Discounted MEAN bound for the optimal searcher path problem with non-uniform travel times

Haye Lau, Shoudong Huang, Gamini Dissanayake
Faculty of Engineering, University of Technology, Sydney
ARC Centre of Excellence for Autonomous Systems (CAS),
PO Box 123, Broadway, NSW 2007, Australia
{hlau,sdhuang,gdissa}@eng.uts.edu.au

Abstract

We consider an extension of the optimal searcher path problem (OSP), where a searcher moving through a discretised environment may now need to spend a non-uniform amount of time travelling from one region to another before being able to search it for the presence of a moving target. In constraining not only where but when the search of each cell can take place, the problem more appropriately models the search of environments which cannot be easily partitioned into equally-sized cells. An existing OSP bounding method in literature, the MEAN bound, is generalised to provide bounds for solving the new problem in a branch and bound framework. The main contribution of this paper is an enhancement, Discounted MEAN (DMEAN), which greatly tightens the bound for the new and existing problems alike with almost no additional computation. We test the new algorithm against existing OSP bounding methods and show it leads to faster solution times for moving target search problems.

Keywords: Search Theory; Branch and Bound; Global Optimization; Markov Processes

1. Introduction

We consider the search for a target moving between discrete regions of interest in a structured environment. At each time step, the target either stays in its current region or shifts to a different location according to a known Markov model. After searching a region for one time period, a searcher can continue in the same place or move to search an adjacent region instead. For each time step the searcher spends in the same region as the target, detection can occur with a given glimpse probability. A search plan respecting searcher travel constraints is sought to maximise the probability of detection within a specified time frame. Much work has been done in the past on this path constrained or optimal searcher path problem (OSP) (Stewart, 1979; Eagle and Yee, 1990; Dell et al. 1996; Hohzaki, 1997; Washburn 1998), for the specific case where the environment is discretised into a grid of identically-sized cells and a searcher can immediately search one cell after another in successive time periods. In contrast, a platform searching a built environment cannot move from one region to another without expending some travel time for each transition. Moreover, the travel times required depend on the source and destination regions and may not be the same across all transitions; the effective total time available for searching therefore becomes linked to the choice of search actions itself. The purpose of this paper is to extend the idealised cell model in the optimal searcher path problem to incorporate non-uniform searcher travel time between cells, such that the formulation can model the search of regions in structured environments while preserving the underlying inter-region travel constraints. The main contribution is a branch and bound procedure using a new bounding method that not only generalises the MEAN method by Martins (1993) to solve the new problem, but also greatly tightens obtained bounds for the optimal searcher path problem in general with almost no added computation.

Unlike earlier problems that assumed the ability to arbitrarily divide search effort between all regions within a single time period (Brown, 1980), the optimal searcher path problem accounts for cases where the speed of redeploying effort is slow with respect to the detection time interval and the searcher (e.g. a ship or manned patrol) cannot concurrently cover multiple regions. Not only is search effort indivisible in this sense, the location where it can be placed at a given time is also constrained by the region searched in the previous step. Although this formulation has mostly been used in literature for maritime scenarios, it is clear that similar constraints also apply for searches in terrestrial or structured settings. Instead of dividing an open area into a uniform grid of cells, a built environment can more sensibly be partitioned according to its structure such that the constituent regions (such as rooms or individual buildings) can then be independently searched. Nevertheless, a searcher, be it human or robotic, still occupies only one region at a time and travel between regions is limited by what the structure (in the form of obstacles and region connectivity) allows. The task of finding the best track through the regions remains.

Viewed in this light, the situation in previous optimal searcher path problems can be seen as a searcher being able to expend some amount of time searching one region, with no intervening "travel time" necessary before effort is then immediately shifted to a neighbouring region. While adequate for searching open areas with a fast platform, this assumption is not directly suitable for our intended applications of searching an indoor area consisting of different-sized rooms or a cluster of separated buildings in a given area - a searcher in such cases would reasonably be expected to devote time moving from one region to another before resuming search.

Although many related problems in literature factor in some incurred costs from a searcher looking in each region, few cases exist where the specific travel times from one region or cell to another are directly taken into account. In one example, Hohzaki and Iida (1997) raised the idea of capturing the cost of travel between the successive cells in a search sequence. This was however envisioned as a purely financial expense, with no impact on the effective time available for further searches. The searching of different cells by Lössner and Wegener (1982), DasGupta et al. (2006) and our previous work (Lau et al., 2005) do incur such a switch time, but all dealt only with stationary targets. Interestingly, Dambreville and Le Cadre (2002) raised a Markovian target search formulation that allows a search resource to renew itself only after spending a certain amount of time for moving. The formulation is nevertheless concerned only with allocating an infinitely divisible search resource whose future availability is not conditioned on the searcher's current position, and therefore still does not meet the travel requirements in the scenarios we consider. Existing optimal searcher path problem formulations can approximate the case of location-dependent, non-uniform travel times by first injecting additional identical cells to render travel times uniform, but this then assumes a very myopic sensor, does not take advantage of any inherent environment structure, and would therefore unnecessarily escalate the overall complexity.

The rest of this paper is organised as follows. Section 2 reiterates the existing OSP problem formulation, while Section 3 describes a branch and bound procedure for the problem as well as the operation of MEAN, an existing bounding method proposed by Martins (1993). Section 4 introduces a new bounding method, Discounted MEAN (DMEAN), which is shown in Section 5 to be superior to other known linear bounds. Section 6 then generalises the OSP formulation to directly incorporate searcher travel times between cells, such that it better reflects the limits of a physically-situated searcher in structured or enclosed environments. The earlier branch and bound procedure and DMEAN method are also extended to find solutions for this Optimal Searcher Path Problem with non-uniform Travel times (OSPT).

2. Optimal Searcher Path Problem Formulation

The searcher and target move through an environment divided into a finite set of cells $C = \{1,...,N\}$(see Figure 1). The target occupies one cell (representing a region) at a time and moves according to a specified Markov model at each time step; a matrix Γ describes the probability that a target will move from any of the cells to another at the next time step. A prior distribution $p(\cdot,1)=[p(1,1),p(2,1),...,p(N,1)]$ of the target at time 1 is initially supplied, where $p(i,t)$ is the probability that the target is in cell i at time t without being detected by any searches before t. In the absence of searches, the (possibly defective) distribution evolves according to the formula $p(\cdot,t+1)=p(\cdot,t)\cdot\Gamma$.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

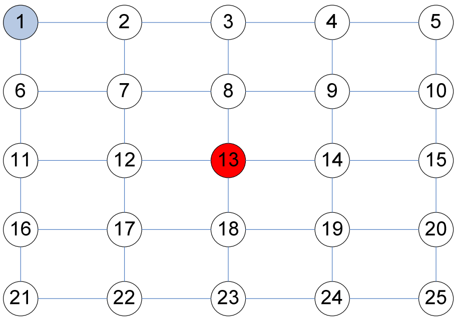

Figure 1 Example OSP search grid and equivalent graph: edges show valid searcher transitions. Results will be illustrated with cases where the searcher begins in cell 1 and the target starts at the centre cell.

Target detection is modelled as follows: if both the searcher and target are in region i during time t, detection occurs with a glimpse probability of $g(i,t)$. As an example, if $p(\cdot,t)=[p(1,t),...,p(N,t)]$ and cell 1 is searched for one time step, then $p(\cdot,t+1)=[p(1,t)\cdot(1-g(1,t)),p(2,t),...,p(N,t)]\cdot\Gamma$. The glimpse function may typically take the form of $g(i,t)=1-e^{-\omega(i,t)}$ (Dell et al., 1996, Eagle and Yee, 1990), with $\omega(i,t)\geq 0$ being a measure of search effectiveness for a given cell i. Any function $0\leq g(i,t)\leq 1$ however can be used. This probability is assumed to be independent of past searches.

