

# Current-Sensitive Path Planning for an Underactuated Free-floating Ocean Sensorweb

Kristen P. Dahl, David R. Thompson, David McLaren, Yi Chao, Steve Chien

**Abstract**— This work investigates multiagent path planning in strong, dynamic currents using thousands of highly underactuated vehicles. We address the specific task of path planning for a global network of ocean-observing floats. These submersibles are typified by the Argo global network consisting of over 3000 sensor platforms. They can control their buoyancy to float at depth for data collection or rise to the surface for satellite communications. Currently, floats drift at a constant depth regardless of the local currents. However, accurate current forecasts have become available which present the possibility of intentionally controlling floats' motion by dynamically commanding them to linger at different depths. This project explores the use of these current predictions to direct float networks to some desired final formation or position. It presents multiple algorithms for such path optimization and demonstrates their advantage over the standard approach of constant-depth drifting.

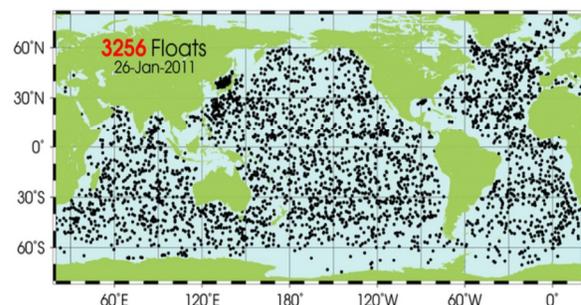
## I. INTRODUCTION

**I**N 1999, the Argo Program began an initiative to place 3,000 free-floating submarine sensors across the world's oceans as part of an integrated global observation strategy [1]. Over 50 research agencies from 18 countries contributed to what is now an array of 3,250 floats. These sensors take temperature and salinity measurements from the top 2000 meters of the ice-free oceans. They collect data during 10 day cycles; at the end of each cycle, they perform a six-hour rise to the surface and upload this data to the satellite Jason. The data collected is immediately uploaded to an open-access database under the control of project GODAE (Global Ocean Data Assimilation Experiment) [1]. An external bladder allows the floats to move to different depths by changing density, but lateral movement is not propelled by the float itself. Instead, strong ocean currents push the sensors along with them.

Argo's coverage goal was to place one float every 3 degrees in latitude and longitude. Typically, research ships deploy the sensors, but sometimes merchant ships are used to extend drop-off areas. The floats are built to last for approximately 150 10-day sampling cycles, and the entire array of 3,000 floats requires over 800 deployments per year to maintain. Aircraft or charter ship deployments are often

necessary to get coverage in the southern-most areas. Sensor coverage is a significant ongoing problem since currents can cause floats to gather in groups or miss areas of the ocean. Proposals for improving coverage involve augmenting the existing network with additional floats. Maintaining this network, and replacing floats after they exhaust their battery power, is a significant ongoing expense.

This work investigated path-planning algorithms to allow the floats to exploit strong, time-varying ocean currents. We used ocean current forecasts to design float mission plans in the form of depth profiles (a list of desired depth at each moment in time). In the proper current field, these plans direct floats purposefully towards a desired configuration and improve coverage. One possible objective is *dispersion*, in which floats spread out from each other so that multiple floats can be deployed in the same location by a single airship and expand to cover a broad area. Alternatively, mission planners could move a float to a certain specific location, facilitating pick-up for repairs or replacement. This can be used whenever a large number of floats need to be retrieved; operators can direct the network to reunite in a central area to facilitate the retrieval process. Exercising purposeful control over float locations can prove useful in maintaining the Argo array coverage and may potentially reduce the cost of sustaining the network.



**Fig 1.** shows the configuration of Argo floats as of January, 2011. Some areas, such as the Sea of Japan, are congested by floats, and others, particularly near the poles, are unmonitored [2].

K. P. Dahl is with the California Institute of Technology, 1200 East California Blvd, Pasadena, CA 91126 USA. kdahl@caltech.edu.

The other authors are with the Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr., Pasadena, CA 91109.

