Who Goes *There*? Selecting a Robot to Reach a Goal Using Social Regret

Meytal Traub
The MAVERICK Group
Computer Science Dept.
Bar-Ilan University, Israel
meytaltra@gmail.com

Gal A. Kaminka
The MAVERICK Group
Computer Science Dept.
Bar-Ilan University, Israel
galk@cs.biu.ac.il

Noa Agmon
Computer Science Dept.
University of Texas at Austin
Austin, TX, USA
agmon@cs.utexas.edu

ABSTRACT

A common decision problem in multi-robot applications involves deciding on which robot, out of a group of N robots, should travel to a goal location, to carry out a task there. Trivially, this decision problem can be solved greedily, by selecting the robot with the shortest expected travel time. However, this ignores the inherent uncertainty in path traversal times; we may prefer a robot that is slower (but always takes the same time), over a robot that is expected to reach the goal faster, but on occasion takes a very long time to arrive. We make several contributions that address this challenge. First, we bring to bear economic decision-making theory, to distinguish between different selection policies, based on risk (risk averse, risk seeking, etc.). Second, we introduce *social regret* (the difference between the actual travel time by the selected robot, and the hypothetical time of other robots) to augment decision-making in practice. Then, we carry out experiments in simulation and with real robots, to demonstrate the usefulness of the selection procedures under real-world settings, and find that travel-time distributions have repeating characteristics.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Algorithms, Economics, Performance, Experimentation

Keywords

Multi-Robot Systems, Decision-Making, Regret

1. INTRODUCTION

A common decision problem in multi-robot settings involves deciding on which robot, out of a group of N robots, should travel to a goal location, to carry out a task there. This decision repeats in many applications: in multi-robot exploration (e.g., deciding who should go to explore a new frontier), in package delivery robots (e.g., deciding who should go to pick up a package), and in other service robotics applications (e.g., in hospitals). In all of these, robots can plan a path to reach their destination, in an environment that is—for the most part—known to them. Thus, in principle, they can analytically predict their travel time to any location.

Cite as: Who Goes *There*? Selecting a Robot to Reach a Goal, Meytal Traub, Gal A. Kaminka and Noa Agmon, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Yolum, Tumer, Stone and Sonenberg (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 91-98.

Trivially, this decision problem can be solved greedily, by selecting the robot with the shortest predicted travel time [12], or using a market-based allocation scheme (see [4]). However, this ignores the inherent variance in the actual path traversal times, due both to motion and sensing errors, as well as multiple factors that affect a robot's velocity (e.g., battery level, unknown obstacles). Solutions that have been proposed to address these challenges include using machine learning to better predict actual travel times under varying conditions [7, 14, 8], or other path-generation techniques that provide estimates [3, 2].

A common thread through previous work is that it focuses on scalar predictions; a single number that denotes the expected travel-time for each robot. Unfortunately, scalar predictions hide important information about the uncertainty in the predictions. In particular, a scalar denoting expected cost ignores information about the distribution of possible costs, best- and worst-case costs, etc. As a result, guarantees on the cost of task execution are not possible.

For instance, supposed that we must send one of two robots to a target location X. Robot A's path to X takes 100 seconds, through a free corridor. But if the corridor is busy with traffic (a rare occurrence), it may take up to 200. In contrast, robot B's travel time is always 150 seconds, through a specialized service way. Since the corridor is normally clear, we might choose robot A for the task. But if we wanted to absolutely guarantee delivery within 150 seconds, we would choose robot B. Note that if we only know the expected (i.e., mean) travel time, we cannot make the necessary distinction that allows this decision.

In this paper, we make several contributions that address the challenge involved in selecting a robot to go to a target location, given that each robot has a distribution over predicted travel times: First, we bring to bear economic decision theory that distinguishes between different selection policies, based on *risk*: risk averse, risk seeking, risk neutral, and bounded-risk selection. Second, we show that under some conditions, the selected robot may still not be a reasonable choice in practice. We thus introduce the use of *social regret* (the difference between the actual travel time by the selected robot, and the hypothetical time of another robot) to augment decision-making. Social regret is inspired by economic notions of regret, though the definitions differ.

Then, we carry out experiments in simulation and with real robots, to demonstrate the usefulness of the selection procedures under real-world settings. We empirically demonstrate that even under static conditions of the environment, when it is completely known to the robots, sensor and actuator errors leads to significant variance in the execution of path-following tasks. This variance leads to non-trivial distribution of costs, which in turn necessitates reasoning about the different optimization criteria when making the selection between robots. Finally, we show empirically that travel time distributions have repeating characteristics (specifically, they fit extreme value distributions).

2. BACKGROUND

There have been several investigations that attempt to predict travel time or related costs. To the best of our knowledge, none addressed complete distributions.

Heero et al. [8] present a method for learning the shortest path in a partially-known environment by using a rectangular grid-based map. For each path they saved four parameters: number of re-plans, travel time, travel distance and deviation from the originally pre-planned path. Chaudhry [3] presented an algorithm for generating paths using matrix representation of the robot's previous path traversals. New paths are created by applying transformations to the matrix, given the new path requirements. Both investigations use previous experiences to generate travel time predictions.

Sofman et al. [13] demonstrated an approach for learning Gaussian distributions associated with the local environments of the robot, to improve navigation and the travel speed. They use the models to generalize to new environments. They do not learn travel time distributions, and in any case as we show, travel time distributions are not Gaussian.

A related—though inverse—problem to ours is the problem of choosing a path, out of k possible paths, for a single robot to reach a goal location. Haigh and Veloso developed ROGUE [7]. It learns situation-dependent rules based on the success or failure in carrying out its tasks, and in particular, learns to take different paths depending on the time of day, expected use of the corridor, etc. ROUGE learns these situated-dependent cost predictions by examining the mean costs of travel for given locations. Thurn et al. developed MINERVA, an interactive tour guide robot for the Smithsonian museum [14]. It used POMDP methods to learn and plan its motion. The use of POMDP is similar to the risk-neutral policy, one of a number we present in this paper.

