Comparison of the PMHT Path Planning Algorithm with the Genetic Algorithm for Multiple Platforms

Brian Cheung†,* brian.cheung@dsto.defence.gov.au
Samuel Davey† samuel.davey@dsto.defence.gov.au
Douglas Gray* dgray@eleceng.adelaide.edu.au
†Defence Science and Technology Organisation, Australia
⋆School of Electrical and Electronic Engineering, The University of Adelaide, Australia

Abstract – *This paper considers the problem of automatically coordinating multiple platforms to explore an unknown environment. The goal is a planning algorithm that provides a path for each platform in such a way that the collection of platforms cooperatively sense the environment in a globally efficient manner. A collection of discrete locales of interest is assumed to be known and the platforms use these as waypoints. The key feature of the method is to treat the assignment of locales to platforms as a target tracking problem. This paper compares the use of the Probabilistic Multi Hypothesis Tracker (PMHT) as a method of performing multiple platform batch data association with the Genetic Algorithm to solve the modified Multi Travelling Salesman Problem.*

1 Introduction

Multiple platform path planning is a problem that arises in many applications including search and rescue, coordinated surveillance, multiple platform simultaneous localisation and mapping (SLAM) and resource dissemination (the travelling salesman). Independent of the application, the goal is to schedule multiple mobile resources with dynamic constraints to cover an area in an efficient manner.

There are many strategies to coordinate a single moving platform to intelligently explore an environment. Most of these strategies take different factors into account before choosing an action, such as the localisation error of the platform's sensors or the information gain potential through the measurements that these sensors may collect. Generally these approaches enumerate a collection of hypotheses based on feasible motion of the platform and use a cost function as the decision criterion. For example, one strategy is to move the platform to build the map information as quickly as possible with localisation being less important, such as [1]. In contrast, under Active Localisation [2], the platform self localisation is most important and the platform's movement is chosen based on what will be likely to reduce the localisation uncertainty. Other methods seek to optimise the view points of the platform to maximise the expected information gain in building the map and to minimise the uncertainty of the platform within the grid cells of the map [3]. These strategies may be greedy one step ahead or N step ahead and are generally for single platforms.

It has been widely acknowledged within the autonomous vehicle research community that the use of multiple cooperating vehicles for exploration tasks has many advantages over a single vehicle architecture. Multiple vehicles have the potential to explore and map an environment more quickly than a single vehicle, are more robust to failures, and provide a broader field of view in dynamic environments. To achieve cooperative exploration, the key problem is to choose appropriate actions for the platforms so that they simultaneously explore different regions of the environment optimally. Similar to the single platform strategies, there are extensions to the multiple platform case, such as [4] to build a map as quickly as possible and [5] using information gain to coordinate the multiple platforms.

A new approach for multiple platform path planning was introduced in [6]. Given known initial conditions for each platform (position, speed, heading) and a set of discrete locales of interest, [6] used the probabilistic multi-hypothesis tracker (PMHT) to design trajectories for the platforms to cooperatively visit the locales.

The novelty in the PMHT path planning approach is to treat the locales as measurements and the platforms as targets and then employ a multi-target tracking algorithm to perform data association, namely to associate the locales with platforms. This ensures that each of the locales is visited by at least one of the platforms, while constraining the motion of the platforms to a realistic dynamic model. The difficulty is that the locales have no intrinsic temporal relationship. Whereas in tracking the usual assumption is that the input information is noisy position estimates collected at known times, here there are no times associated with each locale; there is not even a preferred order in which to visit them. In order to overcome this problem, hypothesised timestamps are treated as hidden variables and the PMHT is used to associate the locales to platforms and times simultaneously.

The PMHT, developed by Streit and Luginbuhl [7], is a data association algorithm derived from the application of the Expectation Maximisation (EM) algorithm [8] to target tracking. The PMHT uses EM to model the assignment of measurements to targets as hidden variables and estimates target states by taking the expectation over the assignments. The advantage of the PMHT over alternative data association techniques is that it has linear complexity in the number of targets, the number of measurements per frame, and in the number of frames. In contrast, other data association methods incur a complexity that grows with the number of permutations of measurements and targets within a frame and then exponentially with time.

In the context of the planning problem, PMHT allows for batch data association over hundreds of time steps with multiple platforms in a field of hundreds of locales. It is also built on an EM framework which is amenable to the incorporation of the missing temporal information.

The initial work of [6] derived the PMHT path planning algorithm and provided some qualitative performance examples. In this paper, the algorithm's performance is assessed quantitatively and compared with a sophisticated alternative method.