The searcher's path is constrained by the structure of the environment. Let $S(i),i\in C$ be the set of cells that a searcher can directly move to from cell i. If the searcher is in cell i at time t, it can only start searching the next cell $j\in S(i)$ at time $t+1$.

Given T time steps to find the target, let ψ denote a valid search plan represented by a series of cells searched in one time unit increments. A searcher following plan ψ first moves to and searches cell $\psi(1)$ for one time period, then travels to search cell $\psi(2)$ for another time step, and continues for the remaining cells until T time periods in total have been expended. For convenience, we use $\psi(0)$ to hold the searcher's given initial position prior to the first search.

Taking into account target motion and the effects of previous cell searches, the undetected target probability mass in a cell at each time period is determined by:

$$p(\cdot,t+1)=p(\cdot,t)\cdot M_{\psi(t)t}\cdot\Gamma, 1\le t<T \tag{1}$$

Where $M_{\psi(t)t}$ is an $N\times N$ identity matrix with the $\psi(t)^{\text{th}}$ diagonal element set to $1-g(\psi(t),t)$.

The objective for the Optimal Searcher Path problem (OSP) is to find the search plan ψ that maximises the cumulative probability of detection $PD(\psi)$ within the T time steps, which can then be stated as:

$$\max_{\psi} PD(\psi)=\sum_{t=1}^{T} p(\psi(t),t)\cdot g(\psi(t),t)$$

Subject to:

$$\psi(t+1)\in S(\psi(t)), t\in 0,...,T-1 \tag{2}$$

The glimpse functions $g(\psi(t),t)$ are given and the undetected target probability $p(\psi(t),t)$ can be obtained using equation (1). Alternative formulations exist in literature for the OSP (Thomas and Eagle 1995; Washburn, 1995), but the above, similar to that used in Dell et al. (1996), is chosen here to make clear the intended generalisation in Section 6.

3. Branch and Bound for the OSP

The complexity of the problem arises because the usefulness of searching an individual cell towards maximising the cumulative probability of detection is governed by the probability of the undetected target being present at that time, which in turn is a function of all previous search actions thus far. Branch and bound approaches (Stewart, 1979; Washburn, 1995; Hohzaki, 1997), which implicitly enumerate feasible searcher actions, have been popular for optimally solving the NP-complete (Trummel and Weisinger, 1986) OSP. Instead of examining every possible solution, the technique takes advantage of the fact that all extensions of a partial search sequence can be ruled out together if they cannot possibly lead to the optimal outcome. Central to the approach is estimating the best payoff achievable from any valid continuation of a given search plan – if the value does not exceed the best known solution, an entire "branch" of related plans can then be safely abandoned.

The algorithm below adopts the approach used in Washburn (1995). $K(s)$ is a set of 3-tuples $\{nextcell, time, upperbound\}$ representing path continuations yet to be explored after a particular sequence of s cells have been searched. The first field refers to the next cell to search for one time step, the second field is the total time expended once the specified cell is searched, and the third contains the upper cumulative probability of detection (PD) bound associated with this particular extension. $p*$ holds the best detection probability hitherto found.

Algorithm for Branch and Bound (OSP):

1. Let $\sigma_0 = \psi(0)$. Set $s = 0, K(s) = \{\{\sigma_s, 0, 0\}\}$ and $p*$ to a value below 0.
2. If $K(s)$ is empty, let $s = s - 1$, else go to 4.
3. If $s = 0$, go to 9, else go to 2.
4. Selection: Choose a tuple $\{\sigma_s, \tau_s, p_s\}$ in $K(s)$ according to a selection criterion and remove it from the set.
5. If $p_s \leq p*$, this extension can be fathomed. Go to 2.
6. Else Branch: For each cell $\sigma_c \in S(\sigma_s)$, if $\tau_s < T$, obtain p_c, the upper PD bound for any plan beginning with the path $\{\sigma_0, ..., \sigma_s, \sigma_c\}$. Add tuple $\{\sigma_c, \tau_s + 1, p_c\}$ to $K(s+1)$.
7. If no tuples were added to $K(s+1)$ in 6, the current extension is a leaf and no more searches can be done. Let $p* = p_s$ and store $\{\sigma_0 ..., \sigma_s\}$ as the new incumbent best path. Go to 2.
8. Else let $s = s + 1$, go to 4.
9. Stop, the last saved path is optimal with the maximum PD of $p*$.

The algorithm employs an eager depth-first branch and bound approach as per Washburn (1995). The selection of partial plans for branching in step 4 can be arbitrary; for the purpose of the examples in this paper, the tuple $\{\sigma_s, \tau_s, p_s\}$ in $K(s)$ with the highest bound p_s is always chosen. It is assumed that bounds for a plan which cannot be extended to include more searches will always equal its actual PD. Step 7 can directly store the new incumbent without checking as its superiority is implied by step 5.

3.1 Obtaining bounds for the OSP

The purpose of finding an upper bound (step 6 in the Algorithm for Branch and Bound) is to quickly estimate the best achievable payoff in a given solution subspace without exhaustively examining each full plan. The tighter a bound is in relation to the actual achievable PD, the more branches can be fathomed in advance. On the other hand, a tight bound which requires too much effort to calculate may actually slow down the whole solution process.

Different aspects of the OSP can be relaxed to facilitate bound evaluation. Washburn (1995, 1998) details a number of known bounding techniques and examines the trade off between bound tightness and calculation speed. The methods reviewed will be referred to by the same names in this paper for consistency. In Stewart (1979)'s depth-first branch and bound framework, path constraints were simplified such that a searcher can visit any cell reachable from the starting location by a given time, even if it is not directly connected to the last cell searched. Although a problem with this distribution of effort relaxation was then iteratively solved with the Forward and Backward (FAB) algorithm (Brown, 1980), the resultant solutions gave heuristic bounds not guaranteed to find the optimal answer. When the searcher is permitted to span multiple cells at the same time, the FAB algorithm itself leads to a true and tight bound. Eagle and Yee (1990) also solved a relaxed problem made convex by no longer confining the searcher to be in one cell at a time. Eschewing sharpness for calculation speed, Martins (1993)'s MEAN method made linear relaxations through transforming the OSP into a longest path problem maximising the expected number of detections that preserves both searcher indivisibility and path constraints. When the path constraints are relaxed to form a reward collection problem, the more easily evaluated bound PROP (Washburn, 1998) is obtained. Lastly, ERGO2 (Washburn, 1998) estimates bounds with even less computation by directly using a stationary target distribution rather than the actual distribution at each time.

As the MEAN bound is also amenable to the new problem to be defined in Section 6, its operation will be described in more detail before an improvement, DMEAN, is introduced in Section 4.

3.2 The MEAN bound for the OSP

The MEAN bound functions by maximising the expected number of detections in the remaining time steps until T. Following Martins (1993), define $P(D_\psi = x)$ as the probability that x detections occur when the searcher follows a sequence ψ from time 1 to T. Taking an expectation of the number of detections yields:

$$ED(\psi) = \sum_{x=1}^{T} xP(D_\psi = x) \geq \sum_{x=1}^{T} P(D_\psi = x) = PD(\psi) \qquad (3)$$

Since the expected number of detections (ED) for a particular plan cannot be smaller than the corresponding probability of detection (PD), the maximal ED across all search plans also provides an upper bound for PD itself.

Under this modified objective, the utility of searching a cell at a given time only depends on the probability of the target being there and is independent of any previous searches that might have already occurred. Prior actions only play a role in limiting where the next searches could take place due to the cell connectivity constraints; calculating the MEAN bound's value can thus be modelled as finding a longest path in a directed acyclic graph (DAG).

