Design and Evaluation of Adverb Palette: A GUI for Selecting Tradeoffs in Multi-objective Optimization Problems

Meher T. Shaikh
Brigham Young University
meherts@byu.edu

Michael A. Goodrich
Brigham Young University
mike@cs.byu.edu

ABSTRACT

An important part of expressing human intent is identifying acceptable tradeoffs among competing performance objectives. We present and evaluate a set of graphical user interfaces (GUIs), that are designed to allow a human to express intent by expressing desirable tradeoffs. The GUIs require an algorithm that identifies the set of Pareto optimal solutions to the multi-objective decision problem, which means that all the solutions are equally good in the sense that there are no other solutions better for every objective. Given the Pareto set, the GUIs provide different ways for a human to express intent by exploring tradeoffs between objectives; once a tradeoff is selected, the solution is chosen. The GUI designs are applied to interactive human-robot path-selection for a robot in an urban environment, but they can be applied to other tradeoff problems. A user study evaluates GUI designs by requiring users to select a tradeoff that satisfies a specified mission intent. Results of the user study suggest that GUIs designed to support an artist's palette-metaphor can be used to express intent without incurring unacceptable levels of human workload.

Keywords

Keywords: human-robot interaction, multi-objective decision making, user interface design, robot path-planning

1. INTRODUCTION

An important part of specifying human intent is identifying acceptable tradeoffs among competing performance objectives. In a multi-objective problem with conflicting performance objectives, the set of Pareto optimal solutions is precisely the set of all possible solutions that satisfactorily tradeoff between the different objectives. Recall that a solution is Pareto optimal if, roughly speaking, there is no other solution that is better for every objective. Since each Pareto solution represents a potentially acceptable tradeoff, specifying intent is roughly equivalent to selecting a desirable Pareto optimal solution.

When a human selects a single solution, making tradeoffs between the objectives creates the need for a robust and intuitive interface that allows a user to select a satisfactory tradeoff without imposing high workload. In a supervisory control problem, given a scenario of (a) what needs to be done — the strategic intent to be accomplished — and (b) a set of ways that a task can be performed, the human (c) determines "how" the task will be done using a GUI.

This paper presents three possible GUI designs that provide a medium to explicitly express human intent for how a task will be done given a set of objectives expressed as adverbs; the adverbs convey what could be important in how the task can be done. The GUI designs allow the human to evaluate different solutions and select one that best matches strategic objectives of the problem. The work in this paper builds from prior work on using Pareto Analysis for exploring tradeoffs [20], which defines the problem as follows:

> The solution points [in] the Pareto [Set] are mathematically indifferent with respect to each other, and thus the selection phase ... is subjectively driven by the human decision maker. This process involves exploration of the [Set], and eventually, the challenge in selecting a solution is to account for gains and losses while adhering to personal preferences.

We discuss different GUI designs and a user study that compares these designs. Generally speaking, the GUI designs are based on the metaphor of an artist's palette, where an artist mixes different colors to produce a desired hue. The adverbs correspond to different objectives to be accomplished; each adverb is a different color, and the mixes of colors represent different tradeoffs between objectives. For example, in a robot path-planning application, consider a command for a robot to "Go quickly and safely from point A to point B." The adverbs associated with these objectives are "quickly" and "safely". Like an artist, the operator can mix the adverbs on the interface such that a *single* path is selected that is both quick and safe from a set of available paths. Results of a user study demonstrate that this

HRI '17, March 6–9, 2017, Vienna, Austria.

ISBN 978-1-4503-4336-7/17/03...$15.00

DOI: http://dx.doi.org/10.1145/2909824.3020225

metaphor can be very useful for helping a human find acceptable tradeoffs between competing objectives.

Figure 1 shows an example Pareto-optimal set for a two-objective problem. Each point in the curve represents a solution and its associated pay-offs for *objective 1* and *objective 2*. The upper left dot represents a solution that has maximum pay-off for *objective 2* at the expense of *objective 1*, and lower right dot represents a solution that has maximum pay-off for *objective 1*. Dots between the extremes represent tradeoffs between the objectives.

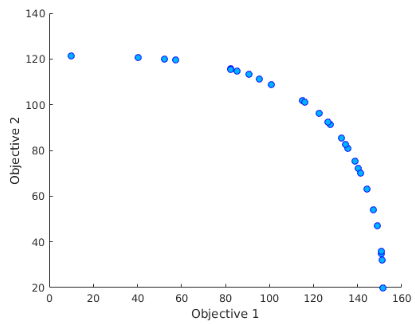


Figure 1: Path Planning with MORRF* for two objectives.

The paper presents GUI designs, *palette*, *sliders*, and *prism*, that are based on the color-blending metaphor and that can be generalized to many problems that require tradeoffs. A fourth GUI design, *waypoints*, is specific to path-planning. The three color-blending designs are described later in the paper.

We apply the GUIs to a robot path-planning problem where multiple performance objectives need to be satisfied. Although there exist many algorithms for multi-objective optimization (see, for example, [16, 6, 7, 25, 15, 2]), we use the MORRF* algorithm [25] because it efficiently generates Pareto optimal solutions specifically for path-planning.

Figure 2 shows an example of one of the designs in AP (the *palette* design) applied to robot path-planning. The left panel of the interface is problem-specific and shows the available alternatives/solutions for the problem. The right panel allows the human to express intent; it is the area where the human can explore many tradeoffs. Based on the human-actions on the right, the left panel updates to show the result/solution. For example, the left side of Figure 2 depicts a map that shows in gray Pareto optimal paths that a robot can take, and the right side of the interface provides an area that can be used by the human to find tradeoffs among the paths. Based on the command issued on right side panel, one of the gray paths gets highlighted on the left panel.

