

A Measure to Match Robot Plans to Human Intent: A Case Study in Multi-Objective Human-Robot Path-Planning*

Meher T. Shaikh¹ and Michael A. Goodrich¹

Abstract—Measuring how well a potential solution to a problem matches the problem-holder’s intent and detecting when a current solution no longer matches intent is important when designing resilient human-robot teams. This paper addresses intent-matching for a robot path-planning problem that includes multiple objectives and where human intent is represented as a vector in the multi-objective payoff space. The paper introduces a new metric called the *intent threshold margin* and shows that it can be used to rank paths by how close they match a specified intent. The rankings induced by the metric correlate with average human rankings (obtained in an MTurk study) of how closely different paths match a specified intent. The intuition of the intent threshold margin is that it represents how much the human’s intent must be “relaxed” to match the payoffs for a specified path.

I. INTRODUCTION

At the heart of multi-objective decision-making is the selection of a solution from a set of alternatives, where each alternative represents a different tradeoff among the objectives. When a human is managing the multi-objective decision problem, the selected solution should match the human’s intent. Intent has been studied in many forms (see the review of related literature), and this paper focuses on how intent can be used in a problem where a ground robot must plan a path from start to goal while balancing multiple objectives. Thus, the paper operationally uses a notion of intent that can be represented as a numerical vector in a multi-objective payoff space.

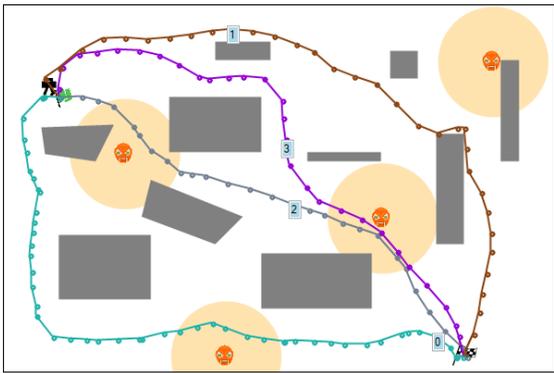


Fig. 1: Example paths; obstacles are gray and enemies orange.

*This paper was partially supported by grants from the US Office of Naval Research under grant numbers N000141613025 and N000141812503. All opinions, findings, and results are the responsibility of the authors and not the sponsoring organization.

¹M. Shaikh and M. A. Goodrich are with the Department of Computer Science, Brigham Young University, Provo, UT, USA mtalats@gmail.com, mike@cs.byu.edu

Consider, for example, the path planning problem illustrated in Figure 1. In the figure, the robot is in the upper left of the map (partially obscured by possible paths) and the goal is in the lower right (also partially obscured by possible paths). Four possible paths are presented which tradeoff between three objectives: safety (stay far from objects), stealth (don’t be seen by the orange “enemies”), and speed (reach the goal as quickly as possible). The bottom path maximizes distance from the gray obstacles, so it satisfies the *safety* intent. The middle gray path minimizes path length so it satisfies the *speed* intent. The top path never passes through the sensor range of the enemies, so it satisfies the *stealth* intent. Each intent can be assigned a numerical value (cumulative proximity to obstacles, path length, portion of path length where the robot can be seen by an enemy, respectively). For mixed intents such as “reach the goal safely and quickly”, a tradeoff between the numerical scores of safety and speed must be found, resulting in the purple path that stays away from obstacles but is still relatively short.

There are many ways to measure how closely a given vector matches another vector in a multi-objective payoff space including Euclidean distance, cosine similarity, TOPSIS, WPM, etc. [3], [26], [16]. Prior work by the authors [19], [20] demonstrated that many existing measures are not useful in determining how closely a planned 2D path matches a human’s intent, when a verbal intent is expressed as a numerical vector in the multi-objective payoff. This prior work also showed that the *cosine similarity* metric provides a useful mapping between payoff vector for different possible paths and the payoff vector for the desired tradeoff in objectives. However, as shown in this paper, the cosine similarity metric has a known limitation when a robot is following a path while objectives change. Specifically, in dynamic worlds it is desirable to be able to use an intent-mismatch metric in order to detect when the current path no longer satisfies the human’s intent. This paper presents an example that illustrates how a favorable change in the world can correspond to a large but undesirable change in the cosine similarity metric. A large change indicates the need to replan even though the current path is objectively better than it was when the path was originally planned.

This paper proposes the *intent threshold margin* (INTHRESH) metric that overcomes the limitation of the cosine similarity metric. The metric is applied to a three-objective path-planning in a known 2D environment. Like cosine similarity, the metric operates by comparing the payoffs of different potential paths to a vector representation

of human intent. When no solution perfectly matches human intent, distance information between the numerical intent vector and achievable payoffs is obtained by relaxing the intent criteria by an ε margin until solution(s) are found that match the relaxed intent.

There are a number of limitations of the paper. First, the paper does not address how a human can express intent in numerical form (though see prior work in [20], [19]). Second, the metric is applied only to 2D robot planning, and future work is needed to understand how and whether it can be used for planning of a manipulator in higher dimensions. Finally, the paper does not address what happens when the intent threshold margin indicates that the planned path no longer matches the human’s intent; future work should explore real-time replanning of a path that will match the intent.

