NON-MYOPIC SENSOR SCHEDULING FOR A DISTRIBUTED SENSOR NETWORK

Himanshu Shah and Darryl Morrell

Department of Electrical Engineering
Arizona State University
hcshah@asu.edu, morrell@asu.edu

ABSTRACT

When tracking a target in a sensor network with constrained resources, significant reductions in sensor communications and usage cost can be realized by using non-myopic sensor scheduling. The use of integer non-linear programming is beneficial for obtaining myopic sensor schedules [1]. In this paper, we extend its benefits to a non-myopic sensor scheduling scenario consisting of a distributed network of bearing sensors. We formulate this problem, which we call the Leader Node Scheduling problem, as an integer non-linear programming problem with the objective of minimizing the total sensor usage and communications cost over a planning horizon subject to tracking error constraints per time step.

Index Terms— tracking, non-myopic sensor scheduling, integer programming, outer approximation, particle filtering

1. INTRODUCTION

In many target tracking applications, it is essential to optimally utilize the sensor resources to minimize their usage and/or communication costs while keeping the target in track. When a sensor or a group of sensors is selected for the next measurement acquisition step, it is denoted as greedy or Myopic Sensor Scheduling (MSS) (see [1]); if we select a sensor or group of sensors for the next M measurement acquisition steps, it is referred to as Non-myopic Sensor Scheduling (NMSS). Certain target tracking scenarios like the ones in [2] benefit from NMSS. For the scenario mentioned in [2], the NMSS problem is solved using efficient search strategies.

We however formulate (a more difficult version of) the entire problem as an Integer Non-Linear Program (INLP) problem and solve it using an optimization strategy known as Outer Approximation (OA) [3]. OA was first used in [1] for MSS; we extend its use for NMSS scenarios.

The paper is organized as follows. In Section 2, we first formulate the problem. Sections 3 and 4 detail the formulation of the objective function and the tracking error constraints respectively for the INLP. We solve this INLP problem in Section 5. Section 6 provides simulation results and the conclusions are made in Section 7.

2. PROBLEM FORMULATION

We consider a sensor network consisting of S sensors, each placed at a known and fixed location $\mathbf{x}^s = (x^s, y^s)$. These sensors acquire bearing measurements from the target moving in the network. L of the S sensors are leader nodes and fuse the measurements obtained from other sensor nodes to update the target state belief.

The target state vector $\mathbf{x}_k$ at time k consists of the 2-D position components x_k and y_k and the 2-D velocity components $\dot{x}_k$ and $\dot{y}_k$ ($\mathbf{x}_k = [x_k \ y_k \ \dot{x}_k \ \dot{y}_k]^T$). The target motion is modeled by a discrete-time nearly constant velocity (NCV) model:

$$\mathbf{x}_{k+1} = \mathbf{F}\,\mathbf{x}_k + \mathbf{w}_k, \tag{1}$$

where $\mathbf{F}$ is the state transition matrix and $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$ and is white. Also, $\mathbf{F} = \left[\begin{array}{c|c} \mathbf{I}_2 & \Delta t\mathbf{I}_2 \\ \hline \mathbf{0}_2 & \mathbf{I}_2 \end{array}\right]$ and $\mathbf{Q}_k = q\left[\begin{array}{c|c} \frac{\Delta t^3}{3}\mathbf{I}_2 & \frac{\Delta t^2}{2}\mathbf{I}_2 \\ \hline \frac{\Delta t^2}{2}\mathbf{I}_2 & \Delta t\mathbf{I}_2 \end{array}\right]$ where Δt is the sampling interval and q is the noise intensity.

The target originated measurement $z_{k,s}$ from sensor s is

$$z_{k,s} = h(\mathbf{x}_k, \mathbf{x}^s) + v_k = \tan^{-1}\left(\frac{y_k - y^s}{x_k - x^s}\right) + v_k \tag{2}$$

where $v_k \sim \mathcal{N}(0, R^s)$ and R^s is the measurement noise variance for sensor s. We track the target using a particle filter (see references in [4]).