Consider the calculation illustrated in the following 1-dimensional example where a searcher can either stay in its current cell or move sideways towards an adjacent one at each time step. Let σ be a fixed sequence of k one-time-unit searches such that the search of cell $\sigma(k)$ takes place at time $k.$ We wish to find the maximum ED possible for any extension of this plan from time $k+1$ up to time $T,$ which can then be added to the known PD for searches during times $1,...,k$ to give the upper PD bound for any sequence beginning with the partial plan $\sigma.$

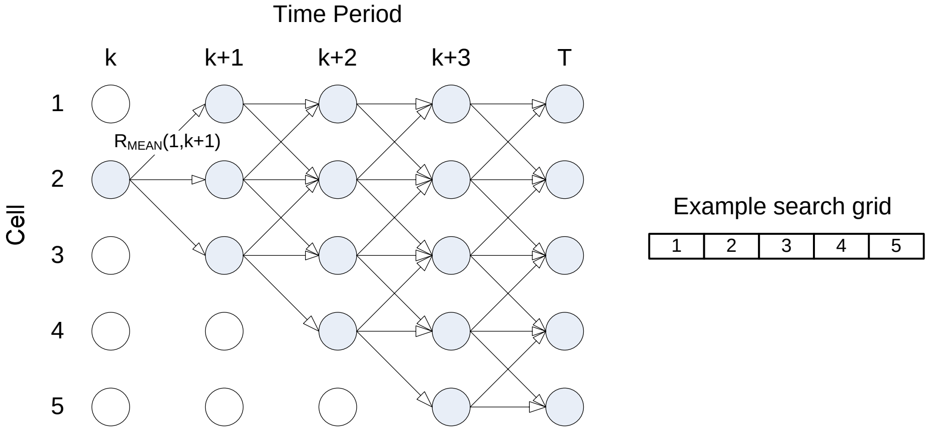

Figure 2 MEAN bound calculation for the OSP ($\sigma(k)=2$). Arcs indicate feasible searcher movements.

Figure 2 shows how the MEAN bound can be found through solving a longest path problem in a network where each node represents the search of a cell at a particular time. Starting with the constraining node $\{\sigma(k),k\}$ denoting the last fixed searcher position, let directed arcs indicate the valid searcher movements from one time to the next. To determine the maximum ED for the interval from time $k+1$ to $T,$ we associate a reward for searching a cell with the arc entering the corresponding node. Let $\pi(k)$ be the undetected target probability mass at time $k,$ taking into effect the searches of $\sigma(1)$ through to $\sigma(k).$ By letting $P(\cdot,t)=\pi(k)\cdot\Gamma^{t-k}, t>k, P(j,t)$ is then the probability that the target was not detected by any searches up to time k and is now in cell j at time t. Accordingly, each arc heading into node $\{j,t\}, j \in C, t \in \{k+1,...,T\}$ is given a weight of $R_{MEAN}(j,t) = P(j,t) \cdot g(j,t),$ reflecting the contribution towards the total ED if cell j is searched at time t. A standard DAG longest path algorithm such as in Cormen et al. (1990) can then be applied on the network to maximise the total reward and obtain an upper bound on the achievable PD. The steps for obtaining the MEAN bound are summarised as follows.

Algorithm for the MEAN Bound

1. For each time step from k to T, create a graph node per cell at that time. Mark node $\{\sigma(k),k\}$ as valid.
2. Use $P(\cdot,t) = \pi(k) \cdot \Gamma^{t-k}$ to calculate $P(\cdot,t)$ for $k < t \leq T$.
3. From each valid node $\{i,t\}$, extend arcs of weight $P(j,t+1) \cdot g(j,t+1)$ to all nodes $\{j,t+1\}, j \in S(i), t+1 \leq T$. Mark the head nodes $\{j,t+1\}$ valid.
4. Repeat 3 until arcs have been extended from all valid nodes.
5. Apply a DAG longest path algorithm to find the maximum length of paths leading from node $\{\sigma(k),k\}$. Add this value to the PD of following sequence $\sigma(1),...,\sigma(k)$ to form the upper PD bound of any continuation.

4. The Discounted MEAN Bound for the OSP

4.1 Motivation

The efficiency of computing MEAN comes from decoupling the reward attributed to an action (i.e., the corresponding arc's weight) from past search history, by assuming that searching a cell does not change the target probability distribution. However, this deliberate forgetfulness comes at a cost when the searcher has a low chance of non-detection or can always visit cells with high target probability. Consider the extreme case where both a searcher with a 100% glimpse probability and all of a stationary target's probability mass begin the same cell. While the maximum PD is clearly 100%, the maximum ED would increase for each additional time period available for search.

Viewed in the context of the longest path problem in Figure 2, the bound's looseness is linked to each arc weight's overestimation of the probability of detecting the target with the corresponding search. In the limit, one can assign accurate weights that fully account for all the cells previously inspected by the searcher, but not without regaining the computational complexity of maximising PD itself. A closer examination of the MEAN bounding method however shows that information already available can be leveraged to cheaply reduce the overestimation of arc weights.

4.2 Method

This section presents an improved bounding technique, Discounted MEAN (DMEAN), which greatly sharpens the MEAN bound at little cost by retaining a limited memory of past actions. Although Martins (1993) calculated the MEAN bound with an ED network that assigned the reward for searching a cell to the weight of a node's outgoing arc, we have thus far employed an equivalent graph where this value is accrued on the arc heading into the corresponding node instead. Each arc's weight is therefore now clearly linked to the paired actions of visiting one particular cell after another, rather than just the current cell in isolation. This suggests that one may discount from the reward of searching the current cell an amount that is known to have already been claimed when the previous cell was searched.

More specifically, the DMEAN relaxation relates the weight of an arc from node $\{i,t\}$ to $\{j,t+1\}$ to the search of both the corresponding cells, while still ensuring the longest path through the network produces a valid upper bound. Based on the previous observation, we can safely reduce the weight to:

$$R_{DMEAN}(j,t+1) = (P(j,t+1) - P(i,t)\cdot g(i,t)\cdot \Gamma_{ij})\cdot g(j,t+1) \tag{4}$$

Algorithm for the DMEAN Bound

The new procedure is obtained by changing step 3 of the Algorithm for the MEAN Bound in Section 3 to:

3'. From each valid node $\{i,t\}$, extend arcs to all nodes $\{j,t+1\}$, $j \in S(i), t+1 \le T$. Mark the head nodes $\{j,t+1\}$ valid.
If $t = k$, assign a weight of $P(j,t+1)\cdot g(j,t+1)$ for each new arc. Else use $R_{DMEAN}(j,t+1) = (P(j,t+1) - P(i,t)\cdot g(i,t)\cdot \Gamma_{ij})\cdot g(j,t+1)$.

Whereas $P(j,t+1)$ is the remaining target probability assuming no searches after k, the discounted term $P(j,t+1) - P(i,t)\cdot g(i,t)\cdot \Gamma_{ij}$ in step 3' denotes the probability that the target has survived the additional search of cell i at time t before reaching cell j and time $t+1$. Not only is the longest path in the discounted network likely to be much shorter in comparison, its length can never exceed the maximum ED.

The DMEAN method thus produces a consistently tighter bound than MEAN regardless of problem parameters.

4.3 Proof of guaranteed upper bound and computational complexity

Proof: Let J indicate detection after time k when following a given plan ψ and I_t be a random variable indicating detection on the t^{th} look by the searcher at time $T_t = t$ [1], disregarding any detections made before that time. $E(J) = E(\sum_{t>k}^{\infty} I_t \prod_{k<u<t} (1-I_u)) = E(I_{k+1} + \sum_{t>k+1}^{\infty} I_t \prod_{k<u<t} (1-I_u)) \le E(\sum_{t>k}^{\infty} I_t)$ then reiterates MEAN's rationale that the PD of a plan is bounded by the corresponding ED. The DMEAN relaxation on the other hand corresponds to $E(I_{k+1} + \sum_{t>k+1}^{\infty} I_t(1-I_{t-1}))$, which therefore necessarily lies between the two values. Evaluating $E(I_t(1-I_{t-1})) = E(I_t) - E(I_t \cdot I_{t-1})$ instead of $E(I_t)$ does not however require much extra computation, as it can be seen that:

$$E(I_t \cdot I_{t-1}) = P(\psi(t-1),T_{t-1}) \cdot g(\psi(t-1),T_{t-1}) \cdot \Gamma_{\psi(t-1)\psi(t)} \cdot g(\psi(t),T_t).$$

Since DMEAN and MEAN differ only in the above calculation of each arc's weight, both methods share a complexity linear to the network size of $O(NcT)$ (Martins, 1993) in scenarios where each cell is connected to at most c others.

