Multi-Objective Ergodic Search for Dynamic Information Maps

Ananya Rao*[1], Abigail Breitfeld*[1], Alberto Candela[1], Benjamin Jensen[1],
David Wettergreen[1], and Howie Choset[1]

***Abstract*— Robotic explorers are essential tools for gathering information about regions that are inaccessible to humans. For applications like planetary exploration or search and rescue, robots use prior knowledge about the area to guide their search. Ergodic search methods find trajectories that effectively balance exploring unknown regions and exploiting prior information. In many search based problems, the robot must take into account multiple factors such as scientific information gain, risk, and energy, and update its belief about these dynamic objectives as they evolve over time. However, existing ergodic search methods either consider multiple static objectives or consider a single dynamic objective, but not multiple dynamic objectives. We address this gap in existing methods by presenting an algorithm called Dynamic Multi-Objective Ergodic Search (D-MO-ES) that efficiently plans an ergodic trajectory on multiple changing objectives. Our experiments show that our method requires up to nine times less compute time than a naïve approach with comparable coverage of each objective.**

I. INTRODUCTION

Robotics has made it possible to explore areas that are inaccessible or unsafe for humans. Robotic explorers can be applied to search and rescue [1]–[3], planetary exploration [4], [5], seafloor mapping [6], [7], agricultural monitoring [8], [9], and other scenarios in which a robot is tasked to collect information about a particular area.

However, robots used for these applications still rely heavily on human input. Mars rovers cover only a few kilometers over their lifetime in part because operators must manually create or verify plans [4]. Similarly, search and rescue robots often rely on humans to choose the most important areas to consider in their search. In these applications, communication with a robot can be costly or impossible. It is advantageous to operate without the need for human input.

Many autonomous motion planning algorithms must address the exploration vs. exploitation problem: should the robot search for new information (explore) or take advantage of what it already knows (exploit). Ergodic search methods find trajectories that effectively balance exploitation and exploration of a single objective by spending time in regions proportional to the expected information value there [10].

In many search applications, robots must consider multiple objectives simultaneously. In the case of planetary exploration, scientists and engineers seek to maximize the amount of information a rover retrieves while avoiding high-risk areas and limiting energy consumption.

Additionally, robots update their belief about the state of their environment as they explore. Many applications also involve multiple objectives that change as the robot explores and gains information. For example, a search and rescue robot can discover a particular area is void of survivors, and the geometry of the rescue site can change with time. An effective search algorithm should take into account how the robot's objectives change as it explores.

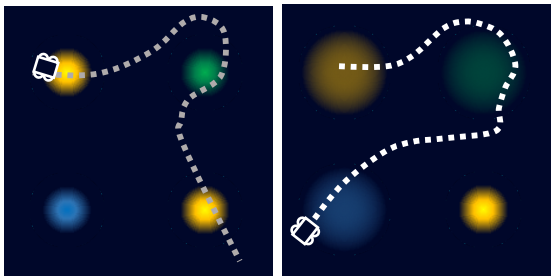

Fig. 1: Left: Map combining three dynamic objectives (blue, green, and yellow), and the original planned trajectory. Rover is in the starting location. Right: Final executed trajectory on the same objectives after re-planning based on change in maps. Rover has traversed the entire trajectory. Each Gaussian peak the robot visits becomes more diffuse, indicating there is less information to be gained in that region once the robot has taken samples there.

This paper describes an approach to autonomously plan a trajectory which effectively covers multiple dynamic (changing) objective maps simultaneously. We introduce a function that selects a new optimal solution after an objective map has changed. Using this function, we maintain coverage of the multiple objective maps, while requiring less compute time than a naïve approach. We demonstrate the efficacy of this approach both on synthetic data (example shown in Fig. 1) and on real data collected by a planetary rover analog.

II. BACKGROUND AND PRIOR WORK

A. Ergodic Planning

This work focuses on the coverage of objective maps, which represent a prior belief about the distribution of information over a spatial region. Methods for coverage of an objective map often concentrate on either exploration or exploitation. Spatial decomposition approaches prioritize exploration by spatially covering an entire region [11]–[13]. Methods based on information theory often focus on exploitation by using a greedy approach to choose the next immediate action [14], [15]. Ergodic search balances exploration with exploitation by designing trajectories that spend time in a region proportional to the expected value of some property in that region [10]. Ergodic search also produces smooth trajectories, as opposed to other methods like rapidly-exploring random tree (RRT) search.

