Analyzing the sensitivity of the optimal assignment in probabilistic multi-robot task allocation

Changjoo Nam and Dylan A. Shell

***Abstract*—We consider multi-robot teams operating in uncertain dynamic settings where the costs used for computing task-allocations are not known exactly. In such cases, the desire to minimize the team's expected cost might need to be curtailed if, in so doing, the risk that results is intolerable. We describe a parameterizable variant of the assignment problem that enables a designer to express such preferences, allowing one to take a risk-averse position if the problem demands it. We consider costs that are random variables, but which need not be independent—a useful setting because it permits one to represent inter-robot couplings. We analyze the sensitivity of assignment optima to particular risk valuations and introduce algorithms that provide an interval for the preference parameter in which all values result in the same optimal assignment. This helps in understanding the effects of risk on the problem, and whether the risk-based model is useful in a given problem domain.**

***Keywords*—Networked Robots; Planning, Scheduling and Coordination**

I. Introduction

Given a team of robots and a set of tasks, multi-robot task allocation (MRTA) is concerned with selecting the best task for each robot to perform. The most common formulation involves, firstly, a cost being estimated for each robot–task pairing, and then the computation of an assignment that minimizes the sum of costs, i.e., the Single-Task robot, Single-Robot task, and Instantaneous Assignment (ST-SR-IA) MRTA problem [1]. When circumstances change or new information comes to light (e.g., the unexpected occurrence of an event) the assignment of robots to tasks may need to be adjusted to reflect these contingencies. By doing so repeatedly and with regularity, the robots can produce dynamic cooperative behavior that befits a team.

The literature on MRTA is expansive but in almost all treatments the estimated costs are scalar values, failing to capture any uncertainty in the states of the robots, or the tasks, or the environment. A further, and also nearly universal, assumption of the ST-SR problems is that the costs are independent, having no interrelationship between values. Practical inefficiencies can result rather easily from ignoring either such interdependencies or uncertainty. Even very straightforward scenarios lead to MRTA problem instances where these assumptions, though standard, are dubious. This paper investigates the problem of optimal assignments when neither of these simplifying assumptions are made.

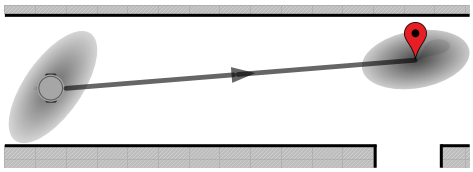


Fig. 1: Cost uncertainty can arise in many ways. For example, owing to position uncertainty of the robot and the task. The sum of two independent normal distributions is also normally distributed, so the distance between a robot and a task is normally distributed. The traveling time is proportional to the distance, so time spent navigating (a useful metric of cost) is also normally distributed.

This paper explores a way to capture specific forms of uncertainty and to incorporate interrelationships between costs because we are interested, generally, in richer cost representations along both dimensions. We consider costs as random variables for which distributional information is available. Such the information could be obtained from robots employing a state estimator or historical measurements from a dataset. For example, the navigation example in Fig. 1 shows position uncertainties represented by 2×2 covariance matrices, output from a Kalman filter. Uncertainty in the robot's pose and its estimate of the task's position mean that traveling time is uncertain too. More specifically, the cost is normally distributed if positions also have normal distributions because the sum of two independent normal distributions is also normal.

In this work, we characterize costs by their mean and Conditional Value-at-Risk (CVaR), the latter is a risk measure suitable for any type of distribution. We describe statistical properties of optimal assignments given the characterizations of costs (i.e., the mean and CVaR) and a precise form of interrelationship in the costs that we can model tractably. We examine an efficient model for computation of optimal assignments subject to a risk preference that determines the relative importance between the mean and CVaR, when there is a tension between them. We show that there is a useful class of assignments that are indifferent to this risk preference. For problems outside of this class, we introduce a fast heuristic algorithm which computes the sensitivity of an optimal assignment to the particular risk preference.

Manuscript received: March 1, 2016; Revised: May 21, 2016; Accepted: June 18, 2016.

This paper was recommended for publication by Editor Nak Young Chong upon evaluation of the Associate Editor and Reviewers' comments. This work was partially supported by National Science Foundation (award numbers IIS-1302393 and IIS-1453652).

Both authors are with the Department of Computer Science and Engineering at Texas A&M University, College Station, Texas, USA. cjnam at cse.tamu.edu

II. Related Work

Assignment problems with random costs are termed *random assignment problems*, an area of extensive research that was first surveyed comprehensively by Burkard and Cela [2] and, more recently, by Krokhmal and Pardalos [3]. Broadly speaking, *exact analysis* studies problems in terms of instance size, n, while *asymptotic analysis* considers $n \to \infty$. These analyses provide the upper and lower bounds, and the expected value of the cost sum of an optimal assignment, as functions of n, allowing one to understand behaviors of the problem according to its size easily. These work has a wide range of applications and gives a useful information to system designing or decision-making process. However, the assumptions of homogeneous distributions and the i.i.d. random variables limit exploring more complex settings where costs are dependent and drawn from various distributional assumptions.

Nikolova and Stier-Moses [4] propose a traffic assignment model incorporating uncertainty by introducing stochastic costs (travel times), which are uncorrelated random variables representing the uncertainty in the time. Each agent chooses a set of routes in a network and has an objective value that is the weighted sum of the expected travel time and the standard deviation of its travel time along the routes. Their work provides equilibria for several versions of routing games. This stochastic cost representation has been successfully used and analyzed in game theoretic perspective of the routing problem, and now we are interested in applying the stochastic representation and the risk-sensitive formulation to MRTA.

The closest work to this paper is that of Ponda et al. [5], who propose a stochastic formulation of task allocation where planning parameters have uncertainties owing to the discrepancies between system models and actual system dynamics. The chance-constrained approach, which maximizes the worst-case cost sum within a risk threshold, is used. The proposed framework allows agents to work in a distributed manner by allocating individual risk threshold based on a global risk threshold. The result shows that their planner outperforms the deterministic and the robust planners for any risk threshold. However, the consequence of a particular value of the global threshold cannot be anticipated directly in terms of the resulting allocation of agents to tasks. Thus, a necessity of analyzing the outcomes of those threshold values arises to consider countermeasures (e.g., reallocations) against the uncertainties.