The non-myopic sensor scheduler plans the sensor schedule over an M step planning horizon to minimize the communications and usage cost subject to predicted tracking error constraints. The sensor scheduler at planning step m of an M step planning horizon selects a leader node l_m and a subset of sensor nodes $\mathcal{S}_m \subset \mathcal{S}$ to acquire measurements to transmit to l_m. If the leader node at the previous planning step is different than the one at the current planning step, then the belief state held by the leader node at the previous planning step must be transferred to the one at the current planning step. The M step sensor schedule is completely executed and the planning procedure is repeated i.e. the sensor scheduler is operated in the Open Loop mode. We call this problem of obtaining the M step sensor schedule the Leader Node Scheduling (LNS) problem.

3. OBJECTIVE FUNCTION FORMULATION

To compute the sensor usage and communication costs, we use the energy model from [2]. Using this model, we develop cost terms $c^1_{l,s}$, the cost of sensors acquiring observations and communicating them to leader node l, and $c^2_{l,l'}$, the cost of transmitting the target belief state from leader node l to node l'.

The objective function for the LNS problem is the total sensor usage and communications cost over the M step planning horizon. Let $a_{l,s,m}$ ($l = 1, 2, \ldots, L;\ s = 1, 2, \ldots, S;\ m = 1, 2, \ldots M$) be a collection of $S \times L \times M$ binary decision variables, such that $a_{l,s,m} = 1$ if a leader node l is chosen along with sensor node s at planning step m and is 0 otherwise. Let $b_{l,m}$ ($l = 1, 2, \ldots, L;\ m = 1, 2, \ldots M$) be a collection of $L \times M$ binary decision variables; $b_{l,m}$ is 1 if a leader node l is chosen at planning step m. The association between $a_{l,s,m}$ and $b_{l,m}$ is made by the following logical proposition:

$$(a_{l,1,m} = 1) \text{ or } (a_{l,2,m} = 1) \text{ or } \ldots\ (a_{l,S,m} = 1) \Rightarrow\ b_{l,m} = 1 \quad \forall\, l, \forall\, m \tag{3}$$

The logical proposition (3) indicates that if at least one of the S variables $a_{l,1,m}, a_{l,2,m}, \ldots, a_{l,S,m}$ is 1, then $b_{l,m} = 1$. We convert (3) into a conjunction of clauses, where each clause is a disjunction of literals. The resulting logical expression is then said to be in Conjunctive Normal Form (CNF) and is given by [4]:

$$\begin{aligned}
&(a_{l,1,m} = 1) \text{ or } (a_{l,2,m} = 1) \text{ or } \ldots \\
&\quad (a_{l,S,m} = 1)\ \Rightarrow\ b_{l,m} = 1 \\
\equiv\ & \Big((a_{l,1,m} = 0) \text{ or } (b_{l,m} = 1)\Big) \text{ and } \\
&\quad \Big((a_{l,2,m} = 0) \text{ or } (b_{l,m} = 1)\Big) \text{ and } \ldots \\
&\text{and } \Big((a_{l,S,m} = 0) \text{ or } (b_{l,m} = 1)\Big),\ \forall\, l, \forall\, m.
\end{aligned} \tag{4}$$

Equation (4) is equivalent to a series of SLM inequalities

$$a_{l,s,m} \leq b_{l,m} \quad \forall\, s, \forall\, l, \forall\, m. \tag{5}$$

Next we define decision variables $d_{l,l',m+1}$($l = 1, 2, \ldots, L,\ l' = 1, 2, \ldots, L,\ m = 1, 2, \ldots, M-1$) which are 1 if leader node l is chosen at planning step m and leader node l' is chosen at planning step $m+1$. There exists a logical expression between $b_{l,m}$ and $b_{l',m+1}$ given by

$$\Big((b_{l,m} = 1) \text{ and } (b_{l',m+1} = 1)\Big) \ \Leftrightarrow\ (d_{l,l',m+1} = 1) \tag{6}$$