The alternative approach considered in this paper is to treat the path planning as a Travelling Salesman Problem (TSP) and find the TSP solution via a Genetic Algorithm. The TSP is a common optimisation problem whereby an agent must visit a set of known locations exactly once and in the order that minimises the distance travelled. The multiple platform path planning problem can be treated as a TSP with multiple agents that must cooperatively visit each location in a manner that minimises the total collective distance travelled. No efficient algorithm exists to find an optimal solution to the TSP. Therefore a randomised approximate optimisation method is used, namely a Genetic Algorithm (GA).

The remainder of this paper reviews the PMHT path planning approach and the GA-TSP alternative. The performances of the two methods are then compared via Monte Carlo simulations.

2 Problem Formulation

Assume that there are M platforms, N locales to be visited, and T time steps.

Let the state of platform m at time t be denoted x_t^m, the set of all states for platform m be denoted X^m, and the set of all states be denoted X. For the purpose of this paper, the state may contain the position, velocity and acceleration in the X and Y domain.

It is assumed that the prior distribution of the state of each platform is known and is given by $\psi_0^m(x_0^m)$ for platform m. The platform dynamics are also assumed to be known and can be described by the evolution probability density function (pdf) $\psi_t^m(x_t^m|x_{t-1}^m)$.

Let the location of the nth locale be denoted by z^n and assume that it is known exactly. Let the platform assigned to locale n be $k_n \in 1 \dots M$ and the time assigned to locale n be $\tau_n \in 1 \dots T$. Both of these are, of course, unknown a-prior. Treat both k_n and τ_n as random variables with priors π_{nm}^k and π_{nt}^τ respectively. These priors are also unknown. Let Z denote the set of all locales, K denote the set of all locale-to-platform assignments and τ denote the set of all locale-to-time assignments. Similarly, Π^k and Π^τ denote the collection of platform and time assignment priors respectively.

Define a fitting penalty function, $\zeta_t^m(z_n|x_t^m)$, where $m = k_n$ and $t = \tau_n$. This function quantifies the cost of the locale assignment and may be used to control how closely the platforms approach each discrete locale position. It is a known function of the position of the locale and the position of the platform.

3 PMHT for multiple platform path planning

The standard PMHT algorithm is derived in detail in [7] and [9]. PMHT is based on the application of expectation-maximisation (EM) [8] to multi-target data association. The major advantage of using PMHT is that the computational complexity increases linearly with the number of targets, measurements and time steps unlike other association algorithms which can grow exponentially. This allows the algorithm to be implemented without approximation and allows for efficient smoothing over time batches when the application requires.

The PMHT algorithm is a method for finding the best estimate of the target states, X, when the measurement source K is unknown. It does this by treating the assignments as missing data using EM. The state estimate is derived iteratively by maximising an auxiliary function

$$Q\left(X|\hat{X}(i)\right) = \sum_K P\left(K|\hat{X}(i), Z\right) \log P(X, K, Z), \quad (1)$$

where i is an iteration index. Upon convergence, the algorithm's output is the state estimate. This auxiliary function $Q(.)$ can be maximised using any appropriate estimator. It can be shown that the auxiliary function is equivalent to the log-likelihood of a known assignment problem with synthetic measurements determined by the expectation step [7]. Thus for linear Gaussian cases, the Kalman filter may be used to solve the equivalent problem. For nonlinear problems such as SLAM, a nonlinear filter must be used, such as the Extended Kalman Filter (EKF).

The PMHT algorithm for path planning was introduced and derived in [6], but will be repeated here for completeness.

The derivation of the PMHT algorithm for path planning follows similar development as the standard PMHT. For the path planning version of the PMHT, the platform states, X, are considered the target states to be estimated, the locales, Z, are treated as the measurements, the fitting cost function, $\zeta_t^m(z_n|x_t^m)$, is treated as a measurement pdf and the assignments of the locales to the platforms K is unknown. In addition, the assignments of the locales to the time they were visited τ is also unknown. The following derivation treats both of these assignments as missing data using EM.

In EM terminology, the complete data are (X, τ, K, Z), the incomplete data are (X, Z) and (τ, K) are the missing data. The auxiliary function is the expectation of the complete data log-likelihood over the missing data, which now takes the form:

$$\begin{aligned} &Q(X, \Pi^\tau, \Pi^k | X(i), \Pi^\tau(i), \Pi^k(i)) \\ &\quad = \sum_K \sum_\tau P(\tau, K | X(i), Z) \log P(X, \tau, K, Z), \end{aligned} \tag{2}$$

where the summation is over all permutations of the assignment variables τ and K.

For compactness, let

$$\prod_{n,t,m} (\cdot) \equiv \prod_{n=1}^{N} \prod_{t=1}^{T} \prod_{m=1}^{M} (\cdot), \tag{3}$$

i.e. a product over all of the measurements at each time and from each sensor, and similarly for $\sum_{n,t,m}$.

