Inferring a multi-stage red-force plan using noisy observations of actions

By Paul R. Horridge
QinetiQ Ltd.

Abstract

In this paper, we suppose that an enemy ("red") force is carrying out some plan. We do not know which plan the red force is following so we wish to infer the probability that each of a pre-specified set of plans is in progress using noisy observations of red-force actions.

Each plan is modelled as having multiple stages with the dependencies between stages being modelled as a directed acyclic graph (DAG). We assume that a stage begins when all of its parents in the DAG have been completed. Multiple stages can be in progress at a particular time and each stage is assumed to take a random amount of time to finish depending on a pre-specified distribution which can be different for each stage.

A typical plan scenario could be as follows: red forces wish to attack a friendly ("blue") position. The attack plan involves using supporting units such as snipers and observers. These units need to be deployed before the main forces can advance, creating a dependence structure on the stages of the plan. Observations may consist of reports of troop movements or suspicious people who may be snipers or observers, as well as other data indicative of red-force activity. Alternatively, these activities may be a diversion from a different plan so we wish to determine as quickly as possible which plan is being followed.

The probability distribution of a plan is represented by a distribution over the set of stages which are either currently in progress or waiting for other stages to complete before the plan can progress. Each of these sets also has a probability distribution on the time remaining for the stages in the set to complete. We show how to propagate a plan distribution forward in time as well as update the distribution with observations. Each stage has a separate likelihood function of observations which are consistent with that stage, and some possible observations that may be associated with several of the stages. There is also the possibility of spurious observations which do not correspond to any stage.

The distribution over which of a set of plans is being followed can be inferred by maintaining a separate probability distribution of progress conditional on each plan, and performing Bayesian inference to update the plan probabilities. Using a null plan which produces solely spurious measurements can be used to determine if the received observations are not consistent with any of the pre-specified plans and the red force is doing something unexpected.

Results will be given to show that the approach can be used to distinguish between two plans consisting of around 30 stages each with a complicated dependence structure, as well as a null plan.

1. Introduction

In military planning, being able to quickly deduce the course of action of an adversary has obvious utility [Matheus et. al (2009)], although it has been argued that overreliance on predicted enemy intentions can be risky [Rector (1995)]. However, this task is complicated by the fact that a military plan is typically composed of multiple stages, with some stages being dependent on earlier stages. Furthermore, measurements of adversary actions are subject to error. In this paper, we formulate a model for each of a number of possible plans as well as a measurement model which incorporates uncertainty, and use Bayesian inference to update the plan probabilities given received measurements of enemy activity.

Mathematically, the problem can be expressed as follows. Let $z_{1:T} = \{z_1, \ldots, z_T\}$ a sequence of received measurements, where z_k is received at time t_k, and the measurements are in sequence, i.e. $t_1 \leq \ldots \leq t_T$. Suppose the plan being followed is labelled π. Then we wish to infer, for each $k = 1, \ldots, T$, the plan probabilities

$$p(\pi | z_{1:k}). \tag{1.1}$$

It will be seen that to accomplish this, it is necessary to specify probabilistic models for the plans and the measurements, and show how to propagate the probability distribution of a plan's progress through time.

The paper is organised as follows. In Section 2, we give a generative model to simulate how a plan will evolve over time by specifying the start and end times of its multiple stages, as well as a probability model for the measurements received from a plan. In Section 3, we show how to represent the probability distribution of the state of a plan at a particular time. In Section 4, we predict the probability distribution forward in time. In 5 we update the plan distribution based on a received measurement. In Section 6 we show how to deal with multiple plans and, for each plan, infer the probability that it is the one being followed. Section 7 gives some results on an example scenario.

2. Probability model for a single plan

Each plan requires a set of stages to be carried out with some stages requiring that other stages have been completed first. A natural way of representing such a dependency structure is with a directed acyclic graph (DAG), which can also be used to represent such dependency structures as dependent modules in a compiler, values in a spreadsheet or species in a food web [Gross et. al (2013)].

In this representation, each node represents a stage in the plan to be carried out, and a directed edge from node m and node n indicates that the stage for node m needs to be completed before the stage for node n can start. A small illustrative example is given in Figure 1. Note that the graph will have no cycles because this would involve a sequence of mutually dependent stages which would be impossible to start.

