Tackling Task Allocation Uncertainty via a Combinatorial Method

Lantao Liu and Dylan A. Shell

Abstract — A task-executing robot may encounter various types of uncertainty born of sensing, actuation, and motion errors. A robot uncertain of its current state is likely to reflect this uncertainty in the calculation of its utilities or performance estimates (*e.g.*, costs, fitnesses) for planning purposes too. While controlling a single robot under uncertainty is challenging, coordinating a group of robots is likely to exacerbate the problem. An efficient and reliable way for assessing the uncertainty in the system in light of how it will affect robot's decisions and subsequent actions can help address the challenge of distributed reasoning under uncertainty. This paper examines our previously developed *Interval Hungarian algorithm*, providing complementary interpretations from the perspective of robotics applications. The method is described step by step via a common scenario: multi-robot navigation with localization uncertainty. Which we provide an extended comparison and analysis of this algorithm, as well as insights that we gained from experience conducting physical robots experiments.

Keywords: *Task allocation, multi-robot systems, uncertainty, combinatorial methods*

I. Introduction

Several new robotics research problems of importance have been produced by reexamining problems which have been extensively studied and successfully addressed under ideal conditions, and introducing variants under less benign conditions. Relaxation of information assumptions is one good example of this: problems arising from noise and estimation uncertainty are quintessential examples of this type of progress in the last decade. Indeed, in the real world, sources for uncertainty are everywhere, assumptions of perfect information are frequently violated. A typical mobile robot, for example, has information collected from sensors and motion realized from actuators can be noisy or even inaccurate; the inferred state of a robot must have some uncertainty due to limits in its knowledge (*e.g.*, Monte Carlo localization in an indoor environment represents this via a distribution over poses [1]). In these circumstances, a risk-averse strategy when selecting actions should reduce the chances of catastrophic failures.

The problem is likely to become more complex for a multi-robot system when multiple robots suffer from uncertainty at the same time, and a joint plan (or action) is required. Classic task allocation methods coordinate a multi-robot system by optimizing the overall performance with given metrics, but without considering uncertainties [2], [3]. If uncertainty exists, a common strategy is to compute the mean or median estimates which are then fed into the optimal assignment solvers. Such

Lantao Liu and Dylan A. Shell are with Dept. of Computer Science and Engineering, Texas A&M University, College Station, TX 77843, USA. {lantao, dshell}@cse.tamu.edu

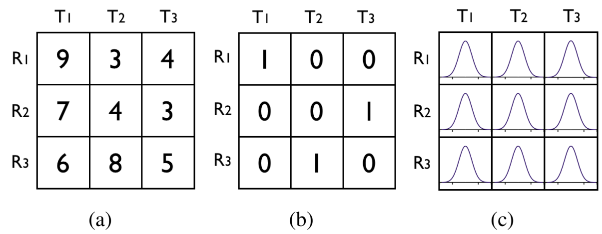


Fig. 1. (a) Utility matrix in a performance maximization problem; (b) Corresponding assignment matrix; (c) The conceptual form in which utilities are distributions instead of fixed values.

estimates may fail to work properly. For example, a robot localizing itself may determine that it is positioned at an emergency exit of the building. If in the map there are two exits, one at the front and the other at the rear of the building, then an estimate such as the mean will position the robot somewhere potentially quite far the exits. Estimates of this sort, thus, can be meaningless when particular situations require estimates of traversal costs.

We suggest that appropriately and efficiently dealing with task allocation under uncertainty, the details of uncertainty need must be considered in within the algorithm that assigns tasks to robots. This means that information regarding the uncertainty should not be reduced to a single value, or a point, but instead one should considered the distributional information which best reflects and describes the nature of the uncertainty. Recently, we proposed an algorithm called the *Interval Hungarian algorithm* [4] to consider multi-robot task allocation subject to perturbations in the input data. This treatment of uncertainty adopts a point of view that differs from popular approaches. (Of which, reviews are provided in a latter section.) Experiments in both simulation and physical robots were also conducted to validate the proposed algorithm [5]. This paper does not repeat the algorithm, but instead, the method is reasoned in detail from a set of multi-robot localization scenarios, aiming to reveal the essence of **why** and especially **how** we designed this method. Also, we outline experience that we gained in applying this method to physical mobile robots, as well as discussions for dealing with some possible issues arising. We believe that this information is valuable more generally in designing risk-averse assignment algorithms and addressing practical task allocation problems. This article complements our preceding descriptions of the *Interval Hungarian algorithm*, by starting from a robotic application, visualizing uncertainty spatially, and examining how the algorithm allows one to address the resulting challenge.