II. RELATED LITERATURE

This paper deals with robot path-planning in 2D worlds with multiple objectives that can be satisfied. There are too many robot path-planning algorithms for a full review, example approaches include sampling-based approaches, graph-based approaches, field-based approaches, and parametric curve-based approaches both for static and dynamic environments [22], [8], [12], [13], [11], [14], [24]. Many such algorithms cater to multiple objectives such as path length, energy consumed, smoothness, stealth, etc. [7], [23]. For the examples in this paper, paths can be planned before the experiment so the speed of the planner isn’t a large constraint. Because future research will include real-time replanning, the work adopted an algorithm that is a modified version of the FMT* algorithm [10], which is a fast sampling-based planner. The modified version, known as O-FMT*, can take advantage of resampling to replan paths in dynamic environments [4].

As with path-planning, the literature on intent and intentionality is vast. Just in the context of human-robot interaction, intent has been interpreted as activity recognition [18], action prediction [9], and goal identification [25]. It could be argued that intention is closely related to so-called legibility [5], though the term legibility was originally used to express how a robot’s plan conveyed its intention to a human partner. Moreover, human intent can be expressed and communicated from human-robot robot explicitly as verbal [15], [17] and non-verbal commands [2], [6]. This paper uses an operational notion of intent that can be expressed as a payoff vector in a multi-objective planning space. The authors subjectively mapped verbal descriptions of intent (which are used in the MTurk study) to their numerical representation using tools from prior work [19].

III. REPRESENTING PATH TRADEOFFS, HUMAN INTENT, AND TRADEOFFS

A path-planning problem is (a) to find a path that goes from an initial location to a goal location (b) given a map of the environment that (c) satisfies a set of user-defined objectives. In general, there may be tradeoffs among

the objectives, meaning that increasing performance on one objective may decrease performance of another.

A. Representing Tradeoffs as Vectors

Consider a path-planning problem with K objectives $\{o_1, o_2, \dots, o_K\}$. Suppose that some path-planning algorithm has generated a set of N solutions to the path-planning problem, $\mathbb{S} = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_N\}$. Each solution \mathbf{S}_i weighs the K objectives differently, yielding a numerical *objective vector* for each path, $\mathbf{o}(\mathbf{S}_i) = [o_1(S_i), \dots, o_K(S_i)]$.

Given the N paths, it is possible to normalize the objective vector so that each path is represented by a normalized payoff vector $\mathbf{p}(\mathbf{S}_i) = [p_1(S_i), \dots, p_K(S_i)]$ where $p_k(S_i) \in [0, 1]$. A value of $p_k(S_i) = 1$ indicates the highest payoff for the specified objective from the set of possible paths (corresponding to the best path for that objective); similarly, $p_k(S_i) = 0$ represents minimum payoff, corresponding to the worst path for that objective.

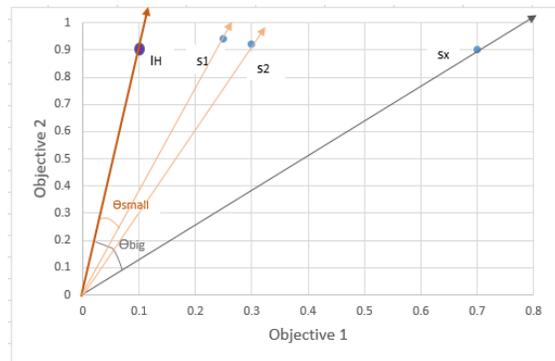


Fig. 2: Vector associated with each path and human intent.

Figure 2 illustrates three possible paths for a two objective problem. Objective 1 and objective 2 are notional, but can be thought of as *stealth* and *speed* respectively; for this problem, *safety* is not important so that objective is not shown. There are three paths shown in the figure: s_1 , s_2 , and s_x ; the s_x notation is meant to indicate that this is an objective that will be important to a later section. For path s_1 , the payoff value for objective 1 (stealth) is about 0.25 and the payoff value for objective 2 (speed) is approximately 0.92. Thus, this path is shown in the payoff space as the vector $\mathbf{p}(s_1) = [0.25, 0.92]$ that begins at the origin and terminates at the dot. Similarly, the payoff vector for path s_2 is $\mathbf{p}(s_2) = [0.3, 0.91]$ and for path s_x is $\mathbf{p}(s_x) = [0.7, 0.9]$. The path \mathbf{S}_i can be thought of as a decision variable with K features.

B. Representing Human Intent as a Vector

Let \mathbf{I}_H be the human intent variable with K features. This paper assumes that intent is already specified and focuses on matching that intent. In the experiments, the authors subjectively chose the intent vector to match a verbal description of intent using tools from prior work [19]. The intent vector \mathbf{I}_H specifies the desired human tradeoff between multiple objectives. The intent vector communicates a preference over the K objectives as $\mathbf{I}_H = [I_1, \dots, I_K]$ where $I_k \in [0, 1]$.

An intent value of $I_k = 1$ indicates that the objective is maximally important and an intent value of $I_k = 0$ indicates that the objective is not important at all.

As mentioned in the example in the introduction, this paper restricts attention (a) to 2D paths from a start location to a goal location and (b) to three *pure* intentions and multiple possible *mixed* intentions:

- *quickly*: preferred paths minimize path length.
- *stealthily*: preferred paths minimize path length in enemy sensor range.
- *safely*: preferred paths maximize cumulative distance from obstacles and world boundaries.
- *mixed*: preferred paths blend objectives such ‘go stealthily and quickly’.

Given these three objectives, $\mathbf{I}_H = [I_1, I_2, I_3]$: 1 indicates stealth, 2 indicates speed, and 3 indicates safety.