III. Definitions and preliminaries

This section describes a mathematical formulation of the MRTA problem, and introduces sensitivity analysis of an optimal assignment, which provides a prescribed region of costs where changes within that region do not impair the optimality of the current assignment. At the end, a brief introduction to risk measures is provided.

A. Multi-robot task allocation

The ST-SR-IA MRTA problem can be posed as an Optimal Assignment Problem (OAP). For n robots and n tasks,[1] we assume we are given costs $c_{ij} \in \mathbb{R}^{\geq 0}$ that represent the cost of the i^{th} robot r_i performing the j^{th} task t_j for $i, j \in \{1, \cdots, n\}$. The robots are to be allocated to tasks: let x_{ij} be a binary variable that equals to 0 or 1, where $x_{ij} = 1$ indicates that the r_i performs t_j, and $x_{ij} = 0$ elsewhere. Then a mathematical description of the problem is

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i, \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j, \tag{3}$$

$$0 \leq x_{ij} \leq 1 \qquad \forall \{i, j\}, \tag{4}$$

$$x_{ij} \in \mathbb{Z}^{+} \qquad \forall \{i, j\}. \tag{5}$$

We let $\mathbf{c}$ and $\mathbf{X}^*$ be $n \times n$ matrices representing a cost matrix and an optimal assignment of the problem, respectively.

B. Sensitivity analysis of optimal assignments

The OAP can be relaxed to a linear programming problem (LP) by removing (5).[2] The LP formulation may make use of Sensitivity Analysis (SA) of an optimal assignment to yield a safe region of costs where all costs within the region preserve the current optimality. We provide a brief interpretation of the analysis for the MRTA problems, based on a comprehensive study of Ward and Wendell [7].

An LP problem corresponding to an MRTA problem can have more than one feasible solution. For each feasible solution, the variables x_{ij} $(i, j = 1, \cdots, n)$ can be divided into *basic* variables and *nonbasic* variables. A variable is basic if it corresponds to one of the vectors in the basis, given a feasible basis to a linear-programming problem; the variable is nonbasic otherwise. For each k, an index of a feasible solution, critical region R_k is a set of costs where the MRTA problem has the same optimal assignment for any cost $c \in R_k$. It is defined as

$$R_k = \{c \in \mathbb{R}^{(n^2)} : \mathbf{c}_{N_k} - \mathbf{c}_{J_k} \mathbf{B}_k^{-1} \mathbf{A}_{N_k} \geq 0\}, \tag{6}$$

where J_k and N_k indicate basic and nonbasic variables of the k^{th} feasible solution, respectively. Matrix $\mathbf{B}_k$ and $\mathbf{A}_{N_k}$ are constraint matrices of basic variables and nonbasic variables. And $\mathbf{c}_{J_k}$ and $\mathbf{c}_{N_k}$ are costs of basic and nonbasic variables.

[1] This is without loss of generality, since if the numbers of robots and tasks are not equal, dummy robots or tasks would be inserted to make them equal. The costs of dummies have very large numbers so that they can be naturally excluded from the optimal assignment.

[2] An optimization problem with a linear objective function has only integer solutions if its constraint matrix is totally unimodular [6].

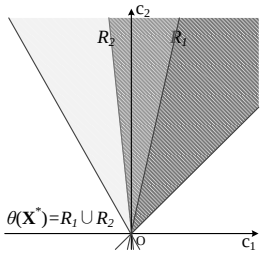


Fig. 2: A 2-D example of $\theta(\mathbf{X}^*)$. Any cost in the set has the same solution.

The critical region R_k is formed by linear boundaries with nonempty interiors.

There is, however, an additional complexity because the MRTA problem is degenerate (see Appendix in [8]) and, consequently, the critical region R_k is not a complete description of the region which preserves optimality of the current assignment. The actual complete set is

$$\theta(\mathbf{X}^*) = \bigcup_{k \in H} R_k, \tag{7}$$

where $H = \{k : \mathbf{X}^*_{J_k} = \mathbf{B}_k^{-1}, \mathbf{X}^*_{N_k} = 0\}$, the union of the critical regions of all degenerate solutions. Note that $\theta(\mathbf{X}^*)$ is also a polyhedral set [7, Theorem 17] with linear boundaries that cross the origin (there might be overlaps among R_k). Fig. 2 shows a two dimensional example of $\theta(\mathbf{X}^*)$ (the smallest nontrivial MRTA problem has four dimensions, the figure is just a pedagogical tool for visualization).

An $n \times n$ MRTA problem has $2n-1$ basic and $(n-1)^2$ nonbasic variables. To compute (7), we must identify the basic and nonbasic variables of the k^{th} feasible solution. The n variables corresponding to costs in the optimal assignment are basic variables, but the degeneracy means that the remaining $n-1$ basic variables cannot be identified directly. Thus, we shall choose the $n-1$ basic variables from the remaining n^2-n variables to complete a feasible solution, yielding a total of $\binom{n^2-n}{n-1}$ feasible solutions. Despite the set being large, the interiors of R_k may overlap so $\theta(\mathbf{X}^*)$ can be covered by a small subset of R_k.

C. Risk measures

In financial mathematics, risk is defined as the variability of the future value of a position [9], and a risk measure maps a set of random variables into the set of real numbers. In [10], variance is employed as the risk measure of a portfolio. However, the variance has been criticized since it treats the positive and negative deviations from the mean identically. For this reason, Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) have been developed.

Given a confidence level $\lambda \in (0,1)$, VaR_λ is the smallest value such that the probability that a loss exceeds the value at most $(1-\lambda)$. Computations of CVaR are not simple (involving integrals), but closed-form expressions of common distributions are provided in [11]. For those distributions with no known closed-form expression of CVaR, one may compute integrals or sample from the distribution and compute the average of the samples greater than or equal to VaR. If one has historical data, CVaR can be computed by fitting a density function to the data or by a nonparametric method by using the data directly. Interested readers are referred to [9].