A series of simulations using predicted ocean currents

from the HYCOM (HYbrid Coordinate Ocean Model) database [4] were run to perform a direct comparison between existing strategies and current-sensitive mission plans.

We modeled float trajectories with records of ocean currents from August 2010, available from GDACs (Global Data Assembly Centers) [4]. We designed paths to achieve two major objectives: (1) to maximize coverage area, and (2) to direct floats to a target endpoint. We tested several path planning algorithms and cost functions designed to optimize these goals. These computed command sequences that would instruct the floats to linger at various depths and durations until the next communications opportunity. We then simulated float trajectories based on the traditional Argo control policy (maintaining a constant, fixed depth) and the new adaptive approach. This indicated how the network configuration was expected to evolve under each of the control strategies. Through these simulations, we directly compared the ability of current-sensitive approaches to achieve Argo mission objectives. These simulations suggest that current-sensitive control strategies can be more effective than the current Argo method for achieving these objectives.

## II. CURRENT PREDICTIONS AND MOTION MODEL

Many projects have explored path planning for autonomous vehicles. However, ocean travel introduces specific challenges. In contrast to path planning on land, the ocean requires a 3D space; depth adds an extra controllable dimension to position [8]. Our model depicted this 3-dimensional space in latitude, longitude, and depth, and tracked the position of gliders as they were propelled by strong ocean currents.

Simulations used ocean current data made available by HYCOM, a part of GODAE that combines efforts from many different organizations. The ocean prediction program runs daily at the Navy DoD Supercomputing Resource Center, and results are published within 48 hours of being generated [3]. Physical locations and their corresponding ocean currents are available to 1/12° equatorial resolution and seven kilometer average horizontal spacing [4]. The data set offers velocity measurements (both magnitude and direction) describing ocean currents for any location specified by time, latitude, longitude, and depth, with currents assumed to be constant over each day. Predictions are provided up to four days in the future. Generally, ocean currents were stronger near the surface and weaker at lower depths. Equatorial latitudes tended to have stronger streams.

These forecasts inform a simple forward model to calculate floats' future positions. Our simulation computes the future network configuration for each of  $T$  timesteps into the future at a time resolution  $\Delta t$ .

Currents are considered discrete in the time dimension, where the velocity takes the value measured at the timestep directly before it. At any exact time in the continuous spectrum, the simulation assumes the time to be the nearest entry within the time vector:

$$t_{sim} = \left\lceil \frac{t_{exact}}{\Delta t} \right\rceil * \Delta t \quad . \quad (1)$$

The simulation involves  $n$  floats over  $T$  timesteps. The pair  $x_t = [latitude, longitude]$  specifies the float's position on a 2D map at time  $t$ . Floats generally move at the velocity of their surrounding current field. For a local current at depth  $d$  given by velocity  $v(d, x_t)$  the motion equations in two dimensions yield a single float's location at the timestep  $t + \Delta t$ :

$$x_{t+1} = x_t + \Delta t \ v(d, x_t) \quad (2)$$

By assuming that currents are constant over each timestep, the simulation can compute each float's trajectory for the entire length of the simulation.

This model fails in several contingency cases including shallow waters (where the floats may attempt to travel to a depth greater than the ocean floor's depth) and coastlines (where the floats may run ashore). In these cases our forward simulation does not move the float but instead flags the collision incident with a token that can be interpreted later by the path cost function. Through this method, the floats are kept contained inside the boundaries of bodies of water.

## III. COST FUNCTIONS

The path planning algorithms used various cost functions to evaluate the utility of a candidate network configuration  $L_t$ . For our purposes, a "configuration" was the spatial location, in latitude and longitude, of all  $n$  floats in the network at a given timestep. We used superscripts to index floats in the network and subscripts to index time, so that a configuration consists of the set  $X_t = \{x_t^1, x_t^2, \dots, x_t^n\}$ .

A cost function mapped the set of float locations onto a real-valued utility score. Such cost functions were written with two priorities in mind: minimizing computational complexity and accurately quantifying the desirability of a float configuration. Cost calculation was the most expensive part of the forward simulation, particularly in simulations involving many floats (on the order of hundreds or thousands), so we designed our evaluation strategies to achieve linear time complexity.