Our notion of regret is inspired by—but different than—notions of regret in economics. Economic regret were introduced by Bell [1] and by Loomes and Sugden [10], who concluded that people do not necessarily maximize their expected utility, but also consider the possible loss they are willing to accept from making a choice. They defined regret as a symmetric function with respect to two choices: choosing A rather than B minus the gain/loss from choosing B rather than A. In our case we calculate the regret with respect to all other choices, yet the comparison between the symmetric cases is done after the calculation (hence our regret function is asymmetric). Foster and Vohra [5] discussed regret in online decision-making, distinguishing between *internal* and *external* regret. Our definition of regret is similar to the definition of internal regret, however we evaluate with respect to the probabilities of costs, rather than on a limited history over time.

Market-based methods are sometimes used to assigning robots to tasks (e.g., a goal location to be reached; see [4] for a survey). In general, these methods rely on scalar cost estimates, and do not utilize information about travel cost distributions. However, they do address self-interest on the part of the robots, while in our work we assume robots are cooperative and truthful. Koenig et al. [9] uses a regret function, different from ours, to improve such auctions.

3. SELECTING A ROBOT

The problem is to select a robot R_i, out of a group of N robots $R_1, \dots R_N$, to carry out a task, while minimizing the cost. We assume that each robot can estimate its cost of task execution with some discrete probability distribution over k cost values $c_1, \dots, c_k$. Each robot R_i has a vector of size k, $< p_1^i, p_2^i, \dots, p_k^i >$ such that p_j^i is the probability that the cost of task execution (travel time, in our case) by robot R_i is c_j and $\sum_{j=1}^{k} p_j^i = 1$ (note that p_j^i can be equal to 0). We use this discrete distribution formalization for simplicity, in lieu of the continuous distribution case which is more natural for estimated travel times. Note also that each robot's travel time is an independent random variable, i.e., the probability of robot R_i having actual cost of c_j does not depend on the probability of some other robot having this or other cost.

We use the following running example throughout this section. Figure 1 shows three robots $\{R_1, R_2, R_3\}$. One of these robots is to be sent to explore a new frontier, F_1, shown in the bottom right corner (circled). Each robot constructs a path (not shown) to the new location, and reports a distribution over estimated travel times. As we show in the experiments (Section 4.1), even in a completely static environment (let alone in dynamic environments), sensor and motion uncertainties cause some variance in this distribution.

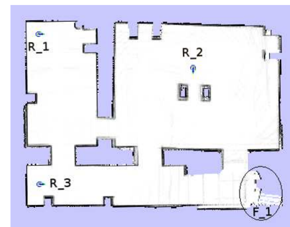


Figure 1: Three robots in exploration task. Map was generated using laser-based SLAM.

Suppose the travel time distributions reported by the 3 robots are as given in Table 1. Each row shows the distribution of a different robot, with different columns denoting different costs. The last column shows the mean (expected) cost for each robot. Given different decision objectives, we would choose different robots to go to F_1. For instance, R_2 is most likely to reach F_1 faster (has a 87% chance of reaching F_1 in 86 seconds). But R_2 may also take up to 134 seconds for the same path. If we wanted to guarantee arrival within 2 minutes, we would choose R_3.

	$c_1 = 86$	$c_2 = 98$	$c_3 = 110$	$c_4 = 122$	$c_5 = 134$	$E(C)$
R_1	$p_1^1 = 0$	$p_2^1 = 0.6$	$p_3^1 = 0.23$	$p_4^1 = 0.17$	$p_5^1 = 0$	104.84
R_2	$p_1^2 = 0.87$	$p_2^2 = 0.03$	$p_3^2 = 0$	$p_4^2 = 0$	$p_5^2 = 0.1$	91.16
R_3	$p_1^3 = 0.6$	$p_2^3 = 0.22$	$p_3^3 = 0.1$	$p_4^3 = 0.08$	$p_5^3 = 0$	93.92

Table 1: Possible cost distribution for $R_{1,2,3}$ for arriving to F_1.

3.1 Risk-based selection

Choosing the robot $R_c \in \{R_1, \dots, R_N\}$ to perform the given task is dependent on a decision policy, which prefers robots—all else being equal—based on the risk involved. For instance, if we have a fixed amount of time to explore a given area, we may want to select a robot that will definitely reach its target within the time allotted. On the other hand, we may decide to take more risks, hoping to reach the target faster than expected.

Such decision policies are well known in economic decision theory. We distinguish four well-defined policies, and outline the selection algorithm for each:

1. Minimize the expected travel time (risk neutral selection).

2. Minimize the expected maximal travel time (risk averse selection).

3. maximize the expected minimal travel time (risk seeking selection).

4. Bound the travel time by a constant A (bounded risk).

Risk-Neutral Selection. Risk-neutral selection implies that we select the robot that minimizes the expected (mean) travel time. To do this, we compute the mean of every robot's distribution, and choose the robot whose mean is minimal

$$MinExp_C = \underset{1 \le a \le N}{\operatorname{argmin}} \{\sum_{i=1}^{k} p_i^a c_i\}$$

where in case of a tie, we choose arbitrarily.

Risk-Averse Selection. In some cases we want to make sure that the worst-case scenario is addressed first, and that we have an absolute guarantee that the task will be carried out within a given amount of time. To do this, we need to look at the robots whose greatest time of arrival is minimal. Of course, the probability of actually taking this long time must also be taken into account. Thus what we want is to find the robot which minimizes the expected maximal cost. This is done in Algorithm 1.

Algorithm 1 MinExpMaxCost(R)

Require: $C = \{c_1, c_2, \ldots, c_k\}, R = \{R_1, R_2 \ldots, R_n\}$
  $v \leftarrow k$
  $Robots_{list} \leftarrow \{R_1, R_2, \ldots, R_n\}$
  while $\exists p_v^j = p_v^h, R_j, R_h \in Robots_{list}$ **do**
    $Robots_{list} \leftarrow \operatorname{argmin}_{R_i \in Robots_{list}} \{p_v^i\}$
    $v \leftarrow v - 1$
  $\mathbf{return}$: $\operatorname{argmin}_{\mathbf{R_i \in Robots_{list}}} \{\mathbf{p_v^i}\}$

Note that ties can be broken in different ways. For instance, we can choose the robot with the lower expected time among those that are returned.