Due to the independence assumptions, the complete data likelihood with classification becomes:

$$P(X, \tau, K, Z) = P(X) P(\tau; \Pi^\tau) P(K; \Pi^k) P(Z | X, \tau, K), \tag{4}$$

where

$$P(X) = \prod_{m=1}^{M} \psi_0^m(x_0^m) \prod_{t=1}^{T} \psi_t^m(x_t^m | x_{t-1}^m), \tag{5}$$

$$P(\tau; \Pi^\tau) = \prod_{n,t} \pi_{nt}^\tau, \tag{6}$$

$$P(K; \Pi^k) = \prod_{n,m} \pi_{nm}^k, \tag{7}$$

$$P(Z | X, \tau, K) = \prod_{t=1}^{T} \prod_{n=1}^{N} \zeta_t^m(z_n | x_t^m)|_{t=\tau_n, m=k_n}. \tag{8}$$

The conditional probability of the missing data, $P(\tau, K | X, Z)$, can be determined using Bayes' Rule:

$$\begin{aligned} &P(\tau, K | X, Z) \\ &\quad = \frac{P(X, \tau, K, Z)}{\sum_{\tau, K} P(X, \tau, K, Z)} \\ &\quad = \frac{P(X) P(\tau) P(K) P(Z | X, \tau, K)}{P(X) \sum_{\tau, K} P(\tau) P(K) P(Z | X, \tau, K)} \\ &\quad = \frac{\prod_{n,t,m} \pi_{nt}^\tau \pi_{nm}^k \zeta_t^m(z_n | x_t^m)}{\sum_{\tau, K} \prod_{n,t,m} \pi_{nt}^\tau \pi_{nm}^k \zeta_t^m(z_n | x_t^m)} \\ &\quad = \prod_{n,t,m} \frac{\pi_{nt}^\tau \pi_{nm}^k \zeta_t^m(z_n | x_t^m)}{\sum_{r=1}^{T} \sum_{s=1}^{M} \pi_{nr}^\tau \pi_{ns}^k \zeta_r^s(z_n | x_r^s)} \equiv \prod_{n,t,m} w_{ntm} \end{aligned} \tag{9}$$

Thus the conditional probability of the assignments is given by the product of individual per measurement *weights*. Each weight, w_{ntm}, is the normalised likelihood of the nth measurement from platform m at time t. The numerator of the weight is simply the product of the assignment priors and the positional measurement likelihood.

Combining the two equations (4) and (9) leads to the auxiliary function to be maximised:

$$\begin{aligned} &Q(X, \Pi^\tau, \Pi^k | X(i), \Pi^\tau(i), \Pi^k(i)) \\ &\quad = \log P(X) + \sum_{n,t,m} w_{ntm} \log \pi_{nt}^\tau + \sum_{n,t,m} w_{ntm} \log \pi_{nm}^k \\ &\qquad + \sum_{n,t,m} w_{ntm} \log \zeta_t^m(z_n | x_t^m) \\ &\quad \equiv Q_X + Q_\Pi^\tau + Q_\Pi^k \end{aligned} \tag{10}$$

The term Q_Π^τ in (10) is given by

$$Q_\Pi^\tau \equiv \sum_{n,t,m} w_{ntm} \log \pi_{nt}^\tau,$$

and is similar to that of the standard multi-sensor PMHT [10]. It is maximised subject to the constraint that $\sum_t \pi_{nt}^\tau = 1$ using a Lagrangian, resulting in the updated prior estimate

$$\pi_{nt}^\tau(i+1) = \sum_{m=1}^{M} w_{ntm}, \tag{11}$$

i.e. the weights' relative frequency for time t.

Similarly, the Q_Π^k term results in a relative frequency estimate for the locale to platform assignment prior

$$\pi_{nm}^k(i+1) = \sum_{t=1}^{T} w_{ntm}. \tag{12}$$

The remaining term, Q_X, couples the platform states and the waypoint measurements and is given by

$$Q_X \equiv \log P(X) + \sum_{n,t,m} w_{ntm} \log \zeta_t^m(z_n | x_t^m). \tag{13}$$

For a Gaussian penalty function, $\zeta_t^m(z_n | x_t^m)$ with covariance R, it can be shown that this function is equivalent to the log likelihood of a tracking problem with known data association [11],

$$Q_X \equiv \log P(X) + \sum_{t,m} \log \tilde{\zeta}_t^m(\tilde{z}_t^m | x_t^m), \tag{14}$$

where the synthetic measurement, $\tilde{z}_t^m$, is given by

$$\tilde{z}_t^m = \frac{1}{\sum_{n=1}^{N} w_{ntm}} \sum_{n=1}^{N} w_{ntm} z_n, \tag{15}$$

and the synthetic measurement function, $\tilde{\zeta}(\cdot)$, is the same as the penalty function with a scaled covariance given by

$$\tilde{R}_t^m = \frac{1}{\sum_{n=1}^{N} w_{ntm}} R. \tag{16}$$

The path for platform m is now refined by smoothing the synthetic measurements and covariances.