Unlike existing approaches, the interval Hungarian algorithm is a combinatorial method which addresses a simplified problem in order to avoid the curses of history and dimension: (1.) the algorithm is strongly polynomial — for any immediate query the algorithm requires $O(n^4)$ time complexity for a team of n robots; (2.) it does not require planning tree structures such as a policy tree or a scenario tree, therefore avoids the curse of dimension.

II. Related Work

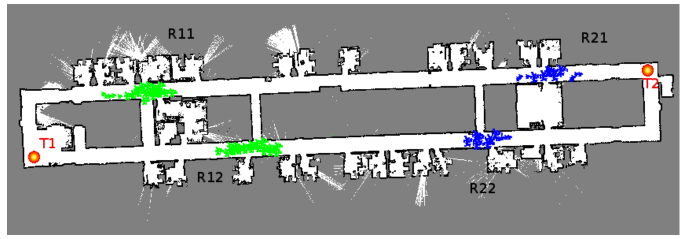
Current multi-robot task allocation methods remain inadequate when considering their performance under uncertainty. Generally, the methods that have garnered significant attention can be classified into two big categories: the (partially observable) Markov decision [6], [7], [8] based approaches, and the stochastic programming methods [9], [10]. It is worthwhile to draw a distinction between multi-robot coordination strategies that employ static notions of expected utility and those that model the task performance itself in greater detail. The latter schemes use a rich model of agents and tasks in order to construct a probabilistic model. For example, stochastic games/decentralized Markov decision problems (MDPs) [6], factored MDPs [11], and partially observable Markov decision problems (POMDPs) [7] permit one to explicitly address the question of when to perform particular actions (movement, sensing, communication) so as to reduce uncertainty if doing will be beneficial for the performance of tasks.

However, these problems do not admit polynomial-time solutions, and often factorization or independence assumptions are introduced in order to make the problem tractable. Further constraints may arise from a distributed solution (*e.g.,* game theoretic models which design individual pay-off matrices so that locally greedy agents maximize global pay-off) and require approximation (*e.g.,* potential games, or a similar treatment) or task-structure assumptions. In essence, both POMDP and stochastic programming methods plan ahead for a number of stages, with computational resources increasing exponentially.

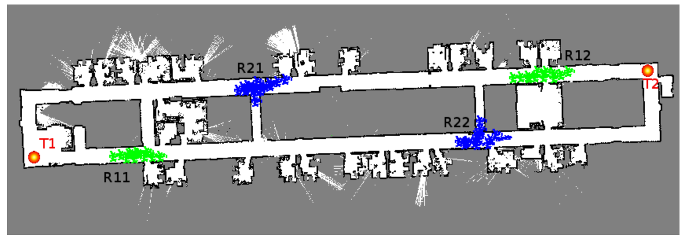
III. Task Allocation under Uncertainty

Task allocation can play an important role in coordinating a multi-robot system. It involves the problem of deciding which robots should do which tasks at specific times so that the team's performance will be optimized. The taxonomy of Gerkey and Matarić [2] describes how multi-robot task allocation systems may be classified. In our work, we are most interested in the category of problems consisting of single-task robots, single-robot tasks, for an instantaneous assignment (ST-SR-IA), which means each robot is assigned with exactly one unique task, and a solution is computed with immediate information and then executed instantly.

