect: "Insider Does Not Have Strength of Resolve to Attack". There are ten actionable events, drawn as the root nodes, in the insider threat model:

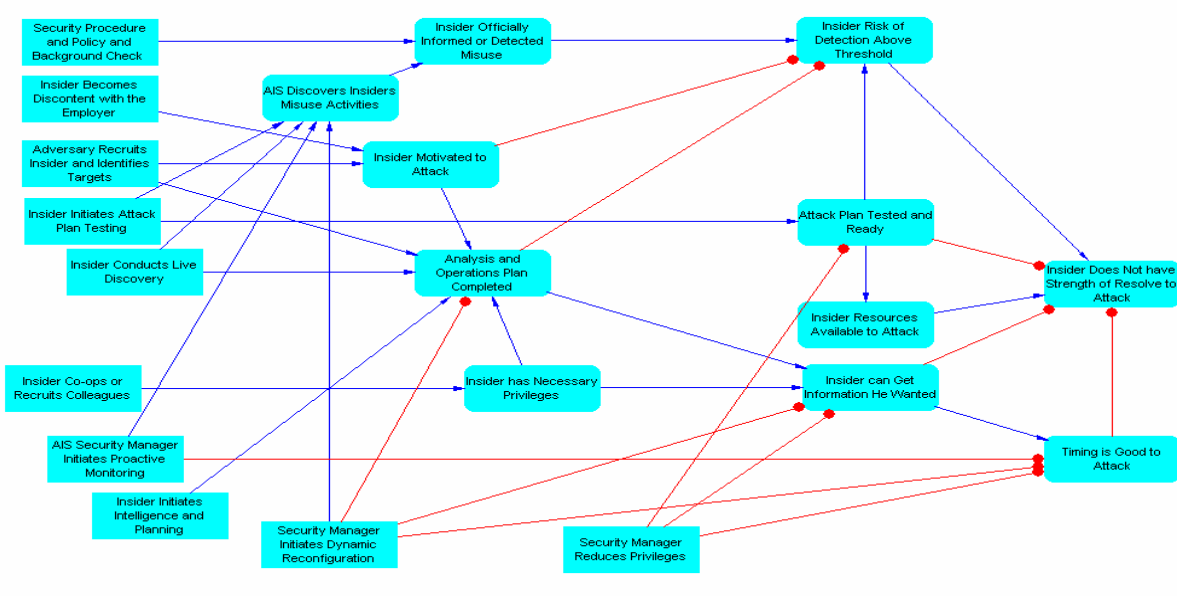


Figure 10: Influence Net for Insider Threat Model

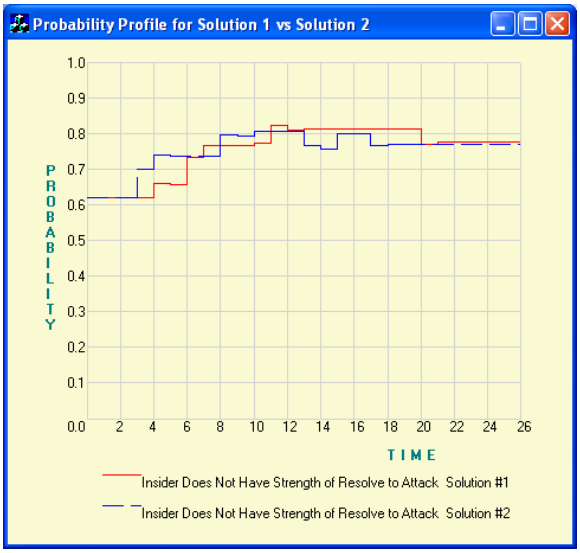
Security Manager Reduce Privileges	(A)
Security Manager Initiates Dynamic Reconfiguration	(B)
Insider Initiates Attack Plan Testing	(C)
AIS Security Manager Initiates Proactive Monitoring	(D)
Insider Co-ops or Recruits Colleague	(E)
Insider Conducts Live Discovery	(F)
Insider Initiates Intelligence and Planning	(G)
Adversary Recruits Insider and Identifies Target	(H)
Insider Becomes Discontent with Employer	(I)
Conduct Security Procedure and Policy and Background Check	(J)

[1] The model was presented to OSD-C3I on Dec 19, 2003 by Lee Wagenhals and Larry Wentz.

