Efficient Tracking and Pursuit of Moving Targets by Heuristic Solution of the Traveling Salesman Problem

Brendan Englot, Tuhin Sahai, and Isaac Cohen

Abstract— **We consider a planning and control problem in which a large number of moving targets must be intercepted by a single agent as quickly as possible. The agent maintains estimates of the instantaneous position and velocity of all targets, and decides which target to pursue next by predicting their future positions. Decision-making is driven by the repeated heuristic solution of the traveling salesman problem (TSP); for which the Lin-Kernighan heuristic (LKH) is compared to a greedy heuristic over different problem parameterizations. We show that the benefit of a non-greedy solution depends on the speed of the targets relative to the agent, and the precision of the measurement process used to track the targets. LKH is superior to a greedy heuristic when the targets are moving at low speed, and its relative performance improves as sensor noise worsens.**

I. INTRODUCTION

Our aim is to develop a computationally efficient planning and control strategy that allows a single agent to quickly intercept each of a large number of targets (25-50) moving in the Euclidean plane. We assume all targets travel at the same speed, and the velocities of the targets occasionally change direction due to collisions with the boundaries of the environment. The agent can measure the positions of targets with precision that varies in the maximum range of its sensor. We assess the suitability of an instantaneous solution of the traveling salesman problem (TSP) for choosing which target to pursue next. Although targets may change position quickly, future positions are predicted and considered in formulating node-to-node costs for the TSP. We formulate this problem as a representative example of a surveillance task to be performed by an unmanned aerial vehicle (UAV), in which there are many targets, but the environment is structured such that the targets remain within a bounded area and do not escape to an extent that they cannot be recovered.

Looking ahead to the greatest extent possible, we could choose the next target to pursue by solving the kinetic traveling salesman problem (KTSP) [3], also known as the moving-target traveling salesman problem [5]. Assuming that the velocities of all targets are known precisely and remain constant, the optimal KTSP tour visits each target once in minimum total time. Even if the targets change their velocities at a future time, following a portion of a KTSP tour may prove advantageous over greedily selecting the target that can be intercepted in the shortest immediate time. Ideally, the KTSP solution would be updated every time a target changes direction, discarding from the problem any targets that have already been intercepted.

B. Englot, T. Sahai, and I. Cohen are with United Technologies Research Center, 411 Silver Lane, East Hartford, CT 06108, USA {englotbj,sahait,coheni} at utrc.utc.com

Unfortunately, only in one-dimensional cases can the KTSP be solved to optimality in polynomial time [5], and polynomial-time approximation schemes only exist for simplified versions of the problem, in which there is a small minority of moving targets [5] or all targets move with identical velocity [3]. Related to these formulations is a problem in which all targets move with identical velocity toward an agent restricted to motion along a perpendicular transect, for which optimal solutions are easily computed or approximated in a variety of limiting cases [1], [15]. To quickly compute a feasible solution to generalized instances of the KTSP, we propose a method that relies on the repeated solution of a static TSP using a computationally efficient heuristic, such as the greedy nearest-neighbor method or the Lin-Kernighan heuristic (LKH) [10].

A standard, static TSP solved over target-to-target distances is based on the state of the system at a single instant in time. It is unlikely that a TSP tour will remain useful after the targets have moved far from their original positions. As a result, we propose the use of a static TSP solution to serve as a temporary decision-making tool, until target positions have changed substantially enough that a new TSP solution is required. We also propose an asymmetric traveling salesman problem (ATSP) formulation that uses instantaneous target velocities to better capture the cost of travel from target to target. In cases where sensor noise and limited field-of-view are imposed on the agent, we consider a metric that weighs both distance and uncertainty in selecting the next target to pursue. Information gain is prioritzed and travel cost is penalized in comparable units.

In the following sections, we present a review of relevant work in target tracking and sensor management, a formulation of the problem, and computational results. These results expose the parameter space in which repeated solution of a static TSP using LKH offers an improved solution over a greedy target selection method. We first give results in the case of perfect sensing, and then consider imperfect sensing with limited viewing range.