*These authors contributed equally to this work
[1]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

The proportion of time a robot spends at a state $x \in \mathcal{X}$, where $\mathcal{X} \subset \mathbb{R}^{\mathrm{d}}$ is the d-dimensional search domain, is called the spatial time-average statistic of the trajectory (γ), and is defined as

$$C^t(x, \gamma(t)) = \frac{1}{t} \int_0^t \delta(x - \gamma(\tau)) d\tau, \tag{1}$$

where the Dirac delta function is denoted as δ. The time-averaged statistics of the trajectory should match the expected information density across the map. The difference between these two distributions is computed using the Fourier decomposition of each. The weighted sum of the difference between the two distributions' Fourier coefficients is called the ergodic metric, $\Phi(\cdot)$, and is defined as

$$\Phi(\gamma(t)) = \sum_{k=0}^{m} \lambda_k \left| c_k(\gamma(t)) - \xi_k \right|^2, \tag{2}$$

where m is the number of Fourier bases chosen, ξ_k are the Fourier coefficients of the spatial distribution of the information map, c_k are the Fourier coefficients of the time-averaged statistics of the trajectory, and λ_k are the weights assigned to each Fourier coefficient. For this work, $\lambda_k = \left(1 + \left\| \frac{k}{\pi} \right\|^2\right)^{-\frac{d+1}{2}}$ is defined to place higher weights on the lower frequency components, which correspond to larger spatial-scale variations in the information distribution.

Ergodic search proceeds by finding optimal controls $u^*(t)$

$$u^*(t) = \mathrm{argmin}_u \Phi(\gamma(t)), \tag{3}$$

for a robot whose dynamics are described by $\dot{x} = f(x(t), u(t))$. We find the solution to Eq. 3 using a direct optimization method. Additional optimization criteria are added to the ergodic metric to penalize control effort ($J_u(u(t))$) and penalize the trajectory leaving a bounded area ($J_b(x(t))$). Eq. 3 becomes $u^*(t) = \mathrm{argmin}_u \Phi(\gamma(t)) + J_u(u(t)) + J_b(x(t))$, with the extra costs defined as

$$J_u(u(t)) = \sum_{t=0}^{T} c_u u(t)^2, \text{ and} \tag{4}$$

$$J_b(x(t)) = c_b \sum_{t=0}^{T} \max(x(t) - L, 0)^2 + \min(x(t), 0)^2, \tag{5}$$

where c_u and c_b are the chosen weighting for the control effort and boundary cost, respectively, and the information map is defined from 0 to L_i in each of the spatial dimensions. In this work, we choose $c_u = 0.03$ and $c_b = 10$. Eqns. 4 and 5 are adapted from the Projection-based Trajectory Optimization (PTO) method of ergodic search [16].

B. Pareto Optimality

Pareto optimality refers to the concept of finding non-dominated solutions for multiple objectives. Non-dominated solutions are those for which there exists no other solution that can perform better in one objective without performing worse in another. The set of non-dominated solutions for the whole solution space is called the Pareto front (Fig. 2).

An analysis of ten methods for selecting an option from a Pareto-optimal set concluded that the technique for order of preference by similarity to ideal solution (TOPSIS) is one of the best performing methods [17]. This work uses TOPSIS as it is intuitive, efficient, and widely used as found in a survey of 266 papers [18]. The TOPSIS technique chooses a solution closest to the "positive-ideal solution" and farthest from the "negative-ideal solution" [19].

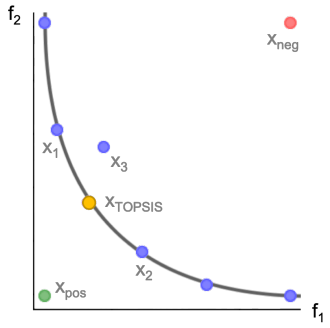

Fig. 2: Example of a set of Pareto optimal solutions in a two dimensional space for two objectives, f_1 and f_2. All blue points except x_3 lie on the Pareto front. The positive-ideal solution (x_{pos}), the negative-ideal solution (x_{neg}), and the solution chosen by TOPSIS (x_{TOPSIS}) are shown.

We find the positive-ideal solution and the negative-ideal solution by respectively combining the best and worst values of each objective within the set of solutions (Fig. 2). Say we have a set of k solutions to a problem with n objectives. For the i^{th} solution in this set, the distance to the positive-ideal is given by

$$S_{i+} = \sqrt{\sum_{j=1}^{n} (v_{ij} - v_j^+)^2}, \tag{6}$$

where v_j^+ is the value of the j^{th} objective in the positive-ideal solution, and v_{ij} is the value of the j^{th} objective in the i^{th} solution in the set. The distance to the negative-ideal (S_{i-}) is calculated in the same way, replacing v_j^+ with v_j^-, the value of the j^{th} objective for the negative-ideal solution. We then find the "closeness" of each solution to the optimal solution using

$$C_i = \frac{S_{i-}}{S_{i-} + S_{i+}}. \tag{7}$$

The solution with the largest C_i value is chosen. Fig.2 shows an example of a set of Pareto optimal solutions with the solution chosen by TOPSIS highlighted.

C. Planning for Dynamic Maps

Candela *et al.* present an approach to science-driven robotic exploration that is guided by dynamic maps that model the current state of knowledge of an area [6], [20]–[22]. The dynamic maps are generated by an adaptive model that consists of Gaussian Processes (GP) and Variational Autoencoders (VAE). The entropy derived from the probabilistic model represents the uncertainty of scientific information (spectra of terrain features) at any point in the map. It is favorable for a robot to sample locations of high uncertainty, in order to lower the total entropy of the map. Entropy is shown to be highly correlated with the error of the spectral reconstructions of terrain features.