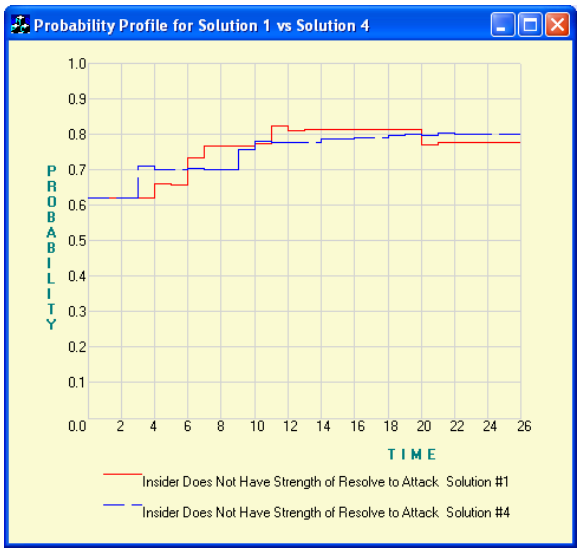
The labels next to the actions are used in the sequel to refer to the corresponding actions. The EA, as explained in Section 4.5, is applied on the TIN developed for insider threat. The parent selection scheme is set to uniform stochastic, while the survival selection is set to truncation. Population and generation sizes are set to 20 and 15, respectively. The window of opportunity is between 1 to 20 time units, while the window of observation is between 1 to 25 time units. The top 4 results produced by the ECAD-EA methodology are presented in Table 9. There are 20 elements in each solution. The odd elements (1, 3, 5, …) represent the state of the actionable events, while the even elements (2,4,6,….) represent the time stamps at which the state of the actions are decided. The order of the actions in a particular solution is the same as described above. The solution described in the first row of Table 9 says that "Security Manager Reduce Privileges" is made True at time 9 while "Security Manager Initiates Dynamic Reconfiguration" is made True at time 4 and so on. . Figures 11(a) and 11(b) show the comparison of Solution 1 to Solution 2 and Solution 1 to Solution 4, respectively. The probability profiles produced by all the solutions are close, i.e., it is hard to say which one is better than the other. In this kind of situation, the best thing is to produce a certain percentage of the top solutions (options) and then let the decision maker consider other characteristics which are not explicitly mentioned while planning a course of action in making his choice.

Table 9: Four Best Solutions for the TIN of Insider Threat Model

S.#																				
1.	1	9	1	4	0	8	1	2	0	19	0	12	0	11	1	17	0	1	1	8
2.	1	6	1	1	1	11	1	8	0	6	1	9	1	12	0	5	1	14	1	6
3.	1	4	1	2	0	18	1	3	0	3	1	9	1	13	1	19	0	14	0	6
4.	1	1	1	7	0	4	1	12	0	6	0	11	0	19	0	15	0	16	0	9



(a) Solution 1 vs. Solution 2



(b) Solution 1 vs. Solution 4

Figure 11: Comparison of Different Solutions

The temporal relationship that exists among the solutions of Table 9 can be generalized to produce a feasible time line that aids a decision maker during the planning stage of a mission. Applying the same approach as explained in Section 4.6, the following temporal relationships are found among the 4 solutions shown in Table 9:

A Before F	A Before G	B Before A
B Before G	B Before H	B Before J
D Before G	J Before F	J Before G

A possible time line based on the above temporal relationships is shown in Figure 12. The above generalization also matches with the intuition one could develop after looking at the set of actions present in the problem. The temporal relationships roughly say that the actions taken by an organization (A, B, D, and J) should be taken before the actions taken by an adversary (F, G, and H).

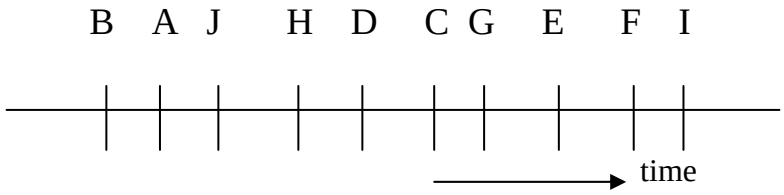
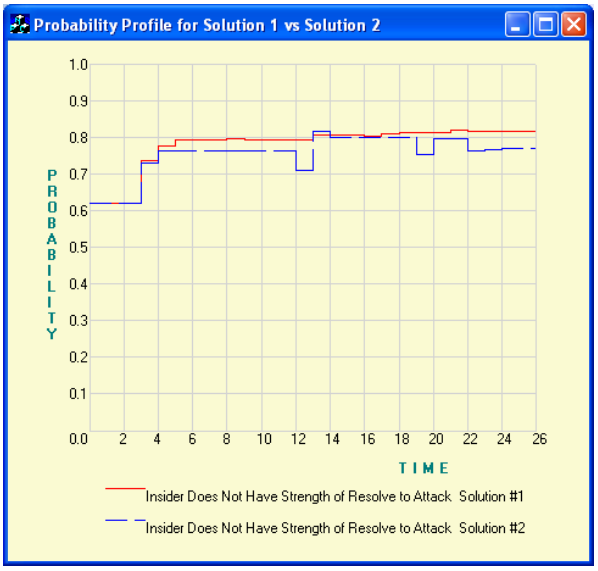