Consider again the example in Figure 1, which shows an example map with four possible paths running from the top left corner to the bottom right corner. Suppose that a human indicates that the stealth is very important, speed is only slightly important, and safety is somewhat important. This verbal expression of intent can be encoded as the intent vector $\mathbf{I}_H = [0.9, 0.1, 0.2]$. In the example, the brown path labeled 1 best matches the given intent from among the four paths because the path keeps away from the enemy. Recall that this paper does not address the way that verbal intentions are translated into the numerical intent vector; this mapping is done by the authors prior to the experiment.

In addition to the example in Figure 1, it is useful to demonstrate how the human intent vector can be represented in the multi-objective space. Consider again Figure 2 and recall that there are two objectives for the problem. Suppose that the human’s intent can be expressed verbally as “speed is very important, stealth is somewhat important, and safety is not important at all”. Since safety is not important, the figure ignores that dimension. Objective 1 corresponds to stealth and objective 2 corresponds to speed, so the human’s intent vector is subjectively represented as $\mathbf{I}_H[0.1, 0.9]$. The vector begins at the origin and terminates at the location indicated by \mathbf{I}_H in the figure; the circle is larger than for the paths to help the reader differentiate between paths and intent.

C. Intent-Matching Metrics

As mentioned in the introduction, prior work evaluated the TOPSIS and WPM multi-objective blending criteria [1], as well as *euclidean distance* and *cosine similarity* for finding path vectors that matched the human intent vector [20]. Cosine similarity either produced better intent matches or produced equivalent matches with greater computational efficiency than the other metrics.

Cosine similarity is the angle between the intent vector and the path payoff vector. If the path vector aligned perfectly with the intent vector, that is, the angle is 0, then $\cos 0$ yields a maximum similarity of 1. This method of checking similarity between the intent vector and the payoff vector works well if all the elements in the intent vector are close

in value to all the corresponding values in the payoff vector. The example in Figure 2 shows that the angle between $\mathbf{p}(s1)$ and \mathbf{I}_H is smaller than both the angle between $\mathbf{p}(s2)$ and \mathbf{I}_H as well as the angle between $\mathbf{p}(sx)$ and \mathbf{I}_H . Thus, the cosine similarity metric would select path $s1$ as the path that most closely matches intent.

Now, rather than interpreting the vectors in the example from Figure 2 as payoffs for different paths, consider a problem where objectives vary over time. Time-varying objectives are important because they mean that a path that once matched intent may not always match intent. For example, suppose that enemies can move, changing the value of the stealthy objectives. Suppose that a path S^* was planned that perfectly satisfied the intent. The vector for this path coincides with the intent vector \mathbf{I}_H . Now suppose that while the robot is following its path the enemies gradually move away from the path so that the planned path decreasingly intersects with the enemies sensor region. After following the path until time $t = 1$, the payoff objective for the planned path S^* is represented by $s1$; objective 2 (speed) increased a bit because there is some noise in its estimate, but objective 1 (stealth) has increased quite a bit because the enemies are moving away from the planned path. The robot continues to follow its path and enemies continue to move away. The vector $s2$ represents the payoff vector at time $t = 2$ and indicates that the payoffs for the path are becoming more favorable for the agent. Finally, at some time in the future, time $t = x$, the planned path has a payoff vector indicated by sx , which is still a very fast path but has also become a very stealthy path.

This example calls attention to the problem with cosine similarity. Because the angle between intent and the payoff vector is increasing (because the path is becoming more stealthy as enemies move away), at some point the angle becomes so high that the cosine similarity metric indicates that the path no longer matches what the human intends. At that point, the robot begins to replan even though there is no need to do so.

Prior work did not identify this limitation of the cosine similarity metric because the problems were constructed such that there were always paths with payoff vectors that were distributed across the Pareto front. Thus, there was always a path that had a small angle between its payoff vector and the intent vector. This limitation of the cosine similarity metric calls for another approach to measuring similarity between the intent vector and a path’s payoff vector.

IV. INTENT THRESHOLD MARGIN

Essentially, the problem with cosine similarity is that it evaluated paths by *how similar* they were to the intent vector. The intent threshold margin doesn’t seek to find how similar a path is to intent, but rather how much of the intent has to be sacrificed before a path becomes satisficing. This section first presents the definition of the intent threshold margin and then shows how it can be used to rank paths relative to the human intent vector.

A. Definition

Consider a subset of paths for which each payoff variable is within an ε -threshold of the corresponding variable in \mathbf{I}_H :

$$\mathbb{T} = \{ \mathbf{S}_i \in \mathbb{S} \mid \forall_k (p_k(\mathbf{S}_i) \geq (I_k - \varepsilon_k)) \} \quad (1)$$

The epsilon threshold ε_k represents a margin, similar to Simon's satisficing aspiration levels [21], by which an intent criterion may be relaxed for an objective k . We call this relaxed margin the *intent threshold margin* (INTHRESH), which is represented by the vector $\mathcal{E} = [\varepsilon_1, \dots, \varepsilon_K]$. Thus, the region specified by \mathbb{T} is a function of the threshold vector, \mathcal{E} , so we can make this dependence explicit by writing $\mathbb{T}(\mathcal{E})$.

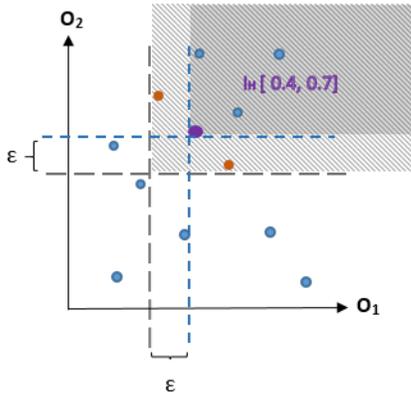


Fig. 3: Paths that make to intent-satisfying-region.