IV. The Probabilistic Cost Representation

A straightforward way to model uncertainty is to treat the c_{ij} in (1) as random variables, C_{ij}. One may further express the observation that uncertainty of a cost arises from three sources: the robot, task, and the environment associated with the robot and task (e.g., the path between a robot and a task), via statistical properties that express interrelatedness in the costs. We consider costs of the form of a sum:

$$C_{ij} = R_{ij} + T_{ij} + E_{ij}, \tag{8}$$

where the R_{ij}, T_{ij}, E_{ij} are random variables, representing costs contributed by the i^{th} robot, the j^{th} task, and an environment between them, subject to the following conditions:

1) every E_{ij} is independent of every R_{kl};
2) every E_{ij} is independent of every T_{kl};
3) every R_{ij} is independent of every T_{kl};
4) each E_{ij} is independent of $\forall k_{k\neq i}\, \forall l_{l\neq j}\; E_{kl}$;
5) each R_{ij} is independent of $\forall k_{k\neq i}\, \forall l\; R_{kl}$;
6) each T_{ij} is independent of $\forall k\, \forall l_{l\neq j}\; T_{kl}$.

Note that there is no assumption of identical distributedness.

Statistical dependencies can exist between factors not precluded by the six conditions above. The term E_{ij} is a factor which influences C_{ij} independently of other factors. But the R_{ij} variable can have dependencies on R_{ik}, it is intended to capture uncertainty born of aspects of the i^{th} robot. The same thinking applies to T_{ij} for the j^{th} task.

Theorem 4.1 For costs of the form (8) subject to the six stated constraints, the costs C_{ij} where $x^*_{ij} = 1$ $(i,j \in \{1,\cdots,n\})$ are independent random variables given an assignment $\mathbf{X}^*$ satisfying (2)–(3).

Proof. Consider distinct C_{kl} and C_{rs}, where $x^*_{kl} = 1$, and $x^*_{rs} = 1$. Clearly E_{kl} is independent from E_{rs}, but R_{kl} can only be dependent on R_{rs} if $k = r$, which contradicts (3). Similarly, T_{kl} can only be dependent on T_{rs} if $l = s$, which contradicts (2). □

The joint distribution of the costs must be known in order to compute the distribution of the sum of random costs. With the independence of the costs, the joint distribution can be obtained by the product of the given marginal cost distributions, which is significantly more convenient than that of the dependent case.

We prove another theorem about CVaR for a computationally tractable formulation of the probabilistic MRTA problem described in the following section. Definitions 4.2, 4.3, and Theorem 4.4 are from [12].

Definition 4.2 (Comonotonic set) The set $A \subseteq \mathbb{R}^n$ is comonotonic if for any y and z in A, either $y \leq z$ or $z \leq y$.

Definition 4.3 (Comonotonic random vector) A random vector is comonotonic if it has a comonotonic support.

Theorem 4.4 (Comonotonic additivity) In Theorem 4.2.1 in [12], it is proven that

$$\text{CVaR}_{\lambda,S^c} = \sum_{i=1}^{n} \text{CVaR}_{\lambda,Z_i}, \tag{9}$$

where $\text{CVaR}_{\lambda,V}$ denotes the CVaR of the random variable V with a confidence level λ. The sum of comonotonic random variables $(Z_1^c, \cdots, Z_n^c)$ is $S^c = Z_1^c + \cdots + Z_n^c$. Thus, the CVaR of the sum of the random variables is equal to the sum of the CVaRs of the random variables if the random variable forms a comonotonic random vector.

From the definitions and theorems, we derive a theorem for a convenient computation of the CVaR of an assignment.

Theorem 4.5 The CVaR of an assignment $\mathbf{X}$ can be computed simply from the sum of each cost distribution's CVaR:

$$\text{CVaR}_{\lambda,C_{\mathbf{X}}} = \sum_{i=1}^{n}\sum_{j=1}^{n} \text{CVaR}_{\lambda,C_{ij}} x_{ij}. \tag{10}$$

Proof. The cost sum of an assignment, $C_{\mathbf{X}}$, consists of the sum of costs (i.e., random variables) whose decisions variables are $x_{ij} = 1$ (i.e., $C_{\mathbf{X}} = \sum_{i,j=1}^{n} C_{ij}x_{ij}$). The support of those random variables is the complete set of, or a subset of, nonnegative real numbers, which is a totally ordered set. A totally ordered set is a comonotonic set by definition. So the sum of costs in an assignment is the sum of comonotonic variables, and Theorem 4.4 holds for the CVaRs of $C_{\mathbf{X}}$ and C_{ij}. □

V. Optimal Assignment with Probabilistic Costs

For $i, j \in \{1, \cdots, n\}$, we use the mean μ_{ij} and $\text{CVaR}_{\lambda,ij}$ of random variable C_{ij} to characterize its distribution. From Theorem 4.5, we can define two objective functions replacing (1), which are the sum of means and CVaRs:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} \quad \mu_{ij}x_{ij}, \tag{11}$$

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} \quad \text{CVaR}_{\lambda,ij}x_{ij}. \tag{12}$$

It is worth noting that, without Theorem 4.1 and 4.5, one must compute the sum of the distributions and the CVaR of the sum, which involve several integrals. The problem of optimizing these functions subject to (2)–(5) is the biobjective assignment problem (BiAP) [13]. The optima are best visualized as a Pareto front. Some assignments may have low summed mean but high CVaR, some with low CVaR but high mean, and others may represent a compromise.

It is not meaningful to seek a single optimum given that there are two objectives unless we can express a preference as a precise trade-off between minimizing the mean and the CVaR. We quantify this as a *risk preference*, viz. a stipulation of the relative importances of the mean and the CVaR. This yields a scalarized BiAP that

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} \left(\alpha\mu_{ij}x_{ij} + (1-\alpha)\text{CVaR}_{\lambda,ij}x_{ij}\right) \tag{13}$$

subject to (2)–(5) where $\alpha \in [0, 1]$ is the risk preference. Thus, with this preference, one may apply a standard assignment algorithm to produce an assignment.