The input data for the allocation problem are called *utilities*. A utility is a value that quantifies the suitability (such as benefit, cost, fitness, *etc.*) of a robot-task pair. Therefore, for a system of m robots and n tasks, a total of $m \times n$ utilities are required. These utilities can be represented by a matrix with m rows and n columns. The matrix is termed a *utility matrix*. To describe the assignment solution, an *assignment matrix* with



(a)



(b)

Fig. 2. Uncertainty is demonstrated with Monte Carlo indoor localization[†]. Two robots localized themselves with uncertain locations in the corridors of a building. (a) Example of uncertainty insensitivity; (b) Example of uncertainty sensitivity.

size of $m \times n$ can be constructed: let the entries be 1s if the corresponding robot-task pairs are assigned and fill all other entries with 0s. Fig. 1(a) and 1(b) show a utility matrix and the corresponding assignment matrix, from which we can see that a valid assignment solution contains at most one assigned entry in any row or column of the assignment matrix.

However, when uncertainty is introduced, the utilities between robot-task pairs are no longer fixed values. In a typical representation, each utility becomes a set of probabilistic values following a certain distribution. Fig. 1(c) shows an example with each utility being replaced with a Gaussian function. For any utility matrix containing uncertain utilities (maybe not all), we need to either make an instantaneous task allocation decision so that robots can start to execute the tasks, or let robots perform immediate actions hopefully improving their state estimate so as to be less uncertain. More specifically, two situations are possible: in some cases, solutions are *insensitive* to uncertainty—despite the variety of possible utilities, the solutions to these various possibilities produce the same allocation and, thus, uncertainty can be safely ignored; in other cases, solutions are *sensitive* to the uncertainty—even slight changes in the utilities associated with a robot-task pair may produce distinct allocations, indicating that it may be better for the robots to take actions to gather information and reduce the level of uncertainty. Fig. 2 utilizes a Monte Carlo localization example to show how uncertainty can affect a task allocation problem. Assume there are only two robots, R_1 and R_2, and we need to dispatch the two robots to the nearest emergency sites (located at the bottom-left and top-right corners) based on their localization results. Now each robot determines that it is located in one of two possible spots: R_1 estimates that it can be located in either region R_{11} or region R_{12}, whereas R_2 has two possible positions: R_{21} or R_{22}. (Since the particle

[†]The map is from *player/stage* source code package [12], and the localization particles are drawn manually for purposes of illustration.

clouds in the corridors of Fig. 2(a) are compact we are treating them as single locations.)

A close examination reveals that robot R_1 should only be assigned to task site T_1 in the bottom-left corner since both possible localized positions are closer than those of R_2. And analysis for robot R_2 is similar. Therefore, the localization uncertainty in this case can be safely ignored, and a decision can be simply made to drive R_1 left towards T_1, and R_2 right towards to T_2. While the robots are moving, localization uncertainty can also be expected to be reduced.

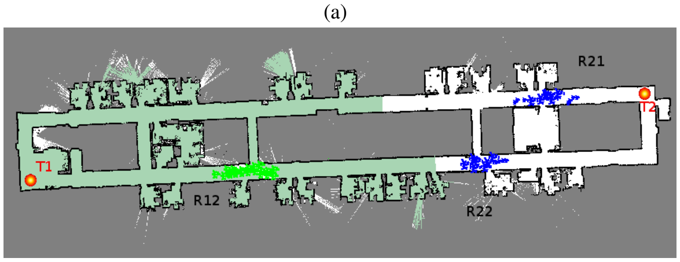
Unlike Fig. 2(a), however, situations like the one depicted in Fig. 2(b) are more complex. Different poses of R_1 can produce different allocations. In these cases, a strategy has to be designed to reduce the chance of mis-allocations, which waste both time and energy.

A natural question arises: is there a boundary which switches from one case to the other? Taking Fig. 2(a) as an example, if everything is unchanged except that robot R_1's one possible location R_{11} appears in a new place, R_{13}, closer to task site T_2, as illustrated in Fig. 3(a), then it is no longer safe to uniformly assign R_1 to T_1 and R_2 to T_2. This is because if R_1's true position is at R_{13} depicted in Fig. 3(a) and R_2's true position is at R_{22}, then the previous assignment conclusion does not hold. Actually, if all other localization information is unchanged, it is always safe to allocate R_1 to T_1 as long as R_{11} appears in the part of the corridor colored in light green (see Fig. 3(b)); otherwise, the assignment solution depends on particular conditions and needs to be considered separately. This concrete examples shows that, indeed, there is a "boundary" between the regions in which the allocation of sensitive and regions which are insensitive to uncertainty.