5. Comparisons and Results

5.1 Uniform OSP search grid

We use the search grids from Eagle and Yee (1990) to compare DMEAN's performance against known OSP bounds. An area is divided into a square of $N = n \times n$ cells (see Figure 1), where at each time step the searcher can only search its current cell or an adjacent (vertical or horizontal) neighbour. The target remains in its cell with a probability d in each step and with the remaining probability moves to one of the adjacent cells at random. Detection occurs with a glimpse probability of g and the sets $S(i), \forall i \in C$ are formed according to the depicted connectivity. For the purpose of the examples here, the searcher initially begins in cell 1 ($\psi(0)=1$) while the target starts at the centre cell S (e.g., $S=13$ for $N=25$) in an odd-numbered grid

[1] Notation used to also cover the case of non-uniform searcher travel times in Section 6.

$(p(S,1)=1, p(i,1)=0, \forall i \in \{1...N\} \setminus S)$. Grids with 11×11 and 15×15 cells will be used.

5.2 Comparison with previous OSP bounds

Tables 1 and 2 show the branch and bound solution times when using DMEAN alongside that of the best bound methods reviewed in Washburn (1998), for a number of uniform search grids with different parameters. In particular the DMEAN, MEAN, PROP and FABC algorithms are implemented and compared. FABC (Washburn, 1995) applies one iteration of the FAB algorithm to solve a version of the problem with the aforementioned distribution of effort and convexity relaxations. It is nevertheless by far the slowest of the tested bounds to individually compute. The PROP method (Washburn, 1998) can be seen as a simplified MEAN bound in which the most profitable node is chosen at each time step without regard to path constraints. While PROP, MEAN and DMEAN all belong to a class of easy to compute linear bounds, the removal of the longest path problem makes a PROP bound the fastest of the three to individually calculate. ERGO2, another linear bound also reviewed in Washburn (1998), is omitted here due to it being dominated by either PROP or FABC in the respective moving and near-stationary target test cases.

The number of bound fathoming attempts (i.e., the number of times that step 5 of the Algorithm for Branch and Bound is executed) in solving the problems with the various bounds are also listed for comparison. Results were obtained using a MATLAB implementation[2] on a 2.6-GHz Opteron 152 processor; values for problem grid size and time horizon were chosen to allow for test completion within reasonable time. Note that the FABC implementation used had not been fully optimised, although the results shown are nevertheless indicative of relative performance. The reader is referred to Washburn (1995, 1998) for performance comparisons between the existing OSP bounding techniques with 1-dimensional problems.

[2] Source code is available on request.

	$g(\cdot,\cdot)=0.3$			$g(\cdot,\cdot)=0.6$			$g(\cdot,\cdot)=0.9$		
	d=0.3	d=0.6	d=0.9	d=0.3	d=0.6	d=0.9	d=0.3	d=0.6	d=0.9
DMEAN	2.54 (10216)	2.91 (11074)	14.52 (51322)	3.03 (10594)	3.14 (10079)	68.21 (256794)	3.13 (9744)	7.24 (17204)	206.81 (941615)
MEAN	5.51 (34329)	8.90 (37054)	72.02 (341929)	12.31 (57924)	12.27 (45457)	324.19 (1980086)	17.77 (79512)	13.59 (49037)	323.98 (1981951)
PROP	4.65 (34329)	6.12 (37054)	54.90 (341929)	9.04 (57924)	8.16 (45457)	281.48 (1980086)	12.96 (79512)	9.01 (49037)	281.05 (1981951)
FABC	40.14 (96244)	18.69 (38354)	9.31 (19684)	216.41 (403667)	62.64 (94034)	13.47 (26849)	2353.84 (3645611)	835.99 (1011430)	65.28 (92687)

Table 1 Branch and bound solution times (in seconds) and number of bounding attempts (shown in brackets) for the 11×11 OSP search grid with *T* = 15. The optimal path is 2, 3, 4, 15, 26, 37, 48, 49, 60, 61, 72, 73, 62, 61, 50 with a detection probability of 0.26491 for the case when glimpse probability *g* = 0.6 and target stay probability *d* = 0.6.

	$g(\cdot,\cdot)=0.3$			$g(\cdot,\cdot)=0.6$			$g(\cdot,\cdot)=0.9$		
	d=0.3	d=0.6	d=0.9	d=0.3	d=0.6	d=0.9	d=0.3	d=0.6	d=0.9
DMEAN	14.56 (58349)	14.53 (52394)	157.30 (380974)	19.07 (49779)	23.76 (47489)	730.96 (2185136)	21.55 (45029)	30.58 (59547)	2902.27 (11299324)
MEAN	47.62 (263314)	41.76 (159392)	708.66 (2526216)	74.01 (313227)	71.57 (166645)	3114.58 (14733399)	131.51 (446140)	113.60 (269479)	11912.70 (74459729)
PROP	37.98 (263314)	27.43 (159392)	466.23 (2526216)	51.50 (313427)	37.20 (167350)	2380.86 (14738284)	82.02 (446340)	60.23 (269479)	10438.66 (74469799)
FABC	416.95 (939229)	135.57 (244199)	29.26 (45974)	2184.70 (3694435)	352.96 (451797)	131.20 (209634)	30297.75 (40372593)	7512.15 (7746490)	396.67 (429665)

Table 2 Branch and bound solution times (in seconds) and number of bounding attempts (shown in brackets) for the 11×11 OSP search grid with *T* = 17. The optimal path is 2, 3, 4, 15, 26, 37, 48, 49, 60, 61, 72, 73, 62, 51, 50, 61, 60 with a detection probability of 0.29785 for the case when *g* = 0.6 and *d* = 0.6.

The above confirms that using the DMEAN bound would outperform MEAN across the board. DMEAN bounds also tend to be much tighter than FABC when $d = 0.3$ and $d = 0.6$, despite being significantly easier to compute. As noted by Washburn (1995, 1998), FABC becomes quite effective when the target is less mobile (i.e. d has a large value). Nevertheless, even in the case of $d = 0.9$ DMEAN is still superior to the linear PROP and MEAN bounds, due to its ability to mitigate the redundant counting of searched target probability mass.

Figures 3, 4 and 5 show the computation times for example problems in a 11×11 grid with increasing time horizons, where $g(\cdot,\cdot) = 0.6$ and the target stay probability d is set at 0.3, 0.6 and 0.9, respectively. When the target is energetic (Figure 3), DMEAN is seen to further outperform the other methods as the time horizon expands. While its performance gap against PROP in the problems where $d = 0.6$ varied occasionally in Figure 4, the advantages appear to hold over time. Note from the number of bounding attempts for MEAN and PROP in Tables 1 and 2 that the problems are already structured to favour the PROP method – PROP bounds are approximately as tight as MEAN when the target roughly follows a random walk and become very loose when for example the target zig-zags from side to side (Washburn, 1998). It needs to be mentioned that in similar comparisons done with a 1-dimensional example and a slower target (Washburn, 1998), FABC computation times exhibited a much more favourable nonlinear growth and eventually fell below that of PROP. The results of Figure 5 confirm FABC's relative sharpness when the target is near stationary and time horizons are large.