We use the following notation for graphs. Given a node n, let the parent set $\mathrm{pa}(n)$ be the set of nodes m for which there is an edge $m \to n$. Hence $\mathrm{pa}(n)$ is the set of prerequisite nodes for n. Similarly, let the child set $\mathrm{ch}(n)$ be the set of nodes m for which these is an edge $n \to m$. Let the ancestor set $\mathrm{an}(n)$ be the set of nodes m for which there is a directed path $m \to n$ and let the descendent set $\mathrm{de}(n)$ be the set of nodes m for which there is a directed path $n \to m$. Given a set of nodes A, we let $\mathrm{pa}(A)$ denote the set of nodes which are in $\mathrm{pa}(n)$ for some n in A, and similarly for $\mathrm{ch}(A)$, $\mathrm{an}(A)$ and $\mathrm{de}(A)$.

We suppose that each stage requires some amount of time to carry out. This is represented by each node n having a probability distribution $f_n(\cdot)$ over the non-negative real numbers. The starting time s_n and ending time e_n for each stage then can be simulated as follows. First we independently sample a duration time $\sigma_n \sim f_n(\cdot)$ for each node. The starting time for the stage corresponding to node n is considered to be the time when all of its prerequisite nodes have finished, or 0 for the nodes with no prerequisites, i.e.

$$s_n = \left(\max_{m \in \mathrm{pa}(n)} e_m \right) \vee 0 \tag{2.1}$$

It is straightforward to compute the topological order of a DAG [Skiena (2009)], so that visiting the nodes in this order means that each node is visited after all of its prerequisites. By doing this, we can ensure that the end times of the prerequisites have already been defined. The end times are just the start times plus the duration, i.e.

$$e_n = s_n + \sigma_n. \tag{2.2}$$

Thus we have a probabilistic model for the stages in progress at each time. It is possible that there might be a delay in starting a stage after its prerequisites have been completed. This can be represented by adding extra "do nothing" stages to the DAG.

We also need to define a model for the measurements we receive. Let $A(t)$ be the set of active stages at time t. This can be specified in terms of the start and end time definitions given in (2.1) and (2.2):

$$A(t) = \{ n \ : \ t \geq s_n \text{ and } t < e_n \} . \tag{2.3}$$

From this, we specify a probability distribution $p(z|A(t))$ on a measurement z received at time t.

There are many possible choices for this distribution and the best choice will depend on the details of the scenario being considered. In this paper we consider the following model:

- A probability distribution $p_n(z)$ is specified for each node n, as well as a distribution $p_C(z)$ for spurious (clutter) measurements. We also specify a probability P_d of a measurement being genuine as opposed to clutter.
- If $A(t)$ is empty, the measurement z is drawn from the clutter distribution $p_C(\cdot)$.
- If $A(t)$ is nonempty, z is drawn from $p_C(\cdot)$ with probability $1 - P_d$. Otherwise, a random node n is drawn from $A(t)$ and z is drawn from $p_n(\cdot)$.

It can be seen that the likelihood for this measurement model is

$$p(z|A(t)) = \begin{cases} p_c(z) & \text{if } A(t) = \emptyset \\ \frac{P_d}{|A(t)|} \sum_{n \in A(t)} p(z|n) + (1 - P_d) p_c(z) & \text{otherwise.} \end{cases} \tag{2.4}$$

3. Representing the probability distribution for a single plan

We represent the progress of a plan as follows. Let a *node-set* be a set of nodes in the DAG representing the nodes either in action or waiting for other nodes to finish. For example, in the tree given in Figure 1, the node-sets are $\{1,5,6\}$, $\{2,5,6\}$, $\{2,3,6\}$, $\{4\}$ and $\{7\}$. We say that a plan is in node-set $\mathcal{N}$ if, for each node in $\mathcal{N}$, the corresponding stage is either in progress or is waiting for one or more other stages to finish before progressing to the next stage, and no other stages are in progress. We also add the node-set $\emptyset$ to represent the state where the plan has completed.

Note that a plan is in exactly one node-set at a given time. Hence we can represent the uncertainty over which node-set it is in with a probability distribution $p(\mathcal{N})$ over the set of node-sets $\mathfrak{N}$. Given a particular node-set, each node n in it has some remaining time τ_n to complete (which may be zero if the corresponding stage is complete and is waiting for other stages). To keep the problem tractable, we assume that these variables are independent conditional on the node-set, so our belief about the progress of the plan at time t_k is represented by the distributions

$$\left.\begin{array}{c} p(\mathcal{N}^k) \\ \{p(\tau_n^k|\mathcal{N}^k) \ : \ n \in \mathcal{N}^k\} \end{array}\right\} \text{for } \mathcal{N}^k \in \mathfrak{N}. \tag{3.1}$$

The distributions $p(\tau_n^k|\mathcal{N}^k)$ can be represented by approximating the continuous time variable by a finite set of discrete bins.