Candela *et al.* investigate multiple planners for generating trajectories based on this dynamically updated entropy map,

including direct (Dijkstra's), random, greedy, and Monte-Carlo Tree Search (MCTS) planners [20]. These methods all plan from a start point to an end point, rather than exploring the map, but still allow for some exploration for scientific information gain. It was found that MCTS performs the best in terms of entropy reduction of the map [20].

Candela and Wettergreen consider the problem of reducing entropy while avoiding risky areas for the robot [23]. A risk model is incorporated into the planning methods in the form of local and budget constraints. Similarly, Hollinger and Sukhatme use an RRT-based approach for planning based on information gain while incorporating other objectives like fuel, energy, and time as constraints [24]. In this work, we seek to incorporate risk and power consumption not as constraints but as a true additional objectives.

Edelson investigates ergodic planning on the entropy map model proposed by Candela *et al.* [25] and concludes that an implementation of ergodic search called a PTO planner [16] exhibits better performance than MCTS, evaluated in terms of the final entropy of the map and the spectral reconstruction error. Edelson shows that a PTO type planner that updates the entropy map and replans after every step along the trajectory exhibits better performance than one that takes multiple samples before re-planning. Our work leverages these conclusions, and extends ergodic search to multiple objective maps.

D. Multi-Objective Trajectory Planning

Many methods exist for multi-objective trajectory planning. Genetic algorithms are popular, but require generating many full solutions, which can be time- and memory-intensive [26]–[33]. Other methods utilize Dijkstra's algorithm, A*, or D* search [34]–[41]. These methods are effective for applications which include trajectory length as a minimization objective and have a well-defined start location and end location, but are less applicable for problems where exploration of a region is important.

Ren *et al.* present the Multi-Objective Ergodic Search (MO-ES) method which finds a set of Pareto-optimal trajectories given multiple information maps [42]. This Pareto-optimal set is generated by employing single-objective ergodic search on combinations of the multiple objective maps. Each Pareto-optimal trajectory has a "weight vector" associated with it, which describes how the information maps were weighted and added together to form a "scalarized" objective map that was used to plan this trajectory. If we have n objective maps c_n and a (unit length) weight vector $\mathbf{\Lambda} = [\lambda_1, \lambda_2, ..., \lambda_n]$, the scalarized map can be described as $\mathbf{C} = \lambda_1 c_1 + \lambda_2 c_2 + ... + \lambda_n c_n$. MO-ES uses adaptive sampling of weight vectors based on the similarity between the multiple information maps in order to approximate the Pareto-optimal front of trajectories. This means fewer weight vectors must be sampled in order to approximate the Pareto front when multiple maps are similar. Ren shows that this method produces better quality solutions than a naïve method and multi-objective genetic algorithms.

MO-ES assumes that all the objective maps are static, and requires a human to choose from the generated Pareto-optimal set of solutions. For applications such as exploration with a planetary rover, the robot should be able to plan with dynamic objective maps that are updated as time progresses or as more regions are explored. The rover should also be able to autonomously choose a solution from the Pareto front in order to minimize costly communication. This work extends MO-ES to take into account dynamic objectives and autonomously select a solution from a Pareto set.

III. D-MO-ES: DYNAMIC MULTI-OBJECTIVE ERGODIC SEARCH

The following section describes our approach to solving the problem of multi-objective ergodic search for dynamic information maps. The method involves three steps.

First, we employ the MO-ES method to generate a set of Pareto optimal solutions based on the initial information maps. Each solution corresponds to a weight vector which describes the single scalarized information map that was used to plan this solution's trajectory. Second, we use the TOPSIS method to choose one of the Pareto optimal solutions based on their proximity to the positive-ideal solution. We then take a single step along the chosen trajectory and update the information maps according to the new robot location. Third, we use a weight update function, pre-computed to approximate the relationship between weight vector and change in dynamic objective, to determine the next weight vector that we should choose. The process of sampling the next point in the trajectory, updating the map, and updating the weight vector is repeated until the desired trajectory length is reached.

A. Objectives Weight Vector Calculation

The naïve approach to using MO-ES for dynamic objective maps would be to allow MO-ES to compute trajectories for a set of weight vectors every time a map is updated, and then use TOPSIS to choose the best weighting scheme from that set. However, this is time-consuming as a trajectory must be planned for each weight vector at every step.

