Assessing Optimal Assignment under Uncertainty: An Interval-based Algorithm

Lantao Liu
Computer Science and Engineering Department
Texas A&M University
College Station, USA
Email: lantao@cse.tamu.edu

Dylan A. Shell
Computer Science and Engineering Department
Texas A&M University
College Station, USA
Email: dshell@cse.tamu.edu

***Abstract*— We consider the problem of multi-robot task-allocation when robots have to deal with uncertain utility estimates. Typically an allocation is performed to maximize expected utility; we consider a means for measuring the robustness of a given optimal allocation when robots have some measure of the uncertainty (e.g., a probability distribution, or moments of such distributions). We introduce a new $O(n^4)$ algorithm, the Interval Hungarian algorithm, that extends the classic Kuhn-Munkres Hungarian algorithm to compute the maximum interval of deviation (for each entry in the assignment matrix) which will retain the same optimal assignment. This provides an efficient measurement of the tolerance of the allocation to the uncertainties, for both a specific interval and a set of interrelated intervals. We conduct experiments both in simulation and with physical robots to validate the approach and to gain insight into the effect of location uncertainty on allocations for multi-robot multi-target navigation tasks.**

I. Introduction

Task-allocation mechanisms are among the most successful non-domain-specific means for coordinating the actions of multiple robots. These involve treating the work to be performed as a set of tasks and the robots themselves as workers to be assigned to particular tasks. By estimating the expected utility of a particular robot's performance of a particular task, algorithms can optimize the allocation of robots to tasks (or vice versa) in order to maximize expected collective performance. The complexity of the allocation problem depends on the particular capabilities required to achieve the task, the capabilities of the robots, and whether the allocation must consider temporal scheduling aspects [1]. Generally the appropriateness of the task-allocation approach and the difficulty of the allocation problem both depend on the degree to which each of the robots and each of the tasks can be considered independent.

We consider the archetype multi-robot task-allocation problem which involves performing an instantaneous assignment of single-task robots to single-robot tasks (following the taxonomic characterization in [1]). This reduces to an instance of the well-studied Optimal Assignment Problem (OAP) for which the Kuhn-Munkres Hungarian algorithm, which was first proposed by H. W. Kuhn [2] in 1955 and improved by J. Munkres [3] in 1957, is a solution.

For multi-robot task-allocation, however, outstanding issues remain. The Hungarian algorithm maximizes the utility for the team because it is provided with an estimate of each robot's expected utility. Calculating this estimate is costly because every robot must provide estimates for each task, and the optimality of the resultant allocation is only meaningful when these estimates are accurate. Furthermore, the robots each have to deal with uncertainty about the state of the world in constructing these estimates. Even if the robots maintain a representation of this uncertainty (*e.g.,* a distribution over potential states) the expected utility is only the first moment of the utility distribution given a particular robot-task assignment pair. Important questions are: (1) How much effort should the robots invest in constructing the utility estimates? For n robots and n tasks, n^2 estimates are provided—but only n elements make up the optimum assignment; not all utility estimates need to be known with equal fidelity. (2) Once an allocation is computed, how stable is that allocation with respect to changes in the matrix of utility estimates? (3) If these utility estimates arise from an underlying probability distribution, what is the likelihood that the assignment is sub-optimal?

This paper makes the following contributions toward addressing these questions:

- Identification of properties of intervals of utility estimates, and proofs of these properties.
- Introduction of a new algorithm, the Interval Hungarian Algorithm, ideally suited to multi-robot systems, although broadly applicable to any OAP where the matrix of utility estimates is subject to uncertainty.
- Introduction of an efficient method for quantifying the effects of uncertainty on an allocation. This includes analysis for instances with a single specific estimate and multiple interrelated estimates.
- Analysis of the impact of uncertainty with concrete examples, using standard localization methods to produce real utility distributions with physical robots and in simulation.

II. Related Work

It is worthwhile drawing a distinction between multi-robot coordination strategies that employ assignment methods in which expected utilities for each robot's task performance is provided, and those that model the task performance itself in greater detail. The former depend on an independence assump-

tion, because a linear function (like the sum) of the group's utilities is optimized. Other effects, like interference or inter-robot synergy, can only be captured in the way they effect particular utilities in the expectation. Centralized and distributed algorithms for performing allocations have been developed, including greedy allocations [4], optimization techniques [1, 5] and auction [6, 7] and market-based approaches [8, 9]. It is within this framework that the present study falls. It uses little information about the domain-specific aspects that lead to the structure of the coordination problem, or even the source of the uncertainty. However, aspects like interrelated utilities (or in the present study, more generally, interrelated uncertainty in the utilities) are not explicitly captured. If the effect of these higher-order interactions on utility values can be calculated, then the presented algorithm can still be of use.

Schemes that use a richer model of agent and task in order to construct a probabilistic model, for example, stochastic games/decentralized-MDPs [10], factored-MDPs,[11], POMDPs [12], permit one to address explicitly the question of when to perform particular actions (movement, sensing, communication) in order to reduce uncertainty if doing so is beneficial for the task performance. However, these problems do not admit polynomial-time solutions, and often factorization or independence assumptions are introduction in order to make the problem tractable.

Algorithm II.1 The Hungarian Algorithm

Input:

A valid $n \times n$ assignment matrix represented as the equivalent complete weighted bipartite graph $G = (X, Y, E)$, where $|X| = |Y| = n$.

Output:

A perfect matching, M.

1: Generate an initial labelling l and matching M in G_e.

2: If M perfect, terminate algorithm. Otherwise, randomly pick an exposed vertex $u \in X$. Set $S = \{u\}$, $T = \varnothing$.