The planning algorithm consists of iteratively calculating assignment weights, w_{ntm}, and estimating the platform paths and assignment priors until convergence.

Several performance examples were presented in [6]. We repeat one here to demonstrate how the PMHT planning algorithm operates. In the example, the locales were set out in a ten by ten grid with four platforms beginning in the middle of the scene. A constant acceleration process model was assumed for the platforms and a linear Gaussian penalty function was used.

Figure 1 shows the evolution of the trajectories for four platforms with the EM iterations. Initially, the assignment implicitly carves the area into quadrants and each platform moves in a straight line due to the symmetry. However, after a period, the paths diverge from the straight and narrow and eventually span each quadrant. The constraints on the platform dynamics inherent in the process model result in smoothly curving paths.

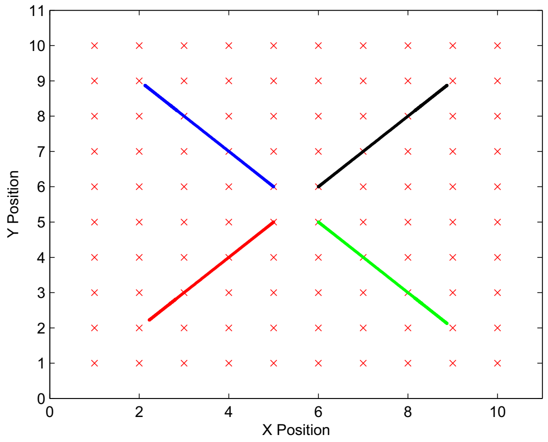
Not all of the locales were closely visited by the platforms. There is a tradeoff between path smoothness and proximity to assigned locales and this is governed by the relative values of the process noise and measurement penalty covariance.

4 Genetic Algorithm Solution to the Travelling Salesmen Problem

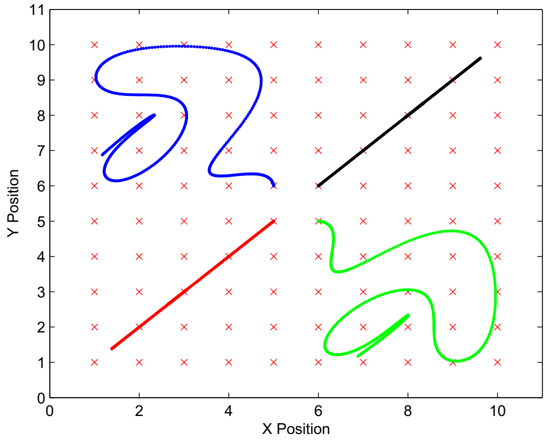
The purpose of this paper is to compare the performance of PMHT path planning with a competent alternative. Because the planning algorithm is a multiple platform batch method, it is appropriate to compare it with a multiple platform batch alternative. One such approach is to treat each of the locales as a city in a Travelling-Salesman-Problem (TSP). In the TSP, a salesman must make a complete tour of a given set of cities in the order that minimises the total distance travelled. Each city must be visited exactly once.

The path planning problem in this paper can be posed as a multi-TSP where multiple travelling salesmen cooperate to complete a tour of the set of cities. As before, each city is visited exactly once and the optimal solution is the set of paths that minimise the combined distance travelled by all salesmen (platforms). We will not require that the platforms return to their starting locations. It is important to highlight that the multi-TSP minimises the distance travelled without any dynamic constraints, this means that there is no smoothness imposed on the solution.

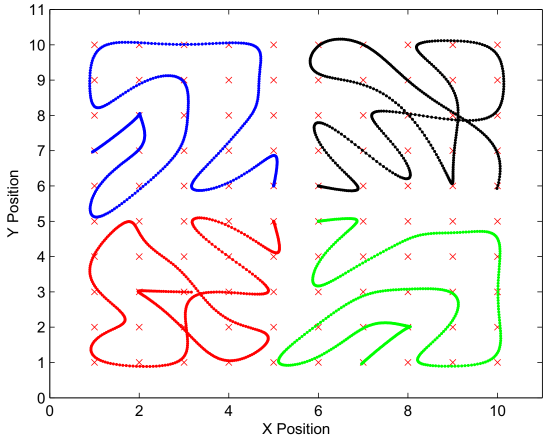
The TSP is well studied in optimisation and belongs to a class of problems where we expect an optimal solution to have exponential complexity in the number of cities. This paper uses a Genetic Algorithm (GA) as an approximate optimisation method to efficiently seek a very good solution, rather than finding the optimal solution. The GA is a randomised search algorithm which is based on the mechanics of natural selection and natural genetics [12]. The GA solution to the multi-TSP is summarised as follows:



(a) Iteration=5



(b) Iteration=20



(c) Converged at Iteration=100

Figure 1: Example assigned trajectories evolution for 4 platforms

1. Create an initial population. In this case, this amounts to a set of random permutations of potential paths for each of the platforms.