In order to efficiently plan on dynamic objective maps, our method to calculate a new chosen weight vector is based on the change in the objective map (Alg. 1). Empirically we see that if one objective is changing, there is relationship between the weight vector used before the map changed, the change in the dynamic objective map, and the new weight vector chosen. This relationship can be locally approximated by a plane, where the independent variables (x and y) are the old weight and the ergodic difference between the old map and the new map, and the dependent variable (z) is the new weight. By fitting a plane we can approximate the new weight vector at each planning step. The plane equation is computed by running MO-ES on a small set of states of the dynamic objective maps, and finding the relationship between how the map changes and the new Pareto-optimal weight chosen. The surface representing the relationship between the weights may not be planar across the entire space, however, we observe that such a surface can be *locally* approximated as a plane (a small change in one of the objective maps leads to a small change in the Pareto front of weight vectors). We show

Algorithm 1 Dynamic Weight Update Function Calculation

$n_{pts} \leftarrow$ number of data points
$M_d \leftarrow$ set of n_{pts} states of the dynamic objective map
$M_s \leftarrow$ set of static objective maps
$\Omega_c \leftarrow []$ ▷ list of weight vectors
$\varepsilon \leftarrow []$ ▷ list of ergodic differences between objectives
$P \leftarrow []$ ▷ list of data points being evaluated
for i in range(0,n_{pts}) **do**
    $\Omega_i \leftarrow$ MO-ES($M_{d,i}$, M_s) ▷ Set of weight vectors
    $\Omega_{c,i} \leftarrow$ TOPSIS(Ω_i) ▷ Chosen weight vector
end for
for i in range(1, n_{pts}) **do**
    $\varepsilon_i \leftarrow$ ErgodicDifference($M_{d,i-1}$, $M_{d,i}$)
    $P_i \leftarrow$ ($\Omega_{c,i-1}$, ε_i, $\Omega_{c,i}$)
end for
plane $\leftarrow$ BestFit(P) ▷ Fit the solution plane

that this method performs well empirically, and leave proving theoretical bounds and exploring other approaches to future work. Once we have the resulting plane equation for the dynamic objective map, the weight vector is updated in two steps: the weight for the dynamic objective map is updated using the plane equation, and then the static objective weights are updated in proportion to their original values (Alg. 2).

In order to calculate this plane equation, we need a few examples of how each dynamic objective map can change. For example in our experiments, we use satellite data to simulate *in-situ* rover measurements, so we can predict how a map of scientific information could change. In some cases, there may not be sufficient information about the environment to simulate how a dynamic objective map changes.

B. Dynamic Multi-Objective Ergodic Search

The dynamic multi-objective ergodic search (D-MO-ES) algorithm leverages the plane equation and weight update function described in the previous section to plan trajectories over multiple dynamic objective maps. D-MO-ES iteratively updates the dynamic objective maps, calculates the new weight vector, and calculates the ergodic trajectory on the new scalarized objective map (Alg. 3).

IV. EXPERIMENTS

D-MO-ES is evaluated on two different kinds of data: synthetic Gaussian information distributions (representing information priors in general search and coverage tasks), and objective maps from real data collected during field tests with a planetary rover analog in Cuprite, NV. This section details how the objective maps are generated and updated.

A. Synthetic Data

The objective maps used for our synthetic data experiments are generated by placing Gaussian peaks at different locations to represent the uncertainty of the corresponding region, with higher values corresponding to higher uncertainty of information. When a sample is taken along a trajectory, the uncertainty of the area around the sample is reduced. For our dynamic synthetic objective maps, we simulate a reduction

Algorithm 2 Weight Vector Update Function

$\omega_{old} \leftarrow$ last chosen weight vector
$\varepsilon \leftarrow$ ergodic difference in the dynamic objective map
$d \leftarrow$ index of the dynamic objective map
$I_s \leftarrow$ set of indices of the other objective maps
plane $\leftarrow$ function of the plane ▷ from Alg. 1
$\omega_{new} \leftarrow \omega_{old}$
$\omega_{new}[d] \leftarrow$ plane($\omega_{old}[d]$, ε)
$\Delta_\omega \leftarrow \omega_{new}[i] - \omega_{old}[i]$
$\omega_{sum} \leftarrow \sum_{i \in I_s} \omega_{old}[i]$
for i in I_s **do**
    $\omega_{new}[i] \leftarrow \omega_{old}[i] + \frac{\omega_{old}[i]}{\omega_{sum}} \Delta_\omega$
end for

Algorithm 3 Dynamic Multi-Objective Ergodic Search

$M_d \leftarrow$ set of initial states of dynamic maps
$M_s \leftarrow$ set of static maps
$f_m \leftarrow$ dynamic maps update function ▷ Describes how each objective map changes with measurements or time
$\Omega_0 \leftarrow$ MO-ES(M_d, M_s) ▷ Initial set of weight vectors
$\omega \leftarrow$ TOPSIS(Ω_0) ▷ Initial chosen weight vector
$t \leftarrow$ trajectory length ▷ Steps in the final trajectory
while $t \geq 0$ **do**
    T $\leftarrow$ ErgodicTrajectory(ω, M_d, M_s) ▷ Find the ergodic trajectory on the combined objective map calculated using the chosen weight vector
    Execute T[0] ▷ Execute first step of new trajectory
    $M_{d,old} \leftarrow M_d$
    $M_d \leftarrow f_m(M_{d,old})$
    $\varepsilon \leftarrow$ ErgodicDifference(M_d, $M_{d,old}$)
    $\omega_{old} \leftarrow \omega$
    $\omega \leftarrow$ WeightVectorUpdate($\omega_{old}, \varepsilon$)
    $t \leftarrow t - 1$
end while

in uncertainty by reducing the heights of the Gaussian peaks within a specified distance of the sample location, to indicate that there is less information to be gained in that location. Since the goal of the planned trajectories is to reduce the overall uncertainty of the area, we evaluate performance of trajectories on these synthetic data maps using reduction in total uncertainty.