The confidence interval λ is another parameter that changes the objective value (13) because it changes the CVaR. Thus, the determination of $\lambda \in (0, 1)$ should be taken into account. We follow one of the conventions (i.e., 95% or 99%) for determining it, but varying its value is still a meaningful direction to consider so discussed in Sec. V-D.

On the other hand, we note that some problems have an optimal assignment $\mathbf{X}^+$ such that

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} \left(\alpha\mu_{ij}x_{ij}^+ + (1-\alpha)\text{CVaR}_{\lambda,ij}x_{ij}^+\right), \tag{14}$$

is minimized for $\alpha \in [0, 1]$. We term these problems *risk preference indifferent* since the outcome is identical for any α. We show a provable instance of such the problems.

Theorem 5.6 Problems with costs of the form (8) subject to the six stated statistical requirements, with the additional fact the $\forall j\,(\forall i\ R_{ij} = R_i)$ and $\forall i\,(\forall j\ T_{ij} = T_j)$ (both R_i and T_j are random variables), and $\forall i, j\ E_{ij} = k_{ij}$ (where each k_{ij} is constant) are risk preference indifferent.

Proof. If R_i has $\text{CVaR}_{\lambda,R_i}$ and T_j has $\text{CVaR}_{\lambda,T_j}$, then all assignments $\mathbf{X}$'s (optimal and otherwise) have a sum of $\sum_{i=1}^{n}\sum_{j=1}^{n} \text{CVaR}_{\lambda,ij}x_{ij}^+ = \sum_{i=1}^{n} \text{CVaR}_{\lambda,R_i}x_{ij}^+ + \sum_{j=1}^{n} \text{CVaR}_{\lambda,T_j}x_{ij}^+$. Hence, the optimal assignment only depends on the first term of (14), that is the one involving μ_{ij}. The minimizer of this assignment is the overall minimizer no matter the value of α. □

Several practical problems fall into this category as the example in Fig. 3, where the distributions are symmetric from the mean of robot or task locations toward the opposite directions. Since CVaR is proportional to the variability (e.g., variance) of a distribution, the sum of CVaRs of the two assignments A and B are the same. Thus, the value of α does not affect the determination of an optimal assignment.

Nevertheless, certainly there remain many problems which are not risk indifferent. Since α weights two sums with different meanings, it is a fairly *ad hoc* parameter. We therefore propose a sensitivity analysis that allows a user to determine how critical their choice of α is in producing the particular optimum. The analysis helps understanding the effect of the preference change and determining its value.

A. Standard sensitivity analysis for the risk preference

Given a risk preference α, we compute its safe interval where any change within the interval does not change the current optimal assignment. Conceptually, this is the same as the sensitivity analysis introduced in Sec. III-B. The difference is that the bounded region of costs is determined by the weighted sum of two objective values:

$$\mathbf{C} = \alpha\mathbf{M} + (1-\alpha)\mathbf{P}, \tag{15}$$

where $\mathbf{C}$ is a matrix representation of C_{ij} for $i, j \in \{1, \cdots, n\}$. $\mathbf{M}$ and $\mathbf{P}$ are $n \times n$ matrices consisting of μ_{ij}

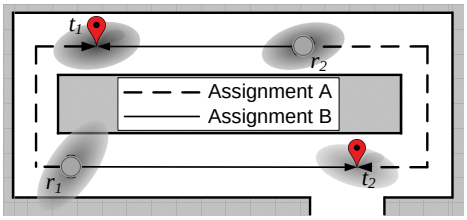


Fig. 3: An example risk preference indifferent problem. Robots navigate to tasks via a road network. By virtue of the symmetric property of the normal distribution, $R_{ij} = R_i$ and $T_{ij} = T_j$ hold for all i and j. Assuming all E_{ij}'s are constant (or equal), two assignments A and B have the same variability, which makes them to have the same CVaR sum.

and $\text{CVaR}_{\lambda,ij}$, respectively. The cost $\mathbf{C}$ varies depending on α and forms a line in the cost space since α is a 1-D parameter.

As discussed, $\theta(\mathbf{X}^*)$ is a polyhedral cone consisting of linear inequalities (6). Since we have a set of linear equations $\theta(\mathbf{X}^*)$ and a linear function $\mathbf{C}$, one can compute intersections between them. An intersection is a particular value of $\mathbf{C}$ that corresponds to a unique α because (15) is injective. Thus, we can compute an interval $\alpha \in [a, b]$ that is the minimum and maximum of those intersections. A straightforward way to achieve this is by linear programming where (15) is minimized or maximized over $\theta(\mathbf{X}^*)$. Since there is only one variable α, we do not need to consider the full cost space of $\mathbf{C}$. We reduce the dimensionality, by minimizing and maximizing α over a set of linear inequalities that are functions of α, obtained by substituting c_{ij} by $\alpha\mu_{ij} + (1-\alpha)\text{CVaR}_{\lambda,ij}$.

This procedure is shown in Alg. 1. Given $\mathbf{M}$ and $\mathbf{P}$, we compute $\mathbf{C}$ (line 1) and compute an optimal assignment (line 2). We use the sensitivity analysis described in Sec. III-B to compute $\theta(\mathbf{X}^*)$ (line 3). Then we rearrange $\theta(\mathbf{X}^*)$ as a function of α. Lines 4 and 5 compute the minimum and maximum intersection, so the objective values yield the exact interval of α where any α within the interval does not destroy the optimality of the current assignment.

Algorithm 1 RiskSA

Input: $n \times n$ matrices $\mathbf{M}$ and $\mathbf{P}$ that consist of μ_{ij} and $\text{CVaR}_{\lambda,ij}$, respectively. A risk preference α.
Output: An exact interval $[a, b]$ of α and the optimal assignment $\mathbf{X}^*$ in that interval.