2. Calculate the fitness function for each of the population member. In this case, the fitness is the total distance travelled by all the platforms.

3. According to the fitness value, select the best population members and used them to form the next generation. Individual candidate solutions may be randomly modified (mutation) or pairs of candidate solutions may be mixed together (crossover).

4. Repeat steps 2 and 3 until the total population fitness converges.

The GA method is a randomised search, so it is not guaranteed to find the global solution in a finite number of iterations. However, the converged solution will be near to the optimal solution. The converged solution may also vary if the GA is applied to the same problem multiple times because of its random nature. In the multiple path planning case, we anticipate that there may be numerous path-sets that have very similar length and it is not critical that the planning algorithm find the best of these.

The GA-TSP algorithm used in this paper was adapted from [13]. This approach produces an ordered list of locales assigned to each of the platforms.

As mentioned above, the TSP formulation does not account for platform dynamics. It also demands that the platforms pass through each of the locales. In contrast, the PMHT-based algorithm plans paths that are sufficiently close to the locales based on a trade-off between path smoothness and the fitting penalty function.

In order to provide a balanced comparison, some modification to the GA-TSP is required. This was done by using the GA-TSP solution as initialisation data for the PMHT algorithm. The PMHT path planning algorithm was then run to produce smooth platform trajectories. By appropriate adjustment of the priors, the PMHT output can be guaranteed to visit the locales in the order supplied by the GA-TSP but with flexibility in the time at which a locale is visited and with the same path smoothness properties. This is termed the GA-TSP PMHT algorithm.

The GA-TSP provides a list of which locales are assigned to each platform that was used to define the locale-to-platform assignment prior. The value of π^k_{nm} is set to unity for the platform assigned to locale n by the GA-TSP and zero for all others.

The GA-TSP also provides an order in which to visit the locales. This order was used to define the locale-to-time assignments. The total batch length was divided into sub-batches and the order of locales was interpreted as an indication of which sub-batch the locale should be assigned to. The value of $\pi^\tau_{nt}(0)$ was uniform within the sub-batch identified by the GA-TSP and zero elsewhere.

5 Comparison

The use of the two algorithms for multiple platform trajectory planning is now illustrated through simulation. The area to be explored was a 10 unit square box. Two locale placement scenarios were considered: in the first, the locales were distributed over a uniform grid; in the second, the locales were distributed randomly with a uniform density. Fixed and randomised initial platform locations were also considered.

The platform states were assumed to follow a constant acceleration model independently in X and Y. Thus the state evolution process, with time-step $dt = 1s$, was defined by

$$\psi^m_t(x^m_t|x^m_{t-1}) = \mathcal{N}\left(x^m_t; F x^m_t, Q\right), \qquad (17)$$

where $\mathcal{N}(t; \mu, \Sigma)$ is a multivariate Gaussian,

$$F = \begin{bmatrix} F_3 & 0 \\ 0 & F_3 \end{bmatrix} \quad \text{with} \quad F_3 = \begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$

and

$$Q = 10^{-8}\begin{bmatrix} Q_3 & 0 \\ 0 & Q_3 \end{bmatrix} \quad \text{with} \quad Q_3 = \begin{bmatrix} \frac{1}{20} & \frac{1}{8} & \frac{1}{6} \\ \frac{1}{8} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{6} & \frac{1}{2} & 1 \end{bmatrix}.$$

For the PMHT algorithm, the fitting penalty function was linear and Gaussian,

$$\zeta^m_t(z_n|x^m_t) = \mathcal{N}\left(z_n; H x^m_t, R\right), \qquad (18)$$

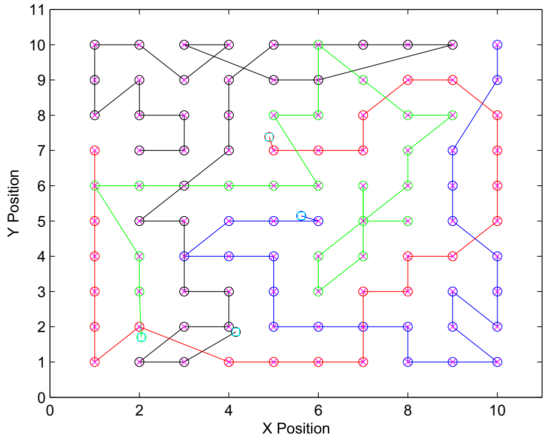
with $H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ and $R = 10^{-2} I$.

$T = 500$ time points were used. In each case, the initial state was assumed known, so the state estimates were initialised at the true position with zero speed and zero acceleration.