The *degree to which a solution satisfies* the given intent \mathbf{I}_H depends on the smallest values in the epsilon vector $\mathcal{E} = [\varepsilon_1, \dots, \varepsilon_K]$ for which the solution is satisficing. Consider Figure 3, which, for illustration purposes, assumes $K = 2$ and two notional objectives o_1 and o_2 (e.g., stealth and speed). Each small (blue or orange) circle in the 2D plane represents the payoff vector for a possible path, placed in the figure according to the normalized payoff vector. Suppose that the human intent vector \mathbf{I}_H is given by $[0.4, 0.7]$; 0.4 for objective o_1 and 0.7 for objective o_2 respectively, expressing intent as preference for paths that favor objective o_2 more than objective o_1 . This example intent is shown as the (larger) solid violet circle at the intersection of the blue dashed lines.

The solid circle at this intersection of the two dashed lines indicates that the payoffs of a desired solution should lie in the region to the right and above the intersection. Thus, the darker gray region to the right and above the intersection corresponds to $\mathbb{T}(\mathbf{0})$ since $\mathcal{E} = \mathbf{0}$. In other words, the dark gray *intent-satisfying-region* satisfies Equation 1 with $\varepsilon_k = 0$ for both objectives o_1 and o_2 . In the figure, the topmost three small blue circles that lie in the dark shaded region all correspond to paths that satisfy intent. That is, the system is able to find a set of three paths with no epsilon relaxed because each of these solutions exceeds the thresholds.

Now suppose that the three paths in the dark shaded region do not exist, indicating that there are no solutions that satisfy the given objectives specified by \mathbf{I}_H . INTHRESH relaxes the values in the intent vector by factor of ε_k for each objective.

The value of ε_k is gradually increased in steps. Referring to Figure 3 again, the relaxation by ε_k causes the intent-satisfying-region to grow a little towards the left and the down, yielding the light gray region. Let $\mathcal{E}' = [\varepsilon_1, \varepsilon_2]$. The lighter gray region is $\mathbb{T}(\mathcal{E}')$. Given the new values of $\varepsilon_k > 0$, two solutions, indicated by the small orange circles make an entry to the set $\mathbb{T}(\mathcal{E}')$.

B. Ranking Solutions

The intent threshold margin can be used to rank paths by how much must be given up before a path becomes satisficing. Formally, we say that a solution \mathbf{S} is satisficing given a threshold vector \mathcal{E} if $\mathbf{S} \in \mathbb{T}(\mathcal{E})$. For an intent \mathbf{I}_H , a solution $\mathbf{S}_x \in \mathbb{S}$ is *superior to* a solution $\mathbf{S}_y \in \mathbb{S}$ if each of the elements of $\mathcal{E}_x = [\varepsilon_{x_1}, \dots, \varepsilon_{x_K}]$ needed to make \mathbf{S}_x satisficing is less than the corresponding counterparts in $\mathcal{E}_y = [\varepsilon_{y_1}, \dots, \varepsilon_{y_K}]$ needed to make \mathbf{S}_y satisficing. That is, if $\forall k \in K, \varepsilon_{x_k} < \varepsilon_{y_k}$ then solutions in $\mathbb{T}(\mathcal{E}_x)$ are ranked higher than solutions in $\mathbb{T}(\mathcal{E}_y) \setminus \mathbb{T}(\mathcal{E}_x)$.

For example, each of the solutions in the darker shaded region, $\mathbb{T}(\mathcal{E}_x)$, in Figure 3 is superior to the ones lying in the lighter shaded region, which is the set difference between the paths above the first set of dashed lines and the lower set of dashed lines, $\mathbb{T}(\mathcal{E}_y) \setminus \mathbb{T}(\mathcal{E}_x)$.

Given an intent vector, \mathbf{I}_H , solutions can be ranked by how much has to be sacrificed, that is, how big ε must become, before a solution becomes satisficing. Algorithm 1 produces a set $\mathbb{E} = \{\mathcal{E}_0, \mathcal{E}_1, \dots, \mathcal{E}_m\}$, which is an indexed set of m intent threshold margins.

Init: $\mathcal{E}_0 = [0, 0, \dots, 0]$, a vector of all zeroes;

$\mathbb{R} = \mathbb{S}$, residual set;

$\varepsilon_k = 0$;

$r = 1$;

while $\mathbb{R} \neq \emptyset$ **do**

$\mathcal{E}_r = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_K]$;

compute $\mathbb{T}(\mathcal{E}_r)$;

if $\mathbb{T}(\mathcal{E}_r) - \mathbb{T}(\mathcal{E}_{r-1}) \neq \emptyset$ **then**

$\mathbb{R} = \mathbb{S} - \mathbb{T}(\mathcal{E}_r)$, update residual set;

$r \leftarrow r + 1$;

end

$\varepsilon_k \leftarrow \varepsilon_k + \delta_k$, for all k objectives ;

end

Algorithm 1: Partitioning Solutions.

Algorithm 1 iteratively increases the value of the threshold variable ε beginning at zero, and uses this value to construct a vector of thresholds \mathcal{E}_r . The amount that ε changes is given by $\delta \ll 1$, which is a small value that slowly lowers the threshold at which solutions become satisficing. For each threshold vector, the set of solutions that are satisficing is computed, $\mathbb{T}(\mathcal{E}_r)$. When the threshold is lowered enough (that is, when ε is high enough) so that a new solution becomes satisficing, the threshold vector \mathcal{E}_r is stored, the iterator value r is increased, and the residual set \mathbb{R} is computed. The residual set consists of all those solutions that are not yet within the satisficing region for the given value

of ε , so when the residual set is empty then all solutions have been partitioned into a satisficing set.