We use Table 1 to illustrate. The algorithm creates a list of all the robots that available to execute the task $\{R_1, R_2, R_3\}$, and starts the run with the highest cost (134). It looks for two robots with the same probability to arrive the goal in cost 134. In this example R_1 and R_3 have the same probability ($p_5^1 = p_5^3 = 0$), so the loop will be entered. The algorithm choose the robots with the minimal probability to execute the task in cost 134 by argmin. By doing it, all the robots with probability higher than 0 will be removed from the list, i.e. robot R_2. The algorithm then examines the next highest cost, 122. While looking on the remaining robots $\{R_1, R_3\}$, their probability to arrive the target is different, therefor the loop will not be entered and the robot with the lowest probability to use this cost will be returned: R_3 ($p_4^1 = 0.17 > p_4^3 = 0.08$).

Risk Seeking Selection. The opposite policy to being risk averse is to be risk seeking; to hope for the best possible travel time of any of the robots. Here the selection is exactly the inverse of the above: We select the robot that maximizes the expected minimal cost. Algorithm 2 is thus the inversion of Algorithm 1.

We again use Table 1 to illustrate. The algorithm starts the run with the lowest cost, i.e. 86. It looks for two robots with the same probability to arrive the goal in cost 86. In this example, there is no two such robots, and so it does not enter the loop and return the robot with the highest probability to arrive the goal in this cost, R_2 ($p_1^1 = 0 < p_1^3 = 0.6 < p_1^2 = 0.87$).

Algorithm 2 MaxExpMinCost(R)

Require: $C = \{c_1, c_2, \ldots, c_k\}, R = \{R_1, R_2 \ldots, R_n\}$
  $v \leftarrow 1$
  $Robots_{list} \leftarrow \{R_1, R_2, \ldots, R_n\}$
  while $\exists p_v^j = p_v^h, R_j, R_h \in Robots_{list}$ **do**
    $Robots_{list} \leftarrow \operatorname{argmax}_{R_i \in Robots_{list}} \{p_v^i\}$
    $v \leftarrow v + 1$
  $\mathbf{return}$: $\operatorname{argmax}_{\mathbf{R_i \in Robots_{list}}} \{\mathbf{p_v^i}\}$

Bounded-Risk Selection. Finally, we may want to choose the robot that maximizes the probability of reaching the target within some limited amount of time. This is different from guaranteeing arrival within this time; it would still be possible that in the worst case, travel time will be longer. Nevertheless, we want to improve its chances of success within the time allotted.

Suppose we are given a time limit T. We can then calculate for each robot the cumulative probability that its travel time be smaller than T, and choose the robot that maximizes this probability.

For each robot R_a, we will calculate the following probability:

$$P[C <= T] = \sum_{c_i <= T, c_i \in C} p_i^a c_i$$

We will choose the robot that maximize the result of this equation. Note, that if only one robot have distribution of cost bellow the constant, then it will be chosen with probability of 1. If there is more than one robot that fits this, then we can select based on any of the other criteria (e.g., the best risk-seeking robot out of the candidates that fit the bound T).

3.2 Regretting the Selection

Despite the economic elegance of the selection policies described above, choosing the robot according to the risk type will not always give us a reasonable selection in practice. To see this, consider the following case (Table 2). Here, we apply the risk-averse policy, and select R_2: It is guaranteed to reach the goal in 199 seconds. However, unless this risk-averseness is somehow extremely strict, R_1 would have been a more reasonable choice: 90% of the time it would have reached the goal in 1 second. And even when it fails, it would do it in 200 seconds, a mere 1 second more than R_2.

	$c_1 = 1$	$c_2 = 199$	$c_3 = 200$	$E(C)$	E_{SoR}
R_1	$p_1^1 = 0.9$	$p_2^1 = 0$	$p_3^1 = 0.1$	20.9	0.1
R_2	$p_1^2 = 0$	$p_2^2 = 1$	$p_3^2 = 0$	199	178.2

Table 2: Robots distributions of costs to arrive at a goal. Expected cost and expected *SoR* are shown. Selecting R_1 over R_2 makes sense in practice.

Note that this is not always the case: It depends very much on the values c_i. If the c_i would have been $1, 2, 200$ rather than $1, 199, 200$, our deliberation would not have reached the same conclusion, and the selection of R_2 would have held.

To conduct this deliberation formally, we define the *social regret* function, which measures, intuitively, the post-hoc payment (in travel time) that we make, given the selected robot.

Social Regret *SoR* is defined as the difference between the *actual* cost c_r of the task executed by robot R_a and the minimal cost of task execution in case some other robot would have executed the task in lower cost. In other words, looking at it from the team's perspective: How bad did the team do by choosing robot R_a to perform the task, given R_a's actual cost was c_r. Formally, *SoR* of robot R_a executing a task with actual cost c_r is *SoR*$(R_a, c_r) = \max_{j \neq i}(c_r - c_j,\ r > j)$.

Since we do not know *SoR* for any specific selection (it is by definition hypothetical), we compute the *expected SoR* for each robot R_a, given all other robots, and all possible outcomes. The expected social regret from choosing R_a, $E_{SoR}(R_a)$, is the probability that some other robot will execute the task with lower cost multiplied by the difference between the costs. We denote the probability that the actual *minimal* cost of task execution by some robot other than R_a is c_i by $PM_i(R_a)$. Note that by *minimal* cost we mean that there is no other robot that executed the task with cost $c_j, j < i$, and that at least one robot executed the task with cost c_i. Therefore, $E_{SoR}(R_a) = p_1^a \times 0 + p_2^a \times PM_1(R_a)(c_2 - c_1) + p_3^a \times [PM_1(R_a)(c_3 - c_1) + PM_2(R_a)(c_3 - c_2)] + \ldots$, and formally

$$E_{SoR}(R_a) = \sum_{i=2}^{k} p_i^a \times \sum_{j=1}^{i-1} PM_j(R_a)(c_i - c_j)$$