As previously noted, a DAG is a natural way to represent multiple interdependent stages such as a plan. However, it is believed that this method of representing a plan's progress as a probability distribution is original.

The problem of performing inference for a single plan involves maintaining the probability distributions (3.1) conditional on the measurements seen so far, i.e. for $k = 1, \ldots, T$, computing

$$\left.\begin{array}{c} p(\mathcal{N}^k|z_{1:k}) \\ \{p(\tau_n^k|\mathcal{N}^k, z_{1:k}) \ : \ n \in \mathcal{N}^k\} \end{array}\right\} \text{for } \mathcal{N}^k \in \mathfrak{N}. \tag{3.2}$$

To do this, we start from the probability distribution at the previous time step:

$$\left.\begin{array}{c} p(\mathcal{N}^{k-1}|z_{1:k-1}) \\ \{p(\tau_n^{k-1}|\mathcal{N}^{k-1}, z_{1:k-1}) \ : \ n \in \mathcal{N}^{k-1}\} \end{array}\right\} \text{for } \mathcal{N}^{k-1} \in \mathfrak{N}. \tag{3.3}$$

This is predicted forward to time t_k without using the measurement z_{t_k}, as described in Section 4:

$$\left.\begin{array}{c} p(\mathcal{N}^k|z_{1:k-1}) \\ \{p(\tau_n^k|\mathcal{N}^k, z_{1:k-1}) \ : \ n \in \mathcal{N}^k\} \end{array}\right\} \text{for } \mathcal{N}^k \in \mathfrak{N}. \tag{3.4}$$

Finally, we condition this distribution on z_{t_k}, giving the distribution in (3.2), as described in Section 5. This pattern of predicting the distribution forward and conditioning on the new measurement is common in target tracking algorithms [Blackman & Popoli (1999)], where the goal is also to infer the probability of an evolving system using sequential measurements.

4. Predicting the plan forward in time

Here we describe how to predict the progress of a plan from time t_{k-1} to t_k. We set $\delta t = t_k - t_{k-1}$ to be the duration of the interval. The plan may pass through several node-sets in the time interval so we need a scheme to deal with this.

Let $\tilde{\mathcal{N}}^{k,m}$ be the node-set for time t_k after we have dealt with at most m transitions through node-sets, and let $\tilde{\tau}_n^{k,m}$ be the corresponding times remaining on each node $n \in \tilde{\mathcal{N}}^{k,m}$ after at most m transitions. We initialise these by keeping the node set distribution the same and subtracting the time interval δt from the time remaining for each node, since no transitions have been processed yet and time δt has elapsed.

$$p(\tilde{\mathcal{N}}^{k,0} = \mathcal{N}|z_{1:k-1}) = p(\mathcal{N}^{k-1} = \mathcal{N}|z_{1:k-1}) \tag{4.1}$$
$$p(\tilde{\tau}_n^{k,0} = x|\tilde{\mathcal{N}}^{k,0}, z_{1:k-1}) = p(\tau_n^{k-1} = x + \delta t|\mathcal{N}^{k-1}, z_{1:k-1}) \tag{4.2}$$

For the rest of this subsection, we suppress the conditioning on $z_{1:k-1}$ to simplify the notation.

A distribution over $\tilde{\tau}_n^{k,m}$ may assign probability to $\tilde{\tau}_n^{k,m} \leq 0$, representing node n having completed. Such a node will either start off one or more child nodes if the other prerequisite nodes have completed and the plan has not concluded, or be waiting for other prerequisite nodes. The time remaining for a newly started node may also have nonzero probability assigned to negative time, so the same probability mass in the plan distribution may transition through several node-sets. The aim is to propagate $\tilde{\mathcal{N}}^{k,m}$ and $p(\tilde{\tau}_n^{k,0}|\tilde{\mathcal{N}}^{k,0})$ through transitions by calculating

$$p(\tilde{\tau}_n^{k,m}, \tilde{\mathcal{N}}^{k,m}) = \sum_{\tilde{\mathcal{N}}^{k,m-1}} p(\tilde{\tau}^{k,m}|\tilde{\mathcal{N}}^{k,m-1}, \tilde{\mathcal{N}}^{k,m}) p(\tilde{\mathcal{N}}^{k,m}|\tilde{\mathcal{N}}^{k,m-1}) p(\tilde{\mathcal{N}}^{k,m-1}) \tag{4.3}$$

until there are no more transitions. Thus we get the final distributions $p(\mathcal{N}^k)$ and $p(\tau_n^k|\mathcal{N}^k)$.