2. RELATED WORK

Making tradeoffs in decision-making is also known as multiple criterion decision-making [24] and multiple attribute decision-making [13]. The goal is to select a decision over available alternatives in a way that balances or trades off between the objectives, i.e, to choose from among a finite set of discrete alternatives [10, 20]. This paper uses three objectives: minimizing distance from the robot's start location to a goal location, avoiding exposure of the robot to one or more enemies, and avoiding collisions with obstacles.

The literature on designing user interfaces for human-machine interaction is vast (see [12, 14, 8, 4, 1] for some examples). There are indeed many graphical interfaces for managing robots in HRI [11, 18, 17, 21]. The interface in the paper differs from these other interfaces in that it is intended to enable interactive decision-making in selecting a solution that satisfies a decision tradeoff. The interfaces in this paper is more similar to decision-support systems than to traditional supervisory control interfaces.

The GUIs presented in this paper are perhaps best classified as ecological interfaces [23, 5] because they seek to enable decision-making easier and more intuitive for a human using a natural metaphor, in this case, a color palette. The metaphor is designed to help a human create a mental model of the tradeoffs and how changing from one solution to another alters how tradeoffs are balanced [22]. The three objectives that we consider are represented by the colors red, green and blue respectively, and the problem domain is supervisory control of a remote robot. Designing interfaces for supervisory control is an important part human-robot interaction (*HRI*), and designing intuitive and efficient interfaces has been a challenging issue in HRI [9, 11].

3. ADVERB PALETTE

The *Adverb Palette (AP)* designs are mouse-based interactive GUIs designed to help a human express intent over Pareto optimal tradeoffs. AP interfaces help a user to trade off among objectives in a way a painter selects colors from a given set of colors. A blend/mixture of colors corresponds to a single tradeoff from the available Pareto optimal tradeoffs. The AP designs are general enough to work for many problems with tradeoffs, and the designs and evaluation are applied to robot path-planning.

The path planning problem is for a robot to go from a start location x_{init} to a goal location x_{goal} within a configuration space (in this paper, a 2-D world). Each GUI has two parts: the *map* in the left panel, which is a task-specific interface that aids visualization of paths, and the *command interface (CI)* in the right panel, which is a general-purpose interface that a user can use to balance different adverbs. The command area allows a user to express intent by balancing tradeoffs, and the map gets updated to show task-specific details by highlighting the corresponding path.

Consider three adverbs, Quickly, Stealthily, and Safely, symbolized by colors *red*, *green* and *blue*, respectively.

- Quickly: prefer shorter paths.
- Stealthily: avoid being viewed by enemies.
- Safely: stay away from obstacles.

Given the Pareto optimal solution set, the goal is to enable a user to find a tradeoff that best expresses his or her intent. Expressing intent has two subproblems to be solved:

1. express a desired tradeoff, and
2. map the tradeoff to a Pareto optimal solution.

We focus on the aspect of human intent that requires satisfactory tradeoffs between competing objectives.

All interfaces include a task-specific map that shows all the routes (paths) in gray. Before tradeoffs are explored, a highlighted path is displayed that gives equal preference to all the adverbs. This paper refines intial *palette*, *sliders*, and *waypoints* from prior work [19] and introduces the *prism* design. This paper also discusses the mapping from intent to solution, and evaluates the designs through a user study. We begin by discussing how to condition the objectives so that they can be expressed using the color-blending metaphor.

3.1 Conditioning Objectives

Objectives may be expressed in incommensurable units, and this causes problems for using the palette metaphor. We perform an affine transformation and normalize objectives so that the multiple objective criterion can be reasonably compared.

Let S denote the set of Pareto optimal solutions, and let the costs associated with a particular solution $s_j \in S$ and objective $k \in \{1, \dots, K\}$ be denoted by $c_k(s_j)$. (Please check our prior work [19] for cost functions computations applied to robotic path planning).

For each solution, we convert the solution costs to solution pay-offs by multiplying by -1 yielding pay-offs for each objective $p_k(s_j) = (-1)c_k(s_j)$ and then normalize them to $[0, 1]$

$$\hat{p}_k(s_j) = \frac{p_k(s_j) - \min_{s_\ell \in S}\{p_k(s_\ell)\}}{\max_{s_\ell \in S}\{p_k(s_\ell)\} - \min_{s_\ell \in S}\{p_k(s_\ell)\}}.$$

The corresponding normalized vector $\boldsymbol{p}(s_j) \in [0, 1]^K$ for a solution is thus given by

$$\boldsymbol{p}(s_j) = [\hat{p}_1(s_j), \hat{p}_2(s_j), \cdots, \hat{p}_K(s_j)]^T. \qquad (1)$$

3.2 Palette

The *palette* displays three initial circles called the "primary dabs," one for each adverb (objective). The user expresses intent by clicking on one of the primary dabs (e.g., take the shortest path) or creates tradeoffs by dragging and dropping adverbs color dabs into the white area of the *CI* to create smaller circles called "paint dab" that blend colors. By creating different paint dabs and then exploring how each dab corresponds to a different path, a user can visualize the consequences of different commands. Line segments connect either the primary dabs and paint dabs or paint dabs to other paint dabs, producing a tree structure that allows the human to see the proportions of each objective.