In order to complete the definition, it is necessary to determine $PM_j(R_a)$, i.e., the probability that some robot R_o, $1 \leq o \leq N$, $o \neq a$ will have minimal cost of c_j. This is the probability that all robots have minimal cost higher than c_{j-1} minus the probability that all robots have minimal cost higher than c_j, i.e., $E_{SoR}(R_a) =$

$$\sum_{i=2}^{k} p_i^a \{\sum_{j=1}^{i-1}(c_i - c_j)[\prod_{h=1}^{N,h\neq a}(\sum_{l=j}^{k} p_l^h) - \prod_{h=1}^{N,h\neq a}(\sum_{l=j+1}^{k} p_l^h)]\}$$

3.2.1 When Should We Overrule The Selection?

Intuitively, $E_{SoR}(R_a)$ measures the potential cost of selecting R_a to carry out a task, given the estimated costs of its peers. Suppose that we have two robots R_i and R_j. What we want, is to compare the difference in the expected *SoR* of the two robots, to the gain from choosing one over the other. If this gain is smaller than the difference in expected *SoR*, then we should consider switching between them.

To illustrate, suppose R_i has been selected by some policy, and has a predicted travel time c_i (this is, for instance, its maximal time). Suppose we want to consider switching to a different robot R_j, with predicted cost c_j. In order to compute the profit form switching two robots we will calculate the distance between the expected *SoR* of R_i and R_j, $E_{SoR}(R_i) - E_{SoR}(R_j)$. We compare this value to the difference in costs between R_i and R_j, which is $(c_j - c_i)$, using the following function.

The Switch function *SwF* is defined as follows:

$$SwF = \begin{cases} 1 & \text{if } (E_{SoR}(R_i) - E_{SoR}(R_j)) > (c_j - c_i) \\ 0 & \text{otherwise} \end{cases}$$

If the *SwF* is 1, the social regret of using R_i is greater than the expected gain of using it, and we should consider switching our selection to R_j instead. We examine this in different selection policies below.

Minimize the expected maximal cost. Table 2 above describes the cost distributions for two robots, R_1 and R_2. As previously discussed, strict risk-averse policy would select R_2 for the task, since it is guaranteed to reach the target in 199 seconds. However, by risking just one additional second, we actually have much better average performance if we choose R_1.

SwF identifies this opportunity. The difference between c_3 (R_1's cost) and c_2 (R_2's cost) is 1, while the distance between the $E_{SoR}(R_2)$ and $E_{SoR}(R_1)$ is 178.1. Thus *SwF* is 1, and we should consider switching our selection to the other robot.

Switching in the case of a bounded risk. Using a bounded-risk policy, we normally select the path that is most likely to carry out the task within the time allotted. But by bounding the cost, we are not bounding the regret function. In other words, choosing the best robot given the bound T, does not reduce our expected *SoR* for the bounded cost, and we can still choose to switch based on *SwF*.

For example, table 3.2.1 shows distributions of costs of two robots, R_1 and R_2. A bound of $T = 7$, yields selection R_1 (with cumulative likelihood 0.2), over R_2 (cumulative likelihood 0). But the expected *SoR* of R_1 is much higher that the expected *SoR* of R_2. Indeed, using the *SwF* we might consider to change the constant T to be higher. By changing the constant T from 7 to 10, R_2 will have higher probability than R_1 to execute the task under the new bound. We will pay 3 in the bound but gain 70.6 in the expected regret $(E_{SoR}((R_1)) - E_{SoR}(R_2))$. The *SwF* will be 1 (70.6 > 3).

	$c_1 = 1$	$c_2 = 5$	$c_3 = 10$	$c_4 = 100$	E_{SoR}
R_1	$p_1^1 = 0.1$	$p_2^1 = 0.1$	$p_3^1 = 0$	$p_4^1 = 0.8$	72
R_2	$p_1^2 = 0$	$p_2^2 = 0$	$p_3^2 = 1$	$p_4^2 = 0$	1.4

Table 3: The bounded cost does not minimizes the expected *SoR*. When should we replace

3.2.2 Minimal Expected Cost is Safe Selection

For one of the policies we introduced, it turns out that we do not need to consider regret. We prove that by minimizing the expected cost, the expected social regret function, *SoR*, is minimized as well, and thus we would not want to switch to a different robot.

First, we show in Lemma 1 that for a pair of robots, minimizing the expected cost leads to minimization of the expected *SoR*. We then complete the proof for N robots in Theorem 2.

LEMMA 1. *For two robots R_1 and R_2 with discrete probability distribution $\{p_1^1, p_2^1, \ldots, p_k^1\}$ and $\{p_1^2, p_2^2, \ldots, p_k^2\}$ (respectively) over possible costs $c_1, \ldots, c_k$ of a given task , $c_i < c_{i+1}$, if the robot minimizing the expected cost of the task execution is chosen, then the expected social regret function SoR is minimized.*

PROOF. Assume, without loss of generality, that robot R_1 minimizes the expected cost of the task execution, i.e., $\sum_{i=1}^{k} p_i^1 c_i < \sum_{i=1}^{k} p_i^2 c_i$, i.e., $\sum_{i=1}^{k} c_i p_i^2 - c_i p_i^2 > 0$. We therefore need to show that $E_{SoR}(R_2) - E_{SoR}(R_1) > 0$.

First, note that since $N = 2$ then $PM_j(R_1) = p_j^2$, and similarly $PM_j(R_2) = p_j^1$, therefore $E_{SoR}(R_2) = \sum_{i=2}^{k} p_i^2 \times \sum_{j=1}^{i-1} p_j^1(c_i - c_j)$ (similarly $E_{SoR}(R_1) = \sum_{i=2}^{k} p_i^1 \times \sum_{j=1}^{i-1} p_j^2(c_i - c_j)$).