The first step is to compute the transitions $p(\tilde{\mathcal{N}}^{k,m}|\tilde{\mathcal{N}}^{k,m-1})$. To do this, we consider which other node-sets can be directly reached from a node-set $\mathcal{N}$. Each of the nodes in $\mathcal{N}$ can either complete or not complete in the time interval. Let $\Theta \subseteq \mathcal{N}$ be the nodes which complete. Then a child node of $\mathcal{N}$ which starts is one with all of parents either in Θ or ancestor nodes of $\mathcal{N}$:

$$V(\mathcal{N}, \Theta) = \{m \in \text{ch}(\mathcal{N}) \ : \ \text{pa}(m) \subseteq \text{an}(\mathcal{N}) \cup \Theta\}. \tag{4.4}$$

These nodes are then added to the node set, and the nodes in the node-set with at least one child starting are removed. Let $W(\mathcal{N}, \Theta)$ be those nodes in $\mathcal{N}$ which have a child in $V(\mathcal{N}, \Theta)$:

$$W(\mathcal{N}, \Theta) = \{n \in \mathcal{N} \ : \ \text{ch}(n) \cap V(\mathcal{N}, \Theta) \neq \emptyset\}. \tag{4.5}$$

Hence the new node-set is:

$$\mathcal{N}(\Theta) = \mathcal{N} \cup V(\mathcal{N}, \Theta) \setminus W(\mathcal{N}, \Theta). \tag{4.6}$$

We can therefore determine which node-sets it is possible to directly reach from a particular node-set. Since these calculations only depend on the structure of the graph, performing them off-line will give a substantial computational saving.

Note that the transitions from a node-set to a different node-set impose a directed graph over the set of node-sets. Moreover, in a transition, at least one node is removed and since nodes are always replaced by children it is impossible for a sequence of transitions to lead back to the the original node-set. Hence this graph is acyclic and thus also a DAG — Figure 2 shows the node-set DAG for the plan in Figure 1.

An important consequence of this DAG structure is that we can carry out the transitions (4.3) with a single pass through the node-sets in topological order, since once a sequence of transitions has visited a node-set, it cannot revisit the same node-set.

The probabilities of the node-set transitions are given by the probabilities of each node completing. By the independence approximation (3.1) we have, for $\Theta \subseteq \mathcal{N}^{k,m-1}$,

$$\begin{aligned} &p(\tilde{\mathcal{N}}^{k,m} = \tilde{\mathcal{N}}^{k,m-1}(\Theta)|\tilde{\mathcal{N}}^{k,m-1}) \\ &\qquad = \prod_{n\in\Theta} p(\tilde{\tau}_n^{k,m-1} \leq 0|\tilde{\mathcal{N}}^{k,m-1}) \prod_{n\in\tilde{\mathcal{N}}^{k,m-1}\setminus\Theta} p(\tilde{\tau}_n^{k,m-1} > 0|\tilde{\mathcal{N}}^{k,m-1}). \end{aligned} \tag{4.7}$$

It therefore remains to calculate, for $\Theta \subseteq \tilde{\mathcal{N}}^{k,m-1}$ and $n \in \tilde{\mathcal{N}}^{k,m-1}(\Theta)$,

$$p(\tilde{\tau}_n^{k,m}|\tilde{\mathcal{N}}^{k,m-1}, \Theta). \tag{4.8}$$

These nodes fall into three cases:

(*a*) Nodes which do not complete, i.e. $n \in \mathcal{N}^{k,m-1} \setminus \Theta$.

(*b*) Nodes which complete but are waiting for other nodes to complete in order to progress, i.e. $n \in \Theta \setminus W(\tilde{\mathcal{N}}^{k,m-1}, \Theta)$.

(*c*) Newly started child nodes, i.e. $n \in V(\tilde{\mathcal{N}}^{k,m-1}, \Theta)$.