3: If $N(S) = T$, update labels:
$\delta = min_{x \in S, y \in Y - T}\{l(x) + l(y) - w(x, y)\}$

$$l'(v) = \begin{cases} l(v) - \delta & if\ v \in S \\ l(v) + \delta & if\ v \in T \\ l(v) & otherwise \end{cases}$$

4: If $N(S) \neq T$, pick $y \in N(S) - T$.
(a) If y exposed, $u \rightarrow y$ is augmenting path. Augment M and go to step 2.
(b) If y matched, say to z, extend Hungarian tree: $S = S \bigcup \{z\}$, $T = T \bigcup \{y\}$, and go to step 3.

$*$ Definitions:

- Equality graph $G_e = \{e(x, y) : l(x) + l(y) = w(x, y)$;
- Neighbor $N(u)$ of vertex $u \in X$: $N(u) = \{v : e(u, v) \in G_e\}$.

III. Hungarian Algorithm

The Hungarian Algorithm treats the optimization OAP as a combinatorial problem in order to efficiently solve an $n \times n$ task assignment problem in $O(n^3)$ time. The utility estimates become edge weights in a complete bipartite graph in which each robot and task becomes a vertex. The Hungarian Algorithm (Algorithm II.1) searches for a perfect matching in a sub-graph of the complete bipartite graph, where the perfect matching is exactly the optimal assignment solution. In step 4 the search process either increases the matching size, or the so-called *equality graph* in which the matching resides. (Graph and matching definitions and notation are adopted from [13].)

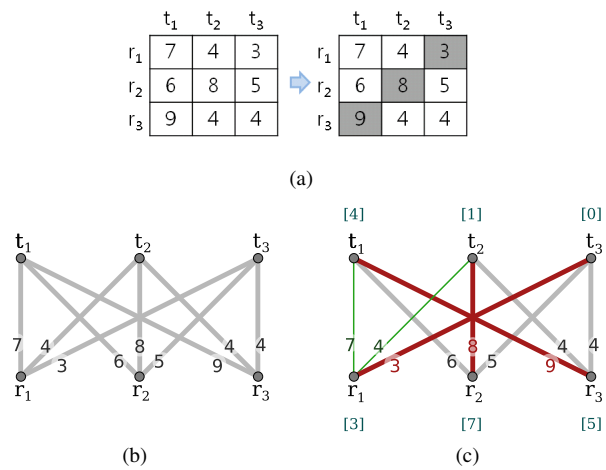


Fig. 1. Hungarian algorithm solves the OAP using a complete bipartite graph. (a) An assignment matrix and solution; (b) A complete bipartite graph; (c) The perfect matching for assignment solution. The state of the graph after an assignment, we term the *resultant bipartite graph*.

Figure 1 shows an example assignment problem and the corresponding perfect matching in form of the associated bipartite graph. In Figure 1(a), the task assignment problem is described as an assignment matrix (which need not be square in general). An element u_{ij} is an estimated utility, meaning that robot r_i has an expected utility when assigned to perform task t_j. The algorithm generates the maximal allocation, shown as shaded cells in the matrix. Figs. 1(b) and 1(c) show the same information but in bipartite graph form. Edges that are used during the calculation are differentiated by 3 colors: red (bold dark) edges comprise the matching, green (thin dark) edges are unmatched but they are in the equality graph G_e, grey (bold light) edges are also unmatched but do not appear the equality graph. Edges within G_e are termed admissible (both red and green edges) and will have weights that satisfy $w_{ij} = l(r_i) + l(t_j)$. Bracketed integers beside each vertex represent the labeling value $l(\cdot)$. Edge set M and scalar m represent a specific perfect matching solution and the corresponding optimum value, respectively.

Since the development of the Hungarian Algorithm many variations have been proposed (see [14] for a recent survey). Two relate directly to this paper: [15] provide an incremental Hungarian Method, which is an $O(n^2)$ technique for inserting a pair of new vertices in the bipartite graph resulting from a previous assignment. An extension to this in [16] also permits deletions, which enables one to solve assignment problems with k utility changes in $O(kn^2)$ time. The current work computes a description of an assignment problem (in $O(n^4)$)

which permits subsequent $O(1)$ queries of whether or not the assignment has changed. When assignments are being recomputed frequently with small absolute changes in value (*e.g.*, frequent replanning of distances as a robot moves) this can be a considerable saving.

IV. Interval Hungarian Algorithm

For each utility value, we compute the interval in which the utility may be (independently) perturbed before the optimality of the computed assignment is violated. Thus, given an input matrix of utilities, the algorithm characterizes a set of inputs which yield the same output. The intervals are computed based on the three categories of edges described above.

A. Interval Analysis for Matched Edges

The allowable intervals for matched edge weights is analyzed as follows: for any such edge $e_m(r_\alpha, t_\beta)$, the interval can be described as $[w_{m\alpha\beta} - \varepsilon_m, +\infty)$, where $w_{m\alpha\beta}$ is the edge weight of $e_m(r_\alpha, t_\beta)$ and ε_m is the *tolerance margin* that the weight can decrease without violating the optimality of the current matching solution. It is safe to increase the weight as this is a maximization problem. We say a matched edge is *hidden* if its weight has decreased so as to no longer form part of a matching solution.

Lemma 4.1: With the resultant matching solution M_0 and bipartite graph of Hungarian algorithm, if a matched edge $e_m(r_\alpha, t_\beta)$ is hidden, then the Hungarian algorithm can be completed with one iteration rooted at exposed node r_α. When a new perfect matching solution M' exists, the labeling reduction of the root r_α satisfies $l(r_\alpha) - l'(r_\alpha) = m_0 - m'$.