IV. Solution Overview: A Strong Polynomial Sensitivity Analysis Technique

The boundary mentioned in Section III is important since it splits the assignment problem into two regimes of behavior and it that can ease the uncertainty analysis. We need to find

(a)



(b)

Fig. 3. (a) An example of violation of insensitivity to uncertainty; (b) Insensitivity regions (light green color) of R_{11} that produce the same assignment solution.

	T1	T2	T3
R1	[8, +∞)	(-∞, 8]	(-∞, 5]
R2	(-∞,8]	(-∞, 6]	[2,+∞)
R3	(-∞,12]	[6,+∞)	(-∞, 7]

Fig. 4. An interval matrix that corresponds to utility matrix in Fig. 1(a).

such a boundary to assess whether it is sufficient to ignore underlying uncertainty.

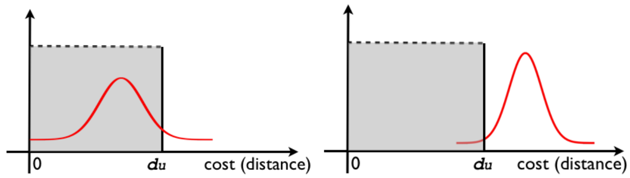
From Fig. 3(b), if a robot is located in the uncertainty-insensitive regions which always result in it being allocated to emergency site T_1 (light green corridors), this means that the distance to site T_1 is within range $[0, d_u]$ (or $(-\infty, d_u]$ in a general case), where the upper bound d_u is the largest allowable distance. Similarly, if a robot is located in the uncertainty-insensitive regions but should never be allocated to emergency site T_2, then the cost/distance to site T_2 should be in the range $[d_l, +\infty)$, where d_l is the lower bound on the distance. Consequently, we may obtain a matrix called an *interval matrix* in which each entry is replaced with an interval. Fig. 4 shows an interval matrix, in a utility maximization problem.

The interval for each matrix entry describes the permissible perturbation which ensures that the current assignment solution does not violate (values inside the same interval will produce the same solution, assuming other entries are known and fixed). This is the technique of *sensitivity analysis* and can be used to check if the solution of a deterministic linear program is reliable even if some of the parameters are not fully known, but are instead replaced by some estimates [10], [9]. However, popular sensitivity analysis methods generally are expensive to compute; in some instances their worst-cases are intractable for moderate sized problems, although on average they can be efficient.

The assignment problem can be formulated as an *integer linear program*. This classic problem has been solved by many strong polynomial algorithms, of which the Hungarian method [13] is probably the most famous. This algorithm uses only $O(n^3)$ time complexity to find the optimal solution for an $n \times n$ utility matrix.

Extending the Hungarian Method, we designed an algorithm called the *Interval Hungarian Algorithm* to compute both the optimal assignment and the associated interval matrix. The algorithm is also extremely efficient with a worst time complexity of $O(n^4)$ to compute all intervals for all entries. We do not repeat this algorithm in this paper due to space limitations, but summarize the idea as follows: the Interval Hungarian Algorithm purposely hides and exposes certain edges on the bipartite graph, which is the main data structure used in the Hungarian Method, and obtains the perturbation tolerances (intervals) by relaxing the searching conditions during the Hungarian Procedure. (See [5] for details.)

Having understood the essence of this method, we can extrapolate the simplified localization problem to more general scenarios; the added complexity is needed to get closer to real world problems of interest. In the examples of Figures 2 and 3, we assumed that each robot has only two uncertain locations,



(a) (b)

Fig. 5. Dark area represents uncertainty-insensitive regions. (a) The robot is located in the uncertainty-insensitive areas with high probability; (b) The robot is more likely positioned in the uncertainty-sensitive regions.

so that it is easy to determine whether an uncertain location falls into the uncertainty-(in)sensitive regions, or equivalently, whether an uncertain utility falls into the corresponding intervals. However, in practice, the estimated poses of a localization program may be subject to some continuous probabilistic distributions. In these sorts of cases, it is inappropriate to determine whether all possible uncertain utilities fall into their corresponding intervals or not. To address this issue, a *reliability score* $\mathcal{P}$ is defined to quantify the **probability** that an uncertain estimated utility belongs to a specific interval. If $\mathcal{P}$ is greater than some predefined threshold $\mathcal{T}$, one can assume it has a sufficiently large chance of falling into the interval. Figures 5(a) and 5(b) depict the idea of measuring the reliability scores for uncertainty-insensitive and uncertainty-sensitive cases, respectively.