For cases (a) and (b), the node n stays in the same node-set so the time remaining remains unchanged: $\tilde{\tau}_n^{k,m} = \tilde{\tau}_n^{k,m-1}$. In case (a), the time remaining is conditioned to be positive:

$$p(\tilde{\tau}_n^{k,m} = x|\mathcal{N}^{k,m-1}, \Theta) = p(\tilde{\tau}_n^{k,m-1} = x|\mathcal{N}^{k,m-1}, \tilde{\tau}_n^{k,m-1} > 0) \tag{4.9}$$

$$\propto p(\tilde{\tau}_n^{k,m-1} = x, x > 0|\mathcal{N}^{k-1}). \tag{4.10}$$

In case (b), the time remaining is conditioned to be negative:

$$p(\tilde{\tau}_n^{k,m} = x|\mathcal{N}^{k,m-1}, \Theta) \propto p(\tilde{\tau}_n^{k,m-1} = x, x \leq 0|\mathcal{N}^{k-1}). \tag{4.11}$$

Case (c) is more complicated since n is in a new node set. Let $S \subseteq \text{pa}(n) \cap \tilde{\mathcal{N}}^{k,m-1}$ be the set of nodes which completed to allow this node to start (note that parent nodes not in $\tilde{\mathcal{N}}^{k,m-1}$ must be ancestors of $\tilde{\mathcal{N}}^{k,m-1}$ and thus already completed). Let ζ_n be the time before the end of the interval before the node started, i.e.

$$\zeta_n = \min_{r\in S}(-\tilde{\tau}_r^{k,m-1}). \tag{4.12}$$

Then the time remaining for node n is given by

$$\tilde{\tau}_n^{k,m} = \sigma_n - \zeta \tag{4.13}$$

where σ_n is the duration of node n. The conditional complementary cumulative function on $-\zeta_n$ is given by

$$p(-\zeta_n \geq x|\mathcal{N}^{k,m-1}, \Theta) = p(\tilde{\tau}_r^{k,m-1} \geq x \text{ for all } r \in S|\mathcal{N}^{k,m-1}, \Theta) \tag{4.14}$$
$$= \prod_{r \in S} p(\tilde{\tau}_r^{k,m-1} \geq x|\mathcal{N}^{k,m-1}, \Theta) \tag{4.15}$$

where

$$p(\tilde{\tau}_r^{k,m-1} \geq x|\mathcal{N}^{k,m-1}, \Theta) = p(\tilde{\tau}_r^{k,m-1} \geq x|\mathcal{N}^{k,m-1}, \tilde{\tau}_r^{k,m-1} < 0) \tag{4.16}$$
$$= \frac{\int_x^0 p(\tilde{\tau}_r^{k,m-1} = y|\mathcal{N}^{k,m-1})\, dy}{\int_{-\infty}^0 p(\tilde{\tau}_r^{k,m-1} = y|\mathcal{N}^{k,m-1})\, dy}. \tag{4.17}$$

Finally, the distribution of $\tilde{\tau}_n^{k,m}$ can be calculated using the convolution

$$p(\tilde{\tau}_n^{k,m} = x|\mathcal{N}^{k,m-1}, \Theta) = \int_0^\infty p(\sigma_n = y) p(-\zeta_n = x - y|, \mathcal{N}^{k,m-1}, \Theta)\, dy. \tag{4.18}$$

5. Updating the plan using the measurement likelihood

To update our probabilistic representation (3.2) of the plan given a measurement z_k at time t_k, we need to compute, for each node set $\mathcal{N}^k$

$$p\left(\mathcal{N}^k|z_k\right) \tag{5.1}$$
$$p\left(\tau_n^k|\mathcal{N}^k, z_k\right) \text{ for } n \in \mathcal{N}^k. \tag{5.2}$$

Again we assume that the previous measurements are implicit and omit them from the notation. By Bayes' Theorem,

$$p\left(\mathcal{N}^k|z_k\right) \propto p\left(z_k|\mathcal{N}^k\right) p(\mathcal{N}^k) \tag{5.3}$$
$$p\left(\tau_n^k|\mathcal{N}^k, z_k\right) \propto p\left(z_k|\tau_n^k, \mathcal{N}^k\right) p\left(\tau_n^k|\mathcal{N}^k\right). \tag{5.4}$$