II. TARGET TRACKING AND SENSOR MANAGEMENT

Prior work in target tracking and sensor management has emphasized many of the same costs and constraints present in our formulation of the problem, but none that we are aware of has considered all within a single piece of work. The sensor management problem [6] often considers the tracking of many targets with a single sensor, but rarely imposes a travel cost from one target to another to collect the measurements. In such scenarios, the sensor is typically commanded to collect the measurement that maximizes immediate information gain [7], [12], [14], as this greedy strategy is found to outperform naive approaches that use deterministic back-and-forth sweeping [8]. Sensor management without a travel cost has also been formulated as a multi-armed bandit problem [6], [9], [17], requiring planning over longer time horizons to maximize the payoff of sensing actions with probabilistic outcomes.

Many studies of target tracking and sensor management have emphasized the data association problem, which we do not consider in this study. In many real-world sensing applications, establishing the identity of a measured target is a challenge when multiple targets are measured and tracked simultaneously [11], [13]. Although travel cost has been considered in the formulation of optimal planning and control for multiple agents in pursuit of a single moving target [16], the study that bears the greatest similarity to ours considers the problem of target tracking and sensor management by a single sensor with upwards of ten targets on the ground measured from the air [8]. Although the cost of travel by the sensor is not considered, a non-myopic planning strategy is explored which maximizes information gain over mutliple sensing actions, rather than acting greedily. The non-myopic strategy proves benefical in reducing tracking error, and offers a compelling reason for further exploration of non-greedy strategies in target tracking and sensor management.

III. Problem Formulation

We first introduce the basic planning and control strategy used to select the next target to pursue, using LKH and a greedy heuristic as alternative selection strategies. We then discuss the formulation of target-to-target travel costs when perfect information about the state of the targets is available. We subsequently discuss the special considerations needed when the agent's field of view is restricted and sensing is corrupted by noise.

A. General Problem Formulation

Algorithm 1 $CollectTargets(P, p_{agent})$

$P_{Active} \leftarrow P$
$P_{Inactive} \leftarrow \emptyset$
while $P_{Active} \neq \emptyset$ **do**
  $p_i \leftarrow NextTarget(P_{Active}, p_{agent})$
  while $Dist(p_{agent}, p_i) > r_{query}$ **do**
    $MoveAgent(p_{agent}, p_i)$
    $P \leftarrow CollectMeasurement(P, p_{agent}, p_i)$
  end while
  $P_{Active} \leftarrow P_{Active} \setminus p_i$
  $P_{Inactive} \leftarrow P_{Inactive} \cup p_i$
end while

The target-tracking problem of interest is solved using Algorithm 1. All point targets p_i belong to the set P, and the agent, referred to as p_{agent}, is not a member of P. We assume that a state estimate is stored in each $p_i \in P$, and the agent's state is known exactly and stored in p_{agent}. As the targets and agent move and the state of the system evolves, it is assumed that p_{agent} and $p_i \in P$ reflect up-to-