Figure 12: Actionable Events on a Single Time Line

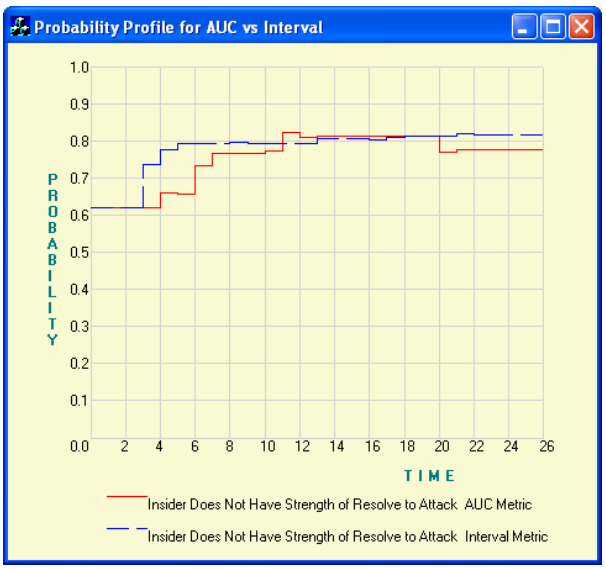
It can be seen from the probability profile of Figure 11(a) that Solution #2 is in fact better than Solution #1 during the time interval 2 and 11. However, during the whole duration of 25 time units, Solution #1 turns out to be better based on the AUC metric. If a system modeler is interested in only the probability profile during the time interval 2 and 11, then the solutions produced by the ECAD-EA would be different. To demonstrate this fact, the EA is run again on the insider threat model, but this time the fitness of a solution is evaluated by the area under a probability profile during time interval 3 and 12. The top 4 solutions are shown in Table 10. The probability profiles produced by the two top solutions are shown in Figure 13(a). Again, there is not much difference between the performances of the two solutions. Figure 13(b) compares the profiles of the best solutions produced using AUC and Interval metric. As expected, the solution based on interval metric performs better during the time interval 3 and 12.

Table 10: Four Best Solutions based on Interval Metric for the Insider Threat Model

S.#	
1.	1 1 1 2 0 6 1 1 0 17 1 13 0 2 0 18 0 15 1 10
2.	1 11 1 1 1 10 1 1 0 4 0 1 1 17 1 19 0 19 1 10
3.	1 4 1 2 1 19 1 17 1 5 0 3 1 14 1 11 0 3 0 13
4.	1 2 1 4 0 12 0 5 0 11 1 4 0 6 1 16 0 15 1 17



(a) Solution 1 vs. Solution 2



(b) Solution 1 vs. Solution 4

Figure 13: Comparison of Different Solutions

Table 11: 4 Best Solutions for Insider Threat Model Having Few Constraints

S.#																			
1.	1	1	1	1	0	15	1	4	0	12	1	6	0	9	0	8	0	1	1	1
2.	1	1	1	1	1	15	1	4	0	12	1	6	0	9	0	8	0	5	1	1
3.	1	1	1	1	0	10	1	4	0	12	1	6	0	2	0	14	0	1	1	1
4.	1	1	1	1	0	15	1	4	0	12	1	6	1	8	0	8	0	5	1	3

The solutions of Table 3.16 are produced without considering any constraints. Suppose it is required that 'Security Manager Initiates Dynamic Reconfiguration' must be true. Based on the temporal constraints, 'Insider Initiates Intel and Planning' must be at least 3 time units before 'Insider Initiates Attack Plan Testing'. During the plan execution, it turns out that 'Insider Conducts Live Discovery' at time 6. Furthermore, 'Security Manager Initiates Proactive Monitoring' at time 4. Finally, it is discovered at time 12 that the insider attempts to "Co-ops or Recruits Colleague" failed. All of these constraints can be written in terms of the Constraint Specification Language described earlier.

Before 'Insider Initiates Intel and Planning' 'Insider Initiates Attack Plan Testing' 3
True 'Security Manager Reduce Privileges'
At_Time 'AIS Sec Mgr Initiates Proactive Monitoring' 4
True 'Insider Conducts Live Discovery'
At_Time 'Insider Conducts Live Discovery' 6
False 'Insider Co-ops or Recruits Colleague'
At_Time 'Insider Co-ops or Recruits Colleague' 12

The top 4 best solutions produced by the ECAD-EA methodology when applied on the insider threat model having the above constraints are presented in Table 11. The results presented so far consider only single desired effect, but the proposed approach works equally well on problems having multiple desired effects.