To compute the likelihood, recall that in Section 2 we defined it in terms of the set $A(t_k)$ of actions in progress at time t_k. The actions in progress are those in the current node-set which are not waiting, i.e. have some time left to complete:

$$A(t_k) = \left\{n \in \mathcal{N}^k \ : \ \tau_n^k > 0\right\}. \tag{5.5}$$

Let $a_n = 1$ if $\tau_n^k > 0$ and zero otherwise, and let $a_{\mathcal{N}^k} = (a_n : n \in \mathcal{N}^k)$. Then

$$p\left(z_t|\mathcal{N}^k\right) = \sum_{a_{\mathcal{N}^k}} p\left(z_k|a_{\mathcal{N}^k}, \mathcal{N}^k\right) p\left(a_{\mathcal{N}^k}|\mathcal{N}^k\right) \tag{5.6}$$

where

$$p\left(a_{\mathcal{N}^k}|\mathcal{N}^k\right) = \prod_{n \in \mathcal{N}^k} p\left(\tau_n^k = 0|\mathcal{N}^k\right)^{1-a_n} p\left(\tau_n^k > 0|\mathcal{N}^k\right)^{a_n}. \tag{5.7}$$

Furthermore,

$$p(z_k|a_n^k, \mathcal{N}^k) = \frac{p(z_k, a_n^k|\mathcal{N}^k)}{p(a_n^k|\mathcal{N}^k)} \tag{5.8}$$
$$= \frac{\sum_{a_{\mathcal{N}^k \setminus \{n\}}} p\left(z_k|a_{\mathcal{N}^k}, \mathcal{N}^k\right) p\left(a^{\mathcal{N}^k}|\mathcal{N}^k\right)}{p\left(a_k^n|\mathcal{N}^k\right)}. \tag{5.9}$$

Plugging (5.6) and (5.9) into (5.3) and (5.4) respectively allows us to compute the updated distributions. Also, we can compute the total likelihood

$$p(z_k) = \sum_{\mathcal{N}^k} p(z_t|\mathcal{N}^k) p(\mathcal{N}^k) \tag{5.10}$$

which will be useful when we need to distinguish between plans later.

6. Multiple plans

Up to this point, we have been concerned with tracking the progress of a single plan. This can provide useful intelligence but in general the plan that an enemy is following is unknown and must be inferred as well. In this section, we suppose that there are a number of plans pre-specified as above and we wish to infer the probabilities of each. We will also show how to define a "null plan" which allows for the possibility that the measurements are not consistent with any of the plans and that the enemy is doing something unexpected.

Suppose we have a discrete variable π representing the plan being followed (which we assume remains unchanged over the scenario). The problem is to infer the probability distribution (1.1) over this variable using measurements received over different times. The updated plan probabilities for each epoch can be computed from the previous epoch probabilities as follows. By Bayes' theorem,

$$p(\pi|z_{1:k}) = \frac{p(z_k|\pi, z_{1:k-1})p(\pi|z_{1:k-1})}{p(z_k|z_{1:k-1})}. \tag{6.1}$$

Here, $p(z_k|z_{1:k-1})$ is just a normalising constant and $p(z_k|\pi, z_{1:k-1})$ is given by (5.10), the total measurement likelihood for plan π. The initial plan probabilities $p(\pi)$ are set to be the prior probabilities representing the relative probability of each plan being followed.

To deal with the possibility of none of the specified plans being followed, we introduce a null plan π_0. This plan has no stages and does not evolve over time. The measurement model for the null plan is taken to be the clutter model from Section 2:

$$p(z_k|\pi_0) = p_C(z_k). \tag{6.2}$$

It may be argued that an enemy's plan will not look very much like clutter, but it is difficult to come up with an accurate model for something unpredicted. Inferring the probability of this "plan" can be performed using (6.1) in the same way as the others.

7. Results

To test this approach, we run it on two plans having 30 and 33 stages respectively, with the DAGs given in Figure 3. It can be seen that each plan has a complicated dependence structure. We suppose that each measurement received is an integer between 1 and 43 corresponding to an activity taking place which may be observed. The measurement likelihood p_n^π for each measurement and each plan assigns probability 0.95 to the measurements listed in Table 1, and probability 0.05 to a random measurement. Note that although this table only lists a single measurement for each node and plan combination, our approach allows a node likelihood to be a more general probability distribution over the measurements. Dashes in the table denote nodes where nothing is happening, so the measurement likelihood for these nodes is the clutter likelihood which we assume is uniform across all measurements. Also, several measurement IDs are common to both plans, adding some ambiguity to the problem. The detection probability P_d of receiving a genuine measurement as opposed to clutter is taken to be 0.9.