The index of the intent threshold margin set determines the extent to which its associated solution(s) satisfies intent. The index gives a *rank* to a solution for a given human command represented as an intent vector. The solution(s) at index 0 all exceed the human's intent. Solution(s) of rank 1 is/are the top ranked solution(s) among those for which at least one part of the human's intent must be relaxed. And the solution(s) at index m has the last rank. Every element in \mathbb{S} is associated with a single index in \mathbb{E} , so every path is ranked.

In the algorithm shown, the δ_k parameter is a small percentage of the intent parameter values specified in \mathbf{I}_H . In other words, thresholds are relaxed in proportion to the magnitude of their weight in the human intent vector. For the experiments in this paper, δ_k was defined as $\delta_k = (p/100) * I_k$ and $p = 15$ was used.

V. EVALUATIONS

We conducted an Amazon Mechanical Turk[®] (MTurk) study with 50 participants in order to assess the intent threshold margin metric. The goal of the study was to evaluate whether the rankings induced from the intent threshold margin correlated with rankings from MTurk participants. Similarly, paths ranked low by participants should have higher indices in \mathbb{E} .

Prior to the study, an indexed set of 14 configuration maps, $\mathbb{C} = \{C_1, \dots, C_{14}\}$ was produced, showing the robot's start location and goal location, the obstacles, and enemy positions. Five of the configurations were used for training, and the remaining nine were used for evaluation, presented to participants in a counterbalanced way. For each configuration, nine paths were planned using an Online-FMT* algorithm presented in [4] using different weights that were uniformly selected from among the three objectives. The different weights created paths that were all in the Pareto set, meaning that no path in was payoff dominated by any other path.

An intent was specified as an English sentence for each configuration. The intent was either a single objective such as 'go quickly', or a multi-objective intent such as 'go stealthily or safely'. The English sentences were designed to either put all weight on a single objective or to give preference for two objectives. Thus, each command resulted in a unique intent vector \mathbf{I}_H . The intent threshold algorithm was then applied to the nine paths and intent vector to produce the set \mathbb{E} .

From the set \mathbb{E} , four paths that corresponded to different indices within \mathbb{E} were selected. These four paths included one path from the lowest index ("best" path), one from the highest index ("worst" path), one from roughly the second quartile, and one roughly from the third quartile. Please refer back to Figure 1 that shows four paths for an example configuration. Thus, each configuration had a set of four paths that were designed to match a specified intent to varying degrees. An IRB-approved pilot study was conducted to determine whether there was sufficient differences in the paths to justify a complete study.

The full study preceded with participants giving informed consent as approved by the university's IRB office. Each participant received \$3 as compensation. Next, participants were trained using (a) illustrations of the configuration, (b) illustrations of the four paths per configuration, and (c) definitions of the path-planning objectives; quickly, stealthily and safely.

MTurk was set up to show participants the path-planning configuration (e.g., robot, goals, enemies, obstacles) and the four paths selected for that configuration; Figure 1 is an image from one of the configurations. Each participant was asked to rank 4 paths from 1 to 4, with 1 indicating the path that best matched the specified intent. 5 configurations were used for training, and the data collected from remaining 9 configurations was used for analysis.

Responses from 47 participants out of the 50 who participated were included in the data analysis. We discarded the results of two participants because they took fewer than 5 minutes required to respond to the survey, indicating that they did not seriously consider each configuration; the median completion time was 14.4 minutes. Further, a technical glitch caused MTurk data to be lost for one participant.

Before proceeding, please note that the term *rank* is used to indicate the ordinal value assigned to a particular path in a particular configuration; ranks are values in the set $\{1, 2, 3, 4\}$. Further, the term *ranking* is used to indicate the ordering of the set, that is, to indicate the vector of ranks. For example, the hypothetical ranking is always the vector $[1, 2, 3, 4]$ for path 1, path 2, etc; but participants might not rank each path the same, so a ranking for a particular participant might be the vector $[1, 3, 2, 4]$ indicating that the participant swapped the ranks of the second and third paths.

A. Hypotheses

We hypothesize the following¹:

- *Hypothesis 1*: For each path in each configuration, there will be no significant difference between the ranks from participants and the rank induced by the intent threshold distance.
- *Hypothesis 2*: For each configuration, the ranking of the four paths from the participants will be strongly and positively correlated with the ranking of the four paths induced by the intent threshold distance.
- *Hypothesis 3*: The smallest value of ε for which the path is part of \mathbb{E} , which is the minimum intent threshold distance, will positively correlate with ranks from participants; small values of epsilon correspond to top ranks (e.g, rank 1), and high values correspond to poor ranks (e.g., rank 4).

B. Data

The following data were gathered for each configuration path:

- *Hypothetical rank*, R_i : a rank between 1 to 4 (inclusive) obtained from Algorithm 1 and ordered from the set

¹Unfortunately, we did not register our hypotheses before the experiment via the Center for Open Science, cos.io/prereg/. We learned about registering hypotheses after the data was gathered.

\mathbb{E} , with 1 indicating the path with the smallest intent threshold distance.

- *User rank, $R_{u,s}$* : a rank between 1 and 4 (inclusive) as selected by the participant.
- *Intent threshold distance*: the smallest value of ε for which the path is part of \mathbb{E} .