Figure 2 shows an example command where the user desires a path that is quick and safe but does not care about being seen by enemies, which is represented numerically as " 50% quickly, 0% stealthily, 50% safely ". The pie graph on the lower left area in the *CI* shows the proportion of each objective in a particular paint dab. Blending in multiple adverbs (colors) is thus equivalent to making tradeoffs with multiple objectives. The default magenta paint dab in Figure 2 is an equal mixture "33.33% quickly, 33.33% stealthily, 33.33% safely".

Let dab_d represent any paint dab on *CI*. Let n_i be the number of times the user has dragged adverb i on dab_d, where $0 > i \leq K$. The total number of drags a user makes for dab_d is $n = \sum_{i=1}^{K} n_i$. The *user's intent* from the palette is the vector $\vec{h}^\text{pal}$

$$\vec{h}^\text{pal} = \left[\frac{n_1}{n}, \frac{n_2}{n}, \dots, \frac{n_K}{n}\right]^T. \qquad (2)$$

3.3 Sliders

Figure 3 shows the *sliders* interface. The user adjusts the trackbars to get to a desired mixture, and the corresponding solution/path from the left panel is selected. The three

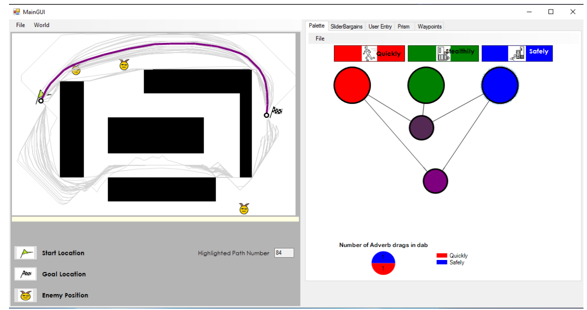


Figure 2: Quick and safe command in lowermost dab: map shows the corresponding highlighted path.

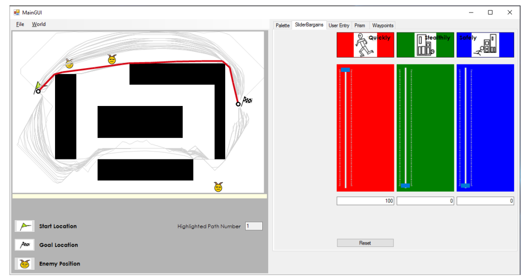


Figure 3: Adverb Palette: Slider interface.

sliders represent the three adverbs. The user can issue any of the three primary commands to the robot (e.g., take the shortest path) by sliding the corresponding slider (e.g., red) to the maximum units. The sum of the units on the sliders does not exceed 100 units, so if the *red*, *green*, or *blue* sliders are at say $33, 33, 34$ units respectively, then moving the *blue* slider to 60 units will cause a change to the slider units to $20, 20, 60$ units respectively. Unlike the palette, the user can explore different tradeoffs while moving a slider, and settle down to a certain position if she desires it. As the user moves one slider the other two sliders move automatically to maintain the sum to 100 units, and the corresponding solution/path gets shown on the map.

Let s_i is the score specified by slider i. The maximum unit on a slider corresponds to the cheapest solution for that objective and the minimum unit corresponds to the most expensive solution. The *human intent* can be represented as a vector $\vec{h}^\text{sli}$ as:

$$\vec{h}^\text{sli} = \left[\frac{s_1}{100}, \frac{s_2}{100}, \cdots, \frac{s_K}{100}\right]^T. \qquad (3)$$

3.4 Prism

Figure 4 shows the *prism* interface. Here the user can move the mouse over different areas of *prism* and discover its associated paths. Each point on the prism is a color corresponding to a certain proportion of adverbial objectives, expressed using a *barycentric coordinate system*.

As a review of barycentric coordinates, consider a triangle defined by three vertices, R, G, and B. Any point P inside or on the triangle can be written as a unique convex combination of the three vertices. Figure 5 illustrates the concept. The dots on the edges and those inside the triangle are example points that P may take. For a point P there is a unique sequence of three numbers such that the sum of these three numbers is 1. The three numbers, denoted by α, β, and γ

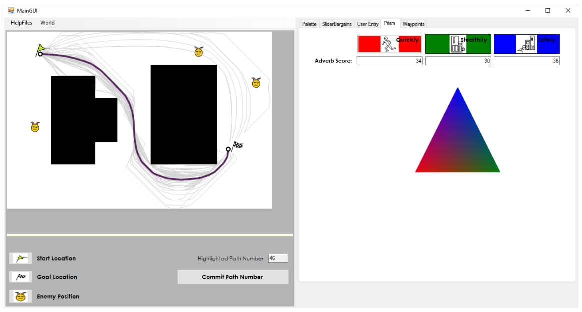


Figure 4: Adverb Palette: Prism interface.

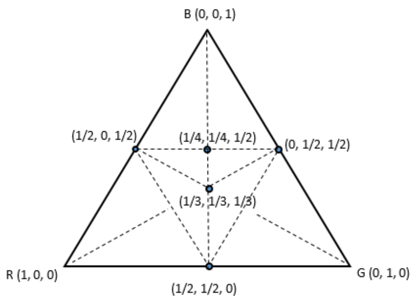


Figure 5: Barycentric coordinates on an equilateral triangle.

indicate the *barycentric* coordinates of point P with respect to the triangle. In the prism interface, α, β, and γ represent the proportion of *quickly*, *stealthily*, and *safely*, respectively. Intent for the prism is represented as

$$\vec{h}^{\text{pri}} = [\alpha, \beta, \gamma]^T. \tag{4}$$

The prism interface only works with three coordinates, and is therefore limited to three objectives.