Each cost function included a term to prevent floats from entering shallow waters or colliding with coastlines. The forward model marked any locations where a float collides with either the ocean floor or the coast. Configurations were penalized in proportion to the number of grounded floats. A separate term in each cost function calculation, denoted as  $A(X_t)$ , distributed a penalty  $p$  to those paths that encounter land.

$$A(X_t) = \sum_f [p \text{ if } x_t^f \text{ is obstructed, } 0 \text{ otherwise}] \quad (4)$$

This caused a preference for paths that result in fewer grounded floats: should two floats be grounded simultaneously, cost minimization dictated that the first would be rewritten to avoid land, even if the second could not be rerouted. If  $A(L_t)$  reflected only the presence or absence of the token, and not the number of appearances, neither path would be rewritten since the revision of only the first path would not improve cost. Our structure for this extra

penalty term allowed algorithms to escape such a loop of multiple grounded floats.

### A. Coverage Maximization

Visiting a wide range of geographic locations provides a more diverse dataset to assimilate into global prediction models and improves the overall utility of the network for modeling. One of the cost functions used in our project favors dispersed configurations. Under weak assumptions about the measured process, such configurations maximize information gain and are therefore preferable from an experimental design perspective. Work done by Krause and Guestrin [5] suggests a method for measuring information gain explicitly using the mutual information of observations with respect to all locations of interest. This calculation could be computationally prohibitive for thousands of floats and millions of potential sampling locations. We desire efficient approximations that will improve sensor placement by distancing floats from each other. This requires pairwise distance calculations between floats, which could be an intractable calculation of  $O(n^2)$  or  $O(n \log n)$  with appropriate caching structures like kd-trees.

For arranging sensors on a 2D plane, a heuristic approximation of the maximum mutual information can be achieved by spreading sensors uniformly throughout the area of interest. This approximation disregards edge effects, which we prefer to handle using coastline penalty terms so that investigators can control the collision penalty according to their tolerance for risk. We employed a maximum-spread cost function that drives the network to separate the floats from each other.

Here we propose a linear time maximum-spread objective function. Rather than compute pairwise distances between all floats, it implicitly computes distances between each float and a fixed number of map grid squares. It places a two-dimensional Gaussian kernel of unit density centered on each float's location, denoted  $g(lat, lon)$ , and evaluates the result for each map grid square. This yields an isotropic "sensing footprint" – a local region of influence for which the float's sensor data is informative. We calculated the total coverage benefit of a single sensor by integrating its footprint over the mapped area. Additional measurements from nearby floats introduce redundancy, so we combine overlapping footprints with an element-wise maximum operator. This yields a density no greater than that of the two floats' footprints if they were taken independently. The element-wise sum of the resulting map yields the total score for that configuration. This cost function could easily be implemented as a parallel operation, or in specialized GPU hardware as filtering operations on a 2D map "image".

```

Input: Float Locations  $X_t$ 
Output: Cost  $C_{cover}(X_t)$ 
1 Initialize map  $M_{180 \times 720}$  to zero
2 for each  $x_t^i = [lat, lon]$  in  $X_t$  do
3    $M = \max(M, g(lat, lon))$ 
4    $M = \max(M, g(lat, lon+360))$ 
5  $K = M_{0:180, 180:540}$ 
6  $C = A(X_t) + ||K||_1$ 
7 return  $-C$ 

```

**Cost Function 1:** Function to maximize coverage. M is expanded to encompass two complete passes through longitude. This eliminates any inaccuracy in distance calculations when passing through the prime meridian.

For the sake of consistency, the final calculated value of this cost function was made negative. Because our optimization favors lesser cost, it prefers a larger kernel sum and therefore a greater covered area.