The computation of an interval assumes other matrix entries are fixed values and are not uncertain. This assumption is too strong for practical problems. One characteristic of the utility matrix particular to multi-robot task allocation problems is that all utilities associated with the same robot-task pair can be correlated if they involve inference over the same underlying state variables. More formally, a row of a utility matrix represents all relevant expected utilities for a specific robot, and the utility estimates of all tasks basically utilize the same function with the same state variable, although with different input data. For instance, a robot with special low resolution sensors will obtain more noisy sensing data and thus always generate more uncertain utility estimates for whatever tasks. We use the term *interrelated utilities* to describe all directly related utilities in a single row or column. Fortunately, a robot-task pair has only two assignment states: assigned or unassigned (reflecting from the assignment matrix, there is only one 1 and all others are 0s in a row/column). Intervals of assigned and unassigned entries have finite lower bounds and upper bounds, respectively. (See Fig. 4 for an example.) To obtain non-conflicting intervals for all interrelated utilities of the same robot, one can opt to reduce the interval sie for both types of entries. Detailed description of method is described in [5].

The preceding discussion described the question of uncertainty and the interrelated utilities for a single robot. How about the uncertainty in the overall multi-robot system? During the execution of tasks, at a certain moment some robots may have little (or even no) uncertainty, whereas some others might suffer a great degree of uncertainty. To tackle this problem, we proposed two solutions: one solution is to simply compute the reliability score for each uncertainty robot and treat the whole system as uncertainty-insensitive only if all scores are greater than the predefined threshold $\mathcal{T}$; the other solution is more conservative, and involves raising the threshold if a larger number of robots are simultaneously subject to uncertainty. This is because the overall uncertainty increases as more uncertainty robots are involved, and hence a higher reliability threshold aims to "compensate" for the increased uncertainty in the system.

V. Validation via Physical Robots

We have conducted experiments to validate the proposed method with physical robots, and the results have shown great success in reducing task mis-allocations caused by uncertainty. The uncertain utilities in our experiments are from robots' localization errors, which are similar to the localization scenarios described in Section III.

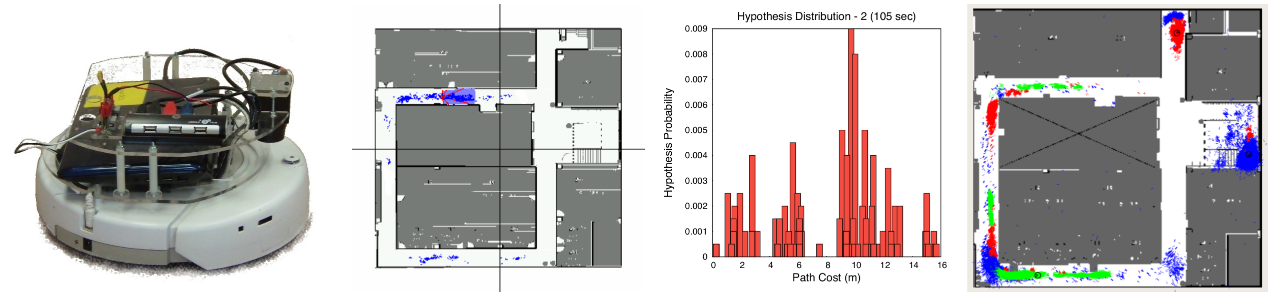
The physical robots that we used are *iRobot create* robots. A Hokuyo URG-04LX-UG01 laser range sensor is mounted on each robot to record range data up to ~5m, and an ASUS EEE netbook is carried by each robot to compute all data including the communication, utility estimation, interval analysis, *etc*. See Fig. 6(a).