By opening the formula of $E_{SoR}(R_2)$ we get that
$E_{SoR}(R_2) = p_2^2\{p_1^1(c_2 - c_1)\} + p_3^2\{p_1^1(c_3 - c_1) + p_2^1(c_3 - c_2)\} + p_4^2\{p_1^1(c_4 - c_1) + p_2^1(c_4 - c_2) + p_3^1(c_4 - c_3)\} + \ldots$
$= c_1\{-p_1^1 p_2^2 - p_1^1 p_3^2 - p_1^1 p_4^2 - \ldots\} + c_2\{p_1^1 p_2^2 - p_2^1 p_3^2 - p_2^1 p_4^2 - \ldots\} + c_3\{p_1^1 p_3^2 + p_2^1 p_3^2 - p_3^1 p_4^2 - \ldots\} + \ldots$
Therefore $E_{SoR}(R_2)$ can be represented as $\sum_{j=1}^{k} c_j[\sum_{i=1}^{j} p_j^2 p_i^1 - \sum_{i=j+1}^{k} p_i^2 p_j^1] = \sum_{j=1}^{k} c_j[\sum_{i=1}^{j} p_j^2 p_i^1 - p_j^1(1 - \sum_{i=1}^{j} p_i^2)] = -\sum_{j=1}^{k} c_j p_j^1 + \sum_{j=1}^{k} c_j[\sum_{i=1}^{j}(p_j^2 p_i^1 + p_j^1 p_i^2)]$. Similarly, $E_{SoR}(R_1) = -\sum_{j=1}^{k} c_j p_j^2 + \sum_{j=1}^{k} c_j[\sum_{i=1}^{j}(p_j^1 p_i^2 + p_j^2 p_i^1)]$, therefore $E_{SoR}(R_2) - E_{SoR}(R_1) = \sum_{j=1}^{k}(c_j p_j^2 - c_j p_j^1) +$

$\sum_{j=1}^{k} c_j[\sum_{i=1}^{j}(p_j^2 p_i^1 + p_j^1 p_i^2) - (p_j^1 p_i^2 + p_j^2 p_i^1)] = \sum_{j=1}^{k}(c_j p_j^2 - c_j p_j^1)$ and by initial assumption this is greater than 0, which completes the proof. □

THEOREM 2. *Given a team of N robots $\{R_1, \ldots, R_N\}$ each with a discrete probability distribution over possible costs $v_1, \ldots, v_k$ for a given task, if we choose a robot R_c that minimizes the expected cost for the task, then the expected social regret function SoR is minimized.*

PROOF. (Sketch). By induction on the number of robots N. Assume, without loss of generality, that robot R_1 minimizes the expected cost of the task execution. For the induction base case $N = 2$, the theorem holds based on the Lemma 1. For the inductive step, suppose the theorem holds for $N - 1$ robots, where $N \geq 3$. We will show that if we choose R_c that minimizes the expected cost, then $E_{SoR}(R_N)$ is minimized.

Let $\{R_1, ..., R_N\}$ be a team of N robots. We will assume, without loss of generality that R_1 minimizes the executed cost of the task execution. Therefore in particular, R_1 minimizes the expected cost for all the possible couples of robots in the environment, i.e., for any given pair of robots (R_1, R_i) where $1 \leq i \leq N$, the selection of R_1 results in a minimal cost compared to the selection of R_i. We will prove by contradiction that R_1 minimizes the expected *SoR* for N robots.

We assume for contradiction that exist robot R_i, where R_1's expected cost is smaller than R_i's expected cost ($E_C(R_1) < E_C(R_N)$). But according to lemma 1, Let R_1 and R_i be a team of two robots, if $E_C(R_1) < E_C(R_i)$ then the expected social regret function ***SoR*** is minimized, R_1's expected ***SoR*** is bigger than R_i's expected *SoR* ($E_{SoR}(R_1) > E_{SoR}(R_i)$). In contradiction to the assumption. Therefore, the theorem holds for N robots, $N \geq 2$. □

Table 2 gives travel times distributions of two robots to a goal. As it shows in the table, if we will choose robot R_1 by minimizing its expected cost, we will minimizes the E_{SoR} as well.

3.2.3 A Short-Cut to Determining SwF

The computation of E_{SoR} for each robot, which is necessary whenever we select robots based on a policy different from risk-neutral selection, is tedious, and potentially time-consuming if the distribution's domains are large, or there are many robots.

Thankfully, it turns out that we do not need to compute E_{SoR} directly. To compute *SwF*, we want the difference $E_{SoR}(R_i) - E_{SoR}(R_j)$ for the two robots R_i, R_j. It turns out that a corollary of Lemma 1 is that this difference is exactly the difference in expected costs of the two robots, which is much easier to compute:

COROLLARY 3. *From Lemma 1, it follows that the difference between the expected costs of any two robots in a given team of N robots $\{R_1, \ldots, R_N\}$ (each with a discrete probability distribution over possible costs $c_1, \ldots, c_k$) is equal to the distance between the expected SoR of the same robots.*

PROOF. Omitted for lack of space. □

4. PATH TRAVEL IN PRACTICE

We experimented with simulated and physical robots, to examine the travel time distributions in practice. We use the results to demonstrate in Section 4.1 that even under ideal conditions, robots do indeed have variance in the time that it takes them to travel a given path, and that this variance needs to be taken into account as described above. In Section 4.3 we show that the travel time distributions have distinctive shapes, and in general fit the Generalized Extreme Value family of distributions, and thus the distributions can be estimated in principle.

4.1 Experiments with Robots

We used our laboratory as the environment for the experiments. First, we used a popular open-source laser-based SLAM package, GMapping [6], to allow the robots to construct a map of the environment. The results of the exploration and mapping process were used as the basis for the experiments; this is to make sure that all path-planning and movements were carried out using a map with realistic quality. For path planning, we used A^* with a fixed 4-neighbor grid laid out over the map. If the robot discovered an unknown obstacle on the way, it tried to go around the obstacle until a timeout occurred, in this case a new path was planned from the current location to the goal, given the new information about the discovered obstacle.

Physical Robot Experiments. We utilized the RV-400 differential-drive robots (see Figure 2) for experiments in our lab. The RV-400 was equipped with a Hokuyu UTM-30LX laser, with nominal range of $30m$ (though in practice effective range was slightly smaller). The RV-400 robot has an approximate size 40×40 (width, length), and so this was used as the grid cell-size. We kept the environment static, with no obstacles or other changes to the environment that are unknown to the robot.