Proof: The proof is based on the Hungarian algorithm solution method. If one matched edge $e_m(r_\alpha, t_\beta)$ is hidden, the bipartite graph remains feasible, but the equality graph G_e loses one matched edge. Hiding $e_m(r_\alpha, t_\beta)$ exposes r_α and t_β, requiring an iteration to complete the Hungarian algorithm. All other unexposed vertices are on corresponding matched edges and, thus, are included in G_e, and moreover the sum of their labels is constant for all subsequent iterations. To grow G_e in bridging a new augmenting path[1], $l(r_\alpha)$ may decrease, but $l(t_\beta)$ remains constant because it is the end of augmenting path and will be reached last. Therefore, $l(r_\alpha)$ is the only variable that can reduce the optimum. This proves that the reduction of $l(r_\alpha)$ is exactly the reduction of optimum from M_0 to M', namely, $l(r_\alpha) - l'(r_\alpha) = m_0 - m'$. ■

*Theorem 4.2 (**Matched Edge Interval**):* Hiding a matched edge from the Hungarian solution leads to a new solution, and the labeling reduction ε_m at the root of the Hungarian tree is the tolerance margin for this element, *i.e.*, the safe interval for matched edge $e_m(r_\alpha, t_\beta)$ is $[w_{m\alpha\beta} - \varepsilon_m, +\infty)$.

Proof: Assigning weight $w = w_{m\alpha\beta} - \varepsilon_m$ to edge $e_m(r_\alpha, t_\beta)$ results in optimum $m_0 - \varepsilon_m$ for the original matching M_0. Any new matching M' which excludes $e_m(r_\alpha, t_\beta)$ also has an optimum of $m' = m_0 - \varepsilon_m$, therefore both M_0 and M' are optimal matching solutions. Whenever the weight of edge $e_m(r_\alpha, t_\beta)$ satisfies $w \geq w_{m\alpha\beta} - \varepsilon_m$, we have $m_0 \geq m'$ implying that M_0 is optimal, since otherwise M_0 would be substituted by M'. Thus, the maximum allowable interval for matched edge $e_m(r_\alpha, t_\beta)$ is $[w_{m\alpha\beta} - \varepsilon_m, +\infty)$. ■

[1]Although the definition of *augmenting path* is not necessary to understand Algorithm II.1, we note that here we deviate from [13, pg. xxxii], where they use the term M-augmenting path.

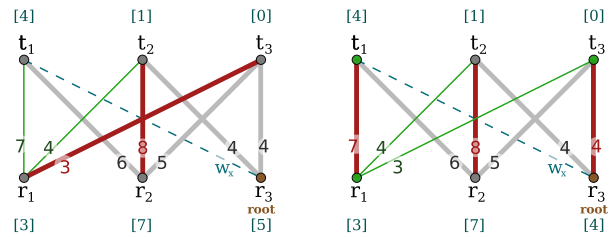


Fig. 2. Interval analysis for a matched edge. (a) Hide an objective matched edge and assign it with weight w_x; (b) New matching solution without the hidden edge.

Algorithm IV.1 Intervals of Matched Edges

Input:

A matched edge $e_m(r_\alpha, t_\beta)$ and the corresponding resultant bipartite graph.

Output:

Interval (lower bound ξ_m) for $e_m(r_\alpha, t_\beta)$.

1: Hide $e_m(r_\alpha, t_\beta)$ by assigning it an unknown weight w_x. Set $S = \{r_\alpha\}$, $T = \varnothing$.

2: If $N(S) = T$, update labels:
$\delta = min_{x \in S, y \in Y - T, e(x,y) \neq e_m(r_\alpha, t_\beta)} \{l(x) + l(y) - w(x, y)\}$

$$l'(v) = \begin{cases} l(v) - \delta & if\ v \in S \\ l(v) + \delta & if\ v \in T \\ l(v) & otherwise \end{cases}$$

$l'(r_\alpha) + l'(t_\beta) - w_x > \delta \Rightarrow w_x < l'(r_\alpha) + l'(t_\beta) - \delta$.
Update $\xi_m = l'(r_\alpha) + l'(t_\beta) - \delta$.

3: If $N(S) \neq T$, pick $y \in N(S) - T$.
(a) If $y = t_\beta$, there must be an augmenting path $r_\alpha \to t_\beta$. Augment matching and terminate algorithm.
(b) If y matched, say to z, extend Hungarian tree: $S = S \bigcup \{z\}$, $T = T \bigcup \{y\}$. Go to step 2.

Lemma 4.1 and Theorem 4.2 permit computation of the interval of a matched edge $e_m(r_\alpha, t_\beta)$ in the following way: first hide $e_m(r_\alpha, t_\beta)$ from the bipartite graph and assign it an undecided weight w_x that satisfies the constraint: $w_x < l(r_\alpha) + l(t_\beta)$. Next, let exposed vertex r_α be the root of a Hungarian tree and construct an augmenting path excluding $e_m(r_\alpha, t_\beta)$. The algorithm terminates when such a path is found that generates a perfect matching. Because $l(t_\beta)$ stays unchanged but $l(r_\alpha)$ is decreased, w_x will decrease per iteration; the lower bound $w_{m\alpha\beta} - \varepsilon_m$ occurs moment when the new perfect matching exists.

Figure 2 illustrates an example. Hiding matched edge $e(r_3, t_1)$ requires construction of a Hungarian tree rooted at newly exposed vertex r_3. Here $l(r_3)$ decreases while searching for an augmenting path, and a new matching solution

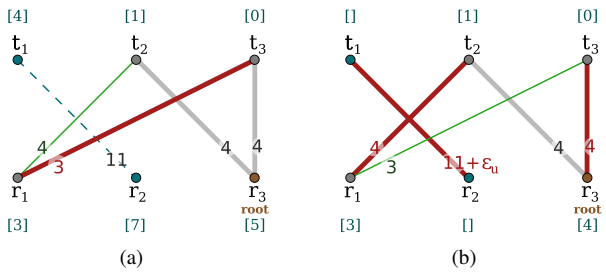


Fig. 3. Interval analysis for an unmatched edge. (a) Hide all associated edges of objective unmatched edge; (b) New matching solution formed with the objective edge and matching solution in auxiliary bipartite graph.

$\langle e(r_1,t_1), e(r_2,t_2), e(r_3,t_3)\rangle$ replaces the original one in augmenting path $r_3 \to t_3 \to r_1 \to t_1$. The reduction of labeling for r_3 is $5-4=1=\varepsilon_m$, and the interval for $e(r_3,t_1)$ is $[8,+\infty)$.