3.5 Waypoints

The *waypoints* interface is path-planning specific while the other three interfaces are generic AP designs. The *waypoints* interface assists a user to construct her own path on the map by allowing her to provide location guidelines that the robot should visit while taking a path. Unlike the other three interfaces, the user here does not make a tradeoff among the available paths from the algorithm but instead makes her own path on the map. She can however compare her path with the best or worst with respect to an adverb based on the Pareto optimal paths' best and worst for that particular adverb. Figure 6 shows a path constructed using the waypoints interface. The graphs on the right panel show how well the user-created path compares to the best and worst objective scores from the solutions in the Pareto set.

4. MAPPING BETWEEN STRATEGIC INTENT AND PAY-OFFS

The *palette* and *sliders* interfaces produce a human intent vector denoted by $\vec{h}^{\text{pal}} = [h_1, h_2, ..., h_K]^T$ and $\vec{h}^{\text{sli}} = [s_1, s_2, ..., s_K]^T$, respectively. The *prism* interface produces the intent $\vec{h}^{\text{pri}} = [\alpha, \beta, \gamma]^T$. We can interpret this tradeoff in a vector space by associating intent with an orientation/direction with respect to some reference frame. More

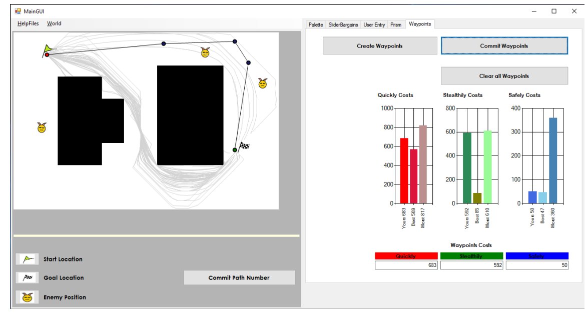


Figure 6: Waypoints interface.

precisely, we operationally define intent as a vector $\vec{h}$ that represents an ideal tradeoff, that is the balance between competing objectives that the human wants to achieve.

The multi-objective optimization algorithm gives a pool of possible solutions. Since each Pareto optimal solution has pay-off values associated with it, it can also be represented as a payoff vector using Eq. (1). Thus we have two vector representations, a human intent vector and the Pareto-optimal solution expressed as a payoff vector.

Given the human intent $\vec{h}$ and the payoff vector for every Pareto optimal solution, we need a mechanism to match the intent to one of the solutions. Intuitively for $\vec{h}$, the solution that has each of the individual pay-off values most closely matching to the corresponding individual intent values would be the one that would be finally selected. The mapping between the tradeoff point and the solutions would then be defined as closeness of the intent to the Pareto optimal solution.

We subjectively evaluated four different mapping strategies. Two of these strategies, WPM and TOPSIS are detailed in [3]. The others that we considered are the popular methods *euclidean distance* and *cosine similarity* for finding similar or matching entities. TOPSIS, WPM, and cosine similarity all gave the same results for mapping. Cosine similarity is the most simple, and subjectively produced better results than euclidean distance.

The cosine similarity between a path vector, $\boldsymbol{p}(s_j)$ and the human intent vector is $\boldsymbol{h}$ is $\frac{\boldsymbol{h} \cdot \boldsymbol{p}(s_j)}{\|\boldsymbol{h}\| \|\boldsymbol{p}(s_j)\|}$. For the given $\boldsymbol{h}$, if the solution $\boldsymbol{p}(s_j)$ ends up with the same orientation, then they have the cosine similarity of 1, and if they are at, say, 90° apart then they end up with the cosine similarity of 0 indicating that they have nothing in common.

Consider Figure 7. The triangle represents the set of possible tradeoffs. Each of the dots on the triangle represents a human intent, and each of the dots to the upper right of the triangle represents a Pareto optimal solution. The dark vectors represent the objective forming the space, stealth, safety, and quickness. The other vectors represent intent and solution vectors. θ_i represents the cosine similarities between the intent vector $\boldsymbol{h}_{\text{ex}}$ and the solution vector $\boldsymbol{p}(s_i)$

5. OBJECTIVE FUNCTIONS

Solving multi-objective optimization problems require computing costs or pay-offs for the involved objectives. Since here we apply AP to robot path-planning, we briefly describe in the following paragraphs the costs computed for a

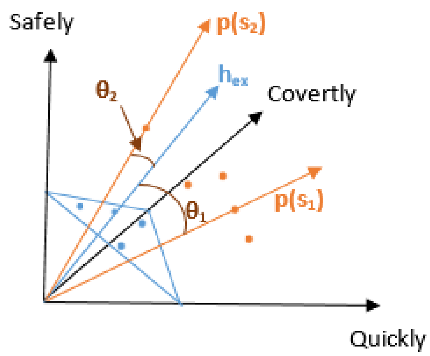
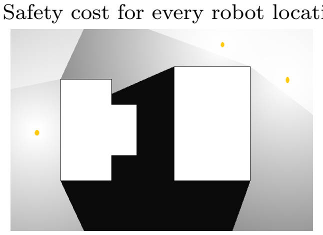


Figure 7: Path Comparison w.r.t example human intent vector $\boldsymbol{h}_{\text{ex}}$.

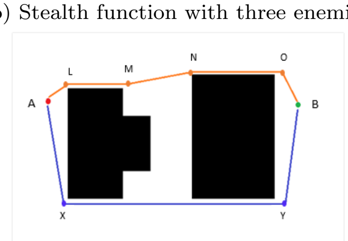
particular robotic-path. Recall that for this application we consider three costs; *quickly*, *stealthily* and *safely*.

