Uncertainty-Aware Task Allocation for Distributed Autonomous Robots

Liang Sun\textsuperscript{1} and Leonardo Escamilla\textsuperscript{1}

\textbf{Abstract}—This paper addresses task-allocation problems with uncertainty in situational awareness for distributed autonomous robots (DARs). The uncertainty propagation over a task-allocation process is done by using the Unscented transform that uses the Sigma-Point sampling mechanism. It has great potential to be employed for generic task-allocation schemes, in the sense that there is no need to modify an existing task-allocation method that has been developed without considering the uncertainty in the situational awareness. The proposed framework was tested in a simulated environment where the decision-maker needs to determine an optimal allocation of multiple locations assigned to multiple mobile flying robots whose locations come as random variables of known mean and covariance. The simulation result shows that the proposed stochastic task allocation approach generates an assignment with 30\% less overall cost than the one without considering the uncertainty.

I. INTRODUCTION

One of the great challenges facing modern society is to develop new technologies that transform how we manage the transportation ecosystem to embrace upcoming large-scale distributed autonomous robots (DARs), such as unmanned aerial vehicles (or drones) and self-driving cars. Autonomous robots have demonstrated their capabilities to complete missions that are too dangerous, dumb, costly, or impossible for humans to handle, such as transportation [1], [2], warehouse sorting [3], emergency response [4], and hazardous chemical spraying [5], to name a few. These challenging missions usually consist of multiple tasks with demands and uncertainty that may vary over time, which pose critical challenges to the coordination of DARs. The advances of optimization theory and computer algorithms have revolutionized approaches (e.g., market-based algorithms [6] and bipartite matching algorithms [7]) to centralized task-allocation strategies for autonomous robots, but these approaches are not directly applicable to DARs. When distributed optimization techniques are paired with probability theory and appropriate coordination mechanisms, potentially transformative gains in stochastic task-allocation and dynamic reallocation for DARs are possible. However, the widespread use of DARs for real-world missions has remained elusive, in large part due to limited system-level understanding and control of the complex changes that DARs undergo dynamic and uncertain environments.

Figure 1 depicts a motivating scenario for this paper, where a team of (heterogeneous) unmanned aerial vehicles (UAVs) collaboratively executes multiple “tasks”, which can be defined as simple actions (e.g., take pictures) or complicated missions (e.g., mobile-target tracking, search and rescue, etc.) that may contain subtasks. These tasks and subtasks may be located at distant places with different priorities and different costs to execute. The yellow triangles denote the sensing range of the UAVs and sensor measurements are assumed to be with uncertainty.

As an integral part of combinatorial optimization [8], [9], task allocation processes often form building blocks of solutions to complex problems and have been widely investigated in the literature [10]–[13], such as resource allocation and coordination of DARs [14]–[17]. Auction and Hungarian algorithms are two well-known centralized task-allocation methods in their original forms, which have been initially applied in assignment problems. The Hungarian algorithm [7] was the first to compute an optimal solution in finite time to the linear sum assignment problem (LSAP) [18] and has been implemented in [19], [20]. It has been utilized in applications, such as moving-target detection and tracking [14], [21], [22], formation generation [10], and fault tolerance for cooperative unmanned systems [23]. The market-based auction algorithm depends on a central auctioneer.

\textsuperscript{1}Liang Sun and Leonardo Escamilla are with Department of Mechanical and Aerospace Engineering, New Mexico State University, Las Cruces, NM 88003, lsun@nmsu.edu; brwnl30@nmsu.edu
that is responsible for assigning tasks according to received bids [6], [10], [24]. Auction algorithms require robots to bid on tasks, with rewards based on corresponding prices [6], [25]. The Auction algorithm has been applied in various applications [26]–[33], such as multi-robot coordination [34], [35] and the network flow problems [6], [36].