For the purposes of this scenario, we take the time each node takes to complete to be a Gaussian truncated at 0 with the Gaussian mean being 10 time units and the standard deviation being 5. We receive measurements every 5 time units from $t = 5$ to $t = 150$. We also add in the null plan as described above, with each measurement being drawn from the clutter distribution.

10 sets of data are simulated from each of the 3 plans, and the algorithm is run on each. For each true plan, we show the average estimated probability of each plan inferred by the algorithm over time for the 10 runs in Figure 4. This shows that the probability estimates converge quickly to the true plan, although Plan 2 takes longer to converge than the others, being harder to distinguish from the null plan.

8. Acknowledgements

This work was supported by the Defence Science & Technology Laboratory, an agency of the UK Ministry of Defence.

Node	Plan 1	Plan 2	Node	Plan 1	Plan 2	Node	Plan 1	Plan 2	Node	Plan 1	Plan 2
1	29	13	10	12	4	19	35	8	28	41	24
2	30	14	11	7	5	20	43	-	29	42	25
3	31	15	12	8	19	21	36	9	30	-	-
4	1	16	13	9	20	22	37	10	31		26
5	2	17	14	10	21	23	-	28	32		27
6	3	18	15	32	-	24	6	6	33		-
7	4	1	16	33	11	25	40	22			
8	5	2	17	34	12	26	38	23			
9	11	3	18	-	7	27	39	-			

TABLE 1. The measurement reported for each node in each plan. If plan π is in progress and a non-clutter measurement is reported from node n, the measurement will be as in the table above. Dashes denote that the node likelihood is the clutter distribution.

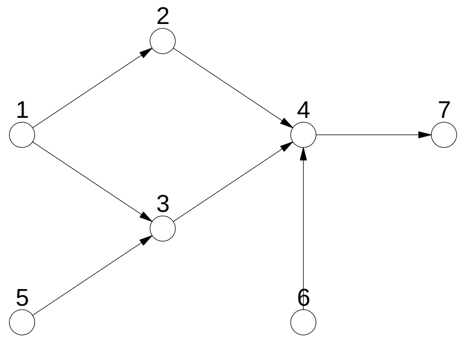

FIGURE 1. An example of a DAG representation of a plan. Here, for example, stages 1 and 5 need to be carried out before stage 3 can be started, whereas stage 2 can begin with just the completion of stage 1. Stage 7 is the final phase of the plan, and requires stage 4 to have been completed.

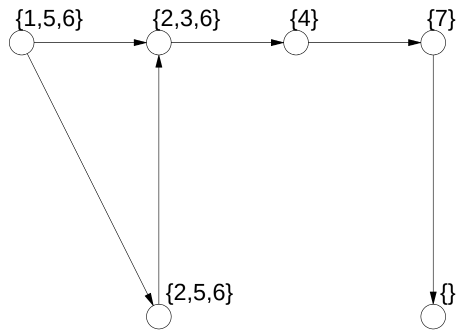

FIGURE 2. The DAG of the node-set transitions for the DAG in Figure 1.

REFERENCES

GROSS, J. L., YELLEN, J. & ZHANG, P. 2013 Handbook of Graph Theory (2nd ed.), *CRC Press*, p. 1181.

SKIENA, S. S. 2009 The Algorithm Design Manual, *Springer*, pp. 179–181.

RECTOR, R. L. 1995 Enemy Course of Action Prediction: Can We, Should We?, *School of Advanced Military Studies, US Army Command and General Staff College, Fort Leavenworth, Kansas.*

MATHEUS, C. J., BRIAN, U., MIECZYSLAW, M. K. & POWELL, G. 2009 On the Formal Representation of Enemy Courses of Action, *Proceedings of the 14th International Command and Control Research and Technology Symposium.*

BLACKMAN, S. & POPOLI, R. 1999 Design and Analysis of Modern Tracking Systems, Artech House.