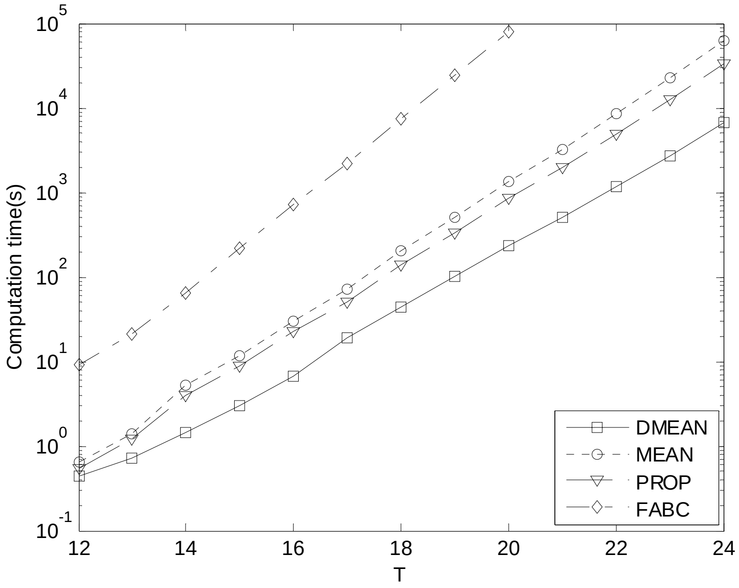

Figure 3 Computation times versus horizon, for 11×11 OSP grid where *g*(.,.)= 0.6 and *d*=0.3.

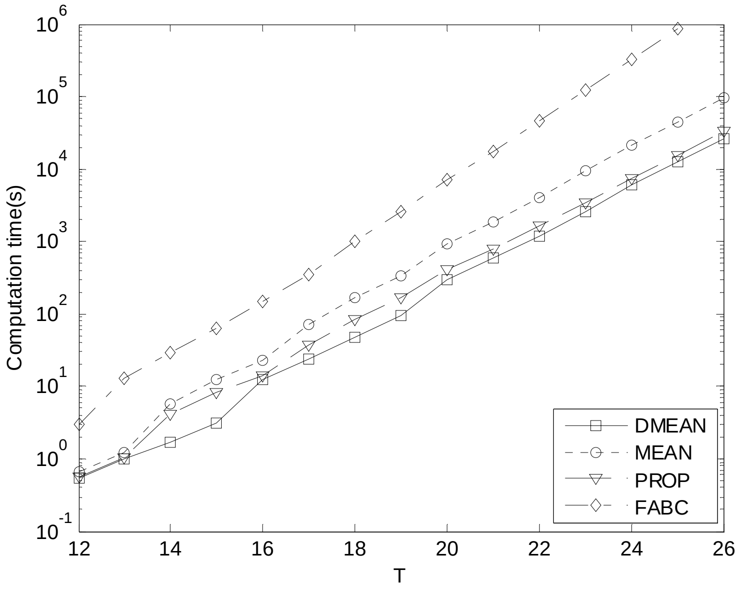

Figure 4 Computation times versus horizon, for 11×11 OSP grid where *g*(.,.)= 0.6 and *d*=0.6.

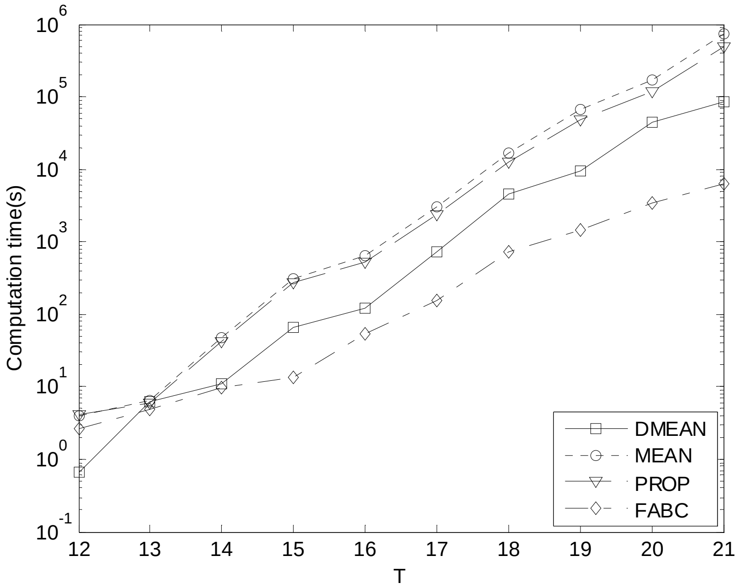

Figure 5 Computation times versus horizon, for 11×11 OSP grid where *g*(.,.)= 0.6 and *d*=0.9.

To better illustrate the scope within which the approach remains feasible on a midrange single-processor computer, Figure 6 plots the computation times with the linear bounds for a number of moving target examples on a larger 15×15 grid, using an unoptimised C++ implementation of the same branch and bound framework.

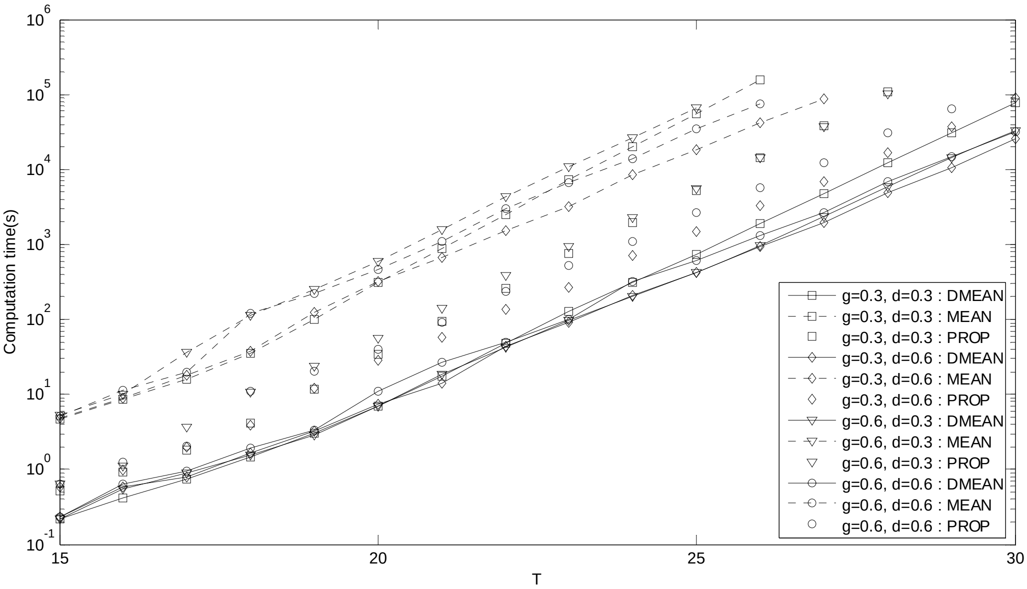

Figure 6 Computation times versus time horizon (C++ implementation) for a 15×15 OSP grid and combinations of *g*=0.3,0.6 and *d*=0.3,0.6. For context, solving with DMEAN when *T*=20, *g*=0.6 and *d*=0.6 requires 10.9 seconds here and 170 seconds in the previous MATLAB implementation.

6. Optimal Searcher Path Problem with Non-uniform Travel Times

This section introduces the Optimal Searcher Path Problem with non-uniform Travel times (OSPT), which generalises the OSP for the search of structured environments where the redeployment of effort from one cell to another may not be necessarily instantaneous. The previously outlined DMEAN bound for the OSP will also be extended to accommodate the OSPT.