The data used for the *intent threshold distance* needs to be explained. Recall from Algorithm 1 that each path produces a vector of epsilon values, one for each objective (stealth, speed, safety). This vector was denoted as $\mathcal{E} = [\varepsilon_1, \dots, \varepsilon_K]$, where K indicated the number of objectives. If a particular objective is not part of an intent, there is no need to find a value of ε for that objective.

This paper considers pure and mixed intents. For pure intents, there is only one ε in \mathcal{E} , that is, $\mathcal{E}^{\text{pure}} = \varepsilon_i$ where $i \in \{1, 2, 3\}$ depending on which objective is chosen. Mixed intents considered only two of the three, so $\mathcal{E}^{\text{mix}} = [\varepsilon_i, \varepsilon_j]$ where $i, j \in \{1, 2, 3\}$ and $i \neq j$. For mixed intents, the *epsilon value* is defined as the maximum of the ε 's needed to make a path satisficing. Thus *epsilon value* = $\max[\varepsilon_i, \varepsilon_j]$. In other words, the worst case threshold is used to measure “how far” a path is from the intent.

VI. RESULTS AND DISCUSSION

A. Comparing InThresh to Cosine Similarity

In static problems with solutions somewhat uniformly sampled from the Pareto front, cosine similarity was shown to be a useful metric [19]. Thus, for problems that satisfy these conditions, compatibility of rankings from INTHRESH and cosine similarity would provide evidence in support of the utility of INTHRESH. The four paths used in each experimental configuration satisfied the conditions. For all but one configuration, the ranks of INTHRESH and cosine similarity were identical, and in the other configuration the two rankings switched second and third ranked paths.

B. Difference Between Hypothesized and Participant Ranks

Consider hypothesis 1. Recall that there are nine configurations $\mathbb{C} = \{C_1, \dots, C_9\}$ and that for each configuration there are ranks produced by each participant and the ranks induced by the epsilon threshold distance. Table I shows the hypothetical ranks and the mean (median) ranks across participants for each configuration.

It is obvious from Table I to see that there is a strong relationship between average user rank and the hypothetical ranks. The mean for the hypothetically best path is higher than one because some participants did not rank this path as the best path; similarly the mean rank for the hypothetically worst path is less than four.

Note that the median rank for all participants for the hypothetically best path is always 1. By contrast, the median rank across participants and the hypothetical ranks does not always agree. One possible explanation is that it might be easier for participants to determine when a path best matches intent than to determine to what degree a path differs from an intent; future work should explore this explanation.

TABLE I: Mean (and median, in parentheses) user ranks for the 9 configurations compared to the hypothetical ranks. The † superscript by the configuration indicates a mixture intent of ‘quick and stealthy’, and the ‡ superscript indicates a mixture of ‘stealthy and safe’; intent for all other configurations are for single attributes.

Configuration	Hypothetical Ranking			
	1	2	3	4
C_1^\dagger	1.3 (1)	2.45 (3)	3.7 (4)	2.55 (2)
C_2^\dagger	1.23 (1)	2.15 (2)	2.81 (3)	3.81 (4)
C_3	1.19 (1)	2.09 (2)	3.34 (3)	3.38 (3)
C_4	1.28 (1)	1.98 (2)	2.91 (3)	3.83 (4)
C_5^\ddagger	1.49 (1)	2.66 (2)	2.55 (3)	3.3 (3)
C_6^\ddagger	1.23 (1)	2.38 (2)	2.96 (3)	3.43 (4)
C_7	1.15 (1)	2.6 (3)	3.26 (3)	3.0 (3)
C_8	1.0 (1)	2.68 (3)	2.66 (3)	3.66 (4)
C_9	1.4 (1)	1.94 (2)	3.22 (3)	3.34 (4)

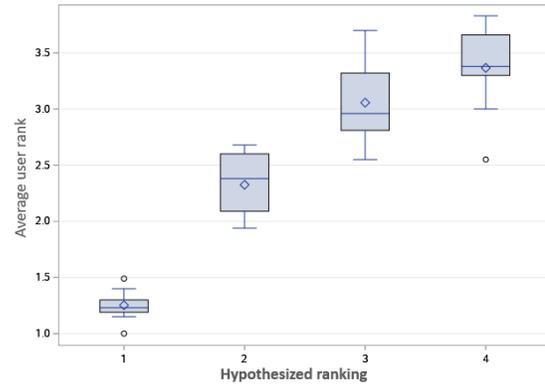


Fig. 4: Average participants ranks across configurations.

Figure 4 gives a box-and-whiskers plot of the hypothetical ranks and average user ranks. The basis for this plot is the average user ranks across configurations. The mean is indicated by the diamond in the box, the median by the horizontal line in the box, the shaded region of the box indicates the range between the first and fourth quartile, the whiskers indicate the span of the 90th percentile, and the circles indicate outliers. For example, the circle above hypothetical rank 1 comes from configuration C_5 whose mean rank is 1.49, the circle below hypothetical rank 1 comes from configuration C_8 whose mean rank is 1.0, and the circle below hypothetical rank 4 is from configuration C_1 whose mean rank is 2.55.

Hypothesis 1 can be evaluated from two perspectives: first, that each participant’s ranks should match the hypothetical ranks, and second, that the average ranks across participants match the hypothetical ranks.

1) *Do Individual Ranks Match Hypothetical Ranks?*: We performed a double sided t-test ($n=47$) with pseudo Bonferroni correction on the difference between user rankings and hypothetical rankings to test whether the differences were statistically different at a level of $p=0.001$.

