2013 American Control Conference (ACC)
Washington, DC, USA, June 17-19, 2013

Risk Allocation Strategies for Distributed Chance-Constrained Task Allocation

Sameera S. Ponda, Luke B. Johnson and Jonathan P. How
Laboratory for Information and Decision Systems, MIT

Abstract— **This paper addresses the issue of allocating risk amongst agents in distributed chance-constrained planning algorithms. Building on previous research that extended chance-constrained planning to stochastic multi-agent multi-task missions, this paper presents a framework for risk allocation and proposes several strategies for distributing risk in homogeneous and heterogeneous teams. In particular, the contributions of this work include: proposing risk allocation strategies that exploit domain knowledge of agent score distributions to improve team performance, providing insights about what stochastic parameters affect the allocations and the overall mission score/performance, and providing results showing improved performance over previously published heuristic techniques in environments with given allowable risk thresholds.**

I. Introduction

The use of autonomous systems, such as unmanned aerial and ground vehicles (UAVs/UGVs), has motivated the development of autonomous task allocation and planning methods that ensure spatial and temporal coordination for teams of cooperating agents. The basic planning problem can be formulated as a combinatorial optimization, often involving nonlinear and time-varying system dynamics. For most problems of interest, optimal solution methods are computationally intractable and approximation techniques are regularly employed [1]. Most of these consist of centralized planning approaches, which usually require high bandwidth communications and react slowly to local changes in dynamic environments, motivating the development of distributed algorithms where agents plan individually and coordinate with each other locally [2]. One class of distributed planning algorithms involves using auction algorithms augmented with consensus protocols, which are particularly well suited to developing real-time conflict-free solutions in dynamic environments [3].

A well-known issue associated with autonomous planning is that algorithms rely on underlying system models and parameters which are often subject to uncertainty. This uncertainty can result from many sources including: inaccurate parameters, model simplifications, fundamentally nondeterministic processes (e.g. sensor readings, stochastic dynamics), and dynamic local information changes. As discrepancies between planner models and actual system dynamics increase, mission performance typically degrades. Furthermore, the impact of these discrepancies on the overall plan quality is usually hard to quantify in advance due to nonlinear effects, coupling between tasks and agents, and interdependencies between system constraints. However, if available, uncertainty models can be leveraged to create robust plans that explicitly hedge against the inherent uncertainty. Several stochastic planning strategies have been considered throughout the literature employing various stochastic metrics (e.g. expected value, worst-case performance, CVAR) [4]–[8]. A particular stochastic metric that can be used when low probability of failure is required is the chance-constrained metric [6,7,9], which provides probabilistic guarantees on achievable mission performance given allowable risk thresholds. This work builds upon our previous efforts to extend chance-constrained planning to distributed environments [10]. In particular, this paper proposes a formal approach to allocating the risk among the agents, and derives risk allocation strategies for homogeneous and heterogeneous teams of agents that can be used within a distributed chance-constrained planning framework.

II. Problem Statement

A. Robust Distributed Planning

Given a list of N_a agents and N_t tasks, the robust task assignment problem can be written as follows:

$$
\begin{aligned}
\max_{\mathbf{x}} \quad & \mathbb{M}_{\boldsymbol{\theta}} \left\{ \sum_{i=1}^{N_a} \left(\sum_{j=1}^{N_t} \mathbf{c}_{ij}(\mathbf{x}_i, \boldsymbol{\tau}_i, \boldsymbol{\theta}) \right) \right\} \qquad (1) \\
\text{s.t.} \quad & \sum_{i=1}^{N_\bullet} x_{ij} \leq 1, \quad \forall j \in \{1, \ldots, N_t\} \\
& \mathbf{x} \in \{0,1\}^{N_\bullet \times N_t}, \quad \boldsymbol{\tau} \in \{\mathbb{R}^+ \cup \emptyset\}^{N_\bullet \times N_t}
\end{aligned}
$$

where $\mathbf{x}$, is a set of binary decision variables, x_{ij}, which are used to indicate whether task j is assigned to agent i; $\mathbf{c}_{ij}$ is the reward agent i receives for task j given the agent's overall assignment and parameters; $\boldsymbol{\tau}$ represents a set of task execution times; $\boldsymbol{\theta}$ is the set of stochastic planning parameters, with joint distribution $\mathbf{f}_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, that affect the score calculation; and finally $\mathbb{M}_{\boldsymbol{\theta}}(\cdot)$ represents a generalized stochastic metric acting upon the overall mission score subject to the uncertainty in $\boldsymbol{\theta}$. The objective of Eq. (1) is to find a conflict-free allocation of tasks to agents (no more than one agent per task), that maximizes the stochastic team score $\mathbb{M}_{\boldsymbol{\theta}}(\cdot)$. If the metric $\mathbb{M}_{\boldsymbol{\theta}}(\cdot)$ allows the sum over agents to be extracted, then a distributed version of Eq. (1) can be written as Eq. (2), where each agent i optimizes its own assignment $\mathbf{x}_i$ subject to its own stochastic score $\mathbb{M}_{\boldsymbol{\theta}}(\cdot)$, and the only coupling between agents is given by the conflict-free constraint (which involves a sum over agents, as shown in

S. Ponda and L. Johnson are with the Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA, {sponda, lbj16}@mit.edu

J. P. How is the R. C. Maclaurin Professor of Aeronautics and Astronautics, MIT, Cambridge, MA, jhow@mit.edu

978-1-4799-0178-4/$31.00 ©2013 AACC

Eq. (2)). Given this distributed framework, robust distributed algorithms can be developed to perform task allocation in these stochastic environments [11].

$$
\begin{aligned}
\max_{\mathbf{x}_i} \quad & \mathbb{M}_{\boldsymbol{\theta}} \left\{ \sum_{j=1}^{N_t} \mathbf{c}_{ij}(\mathbf{x}_i, \boldsymbol{\tau}_i, \boldsymbol{\theta}) \right\} \\
\text{s.t.} \quad & \sum_{i=1}^{N_a} x_{ij} \leq 1, \quad \forall j \in \{1, \ldots, N_t\} \\
& \mathbf{x}_i \in \{0,1\}^{N_t}, \quad \boldsymbol{\tau}_i \in \{\mathbb{R}^+ \cup \emptyset\}^{N_t}
\end{aligned}
\tag{2}
$$