B. Interval Analysis for Unmatched Edges

An unmatched edge $e_u(r_\alpha,t_\beta)$ has an interval $(-\infty,\xi_u]$, where the upper bound ξ_u reflects the maximum value of the utility for it to remain an unmatched robot and task pair.

Lemma 4.3: In the resultant bipartite graph of the Hungarian algorithm, the weight of any unmatched edge $e_u(r_x,t_y)$ can be increased to the sum of two associated labeling values $l(r_x)+l(t_y)$ without affecting the assignment optimum.

Proof: An increment of this form does not violate the feasibility of the bipartite graph, nor does the modification remove any admissible edges from G_e. Therefore, the original matching in G_e remains perfect. ■

However, the upper bound in Lemma 4.3 is not tight. For example, $e(r_2,t_1)$ in Figure 1(c), the weight can safely increase to 12 rather than $4+7=11$ (12 is the upper bound because greater weights result in optimal matching $\langle e(r_1,t_2), e(r_2,t_1), e(r_3,t_3)\rangle$), and there is a tolerance margin of 1. Next we show how to find the tolerance margin ε_u and further improve the upper bound. At present we redefine the interval as $(-\infty, l(r_\alpha)+l(t_\beta)+\varepsilon_u]$. Note that this also shows that all unmatched edges, whether in G_e or not, can be treated uniformly once they become admissible.

To obtain ε_u for unmatched edge $e_u(r_\alpha,t_\beta)$, hide $e_u(r_\alpha,t_\beta)$ and all other edges incident to vertices r_α and t_β from the resultant bipartite graph. This yields a bipartite graph with $n-1$ vertices in each partition. We term this new bipartite graph the *auxiliary* bipartite graph G_a. Notice that auxiliary bipartite graph is associated with a particular edge, and that the auxiliary bipartite graph has only $n-2$ matched edges. It therefore requires the addition of one edge for a matching solution.

*Theorem 4.4 (**Unmatched Edge Interval**):* Any unmatched edge $e_u(r_\alpha,t_\beta)$ in the Hungarian resultant bipartite graph, has interval tolerance margin $\varepsilon_u = m_0-(m_a+l(r_\alpha)+l(t_\beta))$, where m_0 is the optimum of the original solution, and m_a is the optimum of the auxiliary bipartite graph associated with $e_u(r_\alpha,t_\beta)$. The allowable interval for edge $e_u(r_\alpha,t_\beta)$ is $(-\infty, m_0-m_a]$.

Proof: For an arbitrary unmatched edge $e_u(r_\alpha,t_\beta)$ and its associated auxiliary bipartite graph G_a of size $n-1$, we add $e_u(r_\alpha,t_\beta)$ to the matching solution M_a of G_a. This forms a new matching M' of size n. If the weight of edge $e_u(r_\alpha,t_\beta)$ satisfies $w_{u\alpha\beta} > m_0-m_a$, then the matching M' containing $e_u(r_\alpha,t_\beta)$ must satisfy $m' = w_{u\alpha\beta}+m_a > m_0$. But this contradicts the fact that the original matching M_0 was perfect. This proves that the upper bound for unmatched edge $e_u(r_\alpha,t_\beta)$ is m_0-m_a, and that the allowable interval is $(-\infty, m_0-m_a]$. ■

Algorithm IV.2 Intervals of Unmatched Edges

Input:

An unmatched edge $e_u(r_\alpha,t_\beta)$ and the corresponding resultant bipartite graph.

Output:

Interval (upper bound ξ_u) for $e_u(r_\alpha,t_\beta)$.

1: Assume $e(r_\alpha, mate(r_\alpha))$, $e(t_\beta, mate(t_\beta))$ are matched edges, then set $S=\{mate(t_\beta)\}$, $T=\varnothing$.

2: Hide $e_u(r_\alpha,t_\beta)$ and all other edges incident to vertices r_α and t_β, and obtain the auxiliary bipartite graph G_a.

3: In G_a, if $N(S)=T$, update labels:
$\delta = min_{x\in S, y\in Y-T}\{l(x)+l(y)-w(x,y)\}$

$$l'(v) = \begin{cases} l(v)-\delta & if\ v\in S \\ l(v)+\delta & if\ v\in T \\ l(v) & otherwise \end{cases}$$

4: In G_a, if $N(S)\neq T$, pick $y\in N(S)-T$.
(a) If $y=mate(r_\alpha)$, there must be an augmenting path $mate(t_\beta)\to mate(r_\alpha)$. Augment matching and go to step 5.
(b) If y matched, say to z, extend Hungarian tree: $S = S\bigcup\{z\}$, $T=T\bigcup\{y\}$. Go to step 3.

5: $\xi_u = m_0-m_a$.

$*$ Definitions:

- $mate(v)$ is the other ending vertex with regard to vertex v;
- m_0 is optimum of the original solution, m_a is optimum of G_a.

Consider Figure 3 as an example. Hiding $e(r_2,t_1)$ and all edges incident to r_2 and t_1 leaves auxiliary bipartite graph G_a containing vertices r_1, r_3, t_2, t_3 and associated edges (see Figure 3(a)). The Hungarian tree is created rooted at newly exposed vertex r_3, and finally the matching solution M_a of G_a is $\langle e(r_1,t_2), e(r_3,t_3)\rangle$, as shown in Figure 3(b). The tolerance margin and allowable interval for $e(r_2,t_1)$ are $\varepsilon_u = m_0-(m_a+l(r_2)+l(t_1))=1$ and $(-\infty,12]$, respectively.

C. Interval Hungarian Algorithm

Combining the interval analysis of matched and unmatched edges, we have the Interval Hungarian algorithm described in Algorithm IV.3. Figure 4 shows the corresponding intervals for assignment matrix in Figure 1(a).