### B. Motion Toward a Specific Target Location

Another cost function directs the floats towards a target location  $y$  specified by latitude and longitude. This ability may be useful in coverage maintenance: a damaged float can be replaced by sending another to its location. Also, directing a large number of floats to some single, central location would greatly facilitate their collection for retrieval and potential re-use. This cost function might also task floats to cover specific regions of interest or transient phenomena as one element of a dynamic ocean sensorweb. The total cost is given by the mean distance between each float and the target.

$$C_{target}(y, X_t) = A(X_t) + 1/n \sum_f ||y - x_t^f|| \quad (5)$$

## IV. PATH PLANNING ALGORITHMS

Other path planning applications have employed many different algorithms to select the optimal control strategies for dynamic current environments. These various approaches can generally be divided into grid-search and potential field algorithms. Alvarez et. al. explored and implemented a genetic algorithm for a UAV in an ocean environment based on the Darwin theory of natural selection. As this work noted, genetic algorithms are typically vulnerable to local minima. However, the process presented by this group managed to reliably converge to the global minimum, even in fields where local minima exist [9]. Warren explored in great depth the potential field approach to path planning. Positive fields were placed around obstacles and negative fields around destination or goal positions, and vehicles were pulled towards the negative, attractive pole [10]. Wavefront expansion is a third technique commonly used in path planning. In the presence of strong currents, however, the method becomes unreliable and may return paths that are physically impossible for vehicles to traverse. Soullignac et. al. solved this problem by developing a sliding wavefront expansion system. This method was based on the original wavefront expansion but corrected for the influence of

strong air or water currents, and was able to guarantee a feasible path to some specified precision [11].

The float planning problem has several unique challenges that demand entirely new algorithmic approaches. First, float planning involves controlling many tens or thousands of floats simultaneously. This is an order of magnitude larger than any existing AUV fleet and produces a configuration space with thousands of dimensions at any timestep. Additionally, the floats are highly underactuated and most locations on the map are unreachable by any one float. Often, paths are completely determined by current conditions, and reasonably strong currents can render entire subsets of the fleet uncontrollable. Under these circumstances, individual floats deployed at similar locations can exchange plans with no real affect on the resultant paths. Finally, the utility of any one float's plan may be strongly coupled to the data collected by its neighbors, so the maximum coverage problem cannot easily be decomposed into independent single-float planning.

For simplicity, we ignore the time spent transitioning between depths. For Argo floats, this transition time amounts to 6 hours. Our simulations considered timesteps on the order of multiple days, so the time required to rise and sink to depth became insignificant and was therefore ignored. We represent the path plan for a single Argo float  $f$  using a *depth profile*: a list  $D^f = \{d_1^f, d_2^f, \dots, d_T^f\}$  of desired depths at each timestep, with no other constraints on permissible depth levels. Each float was assigned a separate and independent depth profile, and the combination of all  $n$  floats' profiles formed an  $N$ -by- $T$  matrix representing the policy for the entire network over the course of a mission. Each column of this matrix, then, detailed a depth profile  $D^f$  for a single float, with  $D^i$  not necessarily equal to  $D^j$ . The planning problem involves finding an  $N$ -by- $T$  matrix which optimizes the objective function when combined with current predications.

The high-dimensional configuration space of the multi-robot planning environment precludes an exhaustive search. Under these circumstances many path planning systems employ a myopic or heuristic search to improve computational efficiency [6]. In this project, we explored two traditional approaches modified for our particular use: a "constant depth" algorithm and a variable depth, or "greedy" algorithm. The constant algorithm places strong constraints on the flexibility of each vehicle's profile, but evaluates cost based on the entire simulation period. Conversely, the greedy algorithm permits considerable flexibility of motion but looks only one timestep into the future.

#### A. Constant Depth Algorithm

As the name indicates, the constant depth algorithm considers only depth profiles that maintain a single depth throughout the simulation. For a given location in the ocean, there are  $m$  possible depths between the ocean floor and the water's surface. We constructed a depth profile for each float by setting  $d_t = \Psi^f$ , where  $\Psi^f$  is specific to each float but constant across timesteps. We initialize all values of  $\Psi^f$  randomly, and then iteratively update  $\Psi^f$  for each float in turn, selecting a new value that minimizes the cost on the last

simulation timestep given the depth profiles of the other floats. We continued iterating until the depth profiles did not change. Note that the existing Argo control strategy can be described as a universal constant-depth policy for a single, fixed depth of 2000 meters. We will use the Argo policy as a baseline performance benchmark in later experiments.