1 $\mathbf{C} = \alpha\mathbf{M} + (1-\alpha)\mathbf{P}$
2 $\mathbf{X}^* = \text{HUNGARIAN}(\mathbf{C})$
3 $\theta(\mathbf{X}^*) = \text{SA}(\mathbf{X}^*, \mathbf{C})$ // compute $\theta(\mathbf{X}^*)$ by (7)
4 a = LINPROG(α, $\theta(\mathbf{X}^*)$, 0, 1, max) // compute the max $\alpha \in [0,1]$
5 b = LINPROG(α, $\theta(\mathbf{X}^*)$, 0, 1, min) // compute the min $\alpha \in [0,1]$
6 **return** $\mathbf{X}^*$, a, b

B. Heuristic sensitivity analysis for risk preference

The discussion in Sec. III-B anticipates the weakness of Alg. 1: the running time and space complexity are factorial in n because SA in line 3 of Alg. 1 enumerates all feasible solutions, a set whose size has factorial growth. Thus, we describe a heuristic method that produces estimates of the output interval much more quickly, described in Alg. 2.

The algorithm begins with a given α and computes the optimal assignment for α. The algorithm expands the interval downward (lines 5–14) and upward (lines 16–25). In each iteration, α increments or decrements by δ. Within $[0, 1]$, it runs until the optimal assignment is altered. In each direction, the maximum difference between the approximated boundary and the exact boundary is δ. Thus, the approximated interval is smaller than the exact one by at most 2δ. The time complexity is $O(\frac{n^3}{\delta})$ since it calls the Hungarian method ($O(n^3)$) at most $\lfloor \frac{1}{\delta} \rfloor$ times.

Algorithm 2 HeuristicSA

Input: $n \times n$ matrices $\mathbf{M}$ and $\mathbf{P}$ that consist of μ_{ij} and $\text{CVaR}_{\lambda,ij}$, respectively. A risk preference α_0 and the approximation parameter δ.
Output: An approximation interval $[a', b']$ of α and the optimal assignment $\mathbf{X}^-$ in that interval.

1 $\alpha = \alpha_0$
2 $a^- = b^- = \alpha$
3 $\mathbf{C} = \alpha\mathbf{M} + (1-\alpha)\mathbf{P}$
4 $\mathbf{X}^- = \text{HUNGARIAN}(\mathbf{C})$
5 **while** $\alpha - \delta \geq 0$ // expand α downward by δ in each loop
6 $\quad \alpha = \alpha - \delta$
7 $\quad \mathbf{C} = \alpha\mathbf{M} + (1-\alpha)\mathbf{P}$
8 $\quad \mathbf{X}^* = \text{HUNGARIAN}(\mathbf{C})$
9 $\quad$ **if** $\mathbf{X}^* == \mathbf{X}^-$ // if $\mathbf{X}^*$ does not change with the new α
10 $\quad\quad a' = \alpha$ // expand a'
11 $\quad$ **else**
12 $\quad\quad$ **break** // $\mathbf{X}^*$ changed, stop downward expansion
13 $\quad$ **end if**
14 **end while**
15 $\alpha = \alpha_0$
16 **while** $\alpha + \delta \leq 1$ // expand α upward by δ in each loop
17 $\quad \alpha = \alpha + \delta$
18 $\quad \mathbf{C} = \alpha\mathbf{M} + (1-\alpha)\mathbf{P}$
19 $\quad \mathbf{X}^* = \text{HUNGARIAN}(\mathbf{C})$
20 $\quad$ **if** $\mathbf{X}^* == \mathbf{X}^-$
21 $\quad\quad b' = \alpha$ // expand b'
22 $\quad$ **else**
23 $\quad\quad$ **break** // $\mathbf{X}^*$ changed, stop upward expansion
24 $\quad$ **end if**
25 **end while**
26 **return** $\mathbf{X}^-$, a', b'

C. Extension: finding optimal assignments for $\alpha \in [0, 1]$

If a practitioner has doubts when choosing a value of α, it is useful to see how the optimal assignment changes for the entire range. It can be computed via sensitivity analysis of optimal assignments with respect to α. This analysis can be achieved via repeated calls to Alg. 1. The first iteration finds an interval of an arbitrary α for its optimal assignment. Then the next iteration finds another interval outside the initial interval (i.e., choosing an arbitrary value of α outside the interval already found). It finishes once all intervals in $[0, 1]$ are found. Since Alg. 1 finds the exact interval of α for an

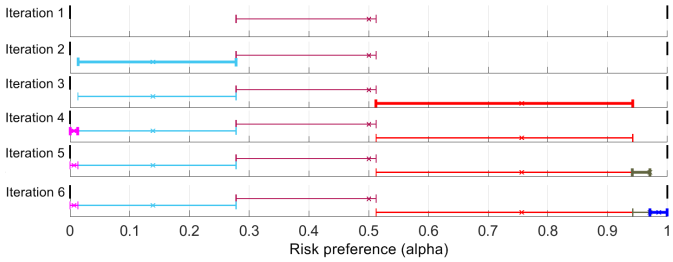


Fig. 4: An example progression of Alg. 3 ($n = 4$). In each iteration, the algorithm finds a new interval (the thick horizontal bar) and its corresponding optimal assignment. The algorithm improves the map of α in each iteration and eventually finds the exact solution (Iteration 6).

optimal assignment, choosing any α outside the range gives a new optimal assignment and its interval.

Although Alg. 1 gives exact intervals, it can be computationally intractable as the problem size increases. Thus, we repeat Alg. 2 until all optimal assignments for all discrete values of α have been explored. It may not find all optimal assignments in the continuous space of α, but quickly give valuable information for determining α.