A safe path is a collision free path. Hence, the safety cost of a robot location in a configuration space is encoded as a function of inverse distance between the robot position and the nearest obstacle in that space. The cost can be computed for every possible point in the configuration space. Therefore, the safety cost of a particular robotic path is the accumulation of the safety cost of individual points that make the path; see Figure 8a for an example safety cost for the polygonal obstacles – darker colors are safer.

(a) Safety cost for every robot location.



(b) Stealth function with three enemies.



(c) Path **ALMNOB** is quicker than the path **AXYB**.

Figure 8: Objective functions.

A stealthy path is less likely to be detected or seen by enemies in the world. The stealth cost function is expressed in terms of the probability of the path being seen by the enemy, and is computed as a function of two factors: the distance of the robot from each of the enemies and the visibility of the robot from the enemies. The resulting effect yields detection likelihood of the robot from the enemies. The stealth cost can thus be determined for every possible point of the robot location in the configuration space. Therefore, the stealthy cost of a path (starting from the initial state to the goal state) can be determined as the sum of the stealthy costs of individual points constituting the path. Figure 8b illustrates a world with three enemies and its corresponding stealthily objective function. If the robot has to travel from the top left corner to the bottom right corner of the configuration space, a path that goes between the obstacles and lower part of the space is more stealthy than a path that goes through the left side of the left obstacle in the space.

A quick path minimizes path length (assuming constant robot speed). The 'quickly' cost is the euclidean distance between the start and the goal position such that the obstacles do not intercept the path. Figure 8c shows two path options going from start location **A** to goal location **B**. The path distance of the path formed by points **ALMNOB** is less than the path distance for **AXYB**, hence the orange path is comparatively quicker than the blue path.

6. USER STUDY

Following a pilot study among university students to refine the AP designs, we conducted an IRB-approved user study to evaluate the four GUI principal designs. The aim was to discover how successful would be the user in finding tradeoffs among the given solutions using the GUI designs.

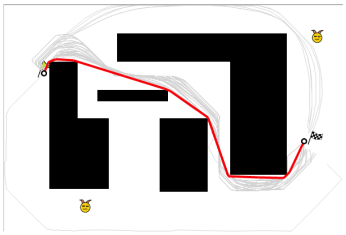
Participants were invited for a one-hour study through an advertisement posted in various departments of the university. 24 people participated, 17 males and 7 females, with a mean age of 24.8. All but one participant was a university student. The participants belonged to diverse majors including food, film, elementary education, nursing, and computer science. Each participant received $15 as compensation. All participants completed all the tasks assigned for the study.

6.1 User Study Procedure

After completing the informed consent process, each participant completed a short demographic questionnaire that included questions on familiarity with using a mouse, exposure to video games, age, and education. The participant then watched a 19 minute video tutorial that described the four GUI designs and showed how to use them in response to a particular scenario or task. Participants were issued a command in written English for the robot to perform such as:

> It is critical to the commander that the robot takes a quick and safe path. Enter the highlighted path number as per this command.

Following the training, participants executed four sets of practice tasks, one for each interface. The practice tasks and the world on which practice tasks were carried out were identical for each interface. Each practice task had an ideal path indicated by a path number, and the task was designed in a way that the user could easily figure out this path on the map in response to the command. The user was allowed three practice attempts to choose the correct path.



(a) Easy world/command: "Issue a command that makes the robot reach its goal as quickly as possible. Enter path number below."

(b) Hard world/command: "It is critical to the commander that the robot takes a path that hides the robot from enemy and that does not come close to buildings. The commander doesn't care if the distance from the start to the goal is big. Enter path number below."

Figure 9: Difficulty level: (a) Easy and (b) Hard.

The experiment was a two-factorial experiment with factors being interface type (palette, sliders, prism, waypoints) and difficulty level (easy, hard). The *difficulty level* is a function of two components: The first component is fairly general, namely, choosing a tradeoff is harder if it has to deal with more objectives/adverbs. The second component is task specific, namely the number of obstacles in the worlds. Hard tasks demand tradeoffs that involve multiple objectives (more than one) and have more obstacles, and easy tasks demand tradeoffs on at the most only two objectives and have fewer obstacles. Four sets of easy tasks and four sets of hard tasks were designed (two tasks in each set), allowing unique pairings of interfaces and worlds. Figure 9(a) and (b) show an example of an easy and hard task, respectively.

Subjective Workload: NASA TLX Scores. Each participant evaluated tasks using all six categories of the NASA TLX questionnaire: *mental demand*, *physical demand*, *temporal demand*, *performance*, *effort* and *frustration*.

Interface Appeal. After completing all tasks, participants ranked the interfaces that reflected their preference for three categories, ranked from most preferred to least preferred. The categories are how appealing the interface is, how easy the interface is to use, and how time-consuming the interface is to use.

Objective Metrics. In addition to subjective workload and user preferences, we evaluated the AP designs using three objective metrics. In each task, a command was given to the participant via the user interface; the command was constructed to describe an ideal path. The first metric evaluates how well participants could express tradeoffs, and the other metrics included both expressing tradeoffs and selecting paths.