#### B. Greedy Algorithm

The greedy algorithm assigns depths based only on the immediate future timestep. This dramatically reduces the expense of its calculations, which expand only a single branch of the expansive policy space. This algorithm is particularly suited for near-term applications using HYCOM, since accurate current predictions are only available over short horizons.

Our greedy algorithm traverses the depth profile one increment at a time, expanding only the current timestep. It selects the depth for time  $t$  which results in the lowest-cost configuration at time  $t+1$ .

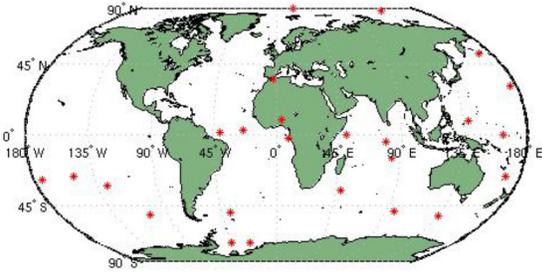
**Input:** Initial float locations  $X_t$   
**Output:** Plan  $M = \{d_t^f$  for each  $f, t\}$

- 1 **for** each timestep  $t$
- 2     **for** each iteration
- 3         **for** each float  $f$
- 4              $c^* \leftarrow \infty$
- 5             **for** each possible depth  $d^*$  **do**
- 6                 forward simulate  $x'_{t+1} \leftarrow x'_t + \Delta t v(d^*, x'_t)$
- 7                 **if**  $C(X_{t+1}) < c^*$  **then**
- 8                      $c^* \leftarrow C(X_{t+1})$
- 9                      $d_t^f \leftarrow d^*$
- 10 **return**  $M = \{d_t^f$  for all floats, timesteps}

**Algorithm 1:** Purely greedy algorithm selects the depth at each timestep that yields the minimum-cost configuration at the timestep directly following. Subscripts  $f$  and  $t$  refer to the float and timestep, respectively. Each iteration looks for the top-performing depth  $d^*$  with cost  $c^*$ .

The myopic algorithm expands as few branches as are necessary. At timestep  $t+1$ , every path considered has identical depths assigned to timesteps  $0$  to  $t$ . This gives the algorithm a narrow scope, but the expense of this process is extremely small since only a small number of positions are computed at each timestep (one for each possible float and depth). It may be worthwhile to employ a multi-step greedy algorithm, which would look forward two or three steps when choosing a depth for a given timestep. For our study, however, inexpensive calculation took priority, so only the next immediate timestep was considered. Additionally, our study used current prediction data from HYCOM which is most reliable for short-horizon planning. Thus we deemed a short-sighted greedy algorithm more appropriate.

## V. EXPERIMENTAL METHOD



**Fig 2.** illustrates the random placement of 25 deployment locations used to test the coverage maximization cost function.

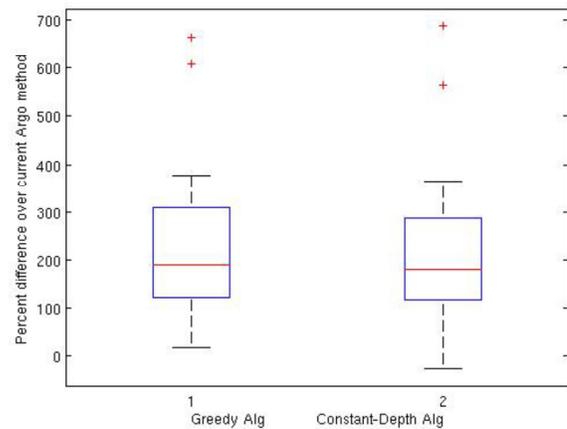
Multiple tests evaluated the two path planning algorithms. We used HYCOM data from 2010 to model ocean currents. We drew 100 random scenarios with locations from a uniform distribution over all ocean locations, as displayed in figure 2. For each chosen start location, 20 floats were evenly dispersed in a grid formation around the selected start point. Starting the floats from slightly varied locations allowed more diverse paths to emerge. To encourage simplicity of calculation (by reducing expense in evaluating both cost function values and algorithms), we chose to model only 20 floats at each location. For this study, we chose to run simulations at 100 different locations was chosen rather than modeling a larger number of floats at each location. This put more weight into investigating reliability in many varied geographical locations, which is valuable to marine monitoring research systems. Also, the target-location cost function is meaningful for small numbers of floats; as  $n$  increases into the hundreds, the objective of moving all floats to the same location becomes less useful. We ran the “mission” for a 30-day period broken into timesteps of 5 days and considered both the coverage maximization cost function and a second trial based on the target-seeking objective with a destination 3 degrees north and east of each deployment.