Alg. 3 analyzes the sensitivity of optimal assignments with respect to α using a modified version of Alg. 2 (a minor modification is limiting the working range of to I instead of the full range $[0, 1]$), and Fig. 4 gives an example use of the algorithm. The algorithm begins with the initial unexplored range of α (line 2). It runs until no range remains unexplored or any of unexplored ranges is greater than δ. In each iteration, one unexplored range is selected and removed from the queue U (line 4). Then α is computed as the midpoint of the selected range (line 5). The optimal assignment $\mathbf{X}_i^*$ for that α and its approximation interval S_i is computed (lines 6–7). In Fig. 4, the cross indicates α, and the horizontal bar represents the interval. If the lower bound of the interval is larger than the lower bound of the previously unexplored range (line 8), a new unexplored range is inserted to the queue (line 9). Lines 11–13 are for the upward expansion. For example, $[a', b'] = [0.278, 0.512]$ in the first iteration in Fig. 4, so U becomes $\{[0, 0.278), (0.512, 1]\}$. The time complexity is $O(\frac{n^3}{\delta^2})$ since it calls Alg. 2 at most $\lfloor \frac{1}{\delta} \rfloor$ times.

D. Remarks

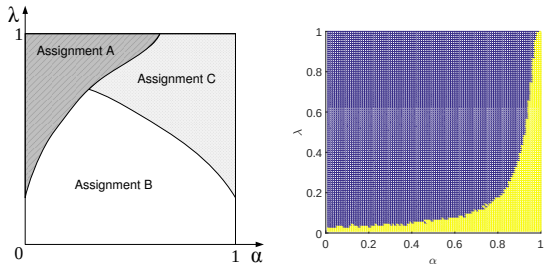
In the BiAP, only empirical methods exist to select α despite its importance. The preceding sections have provided sufficient conditions for problems where the particular value of α is irrelevant. Secondly, they have presented methods that reduce manual labor in exploring different risk preferences.

On the other hand, λ is another parameter that decides the value of the scalarized cost as discussed in Sec. IV. In this work, λ is fixed to a predetermined value (i.e., 95% or 99%) since the parameter has a clear interpretation. If the determination of λ is not convinced, one could perform another sensitivity analysis with respect to the value of λ that is briefly described below.

Algorithm 3 α-hSA

Input: $n \times n$ matrices $\mathbf{M}$ and $\mathbf{P}$ that consist of μ_{ij} and $\mathrm{CVaR}_{\lambda,ij}$, respectively. The approximation parameter δ.
Output: A set of optimal assignments X and their α-intervals S.

1 $i = 0$
2 $U = \{[0, 1]\}$ // `initialize the queue of unexplored` α
3 **while** (U is not empty) || (any range in U is greater than δ)
4 $\quad I = \mathrm{DEQUEUE}(U)$ // `dequeue one unexplored range`
5 $\quad \alpha = \mathrm{MIN}(I) + \frac{\mathrm{MAX}(I) - \mathrm{MIN}(I)}{2}$ // `begin with the midpoint` // `of an unexplored range`
6 $\quad [\mathbf{X}_i^*, a', b'] = \mathrm{HEURISTICSA}'(\mathbf{M}, \mathbf{P}, \alpha, \delta, I)$ // `run Alg. 2`
7 $\quad S_i = [a', b']$ // `the apx. interval of` $\mathbf{X}_i^*$
8 $\quad$ **if** $\mathrm{MIN}(I) < a'$
9 $\quad\quad \mathrm{ENQUEUE}(U, (\mathrm{MIN}(I), a'))$ // `exclude searched range`
10 $\quad$ **end if**
11 $\quad$ **if** $\mathrm{MAX}(I) > b'$
12 $\quad\quad \mathrm{ENQUEUE}(U, (b', \mathrm{MAX}(I)))$ // `exclude searched range`
13 $\quad$ **end if**
14 $\quad$ increment i
15 **end while**
16 **return** $X = \{\mathbf{X}_0^*, \cdots \mathbf{X}_\Pi^*\}$, $S = \{S_0, \cdots S_\Pi\}$



(a) A hypothetical mapping. (b) A mapping from a small-sized instance ($n = 5$).

Fig. 5: A hypothetical and an example mapping of optimal assignments on the α-λ plane, which is analogous to the mapping with respect to α shown in Fig. 4.

We know that (15) given a fixed value of λ forms a line in the cost space while changing α. For all values of $\lambda \in (0, 1)$, one ends of the lines (when $\alpha = 1$, which the CVaR term has no contribution) reach to the same point in the cost space. Changing both parameters produces a (hyper) surface unless $\mathbf{P}$ in (15) is invariant (in this case, the cost still forms a line). The shape of the surface, which could be nonlinear, is determined by the cost distributions. The surface has a point that all constituting lines (i.e., each line is formed when changing α for each value of λ) join at, such as a cone.

Given a cost surface, we can use the standard or the randomized analysis (RANDSA) proposed in [8] to compute the regions of the optimal assignments on the surface. The regions from an analysis correspond to the regions of optimal assignments on the α-λ plane (Fig. 5), which is analogous to the one-dimensional mapping with respect to α (Fig. 4).[3]

[3]Fig. 5(b) is not drawn by a sensitivity analysis but by changing two parameters manually.

VI. EXPERIMENTS

We examine four distinct types of experiments. First, we generate random costs that simulate the situations with large uncertainties. This experiment shows the validity of our formulation of using the mean and CVaR than using scalar values only. Next, we measure the performance of the algorithms with random instances. Next, we examine two robot navigation scenarios, giving a view of the sensitivity of a risk preference in practical applications. First, we capture uncertainties in the state of robots from a simulator using extended Kalman filter (EKF) for localization. Second, we collect traveling times in an urban area using the Google API [14], which provides a traveling time between two points for a specific time in a day. It is an instance of obtaining cost distributions from historical data or measurements. From the data, we compute the mean and CVaR of traveling times.