Conclusions

The paper presents a methodology to identify effective courses of action in complex uncertain situations. The approach works on Timed Influence Nets that are used to model the uncertainties present in such complex situations. The presented approach uses Evolutionary Algorithms to identify effective courses of action. The approach not only provides a single best solution but it also provides several alternate solutions that are close enough to the best solution. These alternate solutions aid a decision maker in understanding the impact of actions' time of executions on the desired effect. The paper suggests a temporal generalization based on the similarities that exist in the temporal relationships among the actions.

A course of action is evaluated based on some pre-defined metrics. The paper discusses a set of metrics that can be considered, either individually or jointly (depending upon the situation), while evaluating a course of action. A constraint specification language that aids a system modeler in specifying the temporal and causal constraints among the actionable events is also presented.

Acknowledgement

The first author is grateful to Prof. Kenneth De Jong for the discussions he had with him while the author was working on the first draft of this paper.

References

Boyen, X. and Koller, D., "Tractable Inference for Complex Stochastic Processes," Proceedings of the Conference on Uncertainty in AI, 1998.

Chang, K. C., Lehner, P. E., Levis, A. H., Zaidi, S. A. K. and Zhao, X., "On Causal Influence Logic," George Mason University, Technical Report Dec. 1994.

Charniak, E., "Bayesian Network without Tears," in AI Magzine, vol. 12, 1991, pp. 50-63

Cooper, G. F., "The Computation Complexity of Probabilistic Inference Using Bayesian Belief Networks," Artificial Intelligence, vol. 42, pp. 393-405, 1990.

Dagum, P. and Luby, M., "Approximating Probabilistic Inference in Bayesian Belief Networks Is Np-Hard," Artificial Intelligence, vol. 60, pp. 141-153, 1993.

DeJong, K. A., "An Analysis of the Behavior of a Class of Genetic Adaptive Systems", Doctoral Dissertation," University of Michigan, 1975.

DeJong, K. A., Evolutionary Computing: MIT Press, 2005 (To Appear).

Fogel, L. J., Owens, A. J. and Walsh, M. J., Artificial Intelligence through Simulated Evolution. Chichester, UK: John Wiley Inc., 1966.

Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning. Massachusetts: Addison-Wesley, Reading, 1989.

Haider, S. and Levis, A. H., "A Heuristic Approach for Belief Updating in Timed Influence Nets," Proceedings of 2004 Command and Control Research and Technology Symposium, San Deigo, 2004.

Haider, S. and Zaidi, A. K., "Transforming Timed Influence Nets into Time Sliced Bayesian Networks," Proceedings of 2004 Command and Control Research and Technology Symposium, San Diego, 2004.
Haider, S., Zaidi, A. K. and Levis, A. H., "On Temporal Analysis of Timed Influence Nets Using Point Graphs," To be Presented in the 18th International Florida AI Research Society Conference (FLAIRS 05), 2005.
Haiying, T., Levchuk, Y. N. and Pattipati, K. R., "Robust Action Strategies to Induce Desired Effects," IEEE Transactions on Systems, Man, and Cybernatics - Part A: Systems and Humans, vol. 34, pp. 664-680, 2004.
Hanks, S., Madigan, D. and Gavrin, J., "Probabilistic Temporal Reasoning with Endogenous Change," in Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (Uai-95), 1995, pp. 245--254.
Kjaerulff, U., "A Computational Scheme for Reasoning in Dynamic Probabilistic Networks," in Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence, 1992, pp. 121--129.
Neapolitan, R. E., Learning Bayesian Networks: Prentice Hall, 2003.
Pearl, J., Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference: Morgan Kaufmann, 1987.
Rosen, J. A. and Smith, W. L., "Influence Net Modeling with Causal Strengths: An Evolutionary Approach," Proceedings of the 1996 Command and Control Research and Technology Symposium, Monterey CA, 1996.
Santos, J. E. and Young, J. D., "Probabilistic Temporal Network: A Unified Framework for Reasoning with Time and Uncertainty," International Journal of Approximate Reasoning, vol. 2, 1999.
Schwefel, H. P., Evolution and Optimum Seeking: John Wiley & Sons, 1995.
Schwefel, H.-P., "Evolutionsstrategie Und Numerische Optimierung," Department of Process Engineering, Technical University of Berlin, 1975.
Wagenhals, L. W., "Course of Action Development and Evaluation Using Discrete Event System Models of Influence Nets," PhD Dissertation, School of Information Technology and Engineering, George Mason University, 2000.
Zaidi, A. K., "On Temporal Logic Programming Using Petri Nets," IEEE Transactions on Systems, Man, Cybernetics A, vol. 29, 1999.
Zaidi, A. K. and Levis, A. H., "Temper: A Temporal Programmer for Time-Sensitive Control of Discrete Event System," IEEE Transactions on Systems, Man, Cybernetics A, vol. 31, 2001.