Centralized task-allocation algorithms rely on a central agent to achieve an optimal assignment by considering the situational awareness of all agents. These algorithms are usually able to produce global optimal solutions, but unfortunately they have limitations in robustness and scalability. Distributed auction-based approaches have been widely explored, such as the Consensus-based bundle algorithm (CBBA) [26], which has been adopted as a benchmark technique for multi-task allocation problems. Consensus-based approaches [26], [37]–[41] typically require the robots to converge on a consistent situational awareness before performing the assignment. Though such methods are robust, they are typically slow to converge and require the transmission of large amounts of data. Although the original Hungarian method outperforms the auction method on the basis of scalability and system requirements [42], the distributed version of the Hungarian method has drawn limited attention [43]–[48]. Also, most existing algorithms for distributed task allocation assume that the convergence speed of the algorithm is faster than the change in system states (e.g., locations of mobile DARs) nor take system uncertainty into the optimization process for task allocation.

The ubiquitous uncertainty in situational awareness makes deterministic models difficult to meet the requirement for evaluation of success for critical missions, thereby stochastic models have been widely adopted in DAR autonomy. The uncertainty in situational awareness is usually represented by random variables with specific probability distributions, e.g., the mean-variance pair, \((\mu, \sigma^2)\). The propagation of a random variable through a nonlinear function and the fusion of two random variables have been studied and utilized in recursive Bayesian filtering techniques [49], which were originally developed for probabilistic inference.

Probabilistic inference is the problem of estimating the hidden variables (states or parameters) of a system optimally and consistently when a set of noisy or incomplete observations of the system becomes available [50]. The optimal solution to this problem is given by the recursive Bayesian estimation algorithm, which recursively updates the posterior density of the system state as new observations arrive. This posterior density constitutes the complete solution to the probabilistic inference problem and allows us to calculate any “optimal” estimate of the state. Unfortunately, for most real-world problems, the optimal Bayesian recursion is intractable and approximate solutions must be used. Within the space of approximate solutions, the extended Kalman filter (EKF) has become one of the most widely used algorithms with applications in state and parameter estimation [51]–[55]. Unfortunately, the EKF is based on a sub-optimal implementation of the recursive Bayesian estimation framework applied to Gaussian random variables. This can seriously impact the accuracy or even lead to divergence of any inference system that is based on the EKF.

Over the past two decades, several novel, more accurate, and theoretically better motivated algorithmic alternatives to the EKF have surfaced in the literature. These include the Unscented Kalman Filter (UKF) [56] and Central-Difference Kalman Filter (CDKF) [57], which are collectively referred to as Sigma-Point Kalman Filters (SPKF). Unlike the EKF that uses a first-order “linearization” approach to deal with nonlinear systems, the Sigma-Point methods involve deterministic sampling-based approximations of the relevant Gaussian statistics to achieve “second-order” or higher accuracy. Remarkably, the computational complexity of an SPKF is the same order as that of the EKF. Furthermore, implementing an SPKF is often substantially easier and requires no analytic derivation of Jacobians (gradients) as in the EKF.

In this paper, we proposed a novel solution framework for generic task allocation with uncertainty for DARs. The proposed framework decouples the uncertainty propagation and the task allocation processes by utilizing the sigma-point sampling. It has great potential to be employed for generic task-allocation schemes, in the sense that there is no need to modify an existing task-allocation method that has been developed without considering the uncertainty in the situational awareness. The proposed framework was tested in a simulated environment where the decision-maker needs to determine an optimal allocation of multiple locations assigned to multiple mobile flying robots whose locations come as random variables of known mean and covariance. The simulation result shows that the proposed stochastic task allocation approach generates an assignment with 30% less overall cost than the one without considering the uncertainty.

The rest of the paper is organized as follows. In Section II the formal formulation for the proposed work is provided. In Section III we present the methodology of the proposed framework. In Section IV we present the simulation results that validate the effectiveness of the proposed method and the concluding remarks are summarized in Section V.
\(N = \{N_1, N_2, \cdots, N_m\}\) is a set of neighbor lists for each robot and \(a_j \in N_i\) if \(g_{ij} \neq 1\). Note that the time variable \(t\) is omitted in \(\{\mathcal{E}, P_\mathcal{E}, \mathcal{G}, N\}\) for simplification.