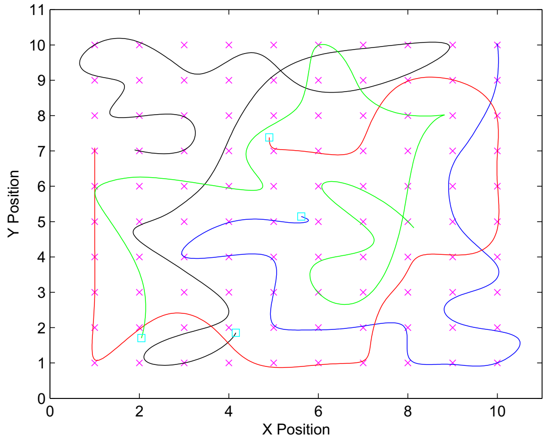
Three scenarios are presented here: a uniform grid of locales with randomised initial platform locations (A); randomised locale placement with fixed initial platform locations (B); and randomised locale placement and initial platform location (C).

Figure 2 shows the paths resulting for a uniform grid of locales and random initial platform locations. The crosses mark the locations of the locales, the cyan boxes show the platform initial locations, and the coloured lines show the paths. With this scenario, the PMHT divides the locales relatively evenly among the platforms with no crossovers whereas the GA-TSP has many platform crossovers.

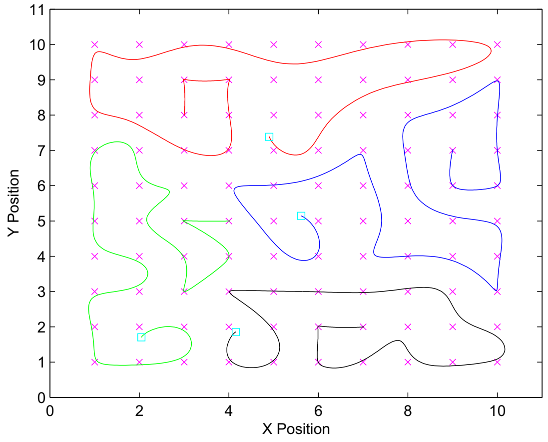
Figure 3 shows a three platform scenario where the platforms all began on the left side outside the map and the locales were randomly distributed. Again the PMHT plan divides the space roughly evenly and the paths do not cross, whereas the TSP paths cross in several places. Both algorithms tend to plan paths that pass through areas of high locale density and avoid areas where locales are sparse. Figure 4 shows a scenario where all four platforms had random initial positions in a field of randomly distributed locales.



(a) GA-TSP



(b) GA-TSP PMHT smoothed



(c) PMHT-pp

Figure 2: Results for 4 platforms and grid of waypoints

Table 1: Monte Carlo results

	A	B	C
Total Distance			
GA-TSP	117.8	118.8	87.8
GA-TSP PMHT smoothed	124.6	93.8	92.8
PMHT	94.6	77.8	78.8
CPU time (s)			
GA-TSP	18.3	17.4	18.0
GA-TSP PMHT smoothed	39.4	33.3	38.9
PMHT	23.1	16.7	22.8

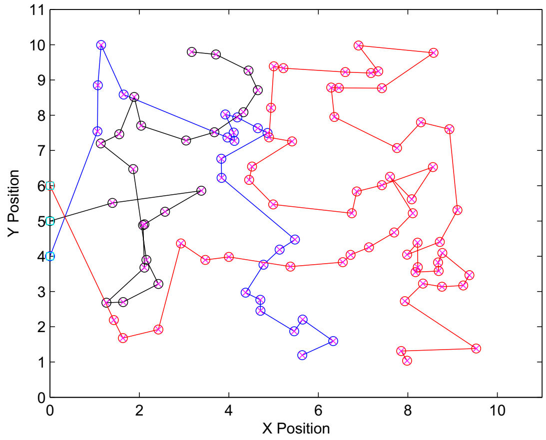
It is clear from the various examples that in the PMHT case, the paths of different platforms do not cross. This is due to a well known property of EM mixture fitting which biases the platform paths away from each other. This may be an undesirable feature of the planned paths. There are several potential methods to address this deficiency and will be the subject of future investigation.

Monte Carlo simulations of the above three scenarios were performed, and the total distance travelled by the platforms averaged over 100 trials. These average total distances and the CPU time incurred for each algorithm are given in Table 1. Clearly the PMHT paths are significantly shorter. The relationship between the GA-TSP paths and the GA-TSP PMHT smoothed paths is less direct. Two factors influence the path length in the smoothed case: the path may need to be extended in order to turn a sharp corner, but the platform is no longer required to pass exactly over the locale. These two factors have opposing effect and the net result depends on the scenario.

