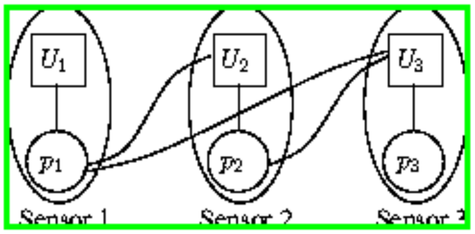

Figure 1: The factor graph for three sensors

We have now transformed the problem so it is amenable to optimisation by the max-sum algorithm. The max-sum algorithm is a decentralised coordination algorithm based on message passing, and has been shown to compute good quality approximate solutions with acceptable computation overhead compared to various optimal decentralised algorithms (Farinelli et al. 2008). Moreover, it has been empirically shown to scale well with the number of sensors, and be robust against message loss and failure of individual sensors. In more detail, the max-sum algorithm requires the problem to be encoded as a factor graph, which is a bipartite graph in which vertices represent variables and functions, and an edges represent the 'is parameter of' relation. Hence, in our problem, an edge exists between vertices p_i and U_j iff $i \leq j$, such that sensor i's utility function U_i has i variables. Figure 1 shows the factor graph for a setting with three sensors.

Now, we can use the max-sum algorithm to find a path for each sensor $[p_1^*, \ldots, p_n^*]$ that maximises the value of the measurements collected by the team of sensors:

$$[p_1^*, \ldots, p_n^*] = \underset{[p_1, \ldots, p_n]}{\arg\max} \sum_{i=1}^{n} U_i(p_1, \ldots, p_i) \qquad (3)$$

This is accomplished in a decentralised fashion by passing messages between the functions and variables as follows:

$q_{i \to j}$ **Message from variable p_i to function U_j ($i \leq j$):**

$$q_{i \to j}(p_i) = \alpha_{ij} + \sum_{k \geq i, k \neq j} r_{k \to i}(p_i) \qquad (4)$$

where α_{ij} is a normalising constant chosen such that $\sum_{p_i} q_{i \to j}(p_i) = 0$, so as to prevents the messages from growing to the point of calculation error in cyclic factor graphs.

$r_{j \to i}$ **Message from function U_j to variable p_i ($i \leq j$):**

$$r_{j \to i}(p_i) = \max_{\mathbf{p}_j \setminus p_i} \left[U_j(\mathbf{p}_j) + \sum_{k \leq j, k \neq i} q_{k \to j}(p_k) \right] \qquad (5)$$

Here we use $\mathbf{p}_j$ as an abbreviation for $\{p_1, \ldots, p_j\}$. Thus, $U_j(\mathbf{p}_j)$ is the utility function of sensor j.

These messages are exchanged until their contents converge, or for a predefined number of iterations if the former does not occur. Then, to obtain its part of the coordinated solution p_i^* defined in Equation 3, sensor i calculates:

$$p_i^* = \arg\max_{p_i} \sum_{k \geq i} r_{k \to i}(p_i) \qquad (6)$$

The motivation here is that this sum is an approximation to the marginal function of Equation 3 with respect to p_i, where for any element e of the domain p_i, this marginal function is equal to the maximum value Equation 3 can attain if $p_i = e$. When the factor graph is acyclic, this marginal function is exact, and as a result, the solution computed in Equation 6 is guaranteed to be optimal. Otherwise, it is an approximation to Equation 3.

In Section 5 we apply our algorithm to two challenging information gathering domains, which are discussed next.

4 Two Information Gathering Domains

We now instantiate the general model and algorithm described in Sections 2 and 3 for two information gathering domains for which no online decentralised algorithm previously existed. In the first, pursuit-evasion, sensors are tasked with capturing a moving object in graph G. In the second, patrolling, the sensors' goal is to prevent intrusions on vertices of G. In both domains, observations y are binary: either a moving object or intrusion is detected, or it is not. Furthermore, we assume that sensors are capable of sensing all locations within a radius of r_s of their position. Sensing is assumed to be imperfect, and is given by a probability of a false positive p_{fp} and a false negative p_{fn} detection. In the upcoming sections, we will derive the measurement value function f we defined in Section 2 for both domains.

4.1 Pursuit Evasion

The pursuit evasion problem is characterised by the presence of a single moving object e (called an evader) that the sensors have to capture as quickly as possible. The evader e has a motion model M (e.g. random, stationary, etc.) that specifies how it moves in graph G. Figure 2 shows an example of this scenario with a randomly moving evader.

Now, to model their belief of the evader's location at time t, denoted by e_t, the sensors maintain a probabilistic map $p_e(e_t = v\ Y_{t-1})$ representing the probability that the evader is at location v given measurement history Y_{t-1}. This model extends the work by Hespanha, Kim, and Sastry (1999) to a setting with an arbitrary evader (as opposed to a randomly moving one), or an unknown evader. Each sensor has a copy of this probabilistic map which is kept consistent by exchanging observations. At each time step t, the sensors take new measurements y_t and obtain $p_e(e_{t+1} = v\ Y_t)$ in two steps:

1. Fuse measurements y_t with $p_e(e_t = v\ Y_{t-1})$ to obtain:

   $$p_e(e_t = v\ Y_t) = \alpha p_e(e_t = v\ Y_{t-1}) p(y_t\ e_t = v, Y_{t-1})$$

   Here, α is a normalising constant, and $p(y_t\ e_t = v, Y_{t-1})$ is computed as:

   $$p(y_t\ e_t = v, Y_{t-1}) = \prod_{y \in y_t} p(y\ e_t = v, Y_{t-1})$$

   and $p(y\ e_t = v, Y_{t-1})$ is given by the sensing model:

   $$p(y|e_t = v, Y_{t-1}) = \begin{cases} 1 - p_{fp} & \text{if } y(m) = T \wedge y(v) = v \\ p_{fp} & \text{if } y(m) = T \wedge y(v) \neq v \\ p_{fn} & \text{if } y(m) = F \wedge y(v) = v \\ 1 - p_{fn} & \text{if } y(m) = F \wedge y(v) \neq v \end{cases}$$