For a non-fully connected DAR system (i.e., a robot not directly connected to all other robots), it takes some time (e.g., several hops) for updated knowledge to be transmitted from robot \(a_i\) to another robot \(a_j\), where \(a_j \notin N_i\), via the communication network defined by \(\mathcal{G}\). So the situational awareness that robot \(a_i\) possesses about the entire DAR system is formally defined by \(\mathcal{S}_k(\mathcal{E}, P_\mathcal{E}, \mathcal{G}, N_i)\), \(i \in \mathcal{I}\), where \(\mathcal{I} = \{1, 2, \cdots, m\}\). In such an asynchronous situation, the situational awareness among robots is not necessarily the same at each time step. So let us assume that \(\mathcal{S}^j_k \neq \mathcal{S}^j_k\) until the DAR system converges to a common \(\mathcal{S}\). Note that, for simplicity of presentation of the proposed framework, \(\mathcal{G}\) and \(N\) are assumed static.

A. Uncertainty Propagation

Unscented transform [49], [55], [58], [59] is a mathematical function used to estimate the result of applying a given nonlinear transformation to a probability distribution that is characterized only in terms of a finite set of statistics. The most common use of the unscented transform is in the nonlinear projection of mean and covariance estimates in the context of nonlinear extensions of the Kalman filter.

Define the dynamics of a system using a state-space model as

\[
x_{k+1} = f(x_k) + w_k,
\]

and a generalized nonlinear function as

\[
y_k = h(x_k) + v_k,
\]

where \(x_k \in \mathbb{R}^L\) is the state vector, \(y_k \in \mathbb{R}^L\) is the output vector, and \(w_k \in \mathbb{R}^L\) and \(v_k \in \mathbb{R}^L\) are independent, zero-mean Gaussian noise processes of covariance matrices \(P_w\) and \(P_v\), respectively. An optimal estimate of \(\hat{x}_k\) is given by

\[
\hat{x}_k = \mathbb{E}[\hat{x}_k],
\]

where \(\mathbb{E}(\cdot)\) denotes the expected value operation and the error covariance matrix is given by

\[
P_{\hat{x}_k} = \mathbb{E}\left[ (x_k - \hat{x}_k)(x_k - \hat{x}_k)^T \right].
\]

Assuming a Gaussian distribution of \(x_k\), an optimal estimate of \(h(x_k)\) is given by

\[
\mathbb{E}[h(x_k)] = \int h(x_k) \mathcal{N}(x_k|\hat{x}_k, P_k) \, dx,
\]

where \(\mathcal{N}(x_k|\hat{x}_k, P_k)\) is a multi-dimensional Gaussian density with mean \(\hat{x}_k\) and covariance matrix \(P_k\). The unscented transform approximates \(\mathbb{E}[h(x_k)]\) using multi-dimensional generalizations of Gaussian quadratures, also referred to as Gaussian cubatures, i.e., \(\mathbb{E}[h(x_k)] \approx \sum_{k} \omega_{k,i} h(\hat{x}_{k,i})\), where weights \(\omega_{k,i}\) and sigma points \(\hat{x}_{k,i}\) are functions of mean \(\hat{x}_k\) and covariance matrix \(P_k\). Formally, the set of sigma points, a \((2L+1)\)-tuple located in the center and on the surface of an \(n\)-sphere, is defined by

\[
\mathcal{X}_k = \left\{ \mathcal{X}_{k,0}, \mathcal{X}_{k,1}, \cdots, \mathcal{X}_{k,2L} \right\}
= \left\{ \hat{x}_k, \hat{x}_k + \gamma \sqrt{P_k}, \hat{x}_k - \gamma \sqrt{P_k} \right\},
\]

where \(\gamma\) is a scalar that determines the spread of the sigma-points around \(\hat{x}_k\) and \(\Sigma_k\) [60]. Then the evolution of \(\hat{x}\) and its covariance matrix is given by

\[
\hat{x}_{k+1} = f(\mathcal{X}_{k,i}),
\]

\[
\hat{x}_{k+1} = \sum_{j=0}^{2L} \omega_{m,j} \mathcal{X}_{k+1,i},
\]

\[
P_{\hat{x}_{k+1}} = \sum_{i=0}^{2L} \sum_{j=0}^{2L} \omega_{c,i,j} \mathcal{X}_{k+1,i} \mathcal{X}_{k+1,j}^T,\]

and the evolution of \(\hat{y}_k\), its covariance matrix, and the cross-correlation of \(x\) and \(y\) are given by

\[
\mathcal{B}_{k,i} = h(\mathcal{X}_{k,i}),
\]

\[
\hat{y}_k = \sum_{i=0}^{2L} \omega_{m,i} \mathcal{B}_{k,i},
\]

\[
P_{\hat{y}_k} = \sum_{i=0}^{2L} \sum_{j=0}^{2L} \omega_{c,i,j} \mathcal{B}_{k,i} \mathcal{B}_{k,j}^T,\]

where \(\omega_{m}\) and \(\omega_{c}\) are scalar weights.