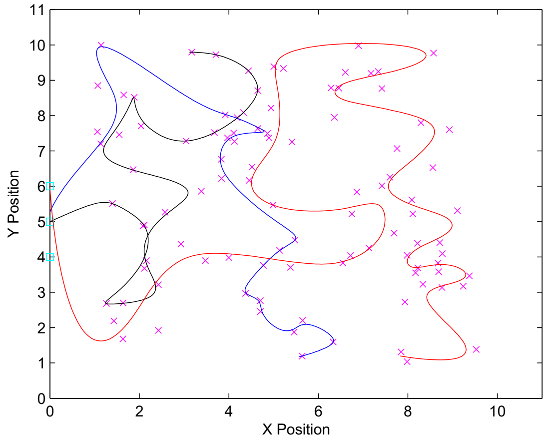
This is a clear advantage of the PMHT algorithm as compared with the GA-TSP. A possible reason why the distance travelled is lower is because the PMHT path is allowed to kill two birds with one stone: under the PMHT model, it is permissable to assign more than one locale to a platform at a particular instant. This allows the PMHT path to pass between two locales and cover them both at the same time, provided that they are close enough together. This is not permitted by the TSP approach. In the context of surveillance, it is quite feasible that the sensor may be able to observe more than one locale in the same frame, so the PMHT behaviour is appropriate provided that the locale fitting function adequately reflects sensor coverage.

The CPU time incurred section in table 1 shows that the PMHT takes a similar amount of time as the GA-TSP to process. Various parameters can be tuned in either algorithm to decrease CPU time such as the number of iterations allowed for convergence. As expected, the average smoothed TSP CPU time is approximately the sum of the TSP and the PMHT CPU times.

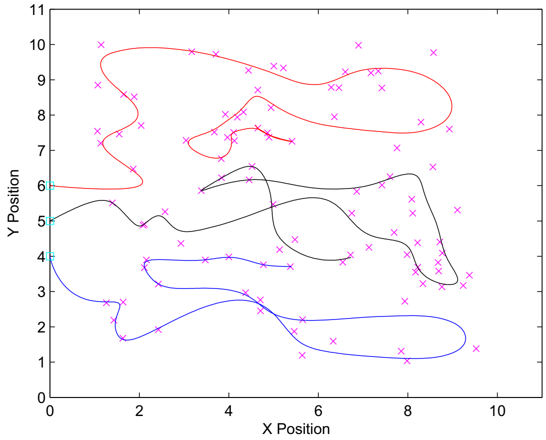
Figure 5 shows a histogram of the number of locales allocated to each platform, averaged over the Monte Carlo trials. For the fixed initial platform scenario (scenario B), this provides little intuition, but for the other two scenarios, the



(a) GA-TSP

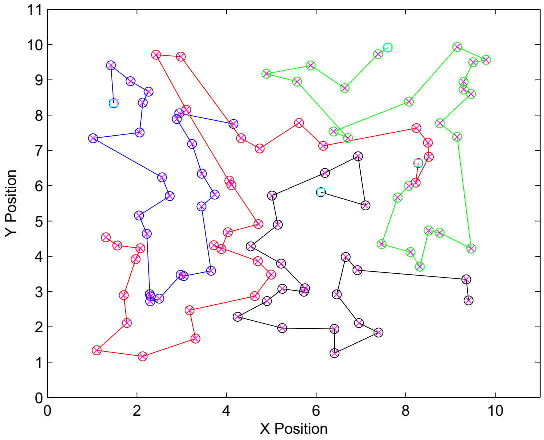


(b) GA-TSP PMHT smoothed

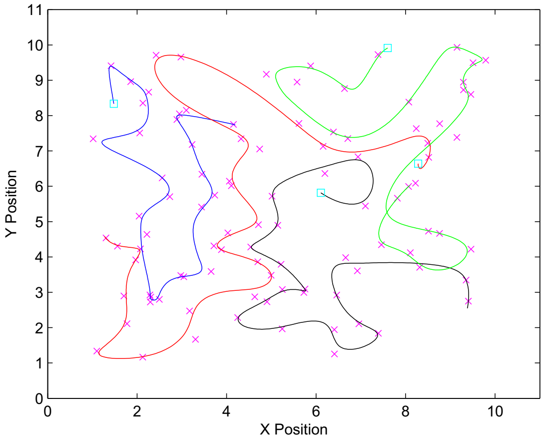


(c) PMHT-pp

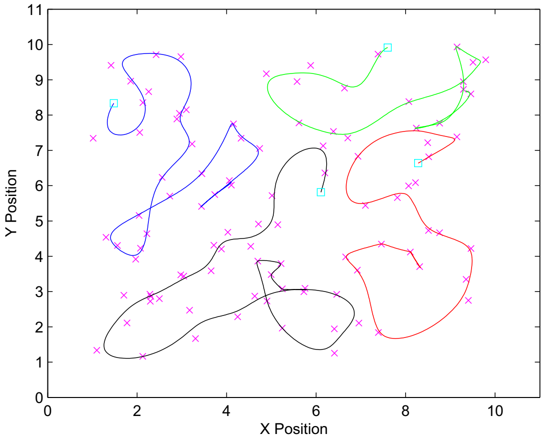
Figure 3: Results for 3 platforms and random locales