Figure 2: RV400 robot.

Figure 3 shows the environment used for the experiments, as mapped by the robot. We tested three paths: A short $6.4m$ path (2 to 3), with a narrow pass; an $8m$ path (1 to 2), through open space; and a $14.6m$ path which combined both (1 to 3). We measured the travel time in each of these paths 10 times.

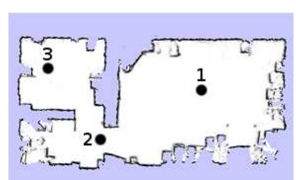
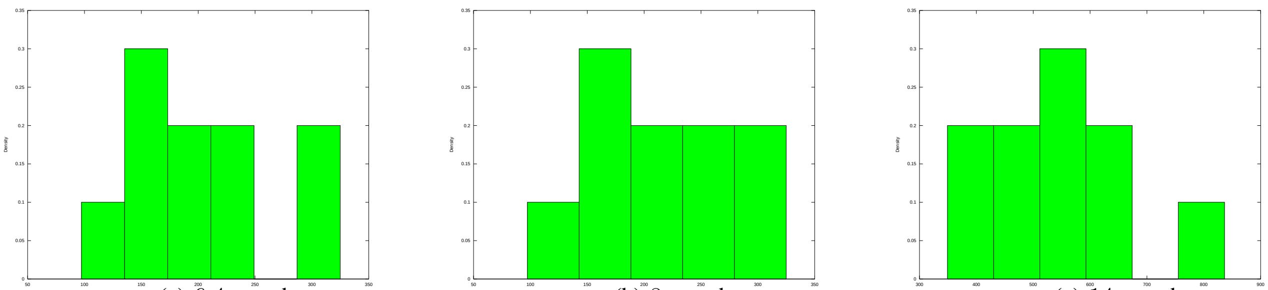


Figure 3: The mapped lab used in the robotics experiments.

Figures 4(a), 4(b), and 4(c) show the distribution, in histogram form, of travel time that were measured in these experiments, for the $6.4m$, $8m$, and $14m$ paths. For each of the settings, the planned path was identical, and the environment kept strictly static. The associated figure shows the path traversal time (X axis) versus its probability (Y Axis). Despite these ideal conditions, the robot took varying amount of time getting to the target locations. This variance is caused because of inaccuracies in the movement and sensing, which lead to actual execution of the path to differ between runs. In addition, changes to battery power also affect the robots linear and angular velocities. Indeed, Figure 4(b) does not include four data points that were removed from the data, because in their associated runs the robot operated with a faulty battery, and was almost twice as slow as in the other runs.

Simulation Experiments. We also conducted experiments in simulation, where we scaled up the number and complexity of the paths. We utilized



(a) $6.4m$ path. (b) $8m$ path. (c) $14m$ path.

Figure 4: RV-400 Travel Time Distributions.

the Webots 3D physics-based robotics simulator [11] to create the virtual world which the robots mapped and navigated as part of the experiments (see Fig. 5 for the resulting map). Webots has high fidelity, and models realistic sensor and motion errors, as we demonstrate below. In the simulation experiments, we simulated three RV-400 robots and their Hokuyu lasers. The openings between the rooms are doors which were open or close according to the evaluated criteria. Minor obstacles (boxes to be bypassed) are not shown. The doorway between the rooms is $1.2m$ wide.

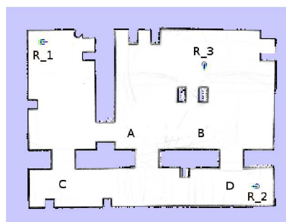


Figure 5: The mapped simulated environment.

The following configurations were used in the simulation experiments: From every robot location, to targets location A, B, C (9 combinations), and robots R_1, R_3 to target location D. we tested 4 obstacle settings: (i) static world (i.e., conforming to the map); (ii) with an unknown obstacle (a box placed on the planned path, that can be avoided and bypassed); (iii) an unexpected closed door blocking the original path (if the path was through an opening); and (iv) two unexpected closed doors blocking the original path, then a re-planned path. Each of the configuration (11 initial-target location pairs, 4 obstacle settings) was repeated 30 times. In all the experiments the robot had a path to the last target.

A small subset of the results from the simulation experiments are shown in Figures 6(a)–6(c). These are the results for one robot R_1, and for a single target point A (results for point B are shown later; other robots and points omitted for lack of space). Our intent is to demonstrate the variance that exists even under idealized simulated conditions.

Figure 6(a) shows the distribution of traversal times of R_1 for arriving at target A in a static world. As seen in the figure, even for a static world, and even under the relative noise-free world of simulation, there is variance in traversal time, due to motion and sensing uncertainties.

Of course, when choosing a robot for executing a task the world cannot typically be assumed to be static. These increase the variance in the actual travel times. Figure 6(b) shows the wider distribution of traversal times when an obstacle was added to the path of the robot, in 50% of 60 cases (the X axis scale is 80 to 350). This obstacle could be locally avoided (bypassed), and thus only a minor change was required to the pre-planned path. Note, that all the distribution of the static environment become a part of the first bin of the new distribution. Figure 6(c) shows the even wider distribution when we also take into account a door that was closed in a third of 90 cases, and which blocked the original path. This requires a new path to be planned and executed from the point where the closed door was discovered, to the target location.

The results above are similar to the distributions collected for the other experiment configurations, i.e., for other robots and other target locations. In all cases, even for paths that involve very few heading changes, and no obstacles or narrow passages, we see distributions that require reasoning about which robots to select, given the decision-maker's policy towards risk.

4.2 Selection Based on Experiment Data

We use the collected data to execute the decision-making policies described earlier, in both the simulated and physical world. To do this, we discretized the collected data into bins of approximately 25 seconds, and chose the robots according to the different decision policies.

Simulation Experiments. Robots $\{R_1, R_2, R_3\}$ compete on reaching targets A, B, C, and robots $\{R_1, R_3\}$ on target D. Table 4 shows the chosen robot using each of the decision policies.