B. Chance-Constrained Distributed Planning

Of interest in this work is the chance-constrained stochastic metric, which provides probabilistic guarantees on achievable mission performance given allowable risk thresholds [6, 7,9], and is useful when low probability of mission failure is required. Substituting $\mathbb{M}_{\boldsymbol{\theta}}(\cdot)$ in Eq. (1) with a chance-constrained metric, the objective function becomes,

$$
\begin{aligned}
\max_{\mathbf{x}} \quad & \mathbf{y} \\
\text{s.t.} \quad & \mathbb{P}_{\boldsymbol{\theta}} \left\{ \sum_{i=1}^{N_a} \left(\sum_{j=1}^{N_t} \mathbf{c}_{ij}(\mathbf{x}_i, \boldsymbol{\tau}_i, \boldsymbol{\theta}) \right) \leq \mathbf{y} \right\} \leq \epsilon
\end{aligned}
\tag{3}
$$

Unfortunately, Eq. (3) introduces an additional constraint to the optimization that couples agents through a probabilistic chance constraint. Using this metric, the sum over agents cannot be easily extracted from the optimization, making distributed implementations nontrivial [3,11]. In previous work [10], we proposed a distributed chance-constrained approximation of the form,

$$
\begin{aligned}
\max_{\mathbf{x}_i} \quad & \mathbf{y}_i \\
\text{s.t.} \quad & \mathbb{P}_{\boldsymbol{\theta}} \left\{ \left(\sum_{j=1}^{N_t} \mathbf{c}_{ij}(\mathbf{x}_i, \boldsymbol{\tau}_i, \boldsymbol{\theta}) \right) \leq \mathbf{y}_i \right\} \leq \epsilon_i
\end{aligned}
\tag{4}
$$

where each agent i solved its own chance-constrained optimization subject to an individual risk threshold ϵ_i. The approximate chance-constrained mission score was then given by the sum over these chance-constrained agent scores, and an equivalence between the two problem formulations was obtained if the individual agent risks ϵ_i satisfied the following *risk constraint*,

$$
\sum_{i=1}^{N_a} \mathbf{F}_{\mathbf{z}_i}^{-1}(\epsilon_i) = \mathbf{F}_{\mathbf{z}}^{-1}(\epsilon) \tag{5}
$$

where ϵ is the mission risk, ϵ_i, $\forall i$ are the individual agent risks, $\mathbf{z}_i = \sum_{j=1}^{N_t} \mathbf{c}_{ij}(\mathbf{x}_i, \boldsymbol{\tau}_i, \boldsymbol{\theta})$, $\forall i$ are the random agent scores subject to the uncertainty in $\boldsymbol{\theta}$ (with distributions $\mathbf{f}_{\mathbf{z}_i}(\mathbf{z}_i)$ and CDFs $\mathbf{F}_{\mathbf{z}_i}(\cdot)$, $\forall i$), and $\mathbf{z} = \sum_{i=1}^{N_a} \mathbf{z}_i$ is the random mission score (with distribution $\mathbf{f}_{\mathbf{z}}(\mathbf{z})$ and CDF $\mathbf{F}_{\mathbf{z}}(\cdot)$). Eq. (5) identifies the relationship between mission risk and agent risks given available distributions for both agent and mission scores, however, in chance-constrained planning problems it is difficult to predict what these distributions will be a priori.

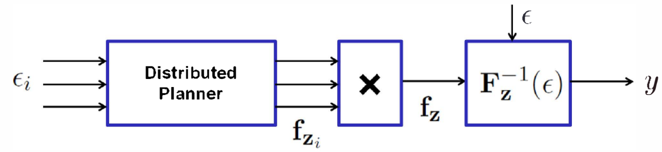


Fig. 1. Relationship between agent risks and chance-constrained score in distributed chance-constrained planning. The main pieces include the distributed planner, which uses the risk allocations to make agent plans, a convolution block that combines the agent score distributions associated with the agent plans to derive the mission score distribution, and a final block that computes the chance-constrained mission score given the score distribution and the allowable mission risk threshold.

There are two main goals associated with allocating risks ϵ_i amongst the agents: (1) to ensure that the global mission risk level is adequately captured (see Eq. (5)), and (2) to find risk allocations that encourage agents to pick "better" plans, such that the chance-constrained mission score $\mathbf{y} = \mathbf{F}_{\mathbf{z}}^{-1}(\epsilon)$ is maximized. This involves finding a distribution for the mission score $\mathbf{z}$ that maximizes $\mathbf{F}_{\mathbf{z}}^{-1}(\epsilon)$ given an allowable risk threshold ϵ. However, as illustrated in Fig. 1, $\mathbf{f}_{\mathbf{z}}(\mathbf{z})$ is a function of the agent score distributions $\mathbf{f}_{\mathbf{z}_i}(\mathbf{z}_i)$ (e.g. convolution if agents are independent), and the distributions $\mathbf{f}_{\mathbf{z}_i}(\mathbf{z}_i)$ are in turn functions of the risk levels ϵ_i and of the inner workings of the planner, which are hard to predict. This severe coupling makes the task of optimizing risk allocations ϵ_i in order to achieve the best plan very difficult. This work presents a formal approach to risk allocation and proposes risk allocation strategies for both homogeneous and heterogeneous teams. Results are provided comparing the performance of these strategies, along with insights into what features affect the chance-constrained team performance in distributed environments.