For static synthetic objective maps, we similarly use Gaussian peaks to represent areas of high information density, but do not update these peaks as the robot moves. We evaluate the coverage of these maps using the ergodic metric described in Section II-A. A lower ergodic metric signals better coverage of the map.

B. Real-World Data

1) Entropy Map: The first objective in our planetary rover analog experiments is to reduce uncertainty in scientific information. We use the entropy map formulation proposed by Candela *et al.*, where "entropy" of each point in the map is related to the variance of a Gaussian process which predicts measurements at each point [20]. We use low-resolution Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) satellite data as the prior and use high-

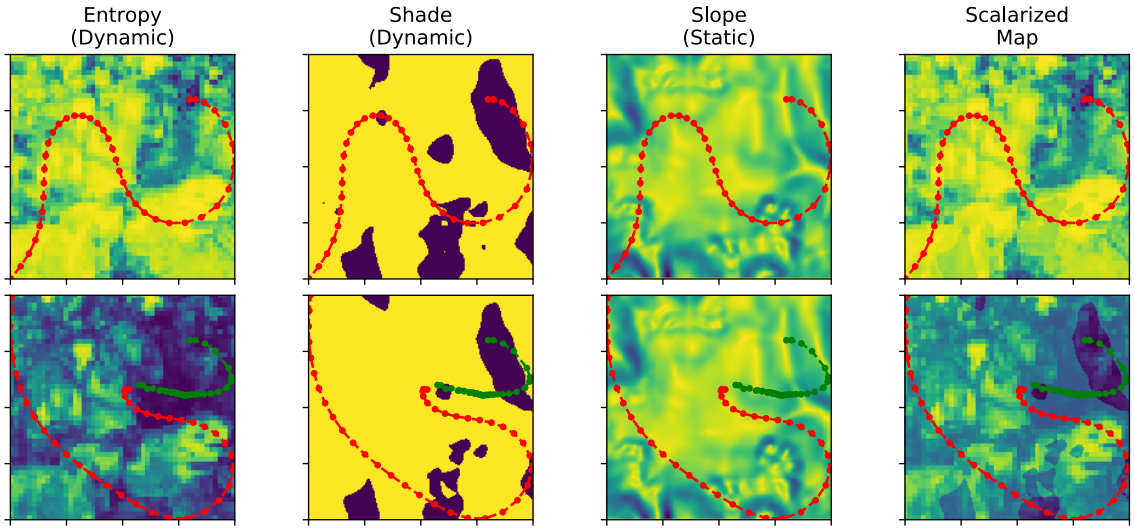

Fig. 3: First row: Initial set of objective maps and their scalarized combination. The planned trajectory based on the scalarized map is shown in red. Second row: Updated set of objective maps and their new scalarized combination after the rover has taken steps along the trajectory. Planned trajectory is shown in red and previously traversed trajectory is shown in green.

resolution Airborne Visible Near Infrared Spectrometer - New Generation (AVIRIS-NG) data as a proxy for *in-situ* samples [43]–[45]. In order to focus the ergodic search on areas of high entropy, we threshold the entropy maps (setting areas of high entropy above 75% of the maximum value to 1, and areas of low entropy below 75% of the maximum value to 0). We find experimentally that thresholding the entropy maps results in a greater reduction of entropy when using an ergodic method. The goal of planning based on the entropy model is to reduce the total uncertainty of the area. Thus, we evaluate performance with the reduction in total entropy. As per the results in Edelson's work, we update the entropy model after every step along the trajectory [25].

2) Shade Map: Our second objective is to avoid shadows. Our rover is solar powered, so in order to increase power generation, the rover should prefer visiting sunlit areas. Using a digital elevation model (DEM) of the field site, we use raycasting based on the angle of the sun to generate a map of shaded regions. Shadows have a low value while sunlit regions have a high value, which encourages the rover to stay in sunny areas. To evaluate performance on the shade maps, we calculate the percentage of the trajectory which falls within shaded regions. We want this value to be low. The shade maps are updated every 4 steps along the trajectory, which corresponds to a sun angle change of one degree.

3) Slope Map: Our third objective is to limit the slope the rover traverses. The slope of the terrain in our test site acts as a proxy for risk, so the rover avoids high sloped areas and prefers driving over low slopes. To generate this risk map, we use a Sobel image filter on a DEM of the region. We opt to use slope as an estimate of risk for simplicity and because of the limited information available for slip characterization [20]. To evaluate performance, we sum all the slope values at every point on the map. We then calculate the slope at every point along a particular trajectory, and calculate the percentage of slope the trajectory visits. A lower slope percentage value is favorable.