- *Accuracy* quantifies the degree to which the tradeoff/solution selected by the user matches the intended tradeoff. Accuracy is measured as the cosine similarity between the intended tradeoff vector and the user-selected tradeoff vector.
- *Interface time* is the time spent performing all tasks required in a particular interface and world.
- *Answer time* is the time spent executing the tasks. For each individual task, answer time is captured from the first click or drag made on the GUI interface to the last click or drag made on the interface. It excludes the time spent to type in the answer for a task. The answer times for each individual tasks are then added to get the total answer times for all tasks on an interface for a particular world.

Participants were not given feedback on whether they executed the tasks correctly or not, and the order of interface/difficulty level was counterbalanced.

6.2 Hypothesis testing

We evaluated the following hypothesis:

1. *Hypothesis 1*: Each AP interface design can be used to successfully complete all assigned tasks.
2. *Hypothesis 2*: Hard tasks have longer completion times and higher subjective workload than easy tasks.
3. *Hypothesis 3*: The interfaces *palette* and *prism* would produce the lowest workload and shortest completion times.

7. RESULTS

Hypotheses were tested using SAS with *Restricted Maximum Likelihood Estimation* for a mixed-design ANOVA using Tukey-Kramer adjustment on subjects.

7.1 F statistics

Table 1 shows the effect of interface, difficulty level, and the combined effect of interface and difficulty on different measures/metrics of user's interaction. The asterisk $*$ denotes significant differences. There were significant differences in interface design and difficulty, but there were few differences for interface plus difficulty level.

7.1.1 Accuracy

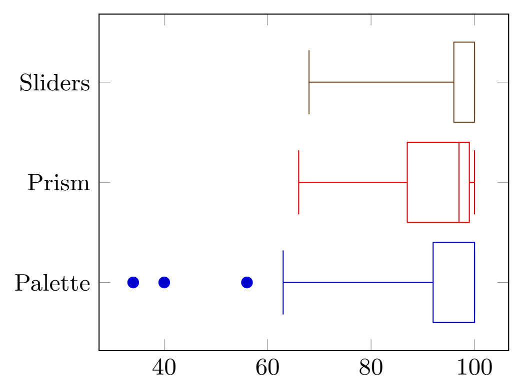


Figure 10: Accuracy of different interfaces.

The waypoints interface was statistically less accurate than all other interfaces, and the other three interfaces had no statistically significant differences; see Figure 10. Difficulty level had no effect on accuracy.

Metrics	Interface (I/F)		Difficulty (DL)		I/F × DL	
	F Value	Pr > F	F Value	Pr > F	F Value	Pr > F
Accuracy	4.28	0.006*	0.02	0.885	3.42	0.018*
Answer Time	44.24	< 0.001*	2.32	0.13	0.16	0.93
Interface Time	98.20	< 0.001*	6.39	0.013*	0.10	0.962
Mental	27.02	< 0.001*	2.33	0.129	0.29	0.832
Physical	8.77	< 0.001*	0.07	0.788	0.16	0.923
Temporal	12.08	< 0.001*	1.25	0.266	0.39	0.762
Performance	17.33	< 0.001*	11.47	< 0.001*	0.32	0.81
Effort	26.05	< 0.001*	5.66	0.0185*	0.21	0.887
Frustation	13.39	< 0.001*	0.87	0.352	0.27	0.846

Table 1: Effect of interface, difficulty level, and interaction of interface and difficulty level on various measures.

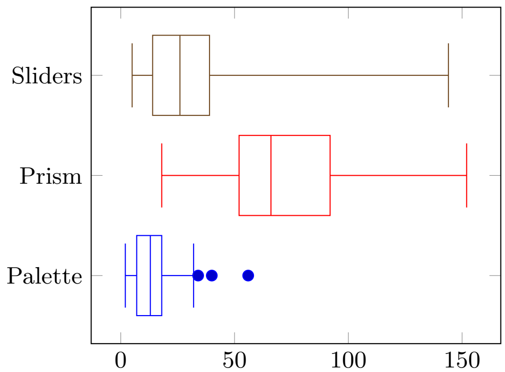


Figure 11: Answer Time in seconds on different interfaces.

Interface	Interface	t value	p
Palette	Prism	-4.31	< 0.001*
Palette	Sliders	-1.43	0.482
Palette	Waypoints	-10.65	< 0.001*
Prism	Sliders	2.88	0.023*
Prism	Waypoints	-6.34	< 0.001*
Sliders	Waypoints	-9.22	< 0.001*

Table 2: Pairwise differences in answer time.

7.1.2 *Answer Time*

Table 2 shows p values for pairwise differences between interfaces for answer time (negative t value indicates that the answer time on *prism* is higher than on *palette*). *Palette* and *sliders* are similar, *prism* is statistically slower than *palette* and *sliders*, and *waypoints* is statistically slower than them all. Figure 11 illustrates the differences for *palette*, *prism*, and *sliders*. Difficulty level has no impact on answer time.

7.1.3 *Interface Time*

Interface	Interface	t value	p
Palette	Prism	-2.48	0.066
Palette	Sliders	-0.79	0.86
Palette	Waypoints	-15.07	< 0.001*
Prism	Sliders	1.69	0.331
Prism	Waypoints	-12.59	< 0.001*
Sliders	Waypoints	-14.28	< 0.001*

Table 3: Pairwise differences of Interface Time.

Except for *waypoints*, which was signficantly slower, interface type did not affect interface time. *Palette*, *sliders* and *prism* were not significantly different (see Table 3).

Task difficulty did have an effect on interface time. *Interface Time* for hard tasks was higher than for easy tasks ($t = -2.55, p = 0.0117$). Table 4 shows the statistics as a function of individual interfaces.