III. Risk Allocation Strategies

Previous work proposed a heuristic risk allocation strategy based on simplified assumptions – Gaussian distributions and identical agent risk allocations – to set the agent risks ϵ_i given a team of heterogeneous agents [10]. This paper presents a formal approach to risk allocation, and proposes several more practical risk allocation strategies for both homogeneous and heterogeneous teams of agents. In particular, the contributions of this work include: (1) presenting a general framework for homogeneous and heterogeneous risk allocation based on Eq. (5), (2) exploiting domain knowledge of agent score distributions to improve performance through risk allocation for both homogeneous and heterogeneous teams (with nonidentical risks), and (3) providing insights on how to compare the different risk allocation strategies, what parameters affect these allocations, and what features affect the performance of the overall chance-constrained mission score given the distributed approximation of Eq. (4).

A. Homogeneous Risk Allocation

The first case considered is for teams of homogeneous agents, where all N_a agents are assumed to have identical score distributions $\mathbf{f}_{\mathbf{z}_i}(\mathbf{z}_i)$, and identical risk allocations ϵ_i. Using these assumptions in Eq. (5) gives the general expression for homogeneous risk allocation,

$$\epsilon_i = \mathbf{F}_{\mathbf{z}_i}\left(\frac{1}{N_a}\mathbf{F}_{\mathbf{z}}^{-1}(\epsilon)\right) \tag{6}$$

Specifying the mission distribution $\mathbf{f}_{\mathbf{z}}(\mathbf{z})$ involves convolving the N_a agent score distributions, which is often analytically intractable. This work uses a Gaussian distribution to approximate the mission score (sum of agent scores) invoking the Central Limit Theorem[1], where $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2)$ with $\mu = N_a\mu_i$ and $\sigma^2 = N_a\sigma_i^2$. For Gaussian random variables, the CDF and its inverse are given by,

$$\begin{aligned} \mathbf{F}_X(x) &= \frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{x-\mu}{\sqrt{2\sigma^2}}\right) \\ \mathbf{F}_X^{-1}(\epsilon) &= \mu + \sqrt{2\sigma^2}\,\mathrm{erf}^{-1}(2\epsilon - 1) \end{aligned} \tag{7}$$

and substituting into Eq. (6) gives,

$$\epsilon_i = \mathbf{F}_{\mathbf{z}_i}\left(\mu_i + \sqrt{\frac{2}{N_a}}\sigma_i\,\mathrm{erf}^{-1}(2\epsilon - 1)\right) \tag{8}$$

Eq. (8) can be used with many different forms of the agent distributions $\mathbf{f}_{\mathbf{z}_i}(\mathbf{z}_i)$. This work explores three different distributions for homogeneous agents based on Gaussian, exponential and gamma distributions. Fig. 2 illustrates the shapes of the distributions used in these strategies and derivation details are provided in the following sections.

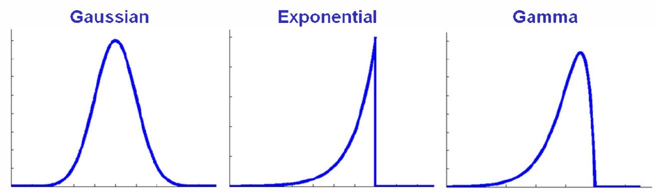
1) Gaussian Risk Allocation: This first risk allocation strategy assumes Gaussian agent scores $\mathbf{z}_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ with mean μ_i and variance σ_i^2. Replacing $\mathbf{F}_{\mathbf{z}_i}(\cdot)$ in Eq. (8) with a Gaussian distribution gives the following agent risks,

$$\begin{aligned} \epsilon_i &= \frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{\mu_i + \sqrt{\frac{2}{N_a}}\sigma_i\,\mathrm{erf}^{-1}(2\epsilon-1) - \mu_i}{\sqrt{2}\sigma_i}\right) \\ &= \frac{1}{2}\left(1 + \mathrm{erf}\left(\sqrt{\frac{1}{N_a}}\,\mathrm{erf}^{-1}(2\epsilon - 1)\right)\right) \end{aligned} \tag{9}$$

This Gaussian risk allocation will only be accurate when agent scores approach Gaussian distributions, however, several scenarios of interest involve nonsymmetric agent distributions. For example, the scenarios explored in this work consist of time-critical missions, where arriving at tasks early or on time results in maximum task rewards, and arriving late yields diminishing rewards. As a result, agent score distributions tend to have higher probability masses concentrated around sums of maximum task rewards, and diminishing tails towards lower scores, motivating the use of nonsymmetric agent distributions within the risk allocation. This work uses gamma and exponential distributions (flipped about the vertical axis and shifted), as shown in Fig. 2.

2) Exponential Risk Allocation: Flipping about the vertical axis and shifting distributions involves applying a linear transformation, $Y = aX + b$, where $a = -1$ (flip) and b is some quantity corresponding to the shift. The CDF of the transformed random variable $\mathbf{F}_Y(y)$ can be computed given the original CDF $\mathbf{F}_X(x)$ as follows,

[1]Note that using the Central Limit Theorem assumes agents are independent. This assumption is valid for agent specific stochastic parameters (e.g. agent velocities), and is also true for task specific parameters (e.g. task service times) assuming conflict-free solutions since no two agents are assigned to the same task



(a) Gaussian Risk (b) Exponential Risk (c) Gamma Risk

Fig. 2. Agent score distributions for three different homogeneous risk allocation strategies.

$$\mathbf{F}_Y(y) = \begin{cases} \mathbf{F}_X\left(\frac{y-b}{a}\right), & a > 0 \\ 1 - \mathbf{F}_X\left(\frac{y-b}{a}\right), & a < 0 \end{cases} \tag{10}$$

For an exponential distribution with parameter λ, the CDF, mean, and variance of the original and transformed distributions are given by,

$$\begin{aligned} \mathbf{F}_X(x) &= 1 - e^{-\lambda x} & & \mathbf{F}_Y(y) = e^{-\lambda(b-y)} \\ \mu_X &= \tfrac{1}{\lambda} & \Rightarrow \quad & \mu_Y = \tfrac{-1}{\lambda} + b \\ \sigma_X^2 &= \tfrac{1}{\lambda^2} & & \sigma_Y^2 = \tfrac{1}{\lambda^2} \end{aligned} \tag{11}$$

which can be used in Eq. (8) to derive the following expression for agent risks,

$$\begin{aligned} \epsilon_i &= e^{-\lambda\left(b - \mu_i - \sqrt{\frac{2}{N_a}}\sigma_i\,\mathrm{erf}^{-1}(2\epsilon-1)\right)} \\ &= e^{-\lambda\left(b + \frac{1}{\lambda} - b - \sqrt{\frac{2}{N_a}}\left(\frac{1}{\lambda}\right)\mathrm{erf}^{-1}(2\epsilon-1)\right)} \\ &= e^{-\left(1 - \sqrt{\frac{2}{N_a}}\,\mathrm{erf}^{-1}(2\epsilon-1)\right)} \end{aligned} \tag{12}$$