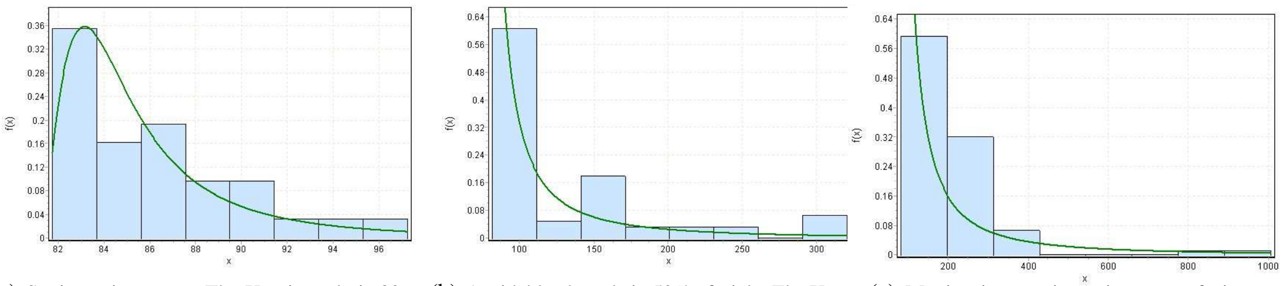
	$MinExp_C$ (risk-neutral)	$MinExpMax_C$ (risk-averse)	$MaxExpMin_C$ (risk-seeking)
A	R_3	R_3	R_1
B	R_3	R_3	R_2
C	R_2	R_2	R_2
D	R_1	R_3	R_3

Table 4: Selected robots for targets, according to each policy.

We find that indeed, the selected robot is not always the closest one to the target. For instance, R_3 is closest to point A. But when selecting a risk-seeking policy, R_1 is chosen. Likewise, R_3 is closest to point B, and yet R_2 is selected when a risk-seeking policy. R_3 is also closer to D, yet R_1 is selected in the risk-neutral policy. This is a direct result of the uncertainty inherent in the robots' movements.

We note that the selected robots for points {A, B, C} in the risk-averse $MinExpMax_C$ criteria were the same as the robots with the minimal expected cost. This is because the robots were in the same rooms with the target locations, and thus the closing and opening of doors–which would otherwise create large worst case travel times (and therefore large expected maximal times)–did not affect the ability of the robots to reach these targets.

Table 5 shows a case where an overruling of the selected robot is recommended by the *SwF* function. When selecting which of the



(a) Static environment. The X axis scale is 80 to 100.

(b) Avoidable obstacle in 50% of trials. The X axis scale is 80 to 350.

(c) Moving in a static environment, facing occasional avoidable obstacles, and sometimes needing to re-plan a path. The X axis scale is 80 to 1050.

Figure 6: R_1 Travel Time Distributions.

robots should reach target A through a risk-seeking policy, both robots R_1, R_2 have a minimal cost of 88. However, robot R_1 is chosen because its probability for this cost is a bit higher.

	p_{88}	E_{SoR}
R_1	0.438679	84.5507
R_2	0.433333	49.2834
R_3	0	29.0444

Table 5: The robots expected minimal cost, and expected *SoR* for the minimal cost of 88, while competing on point A.

But looking at the E_{SoR} of the robots, it is clear that R_2 has lower expected regret than R_1. Plugging these values into the *SwF* function yields the following:

$$\begin{aligned}(E_{SoR}(R_1) - E_{SoR}(R_2)) &= 84.5507 - 49.2834 && (1)\\ &= 35.2673 && (2)\\ &> 0 && (3)\\ &= 88 - 88 && (4)\\ &= \min_C(R_2) - \min_C(R_1) && (5)\end{aligned}$$

In this case, *SwF* returns 1, and we should consider selecting R_2 despite its slightly higher expected minimal traversal time.

Physical Robot Experiments. We utilized the RV-400 data in similar experiments. Abstracting away from the map, we used the distributions for traversal times of $6.4m$, $8m$ and $14.6m$ paths, for three robots: RV_1 positioned $6.4m$ away from a target point, RV_2 positioned $8m$ away from the same point, and RV_3 which is positioned $14.6m$ away. Table 6 shows the chosen robot in each of the decision policies.

$MinExp_C$ (risk-neutral)	$MinExpMax_C$ (risk-averse)	$MaxExpMin_C$ (risk-seeking)
RV_1	RV_2	RV_1

Table 6: Selected physical robot, according to each policy.

The results show that in the physical world as well, the closest robot is not always the robot to choose. Due to the narrow pass in the $6.4m$ path, the worst case travel time for RV_1 was worse (though less likely) than the worst case of RV_2 (which traveled $8m$ through open space).

4.3 Parametric travel time distributions

The experiments conducted reveal repeating characteristics of the emerging distributions, in particular their sharp lower bound and long tail. This is a result of having a clear lower bound on path traversal time (there's a limit as to how quickly a path can be traversed), and the increasingly rare (but still occurring) long arrival times, due to getting stuck by unforeseen obstacles, decreasing battery levels, etc. On such occasions, robots would re-plan their path several times on the way to the goal, and would sometimes need to traverse long distances to bypass a closed door.

We thus hypothesized that in fact known (parametrized) heavy-tailed continuous distributions may fit the data, allowing for improved prediction. We began experimentally, by fitting familiar distributions to the data, and using the Kolmogorov-Smirnov and Anderson-Darling fitness tests to determine the best-fitting distributions.

The fitness results for the best three distributions are shown in Table 7. The table shows the average matching functions, for all the paths that were followed, for the top three matching functions that were found.

	Gen. Log-Logistic	Gen. Extreme Value	Frechet $(3P)$
Kolmogorov-Smirnov	0.132	0.135	0.136
Anderson-Darling	1.051	1.512	0.69

Table 7: The average fitness of the top three matching distributions using Kolmogorov-Smirnov & Anderson-Darling tests.