We consider the problem of dispatching a group of homogeneous robots to a set of emergency locations. Each robot attempts to localize itself in the environment by employing a particle filter-based approach [1] with a given map of our research building. Particles exist only in the obstacle-free area and represent robots' pose hypotheses. The localization system plans global paths using a *wavefront* planner, and plans the local paths via the *VFH+* (Vector Field Histogram) planner [14]; implementations of both were obtained via *player* [12]. The planners return a series of way-points which connect the current pose to the goal pose, and a robot follows these waypoints to reach the assigned emergency task location. The path cost from a pose hypothesis is computed by summing up all the path-segments connecting corresponding way-points.

Fig. 6(b) shows the particles for an uncertainty-robot. A path cost distribution illustrated in Fig. 6(c) shows the effect of the uncertainty on a robots utility estimates, obtained by computing path lengths from all available hypotheses of an uncertainty-robot to a specific task location. We can see that the discrete histograms generally have the shape of Gaussian distributions.

The system is organized in a *centralized* structure in order to assess the uncertainty for the whole system and broadcast task allocation solutions or appropriate action decisions in the simplest and quickest way. A server (or an arbitrary robot) is used for the central computation, and it communicates with the member robots through the UDP protocol. Fig. 6(d) shows the particles produced from three uncertainty-robots (in different colors).

We compared the success ratio between the experiments using our method and the control experiments without any uncertainty processing. The results are shown in Table I, in which the first column represents the number of uncertainty-robots in the system. The reliability threshold $\mathcal{T}$ was slightly



(a) (b) (c) (d)

Fig. 6. (a) The physical robot used for experiments; (b) The particles of uncertain poses are generated throughout the corridors of our research building; (c) An example of cost distribution; (d) Particles generated from multiple uncertainty-robots.

TABLE I
TASK ACCOMPLISHMENT QUALITY COMPARISON

#Uncertainty robots	Method	Successes	Mis-allocation	Other failures
One	HA	50%	35%	15%
	IHA	70%	10%	20%
Two	HA	35%	45%	20%
	IHA	75%	10%	15%
Three	HA	20%	60%	20%
	IHA	60 %	20%	20%

HA and *IHA* denote the Hungarian algorithm and interval Hungarian algorithm, respectively. Each row of data is generated from 20 sets of experiments. $\mathcal{T} = 70\%$, number of particles is 10^4.

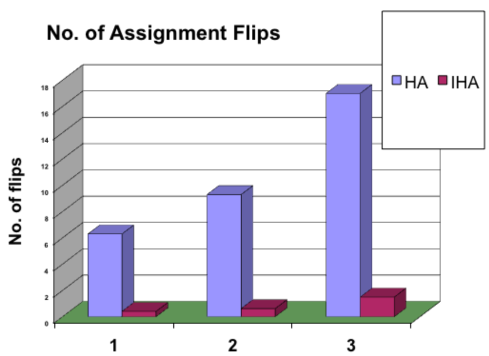


Fig. 7. Assignment flips during task executions. x-axis denotes the number of uncertainty robots in the system.

raised as more uncertainty-robots were involved, and in Table I we can clearly see that the margins of success (quantified as ratios between the proposed method and Hungarian algorithm) also increase as the number of uncertainty-robots grows. Another important advantage of our approach is that it greatly reduces the oscillatory allocation flips, as shown in Fig. 7. This is because the interval Hungarian algorithm always evaluates each uncertain utility and refuses the unreliable assignments. The Hungarian algorithm simply assigns tasks based on the instantaneous localization results, and flips frequently as the localization estimates of individual uncertainty robots change. Therefore, the robots employing our method accomplish the tasks more efficiently, and much time and energy can be spared.