Although this exponential-based distribution is nonsymmetric and captures properties of agent scores better than the Gaussian risk allocation, the shape of the distribution is fixed (the *scale* can be controlled through the parameter λ but the *shape* is fixed). In some situations, it is preferable to use a gamma distribution instead, since it provides more control over the shape of the distribution as well as the scale through the additional parameter p.

3) Gamma Risk Allocation: For this risk allocation, a linear transformation with $a = -1$ and shift value b is again applied, this time to the gamma distribution. Eq. (10) with $a < 0$ can again be used to derive the transformed CDF, and for a gamma distribution with parameters p and θ (controlling the shape and scale respectively), the CDF, mean, and variance of the original and transformed random variables are given by,

$$\begin{aligned} \mathbf{F}_X(x) &= \tfrac{1}{\Gamma(p)}\gamma\left(p, \tfrac{x}{\theta}\right) & & \mathbf{F}_Y(y) = 1 - \tfrac{1}{\Gamma(p)}\gamma\left(p, \tfrac{b-y}{\theta}\right) \\ \mu_X &= p\theta & \Rightarrow \quad & \mu_Y = -p\theta + b \\ \sigma_X^2 &= p\theta^2 & & \sigma_Y^2 = p\theta^2 \end{aligned} \tag{13}$$

where $\Gamma(p) = \int_0^\infty e^{-t}t^{p-1}dt$ is the gamma function and $\gamma(p, x) = \int_0^x e^{-t}t^{p-1}dt$ is the incomplete gamma function. These can be used in Eq. (8) to obtain the following expression for agent risks,

$$\begin{aligned} \epsilon_i &= 1 - \frac{1}{\Gamma(p)}\gamma\left(p, \frac{1}{\theta}\left(b - \mu_i - \sqrt{\frac{2}{N_a}}\sigma_i\,\mathrm{erf}^{-1}(2\epsilon - 1)\right)\right) \\ &= 1 - \frac{1}{\Gamma(p)}\gamma\left(p, p - \sqrt{\frac{2p}{N_a}}\,\mathrm{erf}^{-1}(2\epsilon - 1)\right) \end{aligned} \tag{14}$$

In the case where $p = 1$ the gamma distribution and the exponential distribution are equivalent (where θ is related to λ by $\theta = 1/\lambda$), and thus the gamma and exponential risk allocation strategies return the same values for ϵ_i.

Note that in all the homogeneous risk allocation expressions, the agent risk values are not affected by the shift and scale parameters of the distributions (e.g. μ and σ in the Gaussian case, b and λ for the exponential, and b and θ in the gamma case). Since the mission distribution is a function of the agent distributions, means and variances appear on both sides of Eq. (5) in equal magnitudes and thus cancel out, resulting in constant risk allocations. The intuition behind this observation is that the risk allocation process is affected by the *shape* of the underlying distributions (particularly the tails), and not the scale and shift parameters.

B. Heterogeneous Risk Allocation

Setting risk values for heterogeneous agents is more complex, since the assumptions made in Eq. (6) may no longer hold (i.e. identical distributions, identical risks). In general, there are infinite possible combinations of ϵ_i that are valid solutions to Eq. (5) given a specific value of ϵ, therefore specifying different individual agent risks is nontrivial. In previous work [10], we presented a heuristic risk allocation strategy for heterogeneous agents that assumed Gaussian distributions and identical risk allocations (same $\epsilon_i, \forall i$), which derived from Eq. (5) gives,

$$\epsilon_i = \frac{1}{2}\left(1 + \operatorname{erf}\big(\mathbf{H}\operatorname{erf}^{-1}(2\epsilon - 1)\big)\right), \quad \mathbf{H} = \frac{\sqrt{\sum_{i=1}^{N_a}\sigma_i^2}}{\sum_{i=1}^{N_a}\sigma_i} \qquad (15)$$

where the constant $\mathbf{H}$ represents team heterogeneity with regards to variance in agents' scores. With homogeneous agents, $\mathbf{H} = \sqrt{1/N_a}$ and Eq. (15) is equivalent to the homogeneous Gaussian risk allocation (Eq. (9)). On the other hand, if the variance of the mission distribution was only due to one agent (all other agents deterministic), then $\mathbf{H} = 1$ and $\epsilon_i = \epsilon$ as expected. Thus selecting $\mathbf{H} \in \left[\frac{1}{\sqrt{N_a}}, 1\right]$ was used in [10] to capture team heterogeneity.

This work proposes a more general risk allocation strategy for heterogeneous teams, which involves assigning *different* risk values ϵ_i to different types of agents, lifting the identical risk assumption. This strategy assumes that agents can be grouped into K types, where agents in group k have the same distribution $\mathbf{f}_{\mathbf{z}_k}(\mathbf{z}_k)$, and are assigned identical risk allocations ϵ_k. Given these assumptions, Eq. (5) becomes

$$\sum_{k=1}^{K} N_k\ \mathbf{F}_{\mathbf{z}_k}^{-1}(\epsilon_k) = \mathbf{F}_{\mathbf{z}}^{-1}(\epsilon) \qquad (16)$$

where N_k is the number of agents of type $k \in \{1, \dots, K\}$. Since Eq. (16) is analytically intractable for most distribution types, the Central Limit Theorem is again leveraged and a Gaussian mission score distribution with $\mu = \sum_{k=1}^{K} N_k\mu_k$ and $\sigma^2 = \sum_{k=1}^{K} N_k\sigma_k^2$ is assumed, giving the following approximation to Eq. (16),

$$\sum_{k=1}^{K} N_k\left(\mathbf{F}_{\mathbf{z}_k}^{-1}(\epsilon_k) - \mu_k\right) = \sqrt{2\sigma^2}\operatorname{erf}^{-1}(2\epsilon - 1) \qquad (17)$$