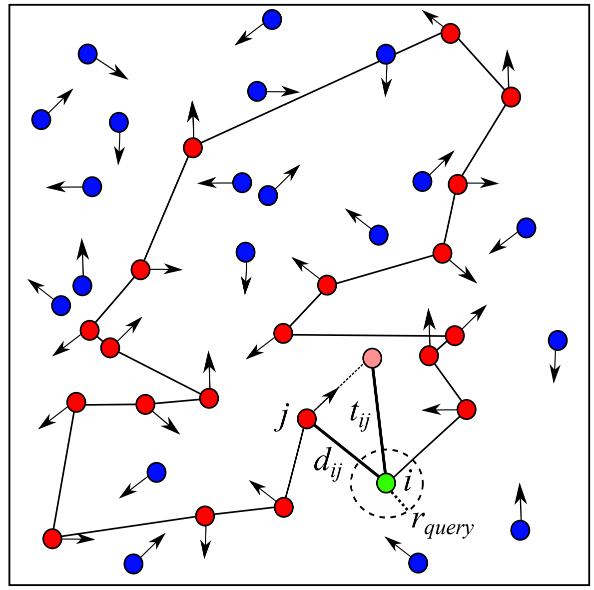

Fig. 1. An illustration of the problem of interest. Targets that have already been intercepted are pictured in blue; targets that remain to be intercepted are pictured in red. The agent pursuing the targets is pictured in green. Two possible formulations of node-to-node cost for the static TSP are illustrated, along with a TSP tour through the targets in pursuit. The pursuit of a target is satisfied once the target falls within the agent's query radius, r_{query}.

Algorithm 2 $p_i \leftarrow NextTarget_{Greedy}(P_{Active}, p_{agent})$

$bestCost \leftarrow \infty$
$bestTarget \leftarrow \emptyset$
for all $p_i \in P_{Active}$ **do**
  $thisCost \leftarrow EvaluateCost(p_{agent}, p_i)$
  if $thisCost < bestCost$ **then**
    $bestCost \leftarrow thisCost$
    $bestTarget \leftarrow p_i$
  end if
end for
return $bestTarget$

Algorithm 3 $p_i \leftarrow NextTarget_{LKH}(P_{Active}, p_{agent})$

$bestTarget \leftarrow \emptyset$
$thisTour \leftarrow LKH(P_{Active}, p_{agent})$
$p_L \leftarrow LeftNode(thisTour, p_{agent})$
$p_R \leftarrow RightNode(thisTour, p_{agent})$
$cost_L \leftarrow EvaluateCost(p_{agent}, p_L)$
$cost_R \leftarrow EvaluateCost(p_{agent}, p_R)$
if $cost_L < cost_R$ **then**
  $bestTarget \leftarrow p_L$
else
  $bestTarget \leftarrow p_R$
end if
return $bestTarget$

date states and estimates. When perfect sensing is assumed, $CollectMeasurement(P, p_{agent}, p_i)$ is not called and the agent is supplied with perfect target state information in P. The set $P_{Active} \subseteq P$ contains all point targets that have not yet been intercepted by the agent. As targets are intercepted, they are removed from P_{Active} and added to $P_{Inactive}$. Once a target is intercepted by the agent, the next target is selected for pursuit using the generic function $NextTarget(P_{active}, p_{agent})$.

In implementation, this function is replaced by Algorithm 2 or Algorithm 3 depending on the strategy of choice for selecting the next target. Algorithm 2 employs a greedy strategy, choosing the target p_i that is the least costly for p_{agent} to pursue. Algorithm 3 solves a static TSP using LKH. Given an adjacency matrix of node-to-node costs, an instance of the TSP is solved using the function $LKH(P_{Active}, p_{agent})$. Although p_{agent} is not a member of P_{active} it is always assumed that the agent is one of the "cities" included in the tour, and its cost to all point targets is computed.

Unlike the typical use of a TSP tour, the agent only visits one node of the tour before recomputing the solution. After the first node is visited, the state of the system may change to an extent that the original tour no longer represents an efficient route. Given a starting position in the tour, two choices are possible in deciding which node to visit. This decision is made greedily and is outlined in Algorithm 3.