Table II shows the outcomes with significant differences marked with an *. Results are that 22 out of 36 average responses are not significantly different than the hypothetical ranking. This provides evidence in support of hypothe-

TABLE II: 22 out of 36 responses indicate that the user ranks were not significantly different from the hypothetical ranks.

	1	2	3	4
C_1	0.0032	0.0011	< .001*	< .001*
C_2	0.0147	0.9	0.0375	0.0375
C_3	0.05	0.103	0.005	< .001*
C_4	0.022	0.57	0.29	0.044
C_5	< .001*	< .001*	0.011	< .001*
C_6	0.02	< .001*	0.71	< .001*
C_7	0.11	< .001*	0.027	< .001*
C_8	.	< .001*	< .001*	0.0011
C_9	< .001*	0.55	0.002	< .901*

sis 1, but there are some individual participant ranks which are obviously different from the hypothetical ranks. Thus, considering only individual participants provides marginal support in favor of hypothesis 1.

2) *Do Average Ranks Match Hypothetical Ranks?:* For each path in a configuration, we computed the average rank across the 47 participants. This gives $4 \times 9 = 36$ different average ranks. Using a Pearson Correlation, we computed the correlation between the average ranks and the hypothetical ranks. The correlation value is $\rho = 0.9118$ ($n=36$) with a p -value of less than 0.001.

The high correlation value indicates that the hypothetical ranks and average ranks across participants are strongly and positively correlated. This provides strong evidence for Hypothesis 1. In other words, if ranks are obtained by a group and then averaged, the results correlate strongly with the ranks induced by the epsilon threshold margin even though some participants differ from the hypothetical ranks.

C. Spearman's Rho for User Ranking

Hypothesis 2 says that the hypothetical rankings are highly, positively correlated with participant rankings. Recall that the term *ranking* means the vector of ranks. Hypothesis 2 is evaluated using Spearman's rank correlation coefficient.

Spearman's rank correlation coefficient provides information on the strength and direction of relationship between two ranked variables. Recall that for each configuration we have the individual user rankings and the ranking induced by INTRESH. For each of the nine configurations and for every participant's ranking, we computed the Spearman's rank correlation coefficient. Thus we computed 423, that is, 47×9 coefficients.

Figure 5 shows the distribution analysis of rhos, that is coefficients, for all the rankings of all the 47 participants. Notice that the mean is of 0.707 indicating a strong positive association between the hypothetical ranking and the user ranking. Notice further that the majority of the rank correlation coefficients exceeded 0.5, showing a positive correlation between individual user ranking and the metric ranking. Negative values show a negative association, and such values rarely occurred in our analysis. Although evaluating the distribution of the coefficients uses only descriptive statistics, the distribution provides good support for Hypothesis 2.

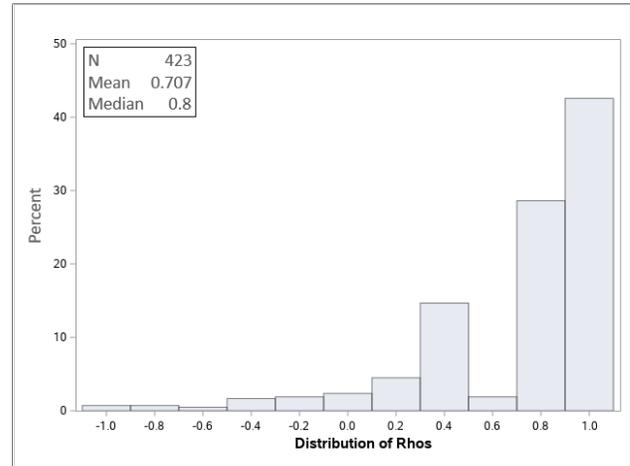


Fig. 5: Distribution analysis of Spearman's rank correlation coefficients for 423 (that is, 47×9) user rankings.

D. Correlation Between Intent Threshold Margin and User Ranks

Hypothesis 3 asserts a positive relationship between the intent threshold margin and the ranks from the participants. Recall that the *epsilon value* is associated with a path is obtained from the L_1 norm of the minimum values of ε required to make a path satisfying; $\text{epsilon value} = \max[\varepsilon_i, \varepsilon_j]$.

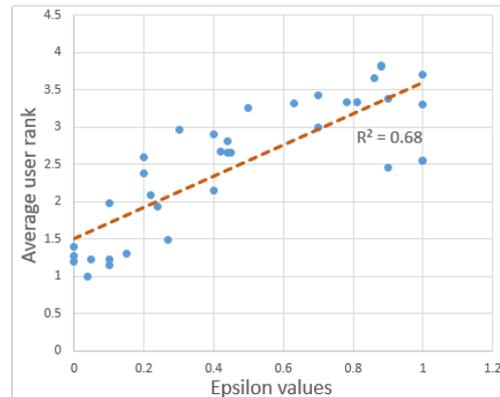


Fig. 6: Correlation between epsilon distances and average user ranks. Pearson's Correlation coefficient 0.823, p -value < 0.001.

Figure 6 shows a strong positive correlation between the epsilon values and the user ranks. The Pearson coefficient is 0.823 ($n=36$, $p < 0.001$, $R^2 = 0.68$). The trendline gives an idea of the fit to the data. The R^2 value gives some confidence that the relationship between rank and epsilon values are linear, and the p -value gives strong confidence that there is a positive correlation between epsilon value and ranks. Thus, we conclude that there is evidence in support of Hypothesis 3.