Even if the agent distributions are available, an issue that still remains is that Eq. (17) involves solving for K unknown variables ϵ_k, for which there are infinite possible solutions. To address this issue, we propose dividing the right hand side of Eq. (17) into shares m_k (with $\sum_{k=1}^{K} m_k = 1$), giving the following set of equations for the different agent types k,

$$N_k\left(\mathbf{F}_{\mathbf{z}_k}^{-1}(\epsilon_k) - \mu_k\right) = m_k\sqrt{2\sigma^2}\operatorname{erf}^{-1}(2\epsilon - 1), \quad \forall k \qquad (18)$$

with associated risk allocations,

$$\epsilon_k = \mathbf{F}_{\mathbf{z}_k}\left(\mu_k + \frac{m_k}{N_k}\sqrt{2\sigma^2}\operatorname{erf}^{-1}(2\epsilon - 1)\right) \qquad (19)$$

Using Eq. (19) typically gives different risk allocations ϵ_k for the different agent types k, depending on the selected shares m_k and the statistics of each agent group. As in the previous section, Eq. (19) can be used with the agent score distributions illustrated in Fig. 2 and described in Eqs. (7), (11), and (13) to give the following heterogeneous risk allocations[2]. For the Gaussian case Eq. (19) becomes,

$$\epsilon_k = \frac{1}{2}\left(1 + \operatorname{erf}\big(\mathbf{H}_k\operatorname{erf}^{-1}(2\epsilon - 1)\big)\right), \quad \mathbf{H}_k = \frac{m_k}{N_k}\frac{\sigma}{\sigma_k} \qquad (20)$$

for the exponential case,

$$\epsilon_k = e^{-\left(1 - \mathbf{H}_k\operatorname{erf}^{-1}(2\epsilon - 1)\right)}, \quad \mathbf{H}_k = \frac{m_k}{N_k}\frac{\sigma}{\sigma_k}\sqrt{2}$$

and for the gamma case,

$$\epsilon_k = 1 - \frac{1}{\Gamma(p)}\gamma\left(p, p - \mathbf{H}_k\operatorname{erf}^{-1}(2\epsilon - 1)\right), \quad \mathbf{H}_k = \frac{m_k}{N_k}\frac{\sigma}{\sigma_k}\sqrt{2p}$$

A final remaining issue involves determining how to partition the shares m_k to get risk allocations for the team that optimize the chance-constrained mission score, a non-trivial endeavor given the complexity alluded to in Fig. 1. This work explores a few different strategies to allocate the shares m_k which are described next for the Gaussian case (similar results can be obtained for the exponential and gamma cases).

1) Equal Shares: Assuming equal shares for all agents, $m_k \propto N_k$, which can be substituted into Eq. (20) to give,

$$m_k = \frac{N_k}{\sum_{k=1}^{K} N_k} \quad \Rightarrow \quad \mathbf{H}_k = \frac{\sigma}{N_a}\frac{1}{\sigma_k} \qquad (21)$$

The first term of $\mathbf{H}_k$ involves statistics affecting all agents equally, whereas the second term is affected by σ_k for agents of type k. As a result, the risk allocations ϵ_k for the different agent types will typically be different.

2) Shares Proportional to Variance: The second case assumes that the shares are proportional to the group variance, $m_k \propto N_k\sigma_k^2$, where substituting into Eq. (20) gives,

$$m_k = \frac{N_k\sigma_k^2}{\sum_{k=1}^{K} N_k\sigma_k^2} \quad \Rightarrow \quad \mathbf{H}_k = \frac{1}{\sigma}\sigma_k \qquad (22)$$

Again the second term is affected by σ_k, so the risk allocations ϵ_k for the different agent types will typically differ.

[2]Note that in these heterogeneous risk allocation strategies, agent risks are independent of the means of the distributions and are only functions of the relative (normalized) variances between agents. This is similar to the observation made for the homogeneous strategies, where the shift and scale parameters did not affect the risk allocation, but only the shape parameters came into play.

3) Shares Proportional to Standard Deviation: The last strategy assumes shares proportional to the group's standard deviation, $m_k \propto \sqrt{N_k \sigma_k^2}$, where $\mathbf{H}_k$ becomes,

$$m_k = \frac{\sqrt{N_k \sigma_k^2}}{\sum_{k=1}^{K} \sqrt{N_k \sigma_k^2}} \quad \Rightarrow \quad \mathbf{H}_k = \frac{\sigma}{\sum_{k=1}^{K} \sqrt{N_k \sigma_k^2}} \frac{1}{\sqrt{N_k}}$$

In this special case, the risk allocations are not affected by each group's variance. The first term of $\mathbf{H}_k$ is constant for all agents, and the second term is only affected by the number of agents in the group. In particular, if each group has the same number of agents, the risk allocations for the team will be identical, even if agents have different variances. Furthermore, taking each agent to be its own type (i.e. $N_k = 1, \forall k$, and $K = N_a$), the expression for $\mathbf{H}_k$ is equivalent to $\mathbf{H}$ in Eq. (15), therefore the risk heuristic proposed in [10] is a specific case of this more general risk allocation framework, where score distributions are Gaussian and shares are proportional to agent standard deviation.