B. Target Tracking with Perfect State Information

A single stage of the proposed TSP-based pursuit strategy, assuming perfect state information, is depicted in Figure 1. The agent is labeled i and an adjacent node in the working TSP tour is labeled j. The two methods used to compute node-to-node TSP costs under perfect state information are illustrated. The first cost, d_{ij}, is based on the current Euclidean distance between specific p_i and p_j. The alternate cost, t_{ij}, is based on the current velocity of p_j and represents the time required by the agent, starting at p_i, to reach p_j when traveling at maximum speed, assuming that the speed of p_j remains constant throughout the pursuit. The velocity-based node-to-node cost t_{ij} can be found by solving (1). There are two roots in the solution of this equation for t_{ij}, one of which is always positive in cases where the agent moves faster than the targets. We assume the agent moves faster than the targets in all problem parameterizations considered in this study.

$$|v_{agent}| = \sqrt{\left(v_{x_j} + \frac{(x_j - x_i)}{t_{ij}}\right)^2 + \left(v_{y_j} + \frac{(y_j - y_i)}{t_{ij}}\right)^2} \quad (1)$$

The term $|v_{agent}|$ represents the maximum speed of the agent, and the other terms in (1) represent position and velocity estimates associated with targets p_i and p_j. When d_{ij} is used as a cost metric, all TSP instances solved are symmetric. If t_{ij} is used instead, the adjacency matrix will be asymmetric. In either case, Helsgaun's implementation of the Lin-Kernighan heuristic is used to produce a high-quality feasible solution quickly [4]. In addition to populating the adjacency matrices used to solve the TSP, specific node-to-node costs d_{ij} and t_{ij} are also computed by the function $EvaluateCost(p_i, p_j)$.

C. Target Tracking Under Noise-Corrupted Sensing

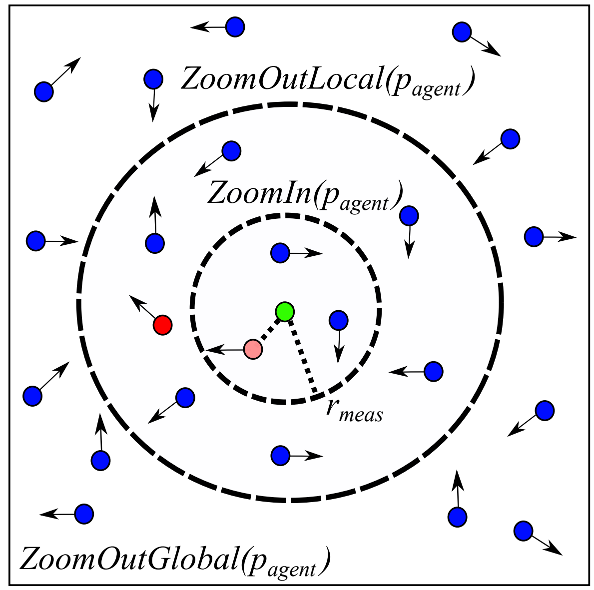

Fig. 2. An illustration of the measurement process described in Algorithm 4. The agent, shown in green, is pursuing the red target. The agent's estimate of the target state is shown in pink. During a typical measurement update, the $ZoomIn(p_{agent})$ function is used and a short-range, low-noise measurement is collected. If the estimate is within a distance r_{meas} of the agent and the target is not measured, $ZoomOutLocal(p_{agent})$ is applied and a medium-range, noisy measurement is collected. If the target is not measured, then $ZoomOutGlobal(p_{agent})$ is applied and a long-range, highly noisy measurement is collected.

Algorithm 4 $P \leftarrow CollectMeasurement(P, p_{agent}, p_i)$

```
P_measured ← ZoomIn(p_agent)
if Dist(p_agent, p_i) < r_meas then
    if p_i ∉ P_measured then
        P_measured ← ZoomOutLocal(p_agent)
        if p_i ∉ P_measured then
            P_measured ← ZoomOutGlobal(p_agent)
        end if
    end if
end if
P ← UpdateEstimates(P_measured)
return P
```