The sigma-point approach differs substantially from general stochastic sampling techniques, such as the Monte-Carlo integration, which requires orders of magnitude more sample points in an attempt to propagate an accurate (possibly non-Gaussian) distribution of the state. The deceptively simple sigma-point approach results in posterior approximations that are accurate to the third order for Gaussian inputs for all nonlinearities. For non-Gaussian inputs, approximations are accurate at least the second-order, with the accuracy of third and higher-order moments determined by the specific choice of weights and scaling factors. Furthermore, no analytical Jacobians of the system equations need to be calculated as is the case for the EKF. This makes the sigma-point approach very attractive for use in “black box” systems where analytical expressions of the system dynamics are either not available or not in a form that allows for easy linearization [60].

B. Distributed Hungarian-Based Algorithm

We adopt the Distributed Hungarian-Based Algorithm (DHBA) [61] to implement the task-allocation function in the proposed framework. DHBA uses an implicit coordination approach for each individual robot to produce task-allocation results. The core the DHBA is the Hungarian algorithm and we adopt one of its implementations as presented in Algorithm F in which given the input as an \(m \times m\) cost matrix, \(C\), a bipartite graph is constructed based on which assignments are admissible. The vectors containing task
labels and agent labels are denoted by $U$ and $V$, both of which are $1 \times m$ vectors with elements $u_i$ and $v_i$, respectively, where $i, j \in \{1, 2, \ldots, m\}$. The edge between $u_j$ and $v_i$ is considered admissible if $v_i + u_j = c_{ij}$. In the bipartite graph, there are either matched admissible edges or unmatched admissible edges (line 7). An edge is considered matched if an agent has chosen a particular admissible task (line 12). More details can be found in [47].

**Algorithm 1 Hungarian Algorithm [19]**

1: procedure HUNGARIAN($C$) $\triangleright C$ is $m \times m$ cost matrix of non-negative integers
2: for all $v_i \in V$, $v_i = 0$
3: for all $u_j \in U$, $u_j = \min(C(:,j)) \triangleright U$ and $V$ are task and agent label vectors of all the tasks and agents
4: while Not Full Match do
5: for all $C_{ij}$
6: if $(v_i + u_j = C_{ij})$:
7: edge is admissible
8: end if
9: end for
10: for all agents $i \in m$, tasks $j \in m$ $\triangleright$ matching phase begins
11: if (edge = admis. and unm.):
12: match agent $i$ and task $j$
13: end if
14: end for
15: Augment(Shortest Path)
16: if (any agent is unmatched):
17: unmatched agent gets marked
18: connected nodes get marked
19: end if
20: for all $j \in J$
21: slack$_j = \min(C_{marked,j}) - (u_j + v_{marked})$
22: end for
23: $\delta = \min$(slack$_j$)/2
24: for all $i \in I$ and $j \in J$
25: if (agent $i$ is marked):
26: $v_i = v_i + \delta$
27: otherwise
28: $v_i = v_i - \delta$
29: end if
30: if (task $j$ is marked):
31: $u_j = u_j - \delta$
32: otherwise
33: $u_j = u_j + \delta$
34: end if
35: end for
36: end while
37: return Match
38: end procedure

**III. METHODOLOGY**

In this section, we propose a framework for stochastic task allocation that explicitly accounts for the uncertainty in situational awareness in the assignment process. In this work, we assume that the numbers of agents and tasks can be the same, while the proposed framework has great potential to be applied to other cases.