Equation (6) is transformed into CNF and the resultant $3L^2(M-1)$ inequalities ($\forall\ l, m = 1, \ldots, M-1$) are:

$$\begin{aligned}
b_{l,m} + b_{l',m+1} - d_{l,l',m+1} &\leq 1 &\text{(7a)}\\
-b_{l,m} + d_{l,l',m+1} &\leq 0 &\text{(7b)}\\
-b_{l',m+1} + d_{l,l',m+1} &\leq 0. &\text{(7c)}
\end{aligned}$$

At time index k, the target belief is held by the leader node $l_0 \in \mathcal{L}$. The planning is then performed for times $k+m$, $m = 1, 2, \ldots, M$. In the inequalities in (7), we have not yet included the decision variables to indicate the transfer of belief state from time step k to time step $k+1$ (i.e $m = 0$ to $m = 1$). This is done by introducing L additional decision variables $d_{l_0,l,1}$ where l_0 is the currently active leader node. However, the 3 inequalities (7) collapse to the equality

$$b_{l',1} - d_{l_0,l',1} = 0 \tag{8}$$

since we do know the currently active leader node i.e. $b_{l_0,m} = 1$ for $m = 0$.

Note that the sensor scheduler at each planning step selects only 1 leader node. This requirement is represented by M equality constraints

$$\sum_{l=1}^{L} b_{l,m} = 1 \quad \forall\, m. \tag{9}$$

To now formulate the objective function, we make the following definitions.

$$\begin{aligned}
\mathbf{a}_{l,m} &\triangleq [a_{l,1,m}\ \ a_{l,2,m}\ \ \ldots\ \ a_{l,S,m}]^T &\text{(10a)}\\
\mathbf{a}_m &\triangleq [\mathbf{a}^T_{1,m}\ \ldots\ \mathbf{a}^T_{L,m}]^T &\text{(10b)}\\
\mathbf{b}_m &\triangleq [b_{1,m}\ \ b_{2,m}\ \ \ldots\ \ b_{L,m}]^T &\text{(10c)}\\
\mathbf{d}_{l,m+1} &\triangleq [d_{l,1,m+1}\ d_{l,2,m+1}\ \ldots\ d_{l,L,m+1}]^T &\text{(10d)}
\end{aligned}$$

Define

$$\begin{aligned}
\mathbf{c}^1_l &\triangleq [c^1_{l,1}\ c^1_{l,2}\ \ldots\ c^1_{l,L}]^T\\
\mathbf{c}^2_l &\triangleq [c^2_{l,1}\ c^2_{l,2}\ \ldots\ c^2_{l,L}]^T,
\end{aligned}$$

where $c^2_{l,s}$ and $c^2_{l,l'}$ are described in Section 3.

Let all the unknown binary variables defined in (10) above be denoted by a vector $\mathbf{\Gamma}$. The overall objective function therefore becomes

$$f(\mathbf{\Gamma}) \equiv \sum_{m=1}^{M}\sum_{l=1}^{L}(\mathbf{c}^1_l)^T\mathbf{a}_{l,m} + \sum_{m=1}^{M-1}\sum_{l=1}^{L}(\mathbf{c}^2_l)^T\mathbf{d}_{l,m+1} + (\mathbf{c}^2_{l_0})^T\mathbf{d}_{l_0,1}. \tag{11}$$

4. FORMULATION OF TRACKING ERROR CONSTRAINTS

For the LNS problem, we constrain the tracking error for each planning step $m = 1, 2, \ldots, M$. Let the predicted error covariance matrix for planning step m in an M step planning horizon and time step $k+m$ i.e. $\mathbf{P}_{k+m|k+m-1}$ be denoted as $\mathbf{P}_{m-}$ and the updated error covariance matrix $\mathbf{P}_{k+m|k+m}$ be denoted as $\mathbf{P}_{m+}$. We know that $\mathbf{P}_{m-}$ and $\mathbf{P}_{m+}$ can be computed recursively as:

$$\mathbf{P}_{m+} = \left(\mathbf{P}_{m-}^{-1} + \sum_{l=1}^{L}\sum_{s=1}^{S} a_{l,s,m}\mathbf{J}_{s,m}\right)^{-1} \quad (12)$$

where $\mathbf{J}_{s,m} = \mathbf{H}_{s,m}^T R_s^{-1}\mathbf{H}_{s,m}$ with

$$\mathbf{H}_{s,m} = \left.\frac{\partial h(\mathbf{x},\mathbf{x}^s)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{F}^m\hat{\mathbf{x}}_{k|k}}.$$