IV. Distributed Chance-Constrained Algorithmic Framework

Once risks have been allocated amongst the agents, a distributed algorithm can be used to solve Eq. (4), where each agent i solves its own chance-constrained optimization subject to its individual risk threshold ϵ_i, and deconflicts assignments with other agents through local communications. This work uses a robust extension to the Consensus-Based Bundle Algorithm (CBBA), a polynomial-time distributed auction algorithm that provides provably good real-time approximate solutions for multi-agent multi-task allocation problems [3]. CBBA consists of iterations between two phases: a bundle building phase where each agent greedily selects a set of tasks to execute, and a consensus phase where conflicting assignments are resolved through local communications with neighboring agents. In recent work, CBBA was extended to account for stochastic metrics within the distributed planning framework [11]. The Robust CBBA algorithm is summarized in Alg. 1. The data structures in $\mathcal{A}_i$ consist of internal decisions made by each agent (including a path sorted by task execution order $\mathbf{p}_i$, and a task bundle sorted in order of task assignment $\mathbf{b}_i$). The data structures in $\mathcal{C}_i$ involve data to be communicated between agents (i.e. a winning agent list $\mathbf{z}_i$, the corresponding winning bid list $\mathbf{y}_i$, and a set of communication timestamps $\mathbf{t}_i$ (for more details see [3,11,12])). Alg. 2 describes the process by which agents decide to add tasks to their current bundle $\mathbf{b}_i$. The main algorithmic steps are as follows: for each available task j, compute the best stochastic score for adding the task to the current path $\mathbf{p}_i$, along with the corresponding best location $n_j^\star$ for task j in the path (line 3); compute the improvement in score and corresponding bid information for communication with other agents (lines 4-6, see [12] for details); select the task that leads to the largest improvement in score (line 8), and add it to the current set of assignments (lines 10-13); repeat until no profitable tasks are available or until a maximum path length L_i has been reached.

Algorithm 1 CBBA($\mathcal{I}, \mathcal{J}$)

Initialize $\{\mathcal{A}_i, \mathcal{C}_i\}, \quad \forall i \in \mathcal{I}$
while $\neg$*converged* **do**
  $(\mathcal{A}_i, \mathcal{C}_i) \leftarrow$ CBBA-BUNDLE-ADD$(\mathcal{A}_i, \mathcal{C}_i, \mathcal{J}), \quad \forall i \in \mathcal{I}$
  $(\mathcal{A}_i, \mathcal{C}_i) \leftarrow$ CBBA-CONSENSUS$(\mathcal{A}_i, \mathcal{C}_i, \mathcal{C}_{N_i}), \quad \forall i \in \mathcal{I}$
  $converged \leftarrow \bigwedge_{i \in \mathcal{I}}$ CHECK-CONVERGENCE$(\mathcal{A}_i)$
end while
$\mathcal{A} \leftarrow \bigcup_{i \in \mathcal{I}} \mathcal{A}_i$
return $\mathcal{A}$

Algorithm 2 CBBA-BUNDLE-ADD($\mathcal{A}_i, \mathcal{C}_i, \mathcal{J}$)

1: **while** $|\mathbf{p}_i| \leq L_i$ **do**
2:   **for** $j \in \mathcal{J} \setminus \mathbf{p}_i$ **do**
3:     $J_{(\mathbf{p}_i \oplus_{n_j^\star} j)} \leftarrow \max_{n_j}$ CC-PATH-SCORE$(\mathbf{p}_i \oplus_{n_j} j)$
4:     $\Delta J_{ij}(\mathbf{p}_i) = J_{(\mathbf{p}_i \oplus_{n_j^\star} j)} - J_{\mathbf{p}_i}$
5:     $s_{ij} = \min(\Delta J_{ij}(\mathbf{p}_i), y_{ik}), \quad \forall k \in \mathbf{p}_i$
6:     $h_{ij} = \mathbb{I}(s_{ij} > y_{ij})$
7:   **end for**
8:   $j^\star = \operatorname{argmax}_{j \notin \mathbf{p}_i} \ \Delta J_{ij}(\mathbf{p}_i)\ h_{ij}$
9:   **if** $(\Delta J_{ij^\star}(\mathbf{p}_i)\ h_{ij^\star} > 0)$ **then**
10:     $\mathbf{b}_i \leftarrow (\mathbf{b}_i \oplus_{\text{end}} j^\star)$
11:     $\mathbf{p}_i \leftarrow (\mathbf{p}_i \oplus_{n_{j^\star}} j^\star)$
12:     $z_{ij^\star} \leftarrow i$
13:     $y_{ij^\star} \leftarrow s_{ij^\star}$
14:   **else**
15:     **break**
16:   **end if**
17: **end while**
18: $\mathcal{A}_i \leftarrow \{\mathbf{b}_i, \mathbf{p}_i\}$
19: $\mathcal{C}_i \leftarrow \{\mathbf{z}_i, \mathbf{y}_i, \mathbf{t}_i\}$
20: **return** $(\mathcal{A}_i, \mathcal{C}_i)$

One complication with evaluating stochastic path scores (as required by Alg. 2, line 3), is that for every realization of the uncertain planning parameters, decisions about optimal task execution times need to be made. In general, given infinite support of the uncertain parameters, this would involve an uncountable number of optimizations. Alg. 3 presents a sampling approximation to the chance-constrained path score that can be used to maintain analytical tractability given this issue regarding optimization of task execution times. The main algorithmic steps involve: selecting N representative samples $\boldsymbol{\theta}_k$ with corresponding probabilistic weights w_k (lines 1-2); computing the optimal execution times $\boldsymbol{\tau}_i^\star$ and corresponding path score $J_{\mathbf{p}_i}^k$ for each sample value of the planning parameters $\boldsymbol{\theta}_k$ (lines 3-6); constructing a discrete approximation to the score distribution by sorting the score samples (with associated weights) in ascending order (line 7); and computing an approximate chance-constrained score given this discrete distribution and the allowable risk threshold ϵ_i (line 8). The next section provides results comparing the different risk allocation strategies presented in this paper for setting the risks ϵ_i given time-critical mission scenarios.