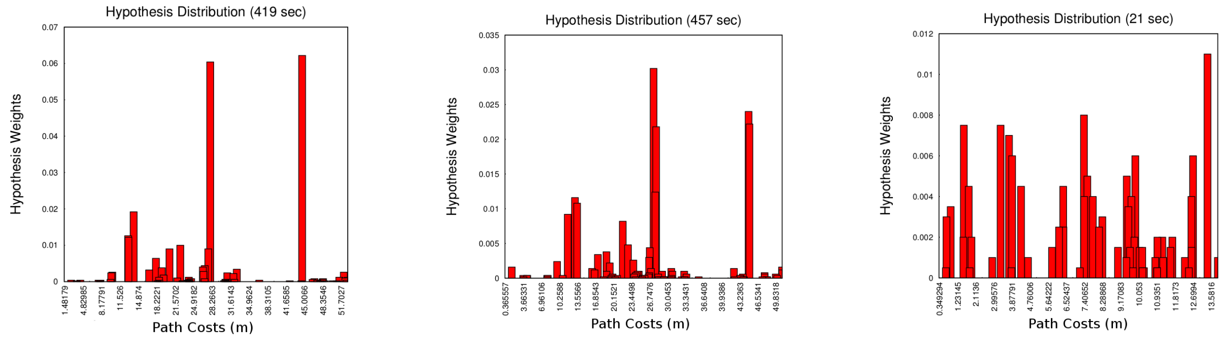
VI. DISCUSSION AND SUGGESTIONS

The results from experiments and the comparison with other popular methods reveal that the proposed method has unique advantages. However, further efforts are necessary to improve the method in order to be more robust in dealing with various uncertainty problems. Based on our experience in conducting the physical robot experiments, we provide the following suggestions in applying this method for specific problems.

A. Preprocess estimates

The Interval Hungarian Algorithm computes intervals based on the optimal assignment solution (matching in bipartite graph). As described in this paper, a slight change in a utility value might not alter the final solution, since such perturbation could be safe, *i.e.*, if it is within its allowable interval. One interesting aspect is that, after having obtained a set of intervals from a given utility matrix, if we make some safe perturbation on an arbitrary utility, and feed it again into the Hungarian method, and then feed the resultant matching solution into interval Hungarian algorithm, then both the re-produced assignment solution and the intervals remain the same. However, if an unsafe perturbation of a utility is made, the assignment solution then turns into a totally new one and the corresponding interval matrix might change dramatically thereafter (remember that the assigned entries and unassigned entries have distinct types of intervals: one goes to $+\infty$ whereas the other is from $-\infty$). This indicates that the input utility matrix has a significant impact on the output intervals. This suggests that one may do well to preprocess on the uncertainty data in order to capture the general uncertainty characteristic before feeding them to the algorithm. Thus we suggest to *get the best possible utility estimates before running the proposed method*. In our experiments, the localized pose of a robot, which is used to estimate corresponding utilities, is the pose hypothesis with largest probability. This estimate is output from the localization program by default, but, in practice, it may not be the best estimate: a larger compact cluster of hypotheses with smaller probabilities might have a larger total probability than that a single small cluster but with larger probabilities. Figures 8(a) and 8(b) illustrate such a condition: in Fig. 8(a) at the 419^{th} second the robot is localized at a place about 45 meters away (corresponding to the highest bar), however a further evaluation reveals that it is more likely that the robot is positioned at a place 25 meters away (corresponding to the second highest bar). This conjecture becomes true 38 seconds later, as shown in Fig. 8(b).

Another benefit from preprocessing is that actions may



(a) (b) (c)

Fig. 8. Distributions of path costs associated with localization hypotheses. (a) At the 419^{th} second; (b) At the 457^{th} second; (c) At the 21^{th} second.

be determined immediately after the preprocessing is done. This happens when the data preprocessing reveals that the utilities are extremely uncertain, *e.g.*, they are almost subject to uniform distributions (see Fig. 8(c), which happens at an early phase of the localization), and estimates of such utilities are meaningless in solving the assignment problem. Thus, actions for reducing uncertainties can be directly determined before bothering to use the Interval Hungarian process.