The measurement and state estimation process employed in the case of limited-range, noise-corrupted sensing is described in Algorithm 4. This algorithm is implemented every time the $CollectMeasurement(P, p_{agent}, p_i)$ function is called in Algorithm 1. Every call to this function collects a new measurement and performs an update to the Kalman filter used for estimating target position and velocity [2]. We assume that the agent must be located within a distance r_{query} to perform the querying operation needed to "intercept" a target. The agent can collect a high-precision measurment using $ZoomIn(p_{agent})$, a noisy measurment using $ZoomOutLocal(p_{agent})$, and a highly noisy measurment using $ZoomOutGlobal(p_{agent})$. High precision measurements can only observe targets within a distance r_{meas} of the agent, as illustrated in Figure 2. The agent typically collects measurments in this zoomed-in sensing mode, since the resulting precision is required to intercept a target at $r_{query} < r_{meas}$. If the estimate of the target being pursued is within r_{meas} of the agent, and the target is not measured, the agent then uses lower-precision, zoomed-out measurements as described in Algorithm 4 and illustrated in Figure 2.

$$x_i^{t+1} = v_{x_i}\,\Delta t + x_i^t \tag{2}$$

$$y_i^{t+1} = v_{y_i}\,\Delta t + y_i^t \tag{3}$$

The simple equations used to propagate a target's state are given in (2) and (3). The velocity of each target is assumed constant, and no process noise is explicity used to propagate the target states. Velocities only change when a target collides with the boundaries of its environment, in which case an elastic collision is assumed. Process noise is assumed in implementation of the Kalman filter, and the process noise covariance matrix Q is tuned appropriately to allow velocity estimates to adjust when collisions occur. A Kalman filter measurement update is performed using a measurement noise covariance matrix R that corresponds to the number of targets measured, and the noise of the sensing mode that is used. The agent can only measure the position of a target; velocity must be deduced from the filter.

The node-to-node cost metric used in the noise-corrupted sensing case is c_{ij}, defined in (4). This metric represents the cost of travel from target to target, weighted by expected information gain σ_j^{-1}.

$$c_{ij} = \frac{|v_{agent}|\, t_{ij}}{\sigma_j} \tag{4}$$

The term σ_j represents the standard deviation associated with target p_j's position error variance in either dimension of the Euclidean plane. We assume that a target's vertical and horizontal position are measured with the same precision. Curbing the uncertainty associated with each target is desirable, and σ_j represents the expense of allowing a target's uncertainty to grow, in units of distance equivalent to the those expended by the agent in traveling from target to target. The use of c_{ij} in $LKH(P_{Active}, p_{agent})$ and $EvaluateCost(p_i, p_j)$ results in an asymmetric formulation of the TSP. Consequently, in all cases of noisy sensing, an asymmetric TSP is solved in each call to Algorithm 3, using Helsgaun's implementation of LKH [4].

IV. Computational Results

A series of Monte Carlo trials was executed to evaluate the performance of greedy target selection versus LKH-based target selection over a variety of problem parameterizations. We first examined the case of perfect sensing with no sensor range limitations, in which the agent has perfect target state information at all times. In this case, performance of the symmetric distance metric d_{ij} relative to that of asymmetric t_{ij} is of interest, as well as performance across different target speeds and target quantities. In each Monte Carlo trial the targets were seeded with uniform random positions, randomly oriented velocities, and a random initial position of the agent. The targets were free to move about a square portion of the Euclidean plane 1 km in dimension, with speeds that varied from 0 to 2 m/s. The agent pursues the targets at a speed of 10 m/s; we assume that all travel occurs at this maximum velocity, with instantaneous changes in direction permitted. A second environment was considered in which the targets were evenly divided and sequestered within 10 smaller square clusters, each 0.2 km in dimension. This alternate environment was added to assess the impact of additional structure and constraints, imposed in real-world environments by fences, buildings, and roadways, on the relative performance of target selection strategies. The agent can pass through the boundaries of the clusters, but the targets stay within their boundaries and collide elastically with the walls of their respective clusters. The locations of the clusters are seeded randomly in each trial such that there is no overlap between the boundaries of any two clusters.