Let $\rho_{T,m}$ be the tracking error threshold for planning step m. The tracking error constraints therefore are:

$$g_m(\mathbf{a}_m) \equiv \rho(\mathbf{a}_m) - \rho_{T,m} \leq 0 \quad m = 1,2,\ldots,M \quad (13)$$

where $\rho(\mathbf{a}_m) \triangleq tr_{1,2}(\mathbf{P}_{m+})$. Here, $tr_{1,2}(\cdot)$ represents the trace of the submatrix formed by rows 1 and 2 and columns 1 and 2 of the matrix $(\cdot)$.

5. SOLVING THE LNS PROBLEM USING OUTER APPROXIMATION

For the LNS problem, we formulate an INLP problem $\mathbb{P}_{INLP-LNS}$ to minimize (11) i.e. the sensor usage and communications cost subject to tracking error constraints (13). The INLP problem is solved using OA [3], which requires that the objective function f and the constraints g_m be convex in order to guarantee a global solution. f for the LNS problem is linear and hence convex. In [4] we show that g_m can be made a convex function of the relaxation of $\mathbf{a}_m$ if $\mathbf{Q}=0$. With $\mathbf{Q}=\mathbf{0}$, $\mathbf{P}_{m+}$ is given as [4]:

$$\mathbf{P}_{m+} = \Big(\big(\mathbf{F}^m\mathbf{P}_{0+}(\mathbf{F}^T)^m\big)^{-1} + \sum_{t=1}^{m}\sum_{l=1}^{L}\sum_{s=1}^{S} a_{l,s,m}\big((\mathbf{F}^T)^{m-t}\big)^{-1}\mathbf{J}_{s,m}(\mathbf{F}^{m-t})^{-1}\Big)^{-1} \quad (14)$$

For tracking applications where $\mathbf{Q}=\mathbf{0}$ is a poor assumption, one can hedge against the use of $\mathbf{Q}=\mathbf{0}$ by replacing $\rho_{T,m}$ with $\rho_{T,m}(1-\epsilon)^m$ where $0<\epsilon\ll 1$.

The solution to $\mathbb{P}_{INLP-LNS}$ provides the non-myopic sensor schedule over the M step planning horizon. The complete INLP formulation $\mathbb{P}_{INLP-LNS}$ of the LNS problem is

$$\min_{\mathbf{\Gamma}} \quad f(\Gamma)$$

subject to constraints (5), (7), (8), (9) and (13).

The master problem at the p^{th} iteration i.e. $\mathbb{P}^p_{MILP-LNS}$ is:

$$\begin{aligned}
&\min_{\mathbf{\Gamma}} \ \zeta \\
&\text{subject to } \zeta \leq B_U^p \\
&\zeta \geq \textstyle\sum_{m=1}^{M}\sum_{l=1}^{L}(\mathbf{c}_l^1)^T\mathbf{a}_{l,m} + \sum_{m=1}^{M-1} \\
&\qquad \textstyle\sum_{l=1}^{L}(\mathbf{c}_l^2)^T\mathbf{d}_{l,m+1} + (\mathbf{c}_{l_0}^2)^T\mathbf{d}_{l_0,1} \\
&0 \geq \rho(\mathbf{a}_m^i) - \rho_{T,m} + [\nabla\rho(\mathbf{a}_m^i)]^T \\
&\qquad (\mathbf{a}_m - \mathbf{a}_m^i) \quad \forall\, m, \forall i \in \mathbf{\Xi}_F^p \text{ or } \mathbf{\Xi}_I^p \\
&\text{and constraints (5), (7), (8), (9), and (13).} \\
&\mathbf{\Gamma} \in \{0,1\}^{\ell(\mathbf{\Gamma})},\ \zeta \in \mathbb{R}
\end{aligned}$$