A. Assignment computation considering uncertainties

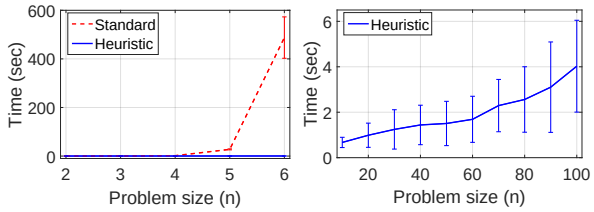
We run an experiment that shows the advantage of using the mean and CVaR than scalar values (e.g, mean) for costs. Our formulation is adequate for uncertain situations, so we set cost distributions to have nontrivial variances. The probability distributions of costs are neither necessarily a particular distribution nor identical. We only need to know their density functions. Then the means and the CVaRs of those distributions can be computed. For the computations of CVaRs, the methods discussed in Sec. III-C can be used.We use normal distributions[4] for costs, where each cost has the mean and standard deviation drawn from uniform distributions $\mathcal{U}(0, 10)$ and $\mathcal{U}(0, 20)$, respectively (e.g., a normal cost distribution $\mathcal{N}(5, 15^2)$). We randomly generate an $n \times n$ matrix of normal distributions and compute an optimal assignment based on both the mean and CVaR (with $\alpha = 0.05$, which is risk-averse). For comparison, we compute another optimal assignment only with the mean values. Then a cost matrix is sampled from the matrix of normal distributions. We compare the cost sum of the both assignments using 10,000 random matrices ($n = 50$ and $\lambda = 0.95$). Our formulation reduces 7.511% of the cost sum (the standard deviation is 0.1578) compared to the case of using the mean only. It shows that our formulation is able to save costs using the richer information about uncertainties.

B. Random instances for performance evaluation

We randomly generate an $n \times n$ matrix of normal distributions where the mean and variance of each distribution are drawn from $\mathcal{U}(0, 1)$. The parameter is $\delta = 0.001$ throughout all experiments. The parameter δ can be adjusted considering the time allowed and the desired degree of accuracy.

Fig. 6, and Tables I and II show the results from random matrices. The running time (averaged over 20 repetitions) of the standard method is prohibitive when $n > 6$. Up to $n = 6$, the standard and the heuristic method use the same problem instance for each repetition. For $n > 6$, we only run the heuristic method. The results in Fig. 6(b) show that the heuristic method is scalable for large instances.

[4]The CVaR of a normal distribution is $-\mu + \frac{\sigma}{\lambda} f(\lambda)$ for a pdf $f(\cdot)$.



(a) Running time up to $n = 6$. (b) Running time for $n > 6$.

Fig. 6: Running time of random instances. (a) The standard algorithm finding the exact interval. (b) The heuristic method for larger instances.

TABLE I: Running time (sec) of Alg. 1 and 2 (20 repetitions).

(a) Running time up to $n = 6$. The running time of the standard algorithm for $n > 6$ is prohibitive.

Alg.	n	2	3	4	5	6
Standard	Mean	0.09980	0.2030	1.724	27.59	487.7
	Std. dev.	0.09450	0.03960	0.05320	2.046	85.19
Heuristic	Mean	0.1465	0.2125	0.2479	0.3508	0.3961
	Std. dev.	0.04642	0.06633	0.08662	0.09421	0.1079

(b) Running time of the heuristic algorithm.

n	10	20	30	40	50	60	70	80	90	100
Mean	0.6706	0.9870	1.244	1.440	1.505	1.688	2.292	2.561	3.103	4.026
Std.	0.2255	0.5324	0.8664	0.8682	0.9744	1.014	1.151	1.441	1.990	2.021

We measure the running time and the solution quality of Alg. 3 for $n = 10, \cdots, 50$ (Table II-a). This result is compared to a basic method that runs the Hungarian method iteratively from $\alpha = 0$ to 1. The number of iterations is determined to reach the same solution quality with Alg. 3 (i.e., α increments by 0.0007). The solution quality (Table II-b) is measured by the searched ranges over the full range of α. Notice that two methods use the same instances.

Alg. 3 shows faster running times than the iterative method while both methods produce high quality solutions. The algorithm is supposed to run offline before operating robots, so 15 sec for 50 robots is reasonable.[5] The solution quality decreases as n grows because the methods with large instances call their subroutines (i.e., Alg. 2 and the Hungarian method) frequently, where each call of a subroutine produces an error at most 2δ.

C. Cost uncertainties from state estimation

We use a robot simulator [15] employing a state estimator (EKF) to capture the uncertainties in robot poses so that costs can be represented by distributions. We place five robots and five tasks (Fig. 7(a)) where the mission is visiting the tasks by the robots without duplicated assignments. Cost is traveling time where the objective is to minimize the sum of traveling times. We simplify that the robots move constantly at 1 m/s. Also, we assume that the locations of the tasks are certain, and the robots locally avoid collisions. For a robot-task pair, the path length between the mean robot position and the task

[5]A system operator (or robots) determines a value of α before execution. Or α may change at run-time by looking at the result in hand.

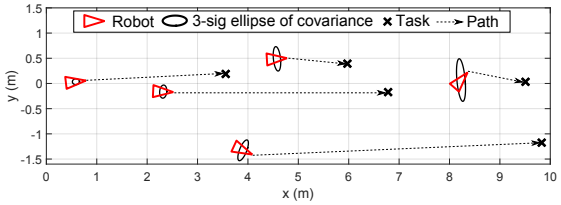
TABLE II: Results of Alg. 3 and an iterative method (20 repetitions).

(a) Running time (sec).

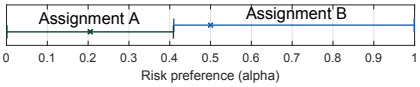
	n	10	20	30	40	50
Alg. 3	Mean	0.8717	2.532	5.061	9.073	15.51
	Std. dev	0.1736	0.5523	0.8147	0.9947	1.474
Iterative method	Mean	1.221	3.523	6.946	12.23	20.40
	Std. dev	0.2579	0.7551	1.090	1.096	1.981

(b) Solution quality (%).

	n	10	20	30	40	50
Alg. 3	Mean	99.37	98.81	98.35	97.71	96.74
	Std. dev	0.1000	0.2300	0.2700	0.4900	0.4400
Iterative method	Mean	99.73	99.29	98.89	98.47	97.77
	Std. dev	0.0900	0.1800	0.2200	0.3000	0.2700



(a) A snapshot from a robot simulator [15] with an example assignment. The robots move to the tasks along the paths.



(b) The result from Alg. 3. Two assignments and their corresponding intervals are found.