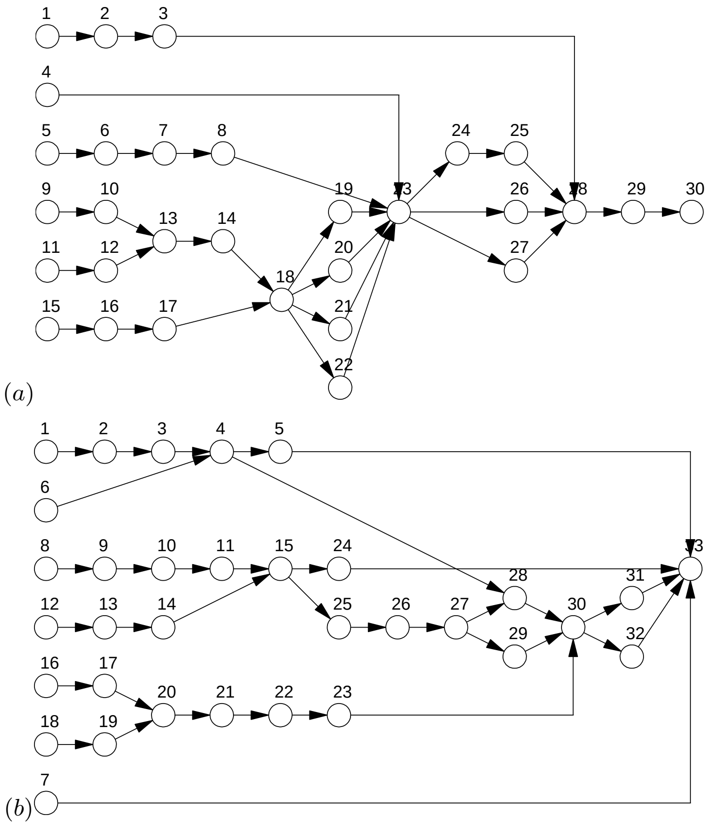

FIGURE 3. The DAGs for (a) Plan 1, and (b) Plan 2.

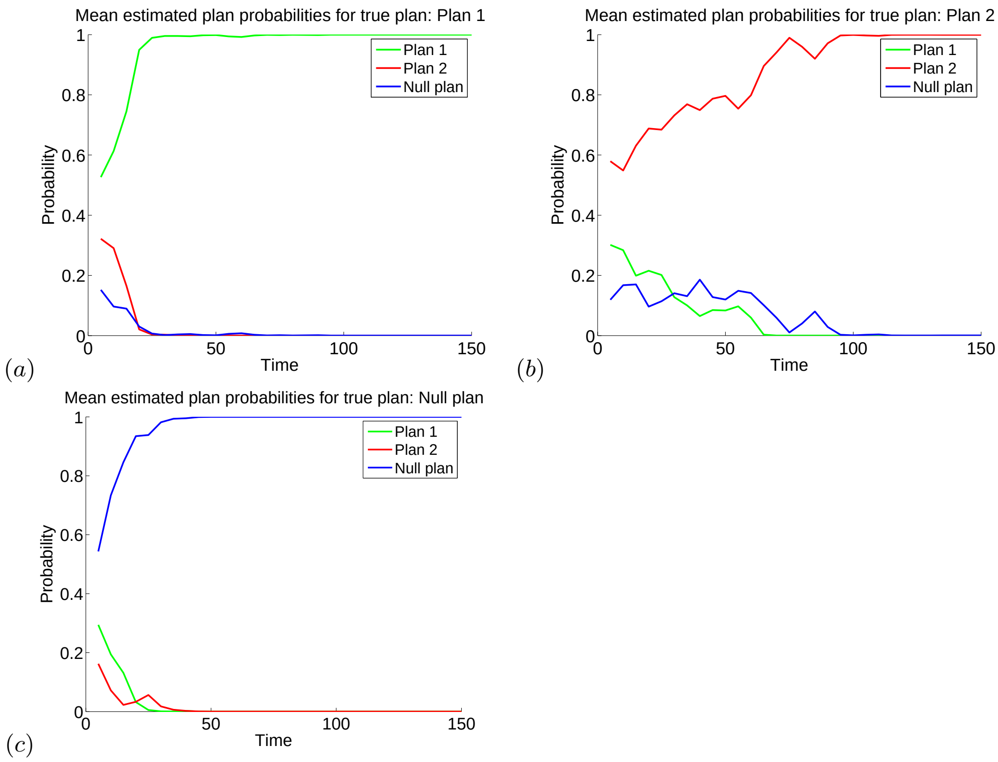

FIGURE 4. The estimated probability of Plan 1 (green), Plan 2 (red) and Plan 3 (blue) averaged over 10 runs with the true plan set to be (a) Plan 1, (b) Plan 2, and (c) the null plan.