where (See [4] for more details)

$$\nabla\rho(\mathbf{a}_m^i) = \left[\begin{array}{ccc} \dfrac{\partial\rho(\mathbf{a}_m^i)}{\partial a_{1,1,1}} & \ldots & \dfrac{\partial\rho(\mathbf{a}_m^i)}{\partial a_{L,S,M}}\end{array}\right]^T,$$

$\ell(\cdot)$ denotes the dimension of the vector $(\cdot)$ and $\mathbf{\Xi}_F$ and $\mathbf{\Xi}_I$ are the sets that contain the feasible and infeasible solutions to $\mathbb{P}_{INLP-LNS}$. The complete OA algorithm therefore proceeds as shown in Table 1.

The solution to the OA problem above gives the leader node l_m to be used along with the sensor node subset $\mathcal{S}_m$ at each planning step m. The selected sensors transmit the measurements to leader node which then updates the target state density. After the M step sensor schedule is executed, the entire process of planning and execution is repeated.

Initialization: $\mathbf{\Gamma}^0=\mathbf{0}, p=0, \mathbf{\Xi}_F^{-1}=\emptyset, \mathbf{\Xi}_I^{-1}=\emptyset, B_U^0=\infty$. Repeat 1. Linearize (13) about $\mathbf{\Gamma}^p$ (see 3^{rd} constraint in $\mathbb{P}^p_{MILP-LNS}$) and set $\mathbf{\Xi}_F^p=\mathbf{\Xi}_F^{p-1}\cup\{p\}$ or $\mathbf{\Xi}_I^p=\mathbf{\Xi}_I^{p-1}\cup\{p\}$ appropriately. 2. If $g_m(\mathbf{\Gamma}^p)\leq 0, \forall\, m$ and $f(\mathbf{\Gamma}^p)\leq B_U^p$, set $\mathbf{\Gamma}^{opt}=\mathbf{\Gamma}^p$ and $B_U^{p+1}=f(\mathbf{\Gamma}^p)$. 3. Solve $\mathbb{P}^p_{MILP-LNS}$ to find the solution $\mathbf{\Gamma}^{p+1}$. Set $p=p+1$. Until ($\mathbb{P}^p_{MILP-LNS}$ is infeasible)

Table 1. OA algorithm for the LNS problem.

6. SIMULATION RESULTS

We perform simulations for tracking a target in a sensor network consisting of $S=6$ bearing sensors and $L=2$ leader nodes as shown in Figure 1. $\mathcal{S}_5$ and $\mathcal{S}_6$ also behave as the leader nodes $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. All sensors have a probability of detection $P_D=1$ and do not give any false alarms.

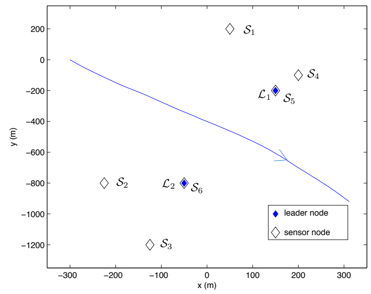

Fig. 1. Simulation Scenario for the LNS problem. $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4, \mathcal{S}_5, \mathcal{S}_6\}$ are the sensor nodes and $\{\mathcal{L}_1, \mathcal{L}_2\}$ are the leader nodes.

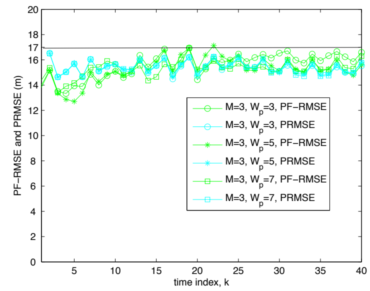

Fig. 2. RMSE plots for $M = 3$ and different values of W_p with $\rho_{T,m} = 289(1-\epsilon)^m$.