6.1 The OSPT problem

The OSPT is identical to the OSP formulation in Section 2 except that a searcher in cell i at time t can only start searching the next cell $j \in S(i)$ at time $t+1+W_{ij}$. The integer value W_{ij} represents the length of time needed for a searcher to travel between the two referenced cells, during which no detection can occur. Travel in different directions may be assigned dissimilar values to capture specific travel or terrain constraints; without loss of generality, we assign $W_{ii}=0, \forall i \in C$ by default. Figure 7 illustrates an example representation of the search space for this new problem.

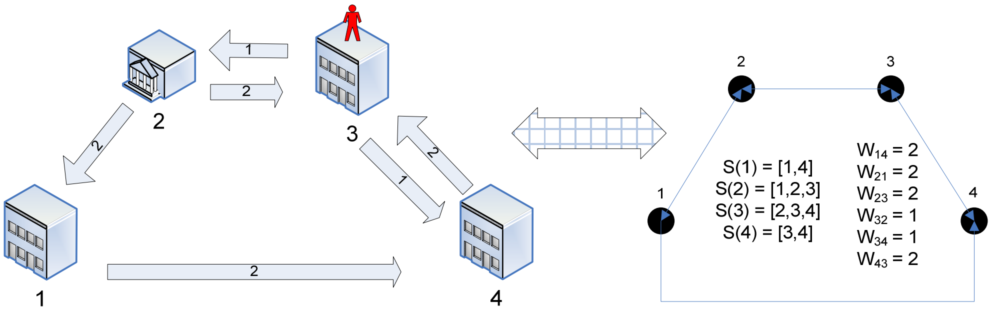

Figure 7 An OSPT search space depicted as a graph. The time required to move from building to building varies with distance.

Note that the number of one-time-unit searches in an OSPT search plan, $|\psi|$, can now be less than T, since some time may be needed for travel between the cells. For example, both $\psi_a=[\psi_a(0),\psi_a(1),\psi_a(2)]=[3,2,1]$ and $\psi_b=[\psi_b(0),\psi_b(1),\psi_b(2),\psi_b(3),\psi_b(4)]=[3,3,2,2,2]$ are valid plans for the problem in Figure 7 when $T=5$. Accordingly, the target distribution p can be more generally updated via:

$$p(\cdot,t)=\begin{cases} p(\cdot,1)\cdot\Gamma^{t-1}, & \text{if } t\le T_1 \\ p(\cdot,T_n)\cdot M_{\psi(n)T_n}\cdot\Gamma^{t-T_n}, & \text{if } T_n<t\le T_{n+1}, 1\le n<|\psi| \end{cases} \tag{5}$$

Where T_n denotes the time period when the n^{th} search occurs.

The objective for the Optimal Searcher Path problem with non-uniform Travel times (OSPT) is then:

$$\max_{\psi} PD(\psi)=\sum_{n=1}^{|\psi|} p(\psi(n),T_n)\cdot g(\psi(n),T_n)$$

Subject to:

$$\psi(n+1) \in S(\psi(n)) \qquad (6)$$

$$T_{n+1} = T_n + W_{\psi(n)\psi(n+1)} + 1,\ n = 0,\ldots,|\psi| - 1 \qquad (7)$$

$$T_{|\psi|} \leq T \qquad (8)$$

We assume $T_0 = 0$ for the purposes of constraint (7) and now employ equation (5) to obtain the undetected target probability $p(\psi(n), T_n)$. When $W_{ij} = 0, \forall i \in C, j \in S(i)$, the problem naturally specialises to an OSP.

As the target is still treated as being in a cell at each time step, longer travel times between regions are represented by setting lower (but non-zero) transition probabilities in its motion model. The problem also assumes that there may be different routes for the target to take in the environment when moving between the regions, such that there is little chance for it to be encountered as the searcher travels. An alternative model is formulated in Lau et al. (2006) for cases where the searcher may glimpse the target during movement.

6.2 Branch and bound for the OSPT

Since the OSP branch and bound algorithm described in Section 3 already maintains the time expended for each prospective path, it can accommodate the OSPT by replacing step 6 with:

6'. Else Branch: For each cell $\sigma_c \in S(\sigma_s)$, if $\tau_s + W_{\sigma_s \sigma_c} < T$, obtain p_c, the upper PD bound for any plan beginning with the path $\{\sigma_0, \ldots, \sigma_s, \sigma_c\}$. Add tuple $\{\sigma_c, \tau_s + W_{\sigma_s \sigma_c} + 1, p_c\}$ to $K(s+1)$

Figure 8 shows an example expansion of search plans using the adapted algorithm for the problem in Figure 7, when the searcher begins in cell 3.

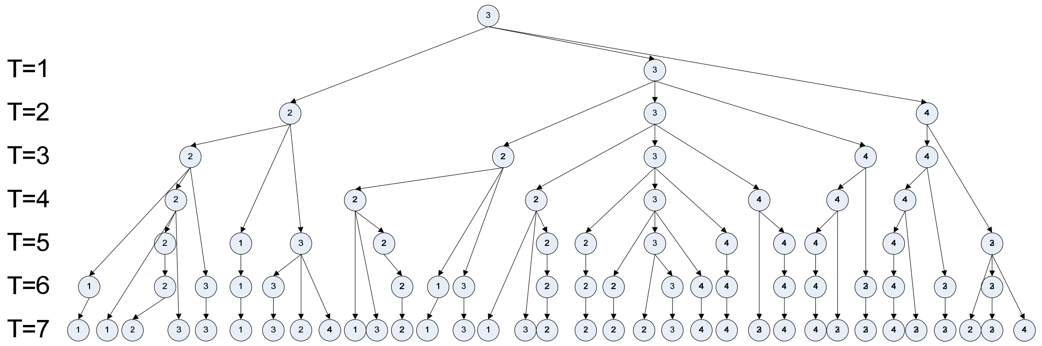

Figure 8 All feasible search plans for the example in Figure 7 when T=7.

6.3 Choice of bounds for the OSPT

As existing OSP bounding methods presume that travel times between cells are zero, they cannot directly help solve the OSPT without first transforming the non-uniform regions into a series of suitably uniform cells, by adding an artificial cell for each travel time unit between regions. This would clearly scale poorly if travel times are large. Hence we seek a bounding method that functions for the OSPT without redefining the search area or sacrificing the corresponding travel constraints that ensure bounds remain tight.

FAB-based methods appear incompatible for the problem without enlarging the map in the manner mentioned. One possible exception is the algorithm by Dambreville and Le Cadre (2002) that optimally allocates a search resource that self-renews after a certain amount of time steps. For the narrow case when all the travel times are identical, this algorithm might therefore be used in a similar way as the continuous FABC to give an upper bound of a discrete path's PD. Nevertheless, the inability to incorporate cell-specific, non-uniform travel times renders the method unsuitable for use with the OSPT.

Given that multiple time steps may elapse as the searcher travels from one cell to another, it is also more likely for the same first-order Markov target to have changed location between successive searches than if travel times were not considered at all. In light of their better performance with energetic targets, PROP and MEAN thus appear to be the best candidates of the OSP bounds for extension. In particular, it can be seen from Section 3.2 that MEAN has a structure amenable to preserving the new time constraints; the next section describes one such generalisation of the MEAN and DMEAN bounds for the OSPT. While PROP (Washburn, 1998) bounds can be calculated by solving an even simpler graph problem, it was not adapted as its form of path relaxation could not be meaningfully extended in the same direct way.

6.4 Generalised MEAN and DMEAN bounds for the OSPT

In the following OSPT example (Figure 9), assume that searching a cell i and then moving to cell $i-1$ requires one time period for travel, reaching $i+1$ expends two time periods, while remaining in the same cell causes no delay.