D. Computational Complexity

The algorithm has worst-case time complexity $O(n^4)$. The Hungarian algorithm has a computational complexity of $O(n^3)$. Obtain the intervals for n matched edges, needs an

Algorithm IV.3 Interval Hungarian Algorithm

Input:

A resultant bipartite graph from running Algorithm II.1.

Output:

An interval matrix $mx_{itv}(n,n)$ storing all intervals.

1: $mx_{itv}(n,n) =$ NULL.
2: for all edges $e(i,j)$ in bipartite graph
  if $e(i,j)$ is matched
    compute interval $I(i,j)$ with Algorithm IV.1.
    $mx_{itv}(i,j) = I(i,j)$.
  else
    compute interval $I(i,j)$ with Algorithm IV.2.
    $mx_{itv}(i,j) = I(i,j)$.
3: return mx_{itv}.

	t_1	t_2	t_3
r_1	(-∞,8]	(-∞,6]	[2,+∞)
r_2	(-∞,12]	[6,+∞)	(-∞,7]
r_3	[8,+∞)	(-∞,8]	(-∞,5]

Fig. 4. An example of an interval matrix

extra $O(n \times n^2) = O(n^3)$ operations since for each edge requires construction and search of the Hungarian tree costing $O(n^2)$. The computation of all the unmatched edges has a worst case cost of $O(n^4)$ since, for each unmatched edge, we do the same searching iterations in G_a, which has a size of $n-1$ and thus needs a computational complexity of $O((n-1)^2)$. There are $n^2 - n$ unmatched edges, yielding a total of $O((n^2-n) \times (n-1)^2) = O(n^4)$.

V. Quantifying the Effect of Uncertainty

When the Hungarian algorithm is applied to a matrix of expected utilities calculated from uncertain data (*e.g.*, using the mean of a utility distribution) one has little idea of the impact the uncertainty has on the resultant assignment. The output from the Interval Hungarian algorithm can be used to analyze the changes in optimal allocation as changes are made to particular utility values. This can be used to evaluate likelihood that the calculated assignment will be sub-optimal.

A. Uncertainty Measurement for a Single Utility

*Theorem 5.1 (**Uncertainty of a Single Interval**):* With regard to any specific single utility value, assuming other utilities are certain, the perfect matching solutions are identical if and only if any specific utility is within its allowable interval.

Proof: Assuming other utilities in the assignment matrix are certain, taken together Theorems 4.2 and 4.4 prove that if the matching solution remains the same, then any specific utility must be within its allowable interval. To prove the converse: suppose the matching solutions are not identical, then there must be a new perfect matching M' that replaces the original one M_0 *i.e.*, $m' > m_0$. However, this cannot happen since any value within interval must produce a matching solution with weight sum less than or equal to m_0. ■

To analyze the effect of uncertainty on a specific utility in the assignment matrix, we assume the other values are certain. Given a probability density function $f(x)$ for this specific expected utility, and associated interval I as output from the algorithm, the probability of a sub-optimal assignment is:

$$P_I = \begin{cases} \int_{\xi_m}^{+\infty} f(x), & \text{when } I = [\xi_m, +\infty) \\ \int_{-\infty}^{\xi_u} f(x), & \text{when } I = (-\infty, \xi_u]. \end{cases} \quad (1)$$

For applications in which robots are actively estimating quantities involved in producing $f(x)$, one may set some threshold T, such that robots only commit to an assignment if $P_I \geq T$, and instead invest resources in reducing the uncertainty in the estimate if it is likely to have a major bearing on the particular assignment. High values of T will ensure the robots only commit to allocations that are robust to errors in estimates of the expected utility.

B. Uncertainty Measurement for Interrelated Utilities

The previous subsection gives an approach for quantifying the effect of uncertainty on the robot-to-task assignment when only one utility was uncertain. Most often, however, multiple utilities are uncertain, and they may all be related if they involve inference over the same underlying state variables. For example, a row in the assignment matrix represents all relevant expected utilities for a specific robot. A change that effects the performance of the robot (*e.g.,* low battery) effects all the entries in the row. Here we use the term *interrelated edges* to represent all directly related utilities in a single row or column. For the same assignment to be preserved, despite n interrelated edges, there must be one and only one edge that remains matched, and all the others should be unmatched.

*Theorem 5.2 (**Uncertainty of Interrelated Intervals**):* Given a set of n interrelated edges, assume e_m is the matched edge with interval $[w_m - \varepsilon_m, +\infty)$, and e_{ui} are unmatched edges with intervals $(-\infty, w_{ui} + \varepsilon_{ui}]$, $(i = 1, 2, ..., n-1)$, then for any $\varepsilon' \leq \varepsilon_m$, the weight of e_m can be safely substituted with $w_m - \varepsilon'$, and the interval for e_{ui} becomes $(-\infty, w_{ui} + \varepsilon_{ui} - \varepsilon']$, $(i = 1, 2, ..., n-1)$.

Proof: $\varepsilon' \leq \varepsilon_m$ indicates that new weight $w'_m = w_m - \varepsilon'$ is within the interval associated with edge e_m, thus a substitution of w'_m will not violate the matching solution. To prove the interval form for e_{ui} $(i = 1, 2, ..., n-1)$: suppose m'_0 is the solution optimum with substituted weight w'_m, then we have $m'_0 = m_0 - \varepsilon'$. From Theorem 4.4, the interval for e_{ui} is $(-\infty, m'_0 - m_a]$, which can be substituted with $(-\infty, m_0 - \varepsilon' - m_a]$. Because $m_0 - m_a = w_{ui} + \varepsilon_{ui}$, the interval for e_{ui} becomes $(-\infty, w_{ui} + \varepsilon_{ui} - \varepsilon']$. ■

Notice that Theorem 5.2 exploits the mutual exclusion property of interrelated unmatched edges: at any time one and only one interrelated unmatched edge can possibly become matched. This means that as the matched edge's weight is decreased, one unmatched edge moves closer to it's interval's upper-bound. However, as the matched edge's value decreases, the unmatched edge's bounds decrease too. To allow the simultaneous occurrence of interrelated matched edge and