I/F	Task	Interface time	
		t value	p
Palette	E vs H	-3.03	0.006*
Prism	E vs H	-2.92	0.008*
Sliders	E vs H	-2.8	0.01*
Waypoints	0.689	-0.53	0.603

Table 4: Interface Time on different interfaces. t value computed as E minus H.

7.1.4 *Subjective Workload*

We used a 20-point scale/score for each of the NASA-TLX question. Except for the performance scale, a value of one corresponds to least workload factor and the score of 20 suggests highest workload. For the *performance* NASA TLX factor, the highest value is the best. Table 5 shows the values obtained from the mixed-design ANOVA on the partcipant's NASA TLX scores. It is seen that the interface type affects workload. The *palette* and *sliders* have similar workload profiles. Furthermore, *waypoints* deviates from every other interface, and *prism* deviates from *palette* and *sliders*. Summarizing, the workload increased roughly in the following order palette $\prec$ sliders $\prec$ prism $\prec$ waypoints.

Difficulty level impacted two NASA TLX scores. Performance is significantly worse on hard tasks ($t = 3.41, p < 0.001$). Effort was also significantly worse on hard tasks ($t = 2.40, p = 0.018$).

7.2 User Preference

Participants ranked the four interfaces with respect to *appeal*, *ease of use*, and *time consuming* on an integer scale of 1 to 4, where 1 is best 4 is worst. Results showed that all interfaces show significant differences.

Table 6 suggests that the most appealing interface to the users was the *palette* and the least appealing was the *waypoints*. The suggested order of appeal is palette $\succ$ sliders $\succ$ prism $\succ$ waypoints.

Each interface differed significantly from the others in terms of ease of use, with *palette* being the easiest to use and *waypoints* being the hardest. Furthermore, *sliders* was

I/F	I/F	Mental	Physical	Temporal	Performance	Effort	Frustration
Palette	Prism	0.022*	0.02*	0.089	0.009*	0.009*	< 0.001*
Palette	Sliders	1	1	0.93	1	0.97	0.98
Palette	Waypoints	< 0.001*	< 0.001*	< 0.001*	$< .001$*	< 0.001*	< 0.001*
Prism	Sliders	0.015*	0.02*	0.018*	0.021*	0.034*	0.002*
Prism	Waypoints	< 0.001*	0.596	0.073	0.011*	< 0.001*	0.648
Sliders	Waypoints	< 0.001*	< 0.001*	< 0.001*	< 0.001*	< 0.001*	< 0.001*

Table 5: Pairwise differences for subjective workload computed using NASA TLX. I/F = Interface.

I/F	I/F	Appeal		Ease ofUse		Time Cons	
		t value	p	t value	p	t value	p
Palette	Prism	-6.65	< 0.001*	-11.4	< 0.001*	-13.79	< 0.001*
Palette	Sliders	-2.52	0.06	-5.87	< 0.001*	-9.19	< 0.001*
Palette	Waypoints	-10.08	< 0.001*	-20.23	< 0.001*	-25.66	< 0.001*
Prism	Sliders	4.12	< 0.001*	5.55	< 0.001*	4.6	< 0.001*
Prism	Waypoints	-3.44	0.004*	-8.81	< 0.001*	-11.87	< 0.001*
Sliders	Waypoints	-7.56	< 0.001*	-14.36	< 0.001*	-16.47	< 0.001*

Table 6: Pairwise differences between interfaces for appeal variables.

easier than *prism*, with both lying between the two extremes of *palette* and *waypoints*.

The *time-consuming* variable for interfaces was very similar to *ease of use*. The p-values suggest that each of the interfaces differed significantly from each other with *palette* being the least time-consuming and *waypoints* being the most time-consuming. Furthermore, *sliders* took less time than *prism*, with both lying between the two extremes of *palette* and *waypoints*.

Thus, all the interfaces were different from each other for the appeal variables, where in each case *palette* was preferred to the interfaces with *sliders* second.

7.3 Discussion

Results indicate that *waypoints* interface is significantly worse than the other three. This is not surprising since the waypoints interface requires participants to both plan a path and explore tradeoffs. It takes more time, induces higher subjective workload, and produces paths that differ significantly from the path intended in the command. We ignore this interface and discuss the others.

Hypothesis 1. Results of accuracy showed that there were no significant differences between *palette*, *sliders*, and *prism*, meaning that the users were able to find an acceptabe tradeoff using each interface. Each interface produced at least 90% accuracy, and difficulty level had no impact on accuracy. We fail to reject hypothesis 1, which suggests that each user can use each of the interfaces successfully.

Hypothesis 2. Difficult tasks took more time and subjective workload than easy tasks. We therefore find support for hypothesis 2.

Hypothesis 3. Both *palette* and *sliders* produced similar interface times, but *prism* required more time to answer the tasks, thereby making it significantly less effective. Similarly, both *palette* and *sliders* produced similar subjective workload, and *prism* had significantly higher subjective workload. The results of users' preferences demonstrated that users preferred *palette* to find tradeoffs among the Pareto optimal solutions. The interface *sliders* followed suit, and then *prism*. Hence, we reject hypothesis 3. Instead *palette* and *sliders* were similarly usable, and *prism* was significantly more challenging.

We hypothesize that participants found it hard to comprehend the mixing of adverbs through *prism*. Note that *prism* used the same optimal solutions that the palette and the sliders used, but participants found it hard to know where to click on the prism to get the solution for a task.