The task allocation problem given situational awareness $\Xi$ is formally defined by $(f_{TG}, \mathcal{F}, P_{\mathcal{F}}, f_{\mathcal{E}}, C, f_{TA}, \Gamma, P_{\Gamma})$, where:

- $f_{TG}$ is an algorithm/function that generates task set $\mathcal{F}$ and corresponding covariance matrices set $P_{\mathcal{F}}$.
- $\mathcal{F} = \{\tau_1, \tau_2, \ldots, \tau_m\}$ is a set of $m$ random variables, in the format of column vectors, representing task states generated by task generator $f_{TG}$ given $(\Xi, P_{\Xi})$.
- $P_{\mathcal{F}} = \{P_{\tau_1}, P_{\tau_2}, \ldots, P_{\tau_m}\}$ is a set of covariance matrices of set $\mathcal{F}$.
- $f_{\mathcal{E}}$ is a function that generates cost matrix $\mathcal{E}$ and corresponding covariance matrices set $P_{\mathcal{E}}$.
- $\mathcal{E} = [c_{ij}] \in \mathbb{R}^m \times m$ is the cost matrix, where $c_{ij}$ denotes the cost that robot $a_i$ pays to execute task $\tau_j$.
- $P_{\mathcal{E}} \in \mathbb{R}^m \times m$ is the covariance matrix associated with $\mathcal{E}$.
- $f_{TA}$ is an algorithm/function that performs task allocation.
- $\Gamma = [\gamma_{ij}] \in \mathbb{R}^m \times m$ is the resulting matrix generated by applying $f_{TA}$ onto $\mathcal{E}$.
- $P_{\Gamma} \in \mathbb{R}^m \times m$ is the covariance matrix associated with $\Gamma$.

The proposed framework includes the following steps (see Fig. 2):

1) Extract the information from situational awareness $(\Xi, P_{\Xi})$ for task allocation and generate tasks in a stochastic form $(\mathcal{F}, P_{\mathcal{F}})$ using function $f_{TG}$;
2) Generate cost information $(\mathcal{E}, P_{\mathcal{E}})$ using function $f_{\mathcal{E}}$;
3) Generate sigma points using Unscented Transform described by Equations (7) and (11);
4) Pass the sigma points through the task allocation process $(f_{TA})$ and calculate the mean and covariance of the result $(\Gamma, P_{\Gamma})$;
5) Generate an executable assignment $\Gamma$ by interpreting the stochastic form of the result $(\Gamma, P_{\Gamma})$.

**A. Interpreted Policy**

Traditional task-allocation algorithms generate a binary assignment matrix, whereas a saliently novel contribution of the proposed stochastic task allocation framework is the evaluation of the confidence, $P_{\Gamma}$, on the result, $\Gamma$. Each resulting
assignment via sigma points, $\Gamma_{s,i}$, $i=0, \cdots, 2L$, is a binary matrix, whereas the result after inverse Unscented Transform, $\Gamma_s = \sum_{i=0}^{2L} \omega_i \Gamma_{s,i}$, would be a non-binary matrix, thereby a policy will be needed to interpret $\Gamma_s$ as an executable result.

We propose a heuristic interpretation policy using the information theory [62] to generate a binary assignment matrix. The proposed policy is as follows:

1) Take the diagonal elements in $P_r$, 
$$\{p_{11}, p_{22}, \cdots, p_{m^2, m^2}\},$$ 
and place them in an $m \times m$ squared uncertainty matrix as
$$\Sigma_s = \begin{bmatrix}
p_{11} & p_{m+1, m+1} & \cdots & p_{m^2-m+1, m^2-m+1} 
p_{22} & p_{m+2, m+2} & \cdots & p_{m^2-m+2, m^2-m+2} 
\vdots & \vdots & \ddots & \vdots 
p_{m,m} & p_{2m,2m} & \cdots & p_{m^2, m^2}
\end{bmatrix},$$ (15)

2) Define a weighted inverse assignment matrix as
$$Q = \begin{bmatrix}
p_{11} & \frac{p_{m+1, m+1}}{\Gamma_s[1,1]} & \cdots & \frac{p_{m^2-m+1, m^2-m+1}}{\Gamma_s[1,m]}
p_{22} & \frac{p_{m+2, m+2}}{\Gamma_s[2,2]} & \cdots & \frac{p_{m^2-m+2, m^2-m+2}}{\Gamma_s[2,m]}
\vdots & \vdots & \ddots & \vdots 
p_{m,m} & \frac{p_{2m,2m}}{\Gamma_s[m,m]} & \cdots & \frac{p_{m^2, m^2}}{\Gamma_s[m,m]}
\end{bmatrix},$$ (16)

where $\Gamma_s[i,j]$ denotes the $ij$th element of matrix $\Gamma_s$.