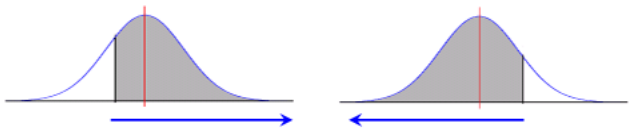


Fig. 5. Reliability levels (shaded area) with Gaussian distribution density. The horizontal axis represents the uncertain utility estimates.

unmatched edges, we compromise between them: intervals of interrelated unmatched edges shrinks by ε', while interval for interrelated matched edge shrinks by $\varepsilon_m - \varepsilon'$. We design the method for measuring the uncertainties of interrelated edges in the following steps:

1) Determine ε_{min} from all interrelated edges: $\varepsilon_{min} = min(\varepsilon_m, \varepsilon_{ui}),\ (i = 1, 2, ..., n-1)$
2) Determine each interrelated interval I_i:

$$I_i = \begin{cases} [w_m - k \cdot \varepsilon_{min}, +\infty) \\ (-\infty, w_{ui} + \varepsilon_{ui} - k \cdot \varepsilon_{min}], (i = 1, 2, ..., n-1) \end{cases}$$

$*I_0$ represents interval for the matched edge. k is an empirical coefficient and $k \in [0, 1]$ which effects the degree to which the matched and unmatched interval's are scaled.

3) Determine probability:

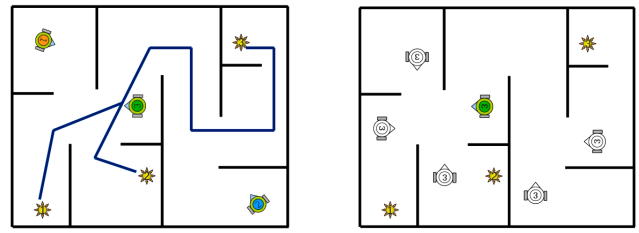
$$P_{Ii} = \begin{cases} \int_{\xi'_m}^{+\infty} f(x),\ (\xi'_m = w_m - k \cdot \varepsilon_{min}) \\ \int_{-\infty}^{\xi'_{ui}} f(x),\ (\xi'_{ui} = w_{ui} + \varepsilon_{ui} - k \cdot \varepsilon_{min}) \end{cases} \quad (2)$$

4) Determine reliability level: The assignment is reliable when $P_{Ii} \geq T$, and unreliable otherwise.

This approach utilizes a parameter k to balance the shrinking intervals (generally, $k \to 0$ when the number of robots is larger, which controls the loss of compromised ranges). In our experiments involving 3 robots, we use $k = 0.5$ to guarantee a margin of at least $0.5\varepsilon_{min}$ from each estimated utility). This is most effective for Gaussian-like distributions in which mean and medians of the distributions are close. Figure 5 illustrates the reliability levels (shaded area) under a Gaussian distribution density for the matched and unmatched edges, respectively. This method can measure the uncertainties for a horizontal row or a vertical column in the assignment matrix, assuming other non-directly interrelated rows or columns are known with certainty. The uncertainty involving multiple rows or columns is complex and beyond the scope of this paper. In practical applications, however, a conservative approach is simply to raise the threshold T so that the inaccuracy arising from multiple interrelated rows or columns can be compensated for.

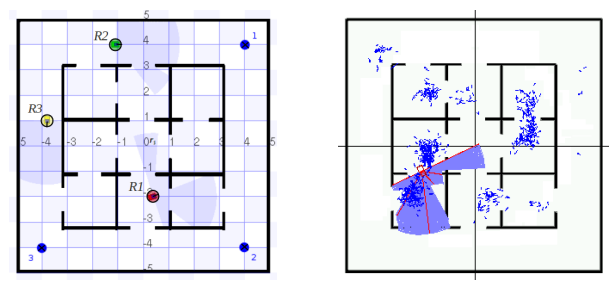
VI. Experiments and Results

We demonstrate, through simulation and physical robot experiments, that the Interval Hungarian algorithm permits the effect of uncertainty on the allocation to be quantified. We consider the problem of dispatching a group of homogeneous robots to a set of destination locations. We selected this task because it (and variations to it) have been used in the literature for the purpose of evaluating task assignment methods (*e.g.,* [7] and subsequent paper by Koenig's group). Additionally, the source of uncertainty and the means for actively estimating this has been well-studied in the last decade. Other tasks (*e.g.,* Parker's [4] Cooperative Multi-robot Observation of Multiple Moving Targets (CMOMMT) task) have different forms of uncertainty, but once a utility matrix is constructed for purposes of robot task-allocation, the method applies directly.



(a) (b)

Fig. 6. Multi-robot task assignment using localization. (a) Planned path with wavefront driver; (b) Possible localization results of robot #3.



(a) (b)

Fig. 7. (a) Multi-robot task assignment and commitment in a maze; (b) Uncertain hypothesis for uncertainty-robot (robot $R1$ in red).

The assignment matrix is a negative estimation of the distance from current location to goal location and the uncertainty in the assignment problem comes from localization error. The robot attempts to localize itself in the environment (with a given map) by employing an particle filter-based approach [17]. Particles are clustered and the weighted means of these clusters taken as pose hypotheses. A planner is used to estimate the cost from estimated location to a given goal location. Figure 6 illustrates three robots being dispatched to three task locations in a maze-like room. The line segments connecting to the task locations in Figure 6(a) are the planned paths, and the path cost is the total length of each path. Figure 6(b) shows the possible poses of robot #3.