(a) GA-TSP



(b) GA-TSP PMHT smoothed



(c) PMHT-pp

Figure 4: Results for 4 platforms and random locales

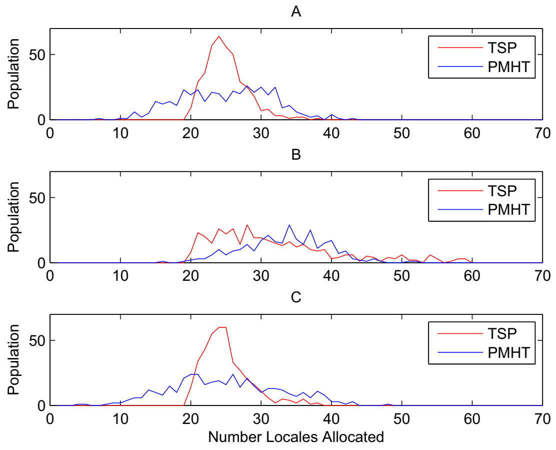


Figure 5: Locale distribution between platforms

GA-TSP variance is much smaller than the PMHT variance: it would appear that the GA prefers to find solutions that allocate a similar number of locales to each platform, whereas the PMHT tends to be more pragmatic.

6 Conclusion

A new path planning method has recently been proposed based on data association in the absence of time stamps using PMHT. This paper has compared the performance of the PMHT path planning method with an alternative based on a variant of the travelling salesman problem.

The PMHT path planning algorithm coordinates multiple platforms by treating the locations to visit as measurements and the platforms as targets. A novel aspect of the data association problem is that the measurements have no inherent temporal relationship. The PMHT association stage determines probabilities for each pairing of locale and platform at each possible time and then estimates the platform states by taking the expectation over these assignments.

The GA-TSP alternative approach was solved using a genetic algorithm. Since the GA-TSP does not incorporate dynamic constraints, the resulting sequences of locales were used as input to the PMHT planning algorithm to produce smoothed trajectories.

Simulation experiments were used to demonstrate the effectiveness of the path planning algorithms. The methods were demonstrated with various platform and locale configurations.

In terms of total distance travelled by all the platforms, the PMHT performed the best which was attributed to the ability of the algorithm to efficiently cover regions where multiple locales of interest were close together.

References

[1] B. Yamauchi, “A frontier based approach for autonomous exploration,” in *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997.

[2] A. Cassandra, L. Kaebling, and J. Kurien, “Acting under uncertainty: Discrete bayesian models for mobile robot navigation,” in *IEEE International Conference on Intelligent Robots and Systems*, 1996.

[3] R. Grabowski, P. Khosla, and H. Choset, “Autonomous exploration via regions of interest,” in *IEEE International Conference on Intelligent Robots and Systems*, 2003.

[4] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, “Collaborative Multi-Robot Exploration,” in *IEEE International Conference on Robotics and Automation*, 2000.

[5] M. Bryson and S. Sukkarieh, “Architectures for cooperative Airborne Simultaneous Localisation and Mapping,” in *Journal of Intelligent and Robotics Systems, Special Issue on Airborne SLAM*, 2009.

[6] B. Cheung, S. J. Davey, and D. A. Gray, “PMHT for Multiple Platform Path Exploration Planning,” in *Proceedings of the 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2009.

[7] R. Streit and T. Luginbuhl, “Probabilistic multi-hypothesis tracking,” NUWC, Newport, RI, Technical Report 10428, Feb. 1995.

[8] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistics Society*, pp. 1–38, 1977.

[9] S. J. Davey, *Extensions to the probabilistic multi-hypothesis tracker for improved data association*. PhD dissertation, School Elect. Electron. Eng., Univ. Adelaide, Australia, 2003.

[10] M. L. Krieg and D. A. Gray, “Multi-sensor probabilistic multi-hypothesis tracking using dissimilar sensors,” in *Proceedings of the Conference on Acquisition, Tracking and Pointing XI, SPIE Symposium on Aerosense*, vol. SPIE 2086.

[11] S. J. Davey, “Simultaneous localization and map building using the probabilistic multi-hypothesis tracker,” *IEEE Transactions on Robotics*, vol. 23, pp. 271–280, 2007.

[12] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley Publishing Company, Inc, 1989.

[13] J. Kirk, “Traveling salesman problem - genetic algorithm,” http://www.mathworks.com/matlabcentral/-fileexchange/13680-traveling-salesman-problem-genetic-algorithm, 2007.