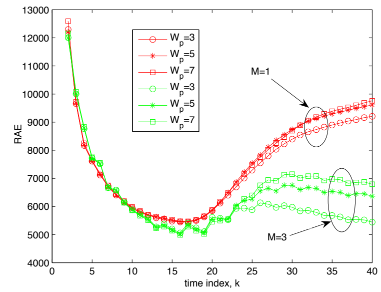

Fig. 3. RAE plots for $M = 1$ and $M = 3$ and different values of W_p with $\rho_{T,m} = 289(1-\epsilon)^m$.

The initial target state parameters are $\mathbf{x}_0 = [-300\ 13\ 0\ -21]$ and $\mathbf{P}_{0|0} = diag\{1000\ 50\ 1000\ 50\}$ and it travels from left to right for a period of 40 seconds. Also, the initial target belief state is held by $\mathcal{L}_1$. The other parameters used are: $q = 1$, $\Delta t = 1$, $R^s = 0.03, \forall\ s$ and $\epsilon = 0.03$. We solve $\mathbb{P}^p_{MILP-LNS}$ using the software lpsolve available freely at [5].

The target state Root Mean Square Error (RMSE) obtained by the Particle filter (i.e. PF-RMSE) and the predicted RMSE (i.e. PRMSE) obtained from (13) (averaged over 500 Monte Carlo (MC) runs) is given in Figure 2 for planning horizon length $M = 3$, with $\rho_{T,m} = 289$ and different values of W_p, where an increasing W_p indicates an increasing cost to transfer target belief. The plot for $M = 1$ is similar to Figure 2 and is not shown.

Figure 3 is the plot for Running Average Energy (RAE) [2] against time index k for $M = 1$ and $M = 3$ for different values of W_p averaged over 500 MC iterations and shows that significant energy savings are accrued by planning over a $M = 3$ step (non-myopic) planning horizon as compared to a $M = 1$ step (myopic) planning horizon. The accrued energy savings is because the belief state is typically transferred from $\mathcal{L}_1$ to $\mathcal{L}_2$ in more simulation runs with M=3 than with M=1; the transfer leads to improved energy performance later in a simulation run. The number of transfers from $\mathcal{L}_1$ to $\mathcal{L}_2$ in a 500 simulation run for different values of M and W_p are shown in Table 2.

7. CONCLUSIONS

We have formulated a NMSS problem for a distributed sensor network as an INLP problem to minimize the total sensor usage and communications cost over an M step planning horizon subject to tracking error constraints at each planning step. We have provided an efficient formulation with mostly linear constraints (apart from the tracking error constraints). Such a formulation was found to be solved much faster using OA than using the efficient search technique provided in [2].

M	W_p	Number of Belief transfers for $\rho_{T,m} = 289(1-\epsilon)^m$
1	3	79
	5	4
	7	0
3	3	412
	5	178
	7	68

Table 2. Number of times the target belief was transferred from $\mathcal{L}_1$ to $\mathcal{L}_2$ in a 500 run simulation.

8. REFERENCES

[1] AS Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Sensor Resource Allocation for Tracking Using Outer Approximation," *IEEE Signal Processing Letters*, vol. 14, no. 3, pp. 213–216, 2007.

[2] AS Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Energy Efficient Target Tracking in a Sensor Network Using Non-myopic Sensor Scheduling," *7th International Conference on Information Fusion, 2005*, vol. 1, 2005.

[3] R. Fletcher and S. Leyffer, "Solving mixed integer nonlinear programs by outer approximation," *Mathematical Programming*, vol. 66, no. 1, pp. 327–349, 1994.

[4] Himanshu Shah, "Non-myopic sensor scheduling using outer approximation," Tech. Rep., Arizona State University, 2007.

[5] Kjell Eikland and Peter Notebaert, "lpsolve C code," Available at: `http://sourceforge.net/projects/lpsolve`.