Fig. 7: An experiment with cost distributions from a state estimator. (a) Cost (traveling time) distributions are computed based on the path lengths and the covariance matrices of the robot poses. (b) Using Alg. 3, the problem of choosing α in the continuous space $[0, 1]$ is converted to choosing one assignment from the discrete set of assignments.

location, which is the mean of the path length distribution, can be computed. The variance of the distribution is determined by the outgoing direction from the current robot pose (i.e., $\sigma^2 = \mathbf{v}^T \boldsymbol{\Sigma} \mathbf{v}$ where $\mathbf{v}$ is the vector representing the outgoing direction of a path and $\boldsymbol{\Sigma}$ is the covariance matrix). From this distance distribution, we calculate the cost (i.e., traveling time) distribution by dividing the mean and the standard deviation by the velocity. Repeating this for all n^2 costs constructs a matrix of distributions.

Fig. 7(b) shows the result from Alg. 3 where the input is the matrix of cost distributions and $\delta = 0.001$. Two assignments and their corresponding intervals are found where Assignment A is risk-averse than B. It shows that the determination of α in the continuous space in $[0, 1]$ is converted to choosing one from the discrete set of two assignments, which is much simpler than the original problem.

D. Cost uncertainties from historical data or measurements

We examine a practical transportation problem of self-driving vehicles in a metropolitan area. We position five vehicles and five tasks at arbitrary locations (Fig. 8). The mission is visiting the task locations by the vehicles where the

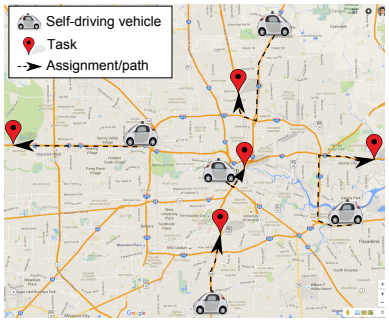


Fig. 8: A transportation problem in a metropolitan area. The mission is visiting the tasks by the vehicles with the minimum sum of traveling times. The traveling times are distributed owing to the varying traffic conditions, so the scalar cost representation is not adequately rich.

traveling times vary depending on traffic conditions. We collect traveling times using Google Directions API [14]. For each pair of a vehicle and a task, 1,439 data points are collected, which are the measurements at every one minute during a day. Note that the path of a pair is the same for all data points, which is marked as an arrow in Fig. 8. These data reflect the traffic condition at the specific time in a day. Thus, the data points are naturally distributed.

Instead of fitting probability density functions to the data, we choose to use a nonparametric method to obtain the mean and CVaR for each vehicle-task pair. We calculate the mean from a set of data point for a pair. For the CVaR, we calculate the mean of the highest 5% of the set (i.e., $\lambda = 0.95$). This nonparametric method does not have the problem of imposing an incorrect assumption about the distribution.

We run Alg. 3 with the matrix of means and CVaRs of traveling times. The algorithm finds a unique optimal assignment for any $\alpha \in [0, 1]$, which means that this problem belongs to the class of risk preference independent. The reason is that the CVaRs are proportional to the length of the paths, so the optimal assignment does not change while α changes. Alg. 3 is able to decide whether a problem instance belongs to the risk preference independent class.

VII. Conclusion

In this paper, we consider multi-robot task allocation under uncertain costs. We use a cost representation incorporating uncertainty and interdependency, via distributional models. We provide conditions to show that the interdependencies among costs do not exist between elements in an assignment. The representation gives a new perspective on optimizing an allocation subject to a risk preference, where uncertainty takes a role in determining an optimal assignment. In addition, we show a problem class where the position taken on risk has no effect on the optimal assignment. For the problems where the risk preference is important, we provide algorithms for analyzing the sensitivity of an optimal assignment with respect to the risk preference. This enables a better understanding of how to determine the risk preference and its consequence. Further, we would like to investigate individual risk preferences than the homogeneous setting.

REFERENCES

[1] B. Gerkey and M. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *Int. J. of Robotics Research*, vol. 23, pp. 939–954, Sept. 2004.

[2] R. Burkard and E. Cela, *Linear assignment problems and extensions*. Springer, 1999.

[3] P. Krokhmal and P. Pardalos, "Random assignment problems," *European J. of Operational Research*, vol. 194, no. 1, pp. 1–17, 2009.

[4] E. Nikolova and N. Stier-Moses, "A mean-risk model for the stochastic traffic assignment problem," *Operations Research*, vol. 62, pp. 366–382, 2014.

[5] S. Ponda, L. Johnson, and J. How, "Distributed chance-constrained task allocation for autonomous multi-agent teams," in *American Control Conference*, 2012, pp. 4528–4533.

[6] T. Dey, A. Hirani, and B. Krishnamoorthy, "Optimal homologous cycles, total unimodularity, and linear programming," *SIAM J. on Computing*, vol. 40, pp. 1026–1044, 2011.

[7] J. Ward and R. Wendell, "Approaches to sensitivity analysis in linear programming," *Annals of Operations Research*, vol. 27, pp. 3–38, 1990.

[8] C. Nam and D. Shell, "When to do your own thing: Analysis of cost uncertainties in multi-robot task allocation at run-time," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2015, pp. 1249–1254.

[9] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, "Coherent measures of risk," *Mathematical finance*, vol. 9, pp. 203–228, 1999.

[10] H. Markowitz, "Portfolio selection," *The J. of Finance*, vol. 7, pp. 77–91, 1952.

[11] A. Andreev, A. Kanto, and P. Malo, "On closed-form calculation of CVaR," 2005, Helsinki School of Economics, working paper.

[12] J. Dhaene, S. Vanduffel, M. Goovaerts, R. Kaas, Q. Tang, and D. Vyncke, "Risk measures and comonotonicity: a review," *Stochastic models*, vol. 22, pp. 573–606, 2006.

[13] E. Ulungu and J. Teghem, "The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems," *Foundations of Computing and Decision Sciences*, vol. 20, pp. 149–165, 1995.

[14] Google, "The Google Directions API," https://developers.google.com/maps/documentation/directions/, 2016.

[15] P. Corke, *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011.