For each problem parameterization, 500 separate computational trials were conducted. A trial terminated when both the greedy agent and non-greedy agent intercepted all required targets. The total distance traveled by the agent to intercept all targets was used to evaluate relative performance between the greedy and LKH-based stragies. Figure 3 displays the results of all trials in which perfect sensing was assumed. A LKH-based strategy achieves the highest margin of superior performance at the lowest speeds, where a minimum-cost tour based on instantaneous state information remains an accurate predictor for a longer period of time. A clustered scenario yields improved LKH performance at higher target speeds, serving to limit the range that targets can wander while an instantaneous TSP tour is partially implemented. The asymmetric velocity-based metric t_{ij} outperformed the distance-based metric d_{ij} in all cases with moving targets. Due to the success of t_{ij}, the distance-based metric d_{ij} is not used in the noise-corrupted sensing case to follow. The relative performance of the LKH strategy worsens slightly as the number of targets is increased from 25 to 50. This reflects the "greediness" that is also inherent in the LKH strategy, which uses only the first step of a TSP tour before recomputing. In no case of perfect sensing does a LKH-based strategy outperform a purely greedy strategy by more than a 7% margin.

We next examined the case of noise-corrupted sensing with the range limitations illustrated in Figure 2. A "noisy" sensing scenario was considered in which the $ZoomIn(p_{agent})$ sensing operation has a 10 m^2 noise variance, and a "clean" sensing scenario was considered in which the same sensing operation has a 0.1 m^2 noise variance. All trials used the information gain-weighted metric c_{ij} for computing target-to-target costs. The benefit of targets sequestered in clusters was once again considered, over the same selection of

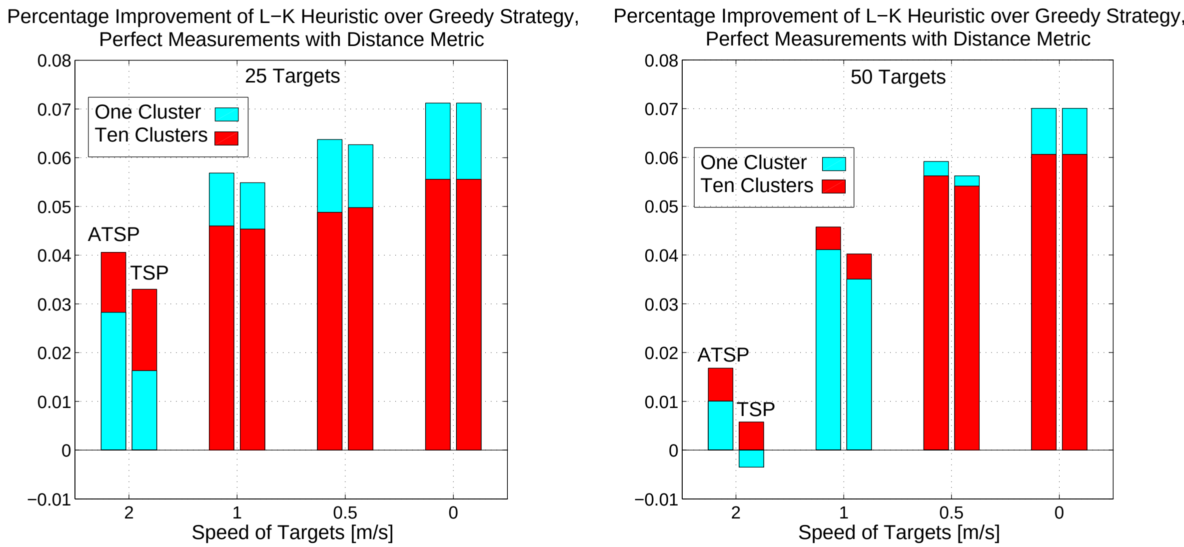

Fig. 3. Mean performance of LKH relative to a greedy algorithm, measured in terms of the total distance traveled by the agent to intercept all targets, is plotted for cases with perfect measurments and no restrictions on sensor range. A 25 target case is shown at left, and a 50 target case is shown at right. Performance of the velocity-based ATSP and distance-based TSP node-to-node cost metrics is compared side-by-side for each target speed. The speed of the agent throughout all trials is 10 m/s.