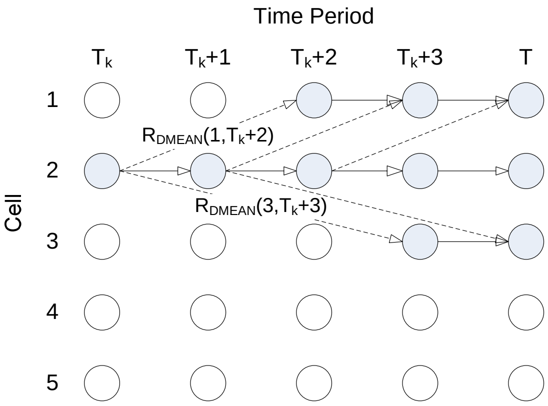

Figure 9 Generalised MEAN/DMEAN bound calculation after k searches ($\sigma(k) = 2$). Arcs can now span multiple time steps (dotted lines) if corresponding searcher travel incurs a time delay.

Using the same set of graph nodes as previously defined, the MEAN and DMEAN bounds can be generalised for the OSPT by forming DAGs in which arcs denote valid searcher transitions from one cell to another, taking also into account the amount of travel time required (Figure 9). The calculation of the generalised DMEAN bound for the OSPT is summarised below:

Algorithm for the generalised DMEAN Bound

1. For each time step from T_k to T, create a graph node per cell at that time.

   Mark node $\{\sigma(k), T_k\}$ as valid.
2. Use $P(\cdot,t) = \pi(T_k) \cdot \Gamma^{t-T_k}$ to calculate $P(\cdot,t)$ for $T_k < t \le T$.
3. From each valid node $\{i,t\}$, extend arcs to all nodes$\{j,\tau\}$,

   $j \in S(i), \tau = t + W_{ij} + 1 \le T$. Mark the head nodes $\{j,\tau\}$ valid.

   If $t = T_k$, assign a weight of $P(j,\tau) \cdot g(j,\tau)$for each new arc. Else use

   $R_{DMEAN}(j,\tau) = (P(j,\tau) - P(i,t) \cdot g(i,t) \cdot \Gamma^{\tau - t}) \cdot g(j,\tau).$
4. Repeat 3 until arcs have been extended from all valid nodes.
5. Apply a DAG longest path algorithm to find the maximum reward for paths leading from node $\{\sigma(k), T_k\}$. Add the reward to the PD of following sequence $\sigma(1),...,\sigma(k)$ to form the upper bound of any continuation.

The proof for the generalised DMEAN bound is omitted here as it is identical to the proof in Section 4.3, given that values for $T_t, t > k$ are as defined for the OSPT and $\Gamma^{T_t - T_{t-1}}_{\psi(T_{t-1})\psi(T_t)}$ is used in place of $\Gamma_{\psi(t-1)\psi(t)}$ for finding $E(I_t \cdot I_{t-1})$. In practice, discounting still does not impose much extra computation as $\Gamma^k, k = \{1 \ldots \max_{i,j \in S(i)} W_{ij} + 1\}$ can be cached in advance. The algorithm also functions for non-homogenous Markov chains when $\Gamma^{T_t - T_{t-1}}_{\psi(T_{t-1})\psi(T_t)}$ is replaced with the probability of the target moving from cell $\psi(t-1)$ at time T_{t-1} to cell $\psi(t)$ at time T_t.

The generalised MEAN bound (not shown) functions in the same way as DMEAN but instead uses $R_{MEAN}(j,\tau) = P(j,\tau) \cdot g(j,\tau)$ as a weight. As with any such problems, there may be many possible ways to obtain bounds for the OSPT. Of the two techniques available, however, the DMEAN bound is clearly the preferred option.

6.5 OSPT example

Figure 10 illustrates an example search environment that requires the use of the OSPT framework. Each node in the figure represents an individual region of interest, for example a specific building or room in the search area, while arcs describe their connectivity and corresponding searcher travel time. Such times are assumed to be symmetric for each transition in this case.

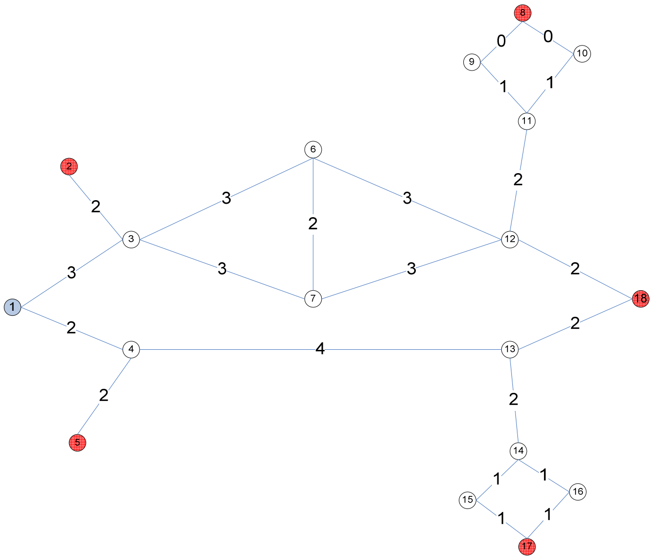

Figure 10 Example OSPT search environment with non-uniform travel times

Consider the case where the searcher begins in cell 1 and the target could begin in cells 2, 5, 8, 17 or 18 with equal probability. The target follows a random walk with $d = 0.8$ amongst the regions. The optimal plan when $g(\cdot,\cdot) = 0.6$ and $T = 40$ is to follow the sequence [3 3 3 3 3 3 3 3 3 3 3 7 12 12 12 18 13 13 14 14 14 15 17 17], which yields a PD of 0.50823 and is computed in 29.47 seconds in a C++ branch and bound implementation using the DMEAN bound. In comparison, the same implementation with the MEAN bound requires 535.55 seconds to compute. Given the inevitable exponential problem complexity growth with increasing search duration, it can be seen that the use of the DMEAN bound allows larger problems to be solved to optimality within a manageable time frame.

In addition to target movement and glimpse probabilities, the values of the travel times along the likely optimal paths also influence the computational effort required. Varying W_{13} or W_{14} for example can alter the relative merits of plans that begin with travel along the top or bottom of the map and accordingly affects how easily either branch could be fathomed.

For large scale problems, ε-optimal solutions that provide an approximately optimal plan with a payoff guaranteed to be within ε of the true optimal can be computed in a fraction of the time required to find the true optimal plan (Washburn, 1995). This can be achieved by changing the fathoming criteria in step 5 of the algorithm given in Section 3 to $p_s \leq p^* + \varepsilon.$ For this particular OSPT example, the solution time falls below 5 seconds when $\varepsilon = 0.1,$ with a corresponding PD reduction to 0.44392. Using an appropriate balance between guaranteed plan quality and computation time, large scale problems could be solved even in cases where the time available for planning is limited.

7. Discussion

7.1 Choice of bounds for the OSP

The effectiveness of a branch and bound solution process depends primarily on the tightness of a chosen bound against its ease of computation. The results show DMEAN to have superior performance to existing bounds for the OSP when the target is reasonably energetic. FABC may still be the preferred bound however for the OSP when the target is known to be very slow and the time horizon is large. Although the choice of bounds always depends on the particular case at hand, the DMEAN bound’s ease of computation and ability to better retain tightness across a range of problem parameter values renders it competitive for a broader number of situations.

7.2 Further discounting

The DMEAN method proposed in this paper may be described as a MEAN bound with a single step look-back, in that the projected gain for searching a cell at a particular time is reduced with respect to the immediate last cell visited. Since this can be calculated by finding a longest path in the same $N \times T$ node network as the original MEAN method, a tighter bound is obtained at little extra computation cost. Looking back one more step using the same approach would however require increasing the size of the ED network, because additional nodes would be needed to uniquely associate the reward of searching a cell to the two previous cells visited instead of one. Figure 11 shows an example of the nodes required for an OSP problem with two cells. In this case, the total number of nodes in the ED network is $2^2 \cdot T = 4T$ instead of $2T$ for the MEAN bound and DMEAN bound computation.