V. Results and Discussion

The distributed chance-constrained CBBA algorithm was implemented in simulation for time-critical UAV target track-

Algorithm 3 CC-PATH-SCORE($\mathbf{p}_i$)

1: $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\} \sim \mathbf{f}(\boldsymbol{\theta})$

2: $\{w_1, \ldots, w_N\} \leftarrow \{w_1, \ldots, w_N\} / \sum_{k=1}^{N} w_k$

3: **for** $k \in \{1, \ldots, N\}$ **do**

4: $\quad \boldsymbol{\tau}_i^\star = \underset{\boldsymbol{\tau}_i}{\operatorname{argmax}} \sum_{j=1}^{N_t} \mathbf{c}_{ij}(\tau_{ij}(\mathbf{p}_i), \boldsymbol{\theta}_k)\, x_{ij}$

5: $\quad J_{\mathbf{p}_i}^k = \sum_{j=1}^{N_t} \mathbf{c}_{ij}(\tau_{ij}^\star(\mathbf{p}_i), \boldsymbol{\theta}_k)\, x_{ij}$

6: **end for**

7: $\left(J_{\mathbf{p}_i}^{\bar{k}}, w_{\bar{k}}\right) \leftarrow$ SORT-SCORES$\left(J_{\mathbf{p}_i}^k, w_k\right)$

8: $\bar{k}^\star = \underset{\bar{k} \in \{1,\ldots,N\}}{\operatorname{argmax}} \left\{ J_{\mathbf{p}_i}^{\bar{k}} \;\middle|\; \sum_{i=1}^{\bar{k}} w_i \leq \epsilon_i \right\}$

9: **return** $(J_{\mathbf{p}_i}^{\bar{k}^\star})$

ing missions. The objective function for each agent was defined as,

$$J_i = \sum_{j=1}^{N_t} \mathbf{R}_{ij}(\tau_{ij})\, x_{ij} - f_i\, d_i(\mathbf{p}_i) \tag{23}$$

where $d_i(\mathbf{p}_i)$ is the distance traveled by agent i given path $\mathbf{p}_i$ and f_i is the fuel cost per unit distance. The time-critical task rewards are given by,

$$\mathbf{R}_{ij}(\tau_{ij}) = \begin{cases} \mathbf{R}_j\, e^{-\lambda_j \Delta\tau_{ij}}, & t_{j_{\text{start}}} \leq \tau_{ij} \leq t_{j_{\text{end}}} \\ -\mathbf{R}_j, & \text{otherwise} \end{cases}$$

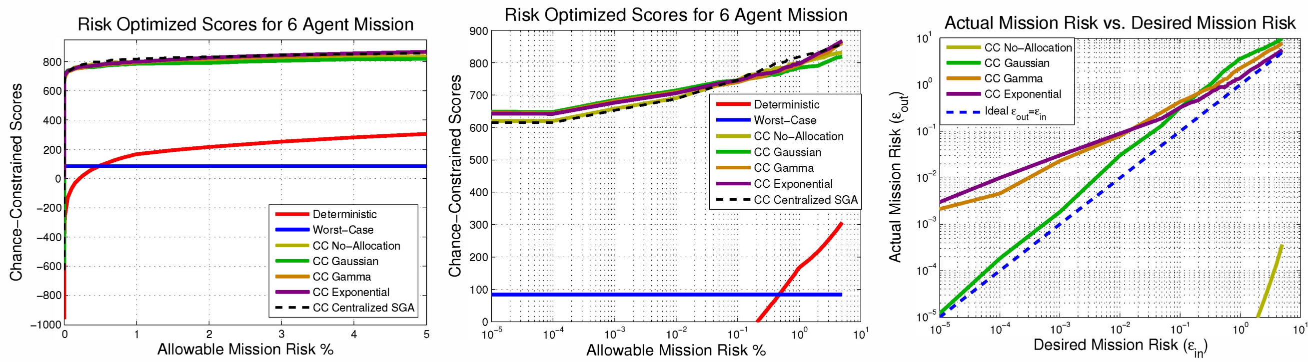
where $\mathbf{R}_j$ is obtained if the task is done on time, an exponential discount penalizes late tasks according to delay $\Delta\tau_{ij} = \max\{0, \tau_{ij} - (t_{j_{\text{start}}} + \bar{t}_{j_{\text{dur}}})\}$, and negative reward $-\mathbf{R}_j$ is incurred for failing to do the task within the time-window. The actual task durations $t_{j_{\text{dur}}}$ were considered random variables sampled from gamma distributions with mean $\bar{t}_{j_{\text{dur}}}$. Three types of tasks were defined: high-reward high-uncertainty tasks, medium-reward tasks with low uncertainty, and low reward tasks but with deterministic service times (same mean duration for all tasks). Two types of teams were considered: homogeneous UAVs with uncertain velocities (uniform distribution over speed), and heterogeneous UAV teams where half the team consisted of fast but unpredictable agents (high mean and variance), and the rest involved slower speed but more predictable agents (lower mean and variance).

Fig. 3 shows Monte Carlo simulation results comparing chance-constrained mission performance for a homogeneous team. The following 7 planning algorithms were compared: a deterministic algorithm (using mean values of parameters), an algorithm optimizing worst-case performance, the chance-constrained CBBA algorithm without explicit risk allocation (all agents planned with mission risk, $\epsilon_i = \epsilon, \forall i$, which is typically conservative), chance-constrained CBBA using the different homogeneous risk allocation strategies (Gaussian, Exponential and Gamma), and a *centralized* chance-constrained sequential greedy algorithm (SGA). The chance-constrained mission scores as a function of mission risk are shown on a linear scale (Fig. 3(a)) and a log scale (Fig. 3(b)) to highlight performance at low risk levels. The 3 risk allocation strategies achieved higher performance than without risk allocation, with Exponential risk performing best on average. At low risk levels, Gaussian risk gave good performance but as the risk level increased the approximation became worse. All chance-constrained planning approaches performed significantly better than deterministic and worst-case planning which did not account for risk. Fig. 3(c) shows the achieved team risk corresponding to the given agent risk allocations ϵ_i, where the dotted line represents a perfect match between desired and actual mission risk. Without risk allocation the team performs conservatively, achieving much lower mission risk than allowed, thus sacrificing performance. With the risk allocation methods, the team is able to more accurately predict the mission risk, where closer matches led to higher scores. Finally, chance-constrained CBBA achieved performance on par with the centralized sequential greedy approach, validating the distributed approximation.