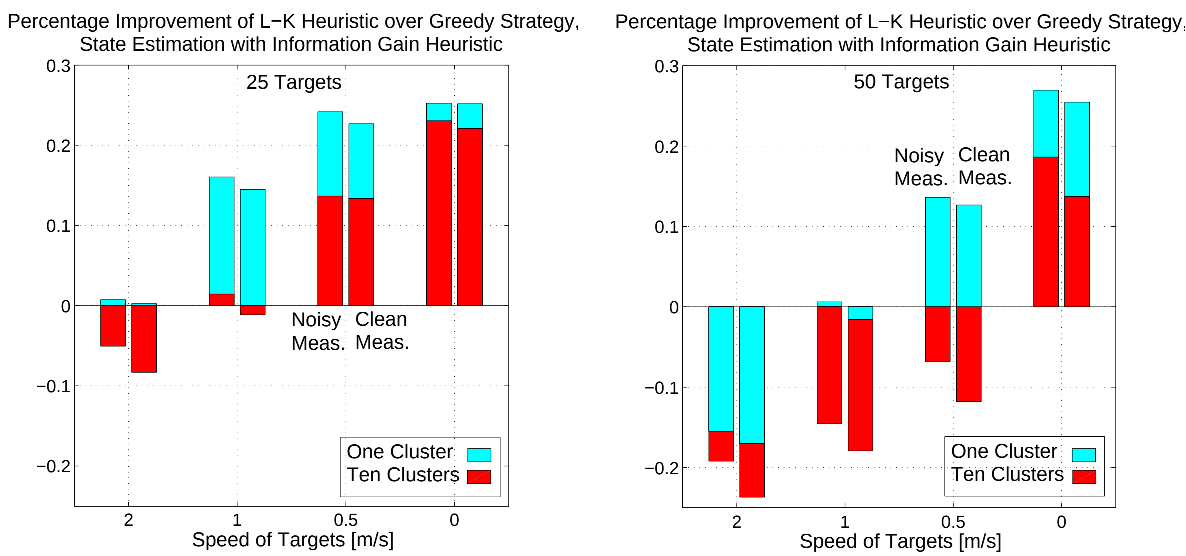

Fig. 4. Mean performance of LKH relative to a greedy algorithm, measured in terms of the total distance traveled by the agent to intercept all targets, is plotted for cases with imperfect measurments and restrictions on sensor range. A 25 target case is shown at left, and a 50 target case is shown at right. Performance of a noisy (10 m^2 noise variance) and clean (0.1 m^2 noise variance) target detection sensor is compared side-by-side for each target speed. The speed of the agent throughout all trials is 10 m/s.

target speeds and target quantitites. Figure 4 displays the results of all noise-corrupted sensing trials. The margin by which a LKH-based strategy outperforms a greedy strategy is significantly greater than the perfect sensing case at low target speeds, and is worse than the perfect sensing case at high target speeds. Uncertainty in target position amplifies the relative benefit of planning ahead versus pursuing the target of greatest immediate benefit, as increased time and distance are expended to locate targets with inaccurate or imprecise estimates. A noisier $ZoomIn(p_{agent})$ measurement leads to a slight improvement in the relative performance of a LKH-based strategy. In no case does a clustered environment outperform an obstacle-free environment under assumptions of noise-corrupted sensing, which is likely due to the increase in the frequency of collisions and the difficulty of accurately estimating target velocity across collisions. The performance of a LKH-based strategy relative to a greedy strategy also worsens more dramatically in increasing target quantity for a noise-corrupted sensing case. Despite this, the margin of improvement in low-target-quantity, low-velocity cases is substantial and would save significant time if implemented in practice, approaching in some instances a 25% savings or more.