In general, discounting $d \geq 1$ steps for a map of n meshed cells would require an ED network with $n^d T$ nodes; it is clear that the exponentially increasing number of nodes in the ED network translates to significant computation burden to obtain the required bounds. A number of problems were solved via bounds that discount for more than one step and the results confirm the high computation cost required.

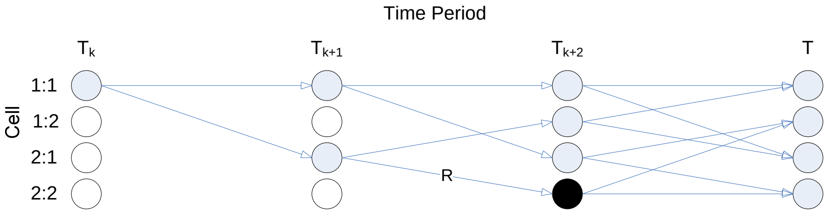

Figure 11 Two-step discounting for a two-cell problem. Each node's label signifies both the cell to be searched at that time and the previous one visited. For example, the reward of the arc entering node {2:2,T$_{k+2}$} (black) is linked to the prior searches of cell 2 at time T$_{k+1}$ and cell 1 at T$_k$.

Certainly, there are particular situations where advantage can be gained by discounting additional steps. From our preliminary simulation results, problems with very small number of cells and a high glimpse probability and/or a slow target (situations in which MEAN/DMEAN perform less well compared with FABC) appeared to benefit. The solution time of the OSPT scenario in Section 6.5 is also halved when an additional discounting step is taken. On the other hand, problems with OSP grids (Section 5.1) larger than 13×13 cells required more time to solve. The choice of the number of discounting steps therefore depends on the individual trade off between the extra complexity and the potential for tighter bounds.

7.3 Application issues

The problem proposed in this paper captures situations where the searcher can most appropriately sense at distinct locations in the environment before moving on to the next, such as buildings in a cluster or rooms with specific vantage points. Under similar assumptions, DasGupta et al. (2006) aggregated a continuous search space into discrete tiles. An alternative problem is formulated in (Lau et al., 2006) for the case where the searcher senses at every time step but must spend sufficient time in each cell as necessary to physically travel to another, with respect to the previous cell visited.

Targeted more towards the search of office environments, the generalised optimal searcher path problem (GOSP) is also amenable to the use of a modified DMEAN bound. An even more generalised framework was provided by Kierstead and DelBalzo (2003), which uses a genetic algorithm to construct continuous paths through arbitrary search environments. One added advantage of the method lies in its ability to support reactive targets when an appropriate target motion model is supplied.

Noting that simple Markov models can lead to seemingly nonsensical movements even for a wandering person when directly applied to structured environments, Moors and Schulz (2006) sampled target paths planned with random destinations before approximating the recorded target intentions in a second-order Markov model. Although time-consuming to learn, the improved models evolve visually more convincing target distributions for indoor areas while remaining principally compatible with existing OSP approaches. Given modifications to account for the multi-step passage of a target through the larger cells, the opportunity exists for a similar approach to be adapted for the OSPT.

Notwithstanding the improved performance provided by the DMEAN bound, whether the time needed to compute the optimal solution is acceptable naturally depends on the individual application. From an implementation perspective, drastically quicker solution times can be obtained with minimal change by accepting suboptimal solutions in the branch and bound process (Washburn, 1995) or implementing a rolling horizon technique and computing individually shorter plans (Dell et al., 1996). Additionally, the number of regions itself can be kept manageable by first hierarchically grouping related regions (Lau et al., 2005). For problems of even larger sizes, alternative heuristics such as genetic algorithms (Kierstead and DelBalzo, 2003; Dell et al., 1996) may then be required.

8. Conclusion and Future Work

This paper proposed an improved bounding method, Discounted MEAN (DMEAN), which is superior to MEAN for solving the optimal searcher path problem (OSP) and also leads to faster branch and bound computation times than other known bounding methods when searching for quick-moving targets. The optimal searcher path problem with non-uniform travel times (OSPT) was formulated in addition to better model the search of structured environments, with solutions found via an adapted OSP branch and bound framework using a generalised DMEAN relaxation.

Future work includes adapting more realistic target motion models for use with the OSPT formulation and the DMEAN method. Plans are also underway to extend the problem to involve multiple heterogeneous searchers, whereby less capable platforms may be able to help find but not “engage” a moving target. Due to the enlarged state space, heuristics and other suboptimal techniques for solving this problem will also be examined.

Acknowledgements

This work is supported by the ARC Centre of Excellence Programme, funded by the Australian Research Council (ARC) and the New South Wales State Government. The authors would like to thank the anonymous reviewers for their helpful comments and suggestions.

References

Brown, S. S., Optimal Search for a moving target in discrete time and space. Operations Research 28 (1980), pp. 1275-1289.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Introduction to algorithms, The MIT Press: Cambridge, Massachusetts; 1990.

Dambreville, F. and Le Cadre, J.-P., Detection of a Markovian target with optimization of the search efforts under generalized linear constraints, Naval Research Logistics 49/2 (2002), pp. 117-142.

DasGupta, B., Hespanha, J. P., Riehl, J. and Sontag, E., Honey-pot constrained searching with local sensory information, Journal of Nonlinear Analysis: Hybrid Systems and Applications 65/1 (2006), pp. 1773-1793.

Dell, R. F., Eagle, J. N., Martins, G. H. A. and Santos, A. G, Using multiple searchers in constrained-path, moving-target search problems, Naval Research Logistics 43 (1996), pp. 463-480.

Eagle, J. N., The optimal search for a moving target when the search path is constrained, Operations Research 32/5 (1984), pp. 1107-1115.

Eagle, J. N., and Yee, J. R., An optimal branch and bound procedure for the constrained path, moving target search problem, Operations Research 38/1 (1990), pp. 110-114.

Hohzaki, R., and Iida, K., Optimal strategy of route and look for the path constrained search problem with reward criterion, European Journal of Operational Research 100 (1997), pp. 236-249.

Kierstead, D. P. and DelBalzo, D. R., A genetic algorithm applied to planning search paths in complicated environments, Military Operations Research 8/2 (2003), pp. 45-59.

Lau, H., Huang, S., and Dissanayake, G., Optimal search for multiple targets in a built environment, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’05), 2005, Edmonton, Canada, pp. 3740-3745.

Lau, H., Huang, S., and Dissanayake, G., Probabilistic search for a moving target in an indoor environment, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’06), 2006, Beijing, China, pp. 3393-3398.

Lössner, U., and Wegener, I., Discrete sequential search with positive switch cost, Mathematics of Operations Research 7/3 (1982), pp. 426-440.

Martins, G., A new branch-and-bound procedure for computing optimal search paths, Master’s Thesis, Naval Postgraduate School, 1993.

Moors, M., and Schulz, D., Improved Markov models for indoor surveillance, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’06), Beijing, China, pp. 3393-3398.

Stewart, T. J., Search for a moving target when searcher motion is restricted, Computers and Operations Research 6 (1979), pp. 129-140.

Thomas, L. C., and Eagle, J. N., Criteria and approximate methods for path-constrained moving-target search problems, Naval Research Logistics 42 (1995), pp. 27-38.

Trummel, K. E., and Weisinger, J. R., The complexity of the optimal searcher path problem, Operations Research 34/2 (1986), pp. 324-327.

Washburn, A. R., Search for a moving target: The FAB algorithm, Operations Research 31/4 (1983), pp. 739-751.

Washburn, A. R., Branch and bound methods for search problems, Technical Report, NPS-OR-95-003, Naval Postgraduate School, Monterey, CA, April 1995.

Washburn, A. R., Branch and bound methods for a search problem, Naval Research Logistics 45 (1998), pp. 243-257.