Fig. 4 shows results for a heterogeneous stochastic mission where the following 8 planning algorithms were compared: deterministic, worst-case, chance-constrained CBBA without risk allocation, chance-constrained CBBA using the risk allocation heuristic proposed in [10] with $\mathbf{H} = \sqrt{2/N_a}$, chance-constrained CBBA using the heterogeneous Gaussian risk allocation strategies (equal shares, shares based on variance, shares based on std. dev.), and the centralized SGA algorithm. All chance-constrained planning approaches did better than the deterministic and worst-case algorithms. The heterogeneous risk allocation strategy proposed in this paper, with shares proportional to std. dev., performed best overall. The heuristic risk allocation of [10] achieved similar performance as well (recall that the two strategies were shown to be equivalent in Sec. III). The other risk allocation approaches performed rather poorly, even though in the equal share case the achieved team risk matched the desired risk well (Fig. 4(c)). The intuition behind these results is that when agent risk allocations were severely unequal, some agents developed very aggressive plans whereas others selected plans that were too conservative, without considering the effect on the mission as a whole. As a result, the achieved score distributions were quite different between agents, and the convolved mission score distribution yielded lower chance-constrained scores. In general, having a more equitable risk distribution for the team led to higher performing plans. Once again, the performance of CBBA was on par with the centralized approach, validating the distributed approximation.

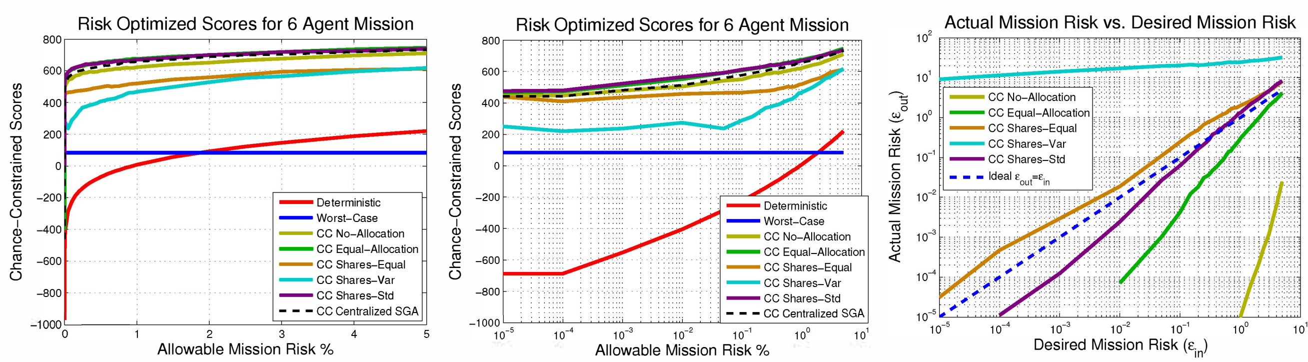
VI. CONCLUSION

This paper proposed a formal approach to allocating risk amongst agents in distributed chance-constrained planning algorithms. Building on previous efforts that extended chance-constrained planning to stochastic multi-agent multi-task missions, this paper presented a generalized framework



(a) Chance-constrained scores (b) Chance-constrained scores (log scale) (c) Achieved mission risk (log scale)

Fig. 3. Monte Carlo results for a stochastic mission with 6 homogeneous agents and 60 tasks.



(a) Chance-constrained scores (b) Chance-constrained scores (log scale) (c) Achieved mission risk (log scale)

Fig. 4. Monte Carlo results for a stochastic mission with 6 heterogeneous agents and 60 tasks.

for risk allocation and proposed several strategies for distributing risk in homogeneous and heterogeneous teams. In particular, the contributions of this work included: proposing risk allocation strategies that exploit domain knowledge of agent score distributions to improve team performance, providing insights about what stochastic parameters affect the allocations and the overall mission score/performance, and providing results showing improved performance over previously published heuristic techniques in environments with given allowable risk thresholds.

ACKNOWLEDGMENTS

This research was supported in part by AFOSR (FA9550-08-1-0086) and MURI (FA9550-08-1-0356). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

REFERENCES

[1] D. Bertsimas and R. Weismantel, *Optimization over integers*. Dynamic Ideas Belmont, MA, 2005.

[2] S. S. Ponda, L. B. Johnson, A. Geramifard, and J. P. How, *Handbook of Unmanned Aerial Vehicles*. Springer, 2012 (to appear), ch. Cooperative Mission Planning for Multi-UAV Teams.

[3] H. Choi, L. Brunet, and J. How, "Consensus-based decentralized auctions for robust task allocation," *Robotics, IEEE Transactions on*, vol. 25, no. 4, pp. 912–926, 2009.

[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. I-II, 3rd Ed.* Belmont, MA: Athena Scientific, 2007.

[5] D. Bertsimas, D. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011.

[6] P. M. Verderame, J. A. Elia, J. Li, and C. A. Floudas, "Planning and scheduling under uncertainty: a review across multiple sectors," *Industrial & engineering chemistry research*, vol. 49, no. 9, pp. 3993–4017, 2010.

[7] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, 2007.

[8] A. Ben-Tal and A. Nemirovski, "Robust solutions of uncertain linear programs," *Operations Research Letters*, vol. 25, no. 1, pp. 1–14, 1999.

[9] E. Delage and S. Mannor, "Percentile optimization for markov decision processes with parameter uncertainty," *Operations research*, vol. 58, no. 1, pp. 203–213, 2010.

[10] S. S. Ponda, L. B. Johnson, and J. P. How, "Distributed chance-constrained task allocation for autonomous multi-agent teams," in *American Control Conference (ACC)*, June 2012.

[11] S. S. Ponda, "Robust Distributed Planning Strategies for Autonomous Multi-Agent Teams," Ph.D. dissertation, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, Sept. 2012.

[12] L. B. Johnson, H.-L. Choi, S. S. Ponda, and J. P. How, "Allowing non-submodular score functions in distributed task allocation," in *IEEE Conference on Decision and Control (CDC)*, Dec 2012.