3) An executable assignment matrix is given by
$$\Gamma_f = \text{HUNGARIAN}(Q).$$ (17)

The heuristic of this proposed policy is that the information of each element in the assignment matrix ($\Gamma_f$) is the inverse of the element in the uncertainty matrix ($\Sigma_s$). Each element in the weighted inverse assignment matrix ($Q$) is the inverse of the weighted assignment matrix, which represents the weighted uncertainty of the assignment. Since the Algorithm 1 is going to minimize the overall cost, we apply it onto $Q$ to find the assignment with the minimized overall uncertainty.

IV. SIMULATION RESULTS

To validate the effectiveness of the proposed framework, MATLAB simulations were conducted. In simulation, the locations of four tasks are specified and assumed known with no uncertainty. Four robots are deployed to complete the tasks. The position of each robot is corrupted by Gaussian noise, with a specified mean and variance.

To verify that the proposed framework generates reasonable results in a simple case with an obvious optimal result, as shown in Figure 3, the tasks (blue crosses) are placed at the following coordinates: $T_1 = (9, 14)$, $T_2 = (1, 1)$, $T_3 = (0, 38)$, and $T_4 = (18, 3)$ . The mean values of the $x$ and $y$ coordinates of the robots are as follows: $D_1 = (10, 15)$, $D_2 = (2, 2)$, $D_3 = (0, 40)$, and $T_4 = (20, 4)$. As for the covariances of each distribution, the $x$ and $y$ coordinates are assumed to be uncorrelated, resulting in a diagonal covariance matrix, represented by four circles of different colors. The variances of the $x$ and $y$ coordinates are assumed to be equal, i.e., $\text{var}X = \text{var}Y = 1.25$. Each robot has the same covariance matrix for their position uncertainty. It can be seen that the optimal solution for the task allocation problem that minimizes the overall distance traveled by each robot is to command agents 1 through 4 to travel to tasks 1 through 4, respectively. The result generated by the proposed framework reveals the same allocation.

Next, we tested the proposed framework in a more complicated scenario where positions of the robots and tasks are close to each other. Figure 4 shows one of many simulations in this case. The positions of the tasks are as follows: $T_1 = (5, 5)$, $T_2 = (2.5, 10)$, $T_3 = (10, 5)$, $T_4 = (5, 3)$. The centers of the robot position distributions are as follows respectively: $D_1 = (1, 5)$, $D_2 = (2, 2)$, $D_3 = (9, 9)$, $D_4 = (8, 4)$. Once again, it is assumed that $\text{var}X = \text{var}Y = 1.25$ for each robot’s distribution. The individual samples associated with each uncertainty circle represent realizations of drone positions following a Gaussian distribution. Using the center of each circle as the estimated location of a drone and the size of a circle represents the amount of uncertainty of an estimated location.

Fig. 3: An example simulation run showing the locations of robots and tasks far away from each other without overlaps of uncertainty circles. The center of a circle represents the estimated location of a drone and the size of a circle represents the amount of uncertainty of an estimated location.


Fig. 4: An example simulation run showing the locations of robots and tasks are close to each other with overlaps of uncertainty circles. The center of a circle represents the estimated location of a drone and the size of a circle represents the amount of uncertainty of an estimated location.

The element of $\Sigma_r$ represents the uncertainty of the corresponding element in $\Gamma_r$. Using the proposed policy in III-A, the executable assignment matrix is given by

$$
\Gamma_f = \begin{pmatrix}
0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
$$

(21)

Since the actually position of each robot can be any realization in the corresponding color zones, we conducted Monte Carlo simulations to compare the overall cost of the assignments generated by both $\Gamma_0$ and $\Gamma_f$. The result shows that the proposed stochastic task allocation approach generates an assignment with 30% less overall cost than using $\Gamma_0$.

V. CONCLUSION

A novel task-allocation framework is proposed in this paper to address the uncertainty in situational awareness for distributed autonomous robots (DARs). The Sigma-Point samples generated by the Unscented transform are passed through a task-allocation algorithm to generate correspond-