A. Simulation

1) Experimental Setup: The simulation environment is designed as in Figure 7(a) within the Stage simulator [18]. The maze was designed to be somewhat symmetric so as to provide multiple uncertain poses for the localization method. Three robots were randomly positioned within the maze and dispatched to the three task locations shown as big dots in the corners of Figure 7(a). Of the three robots, only one

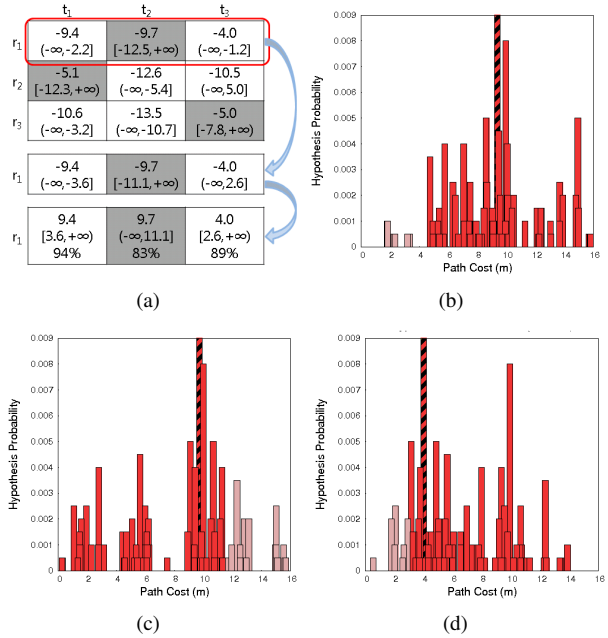


Fig. 8. (a) Uncertainty analysis of interrelated intervals for robot $R1$; (b)—(d) Hypothesis distributions of interrelated utilities captured at 105th second.

(Robot $R1$ in red color, named *uncertainty-robot*) is using the localization algorithm and thus has uncertain poses, the other two ($R2$ in green and $R3$ in yellow, called *localized-robots*) obtain ground-truth poses from the simulator. This facilitates the verification of uncertainty measurement methods provided in Section V, which assume other non-interrelated utilities are known (actually non-interrelated uncertainties are allowed and also can be measured so long as not more than one set of interrelated uncertainties appear concurrently). The uncertainty-robot uses a simulated scanning laser ranging sensor and a map identical to the simulation environment. Figure 7(b) shows the particles for uncertainty-robot.

A path cost distribution shows the effect of uncertainty on a robot's utility estimates; it is obtained by computing path lengths from all available hypothesis of uncertainty-robot to a specific task location.

To capture the uncertainty, we set a fixed period t and empirical reliability threshold T, for example, $t = 5s$ and $T = 80\%$. After every period t, the following was performed: check the number of hypothesis n, if $n = 1$, the the robot is completely localized. If $n > 1$ then check the maximal probability p_{max} among all hypothesis, and if $p_{max} > T$, it indicates the robot is sufficiently certain of its location, otherwise, trigger the uncertainty measurement mechanism provided in Section V.

2) Results and Analysis: Figure 8 is the corresponding captured localization uncertainties at $t = 105s$. Figure 8(a) is the assignment matrix with optimal solution (shaded cells) and allowable intervals for all utilities. Figures 8(b)—8(d) are the path cost distributions for the uncertainty robot. In each sub-figure, the bar with highest probability (colored with stripe) is the path cost for the current most-likely localization result and thus is used in the assignment matrix. Other bars around it are the path costs for all other uncertain hypotheses. Remember that here only Robot $R1$ has imperfect localization, and the intervals for the interrelated utilities (the first row Figure 8(a)) are $\langle(-\infty,-2.2],[-12.5,+\infty),(-\infty,-1.2]\rangle$, thus we can compute its tolerance margin $\varepsilon_m = -9.7-(-12.5) = 2.8$. Using the method discussed in Section V, we get the allowable interrelated intervals $\langle(-\infty,-3.6],[-11.1,+\infty),(-\infty,-2.6]\rangle$. The intervals are flipped from negative to positive to fit the real path cost distributions: $\langle[3.6,+\infty),(-\infty,11.1],[2.6,+\infty)\rangle$.

In Figures 8(b)—8(d), the bars within interrelated intervals are filled red (dark), and the others are in pink (light). The calculated results of P_I are $\langle 94\%, 83\%, 89\%\rangle$, and we use the reliability threshold $T = 80\%$, therefore all utilities are considered reliable and we can trust the assignment solution.

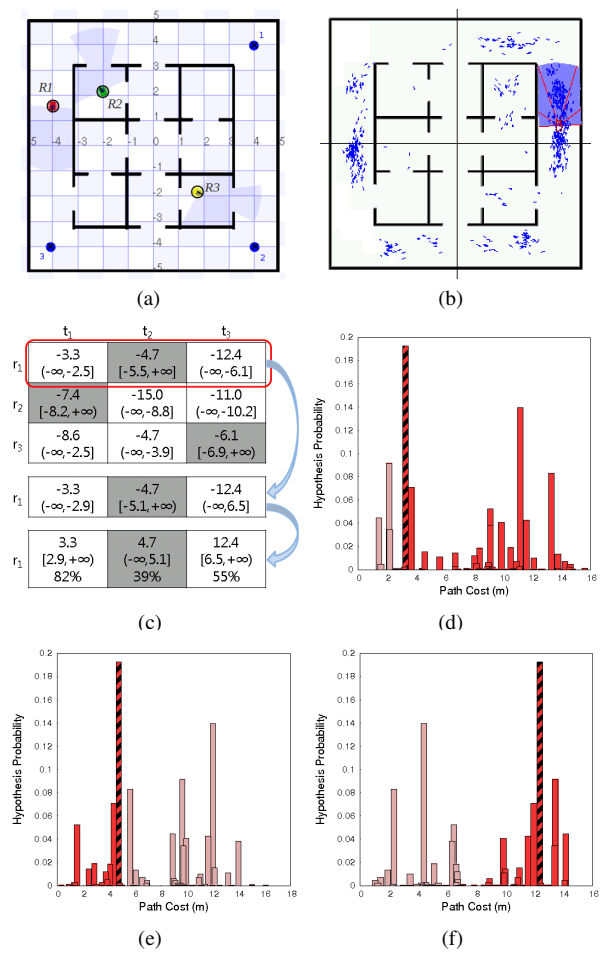
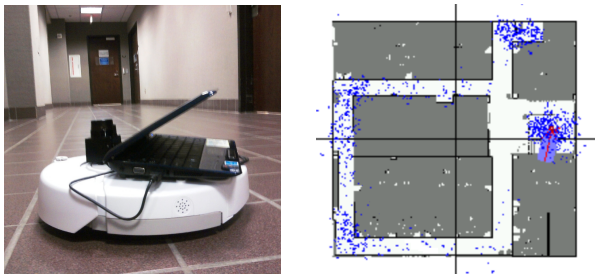


Fig. 9. (a)—(b) Multi-robot assignment simulation captured at 76th second; (c)—(f) Uncertainty analysis of interrelated utilities.