VII. SUMMARY AND FUTURE WORK

This paper proposes the *intent threshold margin* as a measure of how well a solution to a multi-objective problem matches human intent. The need for a new measure was motivated by a limitation in the *cosine similarity* metric that

had proven very useful in prior work. Cosine similarity gave preference to paths that produced small angles between an intent vector and a payoff vector. By contrast, the intent threshold margin gave preference to paths that required less sacrifice from the ideal intent to become satisficing. Results from a Mechanical Turk study indicate that the rankings induced by the *intent threshold margin* are strongly correlated with human rankings in a three-objective problem; the problem was planning a path from a starting location to a goal location in a 2D world.

Future work should address four important problems. First, although the intent threshold margin measure was motivated by a special need in dynamic worlds, the results in this paper are restricted to static worlds. The measure should be used in dynamic world to detect when a path ceases to match intent as the world changes, and a user study should be performed to see if these detections match human expectations. Second, the measure was only applied to a 2D path-planning problem. Work should be done to explore the generalizability of the approach to, for example, planning manipulator trajectories or ranking heterogeneous problem with respect to a multi-objective robot assignment problem. Third, the analyses in this paper compared the way the rankings induced by the measure compared to average user ranking, where averages were either across users or across multiple configurations. Future work should explore how individual differences might affect representation and perception of rank in a multi-objective problem. Finally, analyses only considered mixed intents with two objectives. Future work should evaluate how humans rank solutions when there are four or more objectives, and explore whether the intent threshold margin would correlate to human rankings on these more complicated problems.

REFERENCES

- [1] F. S. Azar, "Multiattribute decision-making: use of three scoring methods to compare the performance of imaging techniques for breast cancer detection," 2000.
- [2] A. Bauer, D. Wollherr, and M. Buss, "Human-robot collaboration: a survey," *International Journal of Humanoid Robotics*, vol. 5, no. 01, pp. 47–66, 2008.
- [3] W. K. M. Brauers, E. K. Zavadskas, F. Peldschus, and Z. Turskis, "Multi-objective decision-making for road design," *Transport*, vol. 23, no. 3, pp. 183–193, 2008.
- [4] B. Chandler and M. A. Goodrich, "Online rrt* and online fnt*: Rapid replanning with dynamic cost," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 6313–6318.
- [5] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2013, pp. 301–308.
- [6] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and autonomous systems*, vol. 42, no. 3–4, pp. 143–166, 2003.
- [7] K. Fujimura, "Path planning with multiple objectives," *IEEE Robotics & Automation Magazine*, vol. 3, no. 1, pp. 33–38, 1996.
- [8] M. Hehn and R. D'Andrea, "Real-time trajectory generation for quadcopters," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 877–892, 2015.
- [9] C.-M. Huang and B. Mutlu, "Anticipatory robot control for efficient human-robot collaboration," in *The eleventh ACM/IEEE international conference on human robot interaction*. IEEE Press, 2016, pp. 83–90.
- [10] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International journal of robotics research*, vol. 34, no. 7, pp. 883–921, 2015.
- [11] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
- [12] L. Kavraki, P. Svestka, and M. H. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. Unknown Publisher, 1994, vol. 1994.
- [13] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [14] T. W. Manikas, K. Ashenayi, and R. L. Wainwright, "Genetic algorithms for autonomous robot navigation," *IEEE Instrumentation & Measurement Magazine*, vol. 10, no. 6, 2007.
- [15] C. Meriçli, S. D. Klee, J. Paparian, and M. Veloso, "An interactive approach for situated task specification through verbal instructions," in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 1069–1076.
- [16] J. M. Merigó and M. Casanovas, "Induced aggregation operators in the euclidean distance and its application in financial decision making," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7603–7608, 2011.
- [17] J. Norberto Pires, "Robot-by-voice: Experiments on commanding an industrial robot using the human voice," *Industrial Robot: An International Journal*, vol. 32, no. 6, pp. 505–511, 2005.
- [18] A. Roitberg, A. Perzylo, N. Somani, M. Giuliani, M. Rickert, and A. Knoll, "Human activity recognition in the context of industrial human-robot interaction," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*. IEEE, 2014, pp. 1–10.
- [19] M. T. Shaikh and M. A. Goodrich, "Design and evaluation of adverb palette: A gui for selecting tradeoffs in multi-objective optimization problems," in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2017, pp. 389–397.
- [20] M. T. Shaikh, M. A. Goodrich, D. Yi, and J. Hoehne, "Interactive multi-objective path planning through a palette-based user interface," in *Unmanned Systems Technology XVIII*, vol. 9837. International Society for Optics and Photonics, 2016, p. 98370K.
- [21] H. A. Simon, *The sciences of the artificial*. MIT press, 2019.
- [22] A. Stentz *et al.*, "The focussed d* algorithm for real-time replanning," in *IJCAI*, vol. 95, 1995, pp. 1652–1659.
- [23] J. Van Den Berg, D. Ferguson, and J. Kuffner, "Anytime path planning and replanning in dynamic environments," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2366–2371.
- [24] D. Yi, M. A. Goodrich, and K. D. Seppi, "MORRF*: Sampling-based multi-objective motion planning," in *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 2015, pp. 1733–1739.
- [25] S.-J. Youn and K.-W. Oh, "Intention recognition using a graph representation," *World Academy of Science, Engineering and Technology*, vol. 25, pp. 13–18, 2007.
- [26] S. Zionts, "A multiple criteria method for choosing among discrete alternatives," *European Journal of Operational Research*, vol. 7, no. 2, pp. 143–147, 1981.