8. SUMMARY AND FUTURE WORK

We have presented four interactive GUI designs for selecting tradeoffs from among Pareto optimal solutions to a multi-objective optimization problem. The AP interface designs provide a novel way of blending objectives and enables users to find and express tradeoffs. The user study indicated that the *palette* and *sliders* designs were usable and relatively easy to use because of its metaphor of mixing colors in an artist's way. A rough aggregation of all measures suggests a slight superiority for the *palette* over *sliders*, and both were superior to the *prism* design, presumably because participants had a hard time understanding this interface.

The results from the *waypoints* interface design suggest that providing an interface that explicitly enables a participant to express tradeoffs is useful. Since expressing tradeoffs are an important part of expressing human intent, an interface that helps a user to understand and express tradeoffs may be useful for many problems.

Some of the participants explicitly disliked *prism*. Future work should be performed on *prism* such as naming or scaling the boundaries of the interface so that people can more easily understand how *prism* works.

Since the GUI designs presented here only consider three adverbs, future work should make GUI designs generic so that they can be applied to a variable number of objectives. It is possible that mixing more colors will make the interface less intuitive, so future work should explore the limitations on the interface as a function of more colors. Also, the current application was robotic-path planning. Future work should explore whether the GUI for other applications that require tradeoffs, such as a social robot that must find a path so that it balances proxemic concerns with energy or safety concerns. Finally, future work should explore palette-based designs that do not rely exclusively on color, adding redundant cues to aid easier human perception.

9. ACKNOWLEDGMENTS

This work was funded by the United States Army Research Laboratory through contract W911NF-14-1-0633. All results, findings, and conclusions are the responsibility of the authors and do not necessarily reflect the opinions of the funding organization.

10. REFERENCES

[1] J. A. Adams. Critical considerations for human-robot interface development. In *Proceedings of 2002 AAAI Fall Symposium*, pages 1–8, 2002.

[2] F. Ahmed and K. Deb. Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms. *Soft Computing*, 17(7):1283–1299, 2013.

[3] F. S. Azar. Multiattribute decision-making: use of three scoring methods to compare the performance of imaging techniques for breast cancer detection. 2000.

[4] M. Baker, R. Casey, B. Keyes, and H. A. Yanco. Improved interfaces for human-robot interaction in urban search and rescue. In *SMC (3)*, pages 2960–2965. Citeseer, 2004.

[5] K. B. Bennett, S. M. Posey, and L. G. Shattuck. Ecological interface design for military command and control. *Journal of Cognitive Engineering and Decision Making*, 2(4):349–385, 2008.

[6] J. Bruce and M. Veloso. Real-time randomized path planning for robot navigation. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference*, volume 3, pages 2383–2388. IEEE, 2002.

[7] A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference*, pages 723–730. IEEE, 2011.

[8] J. Y. Chen, E. C. Haas, and M. J. Barnes. Human performance issues and user interface design for teleoperated robots. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1231–1245, 2007.

[9] F. Driewer, M. Sauer, and K. Schilling. Discussion of challenges for user interfaces in human-robot teams. In *EMCR*. Citeseer, 2007.

[10] T. Gal, T. Stewart, and T. Hanne. *Multicriteria decision making: Advances in MCDM models, algorithms, theory, and applications*, volume 21. Springer Science & Business Media, 2013.

[11] M. A. Goodrich and A. C. Schultz. Human-robot interaction: A survey. *Foundations and trends in human-computer interaction*, 1(3):203–275, 2007.

[12] D. S. Hall, L. G. Shattuck, and K. B. Bennett. Evaluation of an ecological interface design for military command and control. *Journal of Cognitive Engineering and Decision Making*, 6(2):165–193, 2012.

[13] C.-L. Hwang and K. Yoon. *Multiple attribute decision making: Methods and applications a state-of-the-art survey*, volume 186. Springer Science & Business Media, 2012.

[14] M. W. Kadous, R. K.-M. Sheh, and C. Sammut. Effective user interface design for rescue robotics. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot interaction*, pages 250–257. ACM, 2006.

[15] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, 1996.

[16] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.

[17] K. Liu, D. Sakamoto, M. Inami, and T. Igarashi. Roboshop: multi-layered sketching interface for robot housework assignment and management. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 647–656. ACM, 2011.

[18] D. Sakamoto, Y. Sugiura, M. Inami, and T. Igarashi. Graphical instruction for home robots. *Computer*, 49(7):20–25, 2016.

[19] M. T. Shaikh, M. A. Goodrich, D. Yi, and J. Hoehne. Interactive multi-objective path planning through a palette-based user interface. In *SPIE Defense+ Security*, pages 98370K–98370K. International Society for Optics and Photonics, 2016.

[20] O. M. Shir, S. Chen, D. Amid, D. Boaz, A. Anaby-Tavor, and D. Moor. Pareto optimization and tradeoff analysis applied to meta-learning of multiple simulation criteria. In *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, pages 89–100. IEEE Press, 2013.

[21] Y. Sugiura, D. Sakamoto, A. Withana, M. Inami, and T. Igarashi. Cooking with robots: designing a household system working in open environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2427–2430. ACM, 2010.

[22] A. B. Talone, E. Phillips, S. Ososky, and F. Jentsch. An evaluation of human mental models of tactical robot movement. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 59, pages 1558–1562. SAGE Publications, 2015.

[23] K. J. Vicente and J. Rasmussen. Ecological interface design: Theoretical foundations. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(4):589–606, 1992.

[24] J. Wallenius and S. Zionts. *Multiple criteria decision making: From early history to the 21st century*. World Scientific, 2011.

[25] 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*, pages 1733–1739. AAAI Press, 2015.