The three best-fitting functions were found to be the General Log-Logistic (also called the 3-parameter Log-Logistic distribution), the General Extreme Value, a limit distribution of the maximum of a sequence of independent random variables which are identically distributed. and Frechet $(3P)$, a special case of the General Extreme Value distribution. The table shows that The General Log-Logistic distribution has the best average fitness using Kolmogorov-Smirnov test, and Frechet $(3P)$ has the best average fitness using Anderson-Darling test. Both of them, however, are strongly related (special cases) of the General Extreme Value distribution. Figures 6(a), 6(b), 6(c) show a curve which is the best-fit Log-Logistic continuous probability distribution fitting the simulation experiments data. In addition, Figure 7 shows the travel times distribution of robot R_1, traveling to point B in a static environment, 132 times. As seen is the figure, although the number of experiments grow the log-logistic distribution is a good fit.

We focused on the General Log-Logistic distribution. It has three parameters: shape, scale and shift. The shift parameter was found to be almost perfectly linearly correlated with the the minimal travel time of each one on the paths that were traveled. Figure 8 shows the relation between the minimal execution time of

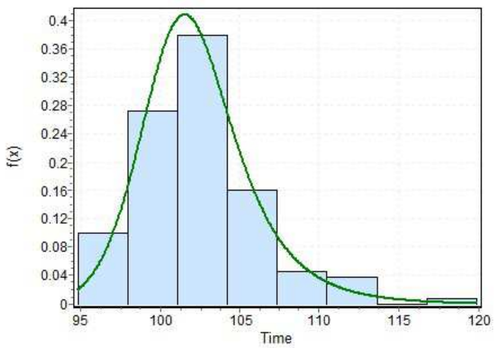


Figure 7: Distribution of R_1's travel times to point B in a static environment, over 132 path following experiments. The line shows the fitted log-logistic distribution.

all the paths that were traveled, and the shift parameter of the General Log-Logistic distribution that was fitted to the histogram of the path travel times. It is clear from the figure that there exist a direct relation between the two.

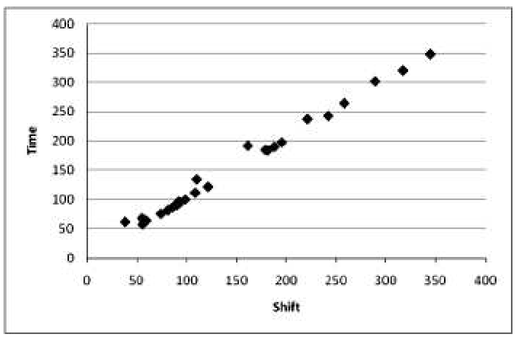


Figure 8: Measured minimal travel time versus fitted shift.

Looking on the other parameters, we found that the shape parameter was quite steady on values between 0.89304 to 3.3684 while its declaration is $(-\infty, \infty)$. We did not find a consistent value for the scale parameter.

5. DISCUSSION AND FUTURE WORK

The techniques developed in this paper allow robot selection that maintains bounds and guarantees as to travel times, even under uncertainty. We showed that even under static environment conditions, it takes the robots varying amount of time getting to a target location. Due to this variance, choosing a robot to preform a task cannot be done based on greedy selection (shortest path).

Thus, we introduced a decision making technique, inspired by economic decision theory, to distinguish between different policies based on risk. The experiments in simulated and physical robots demonstrated that different robots were chosen according to the different policies because of time travel variance: And indeed sometimes the closest robot is not the one to be selected, given the decision-making policy. Furthermore, we have shown that under some conditions, choosing the robot according to the selection policies will not always give a reasonable selection in practice. We defined the social regret function *SoR* which measure the cost of choosing specific robot over all other robot, and allow us to evaluate the gain from switching the chosen robot to a robot that will preform better. In our future work we plan to expand this techniques for allocating teams of N robots to K tasks.

While examining the experiments' data, we found that the data distributions had a good fit to the family of General Extreme Value distributions, and specifically to the General Log-Logistic distribution. We plan to explore this fit and learn to predict the parameters of the distributions for future real world paths and obstacles.

Acknowledgments. We thank the anonymous reviewers, and ISF grant #1357/07. As always, thanks to K. Ushi.

6. REFERENCES

[1] D. E. Bell. Regret in decision making under uncertainty. *Operations Research*, 30(5):961–981, 1982.

[2] J. Bobrow. Optimal robot path planning using the minimum-time criterion. *IEEE Journal of Robotics and Automation*, 4(4):443–450, 1988.

[3] A. Chaudhry. Path generation using matrix representations of previous robot state data. In *2006 45th IEEE Conference on Decision and Control*, pages 6790–6795, 2006.

[4] M. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.

[5] D. Foster and R. Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 29(1–2):7–35, 1999.

[6] G. Grisettiyz, C. Stachniss, and W. Burgard. Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *ICRA-05*, pages 2432–2437, 2005.

[7] K. Z. Haigh and M. M. Veloso. Planning, execution and learning in a robotic agent. In *Proceedings of the International Conference on Artificial Intelligence Planning Systems*, pages 120–127. AAAI Press, 1998.

[8] K. Heero, J. Willemson, A. Aabloo, and M. Kruusmaa. Robots find a better way: A learning method for mobile robot navigation in partially unknown environments, 2004.

[9] S. Koenig, X. Zheng, C. Tovey, R. Borie, P. Kilby, V. Markakis, and P. Keskinocak. Agent coordination with regret clearing. In *Proceedings of the Twenty-Third National Conference on Artificial Intelligence (AAAI-08)*, pages 101–107, 2008.

[10] G. Loomes and R. Sugden. Regret theory: An alternative theory of rational choice under uncertainty. *The Economic Journal*, 92(368):805–824, 1982.

[11] O. Michel. Webots™: Professional mobile robot simulation. *CoRR*, abs/cs/0412052, 2004.

[12] R. Simmons, D. Apfelbaum, W. Burgard, M. Fox, D. an Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *AAAI-00*, 2000.

[13] B. Sofman, E. Lin, J. A. Bagnell, J. Cole, N. Vandapel, and A. Stentz. Improving robot navigation through self-supervised online learning. *Journal of Field Robotics*, 23(11-12):1059–1075, 2006.

[14] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, et al. Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *The International Journal of Robotics Research*, 19(11):972—999, 2000.