C. Experiment Scenarios

For each data set (synthetic and real-world) we test our approach on two different scenarios. In the first, we plan on two objective maps, one static and one dynamic. In the second scenario, we use three objective maps, where two of them are dynamic and one of them is static. Fig. 3 shows an example of our approach on real-world data, with two dynamic objective maps and one static map. We run each scenario for 40-50 trials using a different starting point in the map for each trial. We chose these scenarios to demonstrate our method's ability to operate on a mix of different numbers of static and dynamic maps, though our method will function without any static objectives. For each of the scenarios, we compare two different methods and plan a trajectory with a total length of 10, 15, and 30 steps. At each planning step, the ergodic search algorithm has a lookahead of 30 steps.

1) MO-ES + TOPSIS: The first approach uses the MO-ES adaptive scalarization method to choose from a set of possible weight vectors at each step along the trajectory. MO-ES plans a candidate trajectory for each weight vector, and then chooses one of these trajectories using the TOPSIS method described in Section II-B. This is our naïve approach.

2) D-MO-ES: The second method is our proposed approach, described in Section III.

V. RESULTS AND DISCUSSION

We empirically show that D-MO-ES results in comparable performance on multiple dynamic objectives to MO-ES + TOPSIS with significantly less compute time.

Tables I and II detail the performance of D-MO-ES versus MO-ES + TOPSIS on experiments using synthetic data and data from Cuprite, NV respectively. For trajectory lengths of 10 and 15 steps, we observe comparable performance between the two methods in both two and three objective scenarios for both synthetic and real-world data. D-MO-ES slightly outperforms MO-ES + TOPSIS in some cases, as in the case of a trajectory length of 10 steps for two

Two Objectives (Dynamic Objective 1 and Static Objective 2)

	Percent Reduction of Objective 1			Ergodic Metric of Objective 2			Ergodic Metric of Objective 3			Average Compute Time Per Step (s)		
Steps in Trajectory	**10**	**15**	**30**	**10**	**15**	**30**	**10**	**15**	**30**	**10**	**15**	**30**
MO-ES + TOPSIS	5.88	7.15	14.97	8.98	7.43	5.03	–	–	–	31.72	31.71	31.56
D-MO-ES	5.15	5.15	12.09	9.65	8.47	7.04	–	–	–	10.21	10.38	10.41
p-value	0.53	0.53	0.52	1.7e-3	2.1e-5	2.2e-6	–	–	–	**average speedup: 3.1x**		

Three Objectives (Dynamic Objective 1, Dynamic Objective 2, Static Objective 3)

	Percent Reduction of Objective 1			Percent Reduction of Objective 2			Ergodic Metric of Objective 3			Average Compute Time Per Step (s)		
Steps in Trajectory	**10**	**15**	**30**	**10**	**15**	**30**	**10**	**15**	**30**	**10**	**15**	**30**
MO-ES + TOPSIS	11.02	14.30	28.62	7.35	9.56	15.04	9.07	7.84	5.51	83.20	83.46	87.09
D-MO-ES	7.35	8.82	17.97	5.88	13.96	19.53	9.52	8.28	6.59	9.03	9.17	9.18
p-value	0.05	0.03	0.01	0.93	0.02	0.06	0.07	0.09	0.01	**average speedup: 9.3x**		

TABLE I: Comparative evaluation with synthetic data.

Two Objectives (Dynamic Entropy and Static Shade)

	Entropy Reduction			Average % of Trajectory in Shade			Average % of Slope Visited			Average Compute Time Per Step (s)		
Steps in Trajectory	**10**	**15**	**30**	**10**	**15**	**30**	**10**	**15**	**30**	**10**	**15**	**30**
MO-ES + TOPSIS	4.76	5.75	7.43	24.40	23.07	19.07	–	–	–	23.24	26.41	32.32
D-MO-ES	4.31	5.27	6.45	23.00	19.87	15.07	–	–	–	8.05	8.28	8.51
p-value	0.01	0.01	7.4e-4	0.60	0.20	0.05	–	–	–	**average speedup: 3.3x**		

Three Objectives (Dynamic Entropy, Dynamic Shade, Static Slope)

	Entropy Reduction			Average % of Trajectory in Shade			Average % of Slope Visited			Average Compute Time Per Step (s)		
Steps in Trajectory	**10**	**15**	**30**	**10**	**15**	**30**	**10**	**15**	**30**	**10**	**15**	**30**
MO-ES + TOPSIS	4.38	5.74	8.12	1.25	1.00	0.92	0.021	0.032	0.068	41.04	49.85	70.72
D-MO-ES	4.39	5.52	6.96	2.75	0.33	0.17	0.022	0.033	0.064	7.33	7.42	7.59
p-value	0.70	0.48	1.0e-3	0.03	0.10	0.01	0.05	0.40	0.71	**average speedup: 7.2x**		

TABLE II: Comparative evaluation with real-world data from Cuprite, NV.

objectives on data from Cuprite, NV. In other cases, MO-ES + TOPSIS slightly outperforms D-MO-ES, as in the case of a trajectory length of 15 steps for two synthetic objective maps. The coverage performance of both methods is generally equivalent, with neither method vastly outperforming the other in every scenario. We calculate the p-values between MO-ES + TOPSIS and D-MO-ES using the double sided Wilcoxon signed-rank test [46], and find that there is no statistically significant difference ($p \geq 0.05$) between the two methods for all scenarios with a few exceptions where either MO-ES + TOPSIS or D-MO-ES performs better. A higher p-value means we accept the null hypothesis, which in this case is that using our method and using the naïve method results in a similar outcome on a particular objective.