B. Accounting for various levels of uncertainty

We tested the Interval Hungarian Algorithm with different numbers of (homogeneous) robots with similar uncertainty in their localizations [5]. However, in practice, multi-robot systems can involve much more complex problems. Robots may be heterogeneous in such that the type of utility distributions and the levels of uncertainty can be quite distinct. In fact, sometimes it is hard to determine whether a utility is "certain" or "uncertain" since the borderline is ill-defined. (In our experiments, we considered a robot to be completely localized without uncertainty if all particles were densely clustered, but particles would never converge into a single point which in fact is the ideal absolutely localized pose.) If the robots have different levels of uncertainty (*e.g.*, Gaussian distributions with distinct standard deviations), then the reliability threshold determination step can be modified to further improve the assessment performance. More formally, in the preceding work we used a conservative approach to coarsely measure the system level uncertainty, with such a formula:

$$\mathcal{T} = \mathcal{T}_0 + \exp[-(n - n_x)] \cdot (1 - \mathcal{T}_0), \quad (1 < n_x \leq n) \quad (1)$$

where n, n_x and $\mathcal{T}_0$ are the total number robots, the number of uncertainty-robots, and the initial threshold for a single uncertainty-robot, respectively. The parameter n_x treats all uncertainty-robots uniformly and does not take their respective levels of uncertainty into account. A more sophisticated way to compute n_x is through weighted counting, where the weights are obtained by normalizing the levels of uncertainty associated with robots. For example, the uncertainty level of a Gaussian distribution can be associated with its standard deviation.

VII. Conclusion

An efficient and reliable method to assess the impact of uncertainty on system-wide goals is very important. This paper provides new interpretations of our previously introduced Interval Hungarian algorithm to complement prior descriptions: it grounds the discussion of uncertainty and its meaning in a concrete robotic problem, *i.e.,* dispatching to particular emergency exits in a building. The algorithm is described step by step using the uncertainty arising from scenarios in which probabilistic localization is employed. We outline why and how we designed this method. Then we provide an extended comparison and analysis of this algorithm, and outline lessons learned in applying this method on physical mobile robots. This information is valuable in designing the risk-averse assignment algorithms and addressing the practical task allocation problems under uncertainty.

References

[1] D. Fox, "KLD-sampling: Adaptive particle filters," in *Advances in Neural Information Processing Systems (NIPS-14)*, 2001, pp. 713–720.

[2] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, Sept. 2004.

[3] M. B. Dias and A. T. Stentz, "A market approach to multirobot coordination," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI -TR-01-26, August 2001.

[4] L. Liu and D. Shell, "Assessing optimal assignment under uncertainty: An interval-based algorithm," in *Proc. of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[5] ——, "Assessing Optimal Assignment under Uncertainty: An Interval-based Algorithm," *Intnl. Journal of Robotics Research 30(7)*, pp. 936–953, 2011.

[6] D. S. Bernstein, S. Zilberstein, and N. Immerman, "The complexity of decentralized control of Markov Decision Processes," in *Conference on Uncertainty in Artificial Intelligence*, June 2000.

[7] M. Roth, R. Simmons, and M. Veloso, "What to Communicate? Execution-Time Decision in Multi-agent POMDPs," in *Proceedings of the eighth International Conference on Distributed Autonomous Robotic Systems (DARS'06)*, Minneapolis, MN, U.S.A., July 2006, pp. 177–186.

[8] J. Capitan, M. T. J. Spaan, L. Merino, and A. Ollero, "Decentralized multi-robot cooperation with auctioned pomdps." in *ICRA*. IEEE, 2012.

[9] J. L. Higle and S. W. Wallace, "Sensitivity analysis and uncertainty in linear programming," *Interfaces*, vol. 33, no. 4, pp. 53–60, July 2003.

[10] S. W. Wallace, E. Nationale, and S. G. Industriel, "Decision making under uncertainty: Is sensitivity analysis of any use?" *Operations Research*, vol. 48, 1998.

[11] C. Guestrin, S. Venkataraman, and D. Koller, "Context-Specific Multiagent Coordination and Planning with Factored MDPs," in *Proceedings of the eighteenth AAAI National Conference on Artificial Intelligence (AAAI'02)*, Edmonton, Alberta, Canada, July 2002, pp. 253–259.

[12] B. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proc. Intl. Conf. on Advanced Robotics*, 2003.

[13] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistic Quarterly*, 1955.

[14] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots," in *Proc. Intl. Conf. on Robotics and Automation*, Leuven, Belgium, 1998, pp. 1572–1577.