V. Conclusion

We have considered a problem in which a large number of moving targets must be tracked and intercepted by a single agent, proposing a decision-making strategy for target pursuit based on the iterative solution of a traveling salesman tour over the moving targets. Unlike a typical sensor management application, this problem requires extensive physical travel by the agent. As a result, travel distance has been considered in the formulation of all cost metrics, including scenarios with noisy sensing and significant postition uncertainty. To manage uncertainty in tracking a large number of targets, an information gain metric has been proposed that penalizes each unit of travel required per each corresponding unit of uncertainty in position. It is evident that in cases of fast-moving targets, a non-greedy TSP tour is of limited benefit due to the rapidly-changing state of the system. However, when the movement of the targets is less aggressive, a LKH-based solution can yield a substantial gain in performance over a greedy target selection strategy.

References

[1] S.D. Bopardikar, S.L. Smith, F. Bullo, and J.P. Hespanha. Dynamic vehicle routing for translating demands: Stability analysis and receding-horizon policies. *IEEE Transactions on Automatic Control*, 55(11):2554–2569, 2010.

[2] A. Gelb, editor. *Applied Optimal Estimation*. MIT Press, 1974.

[3] M. Hammar and B.J. Nilsson. Approximation results for kinetic variants of TSP. In *Proceedings of the International Colloquium on Automata, Languages, and Programming*, volume 1644 of *Lecture Notes in Computer Science*, pages 392–401, Berlin, Germany, 1999. Springer.

[4] K. Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operations Research*, 126(1):106–130, 2000.

[5] C.S. Helvig, G. Robins, and A. Zelikovsky. The moving-target traveling salesman problem. *Journal of Algorithms*, 49:153–174, 2003.

[6] A.O. Hero III, D.A. Castanon, D. Cochran, and K. Kastella, editors. *Foundations and Applications of Sensor Management*. Springer, 2008.

[7] M.K. Kalandros, L. Trailovic, L.Y. Pao, and Y. Bar-Shalom. Tutorial on multisensor management and fusion algorithms for target tracking. In *Proceedings of the American Control Conference*, pages 4734–4748, Boston, MA, 2004.

[8] C. Kreucher, K. Kastella, and A.O. Hero III. Sensor mangement using an active sensing approach. 85(3):607–624, 2005.

[9] V. Krishnamurthy and R.J. Evans. Hidden Markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking. *IEEE Transactions on Signal Processing*, 49(12):2893–2908, 2001.

[10] S. Lin and B.W. Kernighan. An effective heuristic for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973.

[11] J. Liu, M. Chu, and J.E. Reich. Target tracking in distributed sensor networks. *IEEE Signal Processing Magazine*, pages 36–46, May 2007.

[12] G.A. McIntyre and K.J. Hintz. Information theoretic approach to sensor scheduling. In *Proceedings of SPIE Signal Processing, Sensor Fusion, and Target Recognition V*, volume 2755, Orlando, FL, 1996.

[13] S. Oh, L. Schenato, and S. Sastry. A hierarchical multiple-target tracking algorithm for sensor networks. In *Proceedings of the International Conference on Robotics and Automation*, pages 2197–2202, Barcelona, Spain, 2005.

[14] W.W. Schmaedeke. Information-based sensor management. In *Proceedings of SPIE Signal Processing, Sensor Fusion, and Target Recognition II*, volume 1955, Orlando, FL, 1993.

[15] S.L. Smith, S.D. Bopardikar, and F. Bullo. A dynamic boundary guarding problem with translating targets. In *Proceedings of the IEEE International Conference on Decision and Control*, pages 8543–8548, Shanghai, China, 2009.

[16] J.R. Spletzer and C.J. Taylor. Dynamic sensor planning and control for optimally tracking targets. *International Journal of Robotics Research*, 22(1):7–20, 2003.

[17] R.B. Washburn, M.K. Schneider, and J.J. Fox. Stochastic dynamic programming based approaches to sensor resource management. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 1, pages 608–615, 2002.