Further, we see that the performance of D-MO-ES generally decreases for trajectories of length 30 steps. This follows from our formulation of D-MO-ES. The weight update function is estimated by using a few examples of how each objective map will change. Thus, we expect that as the objective map continues to evolve the estimate will no longer be an accurate approximation for updating the weights.

The strong advantage of D-MO-ES is its faster compute time as compared to the naïve MO-ES + TOPSIS method. In every scenario tested, we see a 3-9 times speedup in D-MO-ES versus MO-ES + TOPSIS. MO-ES + TOPSIS requires planning multiple trajectories at each step and then choosing the best one, while D-MO-ES is able to predict the optimal solution, and only plan one trajectory. Thus, D-MO-ES shows faster performance while maintaining good coverage on each of the objective maps.

VI. CONCLUSIONS

In this work we present an approach that enables autonomous multi-objective ergodic planning with dynamic objective maps. We present two novel ideas: 1) a function that predicts the new optimal multi-objective weight vector given the amount of change in an objective map, and 2) a method that utilizes this function to efficiently plan trajectories on multiple dynamically updating objective maps. We incorporate the TOPSIS method of choosing a solution from a set of Pareto optimal solutions in order to autonomously plan trajectories on updating maps without human input. Our approach reduces the required computation time as compared to a naïve method without sacrificing the coverage performance of the chosen trajectory. We experimentally verify that this approach can be applied to both synthetic Gaussian maps, and complex real-world data modelling a planetary rover scenario.

Numerical results indicate that the approximations used in our approach perform best with small changes in objective maps. Future work will investigate at what point our function approximation becomes invalid, and will explore more accurate models of the relationship between change in objectives and the weight vector. Further, we will demonstrate this method on different applications and real robotic systems.

References

[1] S. Grogan, R. Pellerin, and M. Gamache, "The use of unmanned aerial vehicles and drones in search and rescue operations–a survey," *Proceedings of the PROLOG*, 2018.

[2] D. S. Drew, "Multi-agent systems for search and rescue applications," *Current Robotics Reports*, vol. 2, no. 2, pp. 189–200, 2021.

[3] B. Shah and H. Choset, "Survey on urban search and rescue robots," *Journal of the Robotics Society of Japan*, vol. 22, no. 5, pp. 582–586, 2004.

[4] A. Rankin, M. Maimone, J. Biesiadecki, N. Patel, D. Levine, and O. Toupet, "Driving curiosity: Mars rover mobility trends during the first seven years," in *2020 IEEE Aerospace Conference*, 2020, pp. 1–19.

[5] J. Balaram and M. Golombek, "The ingenuity helicopter on the perseverance rover," *Space Science Reviews*, vol. 217, Jun. 2021.

[6] A. Candela, K. Edelson, M. Gierach, D. Thompson, G. Woodward, and D. Wettergreen, "Using remote sensing and in situ measurements for efficient mapping and optimal sampling of coral reefs," *Frontiers in Marine Science*, vol. 8, Sep. 2021.

[7] N. Cruz, N. Abreu, J. Almeida, *et al.*, "Cooperative deep water seafloor mapping with heterogeneous robotic platforms," in *OCEANS 2017-Anchorage*, IEEE, 2017, pp. 1–7.

[8] R. R Shamshiri, C. Weltzien, I. A. Hameed, *et al.*, "Research and development in agricultural robotics: A perspective of digital farming," 2018.

[9] S. Fountas, N. Mylonas, I. Malounas, E. Rodias, C. Hellmann Santos, and E. Pekkeriet, "Agricultural robotics for field operations," *Sensors*, vol. 20, no. 9, p. 2672, 2020.

[10] G. Mathew and I. Mezić, "Metrics for ergodicity and design of ergodic dynamics for multi-agent systems," *Physica D: Nonlinear Phenomena*, vol. 240, no. 4, pp. 432–442, 2011, ISSN: 0167-2789.

[11] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, 2002.

[12] M. Santos, Y. Diaz-Mercado, and M. Egerstedt, "Coverage control for multirobot teams with heterogeneous sensing capabilities," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 919–925, 2018.

[13] M. Schwager, D. Rus, and J.-J. Slotine, "Decentralized, adaptive coverage control for networked robots," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009.

[14] W. Chen and L. Liu, "Pareto monte carlo tree search for multi-objective informative planning," in *Robotics: Science and Systems XV*, Robotics: Science and Systems Foundation, Jun. 2019.

[15] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1134–1154, 2012.

[16] L. M. Miller and T. D. Murphey, "Trajectory optimization for continuous ergodic exploration," in *2013 American Control Conference*, 2013, pp. 4196–4201.