Figure 9 is another example that illustrates the unreliable assignment solution. The reliability levels of interrelated utilities are $\langle 82\%, 39\%, 55\%\rangle$, as shown in Figure 9(c). There are two utilities below the reliability threshold, thus the assignment solution is considered unreliable. In fact, from Figure 9(a) and 9(b) we see that, if the localization of Robot $R1$ is at its true location, then the assignment solution should be



(a) (b)

Fig. 10. Physical robot localizing in a floor legend

$r_1 \rightarrow t_3$ (Robot $R1$ is assigned to the task location #3), $r_2 \rightarrow t_1$ and $r_3 \rightarrow t_2$. The case illustrates the sensitivity of the assignment solution to uncertainty, the algorithm's output suggests refining of the localization before committing to an unreliable assignment.

B. Simulation vs. Real Robots

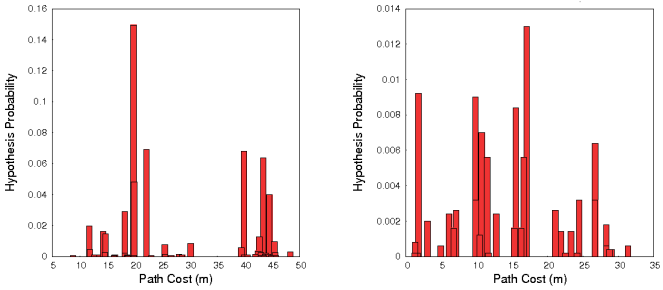
1) Experimental Setup: We also conducted experiments with physical robot in our research building (Figure 10(b) shows the floor plan, as drawn by hand, with pose estimates.) A single iRobot create robot with Hokuyo URG-04LX-UG01 laser sensor, as shown in Figure 10(a), was used to collect pose estimates, which were in turn used to compute path cost distributions. To best compare the distributions we used the same parameters for the localization parameters and randomly placed the robot in a corridor with the same initial poses and goal positions at the same locations.

2) Results and Analysis: Compared with simulation, the physical robot shows greater certainty and converges more slowly. Figure 11(a) is a typical simulation result of the hypothesis distribution captured at $t = 117s$, and Figure 11(b) is the result from physical robot captured at $t = 319s$.

Qualitatively both distributions are clustered Gaussian-distributed bars. The simulation data have fewer clusters than the physical robot data. The greater uncertainty and slower convergence suggests that using the Interval Hungarian Method can be particularly important for physical robots with significant measurement error.

VII. Conclusion

This paper presents the Interval Hungarian algorithm and an approach for measuring the effect of uncertainty on the optimality and stability of the output from the OAP with direct application to task-allocation in multi-robot systems. The Interval Hungarian algorithm is based on the bipartite matching variant of the Hungarian algorithm. Given an input utility matrix it outputs an assignment matrix along with intervals which are associated with each utility in the input. Each interval conveys a tolerance of the optimal assignment to perturbations in the utility value. We illustrated how uncertainties in multi-robot assignment problems can be quantified when probability distributions describe the utility estimates. Data from simulated and physical robot implementations were presented and compared.



(a) Path cost distribution from the simulated robot after 117 seconds.

(b) Path cost distribution from the physical robot after 319 seconds.

Fig. 11. Comparison of hypothesis distributions between simulation (left) and physical robot (right)

References

[1] 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.

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

[3] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Soc. for Industrial and Applied Math.*, March 1957.

[4] L. E. Parker, "Alliance: An architecture for fault-tolerant multi-robot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.

[5] N. Atay and B. Bayazit, "Mixed-Integer Linear Programming Solution to Multi-Robot Task Allocation Problem," Department of Computer Science & Engineering, Washington University, Tech. Rep. WUCSE-2006-54, 2006.

[6] D. P. Bertsekas, "The auction algorithm for assignment and other network flow problems: A tutorial," *Interfaces*, 1990.

[7] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. J. Kleywegt, "Robot Exploration with Combinatorial Auctions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03)*, Las Vegas, NV, U.S.A., Oct. 2003, pp. 1957–1962.

[8] 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.

[9] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: a survey and analysis," *Proceedings of the IEEE—Special Issue on Multi-robot Systems*, vol. 94, no. 7, pp. 1257–1270, July 2006.

[10] 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.

[11] C. Guestrin, S. Venkataraman, and D. Koller, "Context-Specific Multi-agent 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] 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.

[13] L. Lovász and M. D. Plummer, *Matching Theory*. North-Holland, 1986.

[14] D. W. Pentico, "Assignment problems: A golden anniversary survey," *European J. of Op. Research*, vol. 176, no. 2, pp. 774–793, 2007.

[15] I. H. Toroslu and G. Üçoluk, "Incremental assignment problem," *Information Sciences*, March 2007.

[16] G. A. Mills-Tettey, A. T. Stentz, and M. B. Dias, "The dynamic hungarian algorithm for the assignment problem with changing costs," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-07-27, July 2007.

[17] D. Fox, "Kld-sampling: Adaptive particle filters," *In Advances in Neural Information Processing Systems*, 2001.

[18] B. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," *Proceedings of the 11th International Conference on Advanced Robotics*, June 2003.