[17] Z. Wang and G. Rangaiah, "Application and analysis of methods for selecting an optimal solution from the pareto-optimal front obtained by multi-objective optimization," *Industrial & Engineering Chemistry Research*, vol. 56, Dec. 2016.

[18] M. Behzadian, S. Khanmohammadi Otaghsara, M. Yazdani, and J. Ignatius, "A state-of the-art survey of topsis applications," *Expert Systems with Applications*, vol. 39, no. 17, pp. 13 051–13 069, 2012, ISSN: 0957-4174.

[19] C. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications : a State-of-the-art Survey* (Lecture notes in economics and mathematical systems). Springer-Verlag, 1981, ISBN: 9783540105589.

[20] A. Candela Garza, "Bayesian models for science-driven robotic exploration," Ph.D. dissertation, 2021.

[21] A. Candela, S. Kodgule, K. Edelson, *et al.*, "Planetary rover exploration combining remote and in situ measurements for active spectroscopic mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 5986–5993.

[22] A. Candela, K. Edelson, and D. Wettergreen, "Mars rover exploration combining remote and in situ measurements for wide-area mapping," in *i-SAIRAS 2020*, Oct. 2020.

[23] A. Candela and D. Wettergreen, "An approach to science and risk-aware planetary rover exploration," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9691–9698, 2022.

[24] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014.

[25] K. Edelson, "Ergodic trajectory optimization for information gathering," M.S. thesis, Oct. 2020.

[26] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.

[27] C. Fonseca and P. Fleming, "Genetic algorithms for multiobjective optimization: Formulation discussion and generalization," *the fifth Intl conference on Genetic Algorithms*, vol. 93, Feb. 1999.

[28] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006, Special Issue - Genetic Algorithms and Reliability, ISSN: 0951-8320.

[29] D. Cvetkovic and I. Parmee, "Preferences and their application in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 42–57, 2002.

[30] H. Jun and Z. Qingbao, "Multi-objective mobile robot path planning based on improved genetic algorithm," in *2010 International Conference on Intelligent Computation Technology and Automation*, vol. 2, 2010, pp. 752–756.

[31] S. Geetha, G. M. Chitra, and V. Jayalakshmi, "Multi objective mobile robot path planning based on hybrid algorithm," in *2011 3rd International Conference on Electronics Computer Technology*, vol. 6, 2011, pp. 251–255.

[32] B. Moradi, "Multi-objective mobile robot path planning problem through learnable evolution model," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 31, no. 2, pp. 325–348, 2019.

[33] C. Yang, T. Zhang, X. Pan, and M. Hu, "Multi-objective mobile robot path planning algorithm based on adaptive genetic algorithm," in *2019 Chinese Control Conference (CCC)*, 2019, pp. 4460–4466.

[34] A. Lavin, "A pareto front-based multiobjective path planning algorithm," *CoRR*, vol. abs/1505.05947, 2015.

[35] S. Gautam, B. S. Roy, A. Candela, and D. S. Wettergreen, "Science-aware exploration using entropy-based planning," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3819–3825, 2017.

[36] A. Lavin, "A pareto optimal d* search algorithm for multiobjective path planning," *CoRR*, vol. abs/1511.00787, 2015.

[37] K. Jeddisaravi, R. J. Alitappeh, and F. G. Guimarães, "Multi-objective mobile robot path planning based on a* search," in *2016 6th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2016, pp. 7–12.

[38] S. Aine, S. Swaminathan, V. Narayanan, V. Hwang, and M. Likhachev, "Multi-heuristic a*," *International Journal of Robotics Research: Special Issue on RSS '14*, vol. 35, no. 1, pp. 224–243, Jan. 2016.

[39] F. Guo, H. Wang, and Y. Tian, "Multi-objective path planning for unrestricted mobile," in *2009 IEEE International Conference on Automation and Logistics*, 2009, pp. 1046–1051.

[40] L. Mandow and J.-L. Pérez-de-la-Cruz, "A new approach to multiobjective a* search," in *IJCAI*, 2005.

[41] B. S. Stewart and C. C. White, "Multiobjective a*," *J. ACM*, vol. 38, no. 4, pp. 775–814, Oct. 1991, ISSN: 0004-5411.

[42] Z. Ren, A. K. Srinivasan, H. Coffin, I. Abraham, and H. Choset, "A local optimization framework for multi-objective ergodic search," in *Robotics: Science and Systems XVIII*, Robotics: Science and Systems Foundation, Jun. 2022.

[43] H. Fujisada, F. Sakuma, A. Ono, and M. Kudoh, "Design and preflight performance of aster instrument protoflight model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 4, pp. 1152–1160, 1998.

[44] R. O. Green, M. L. Eastwood, C. M. Sarture, *et al.*, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (aviris)," *Remote Sensing of Environment*, vol. 65, no. 3, pp. 227–248, 1998, ISSN: 0034-4257.

[45] L. Hamlin, R. O. Green, P. Mouroulis, *et al.*, "Imaging spectrometer science measurements for terrestrial ecology: Aviris and new developments," in *2011 Aerospace Conference*, 2011, pp. 1–7.

[46] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945, ISSN: 00994987.