The experiments included two distinct simulation scenarios. First, we considered a “long-horizon” planning scenario that might be implemented with long-term current predictions extrapolated from seasonal or historical data. Here, we assumed long-range forecasts that were available for the entire duration of the simulation, and were are uploaded once at the start but never revised. No current prediction product currently exists that could permit this long-duration planning, so its purpose was purely exploratory. As a more realistic case, we also considered a “short-horizon” simulation that could be implemented now using HYCOM 5-day forecasts. Here, plans were updated at 5-day intervals based on the immediate results of the previous communications cycle. This planning cycle was identical to the length of each planning timestep. Without the ability to predict beyond the next timestep, all algorithms reduced to the greedy case. Thus, the short-horizon trials will only compare the baseline Argo and greedy approaches.

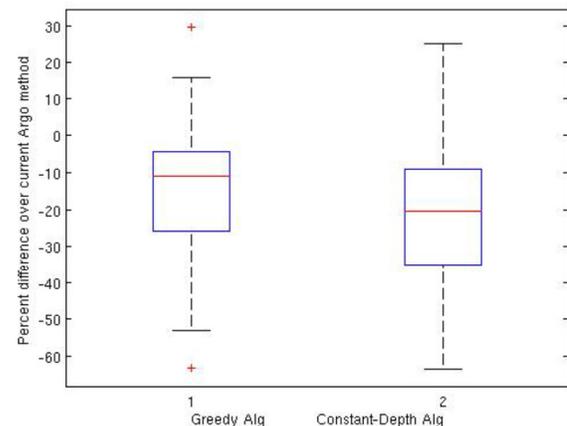
In practice, the actual currents experienced by the float will differ from the predictions. To account for this, we did a second simulation based on *hindcast* products also produced by HYCOM. The hindcast model assimilates data from the immediate past and infers the most likely currents for the previous five days. These retrospective estimates are presumably more accurate than the forecast predictions. The experiments that follow replicate forecast inaccuracy by generating plans using realistic forecast data and then running the actual simulation on the hindcast data.

## VI. RESULTS

Each current-sensitive algorithm generated paths with lower cost than the baseline Argo method. The magnitude of the advantage varied between the two cost functions.



**Fig 3a.** charts performance over all trials using the coverage maximization cost function. Results are measured using percent difference between the algorithmic path’s utility values and the baseline. Any positive value indicates an improvement in utility.

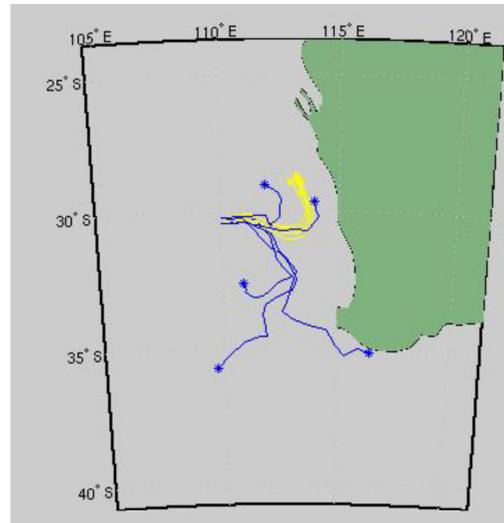


**Fig. 3b.** charts performance over all trials using the target location cost function. Again, results are expressed as percent difference between algorithmic and naïve planning. In this case, cost is measured by distance to target, where a smaller number is better. Negative percent differences, then, indicate improvement.

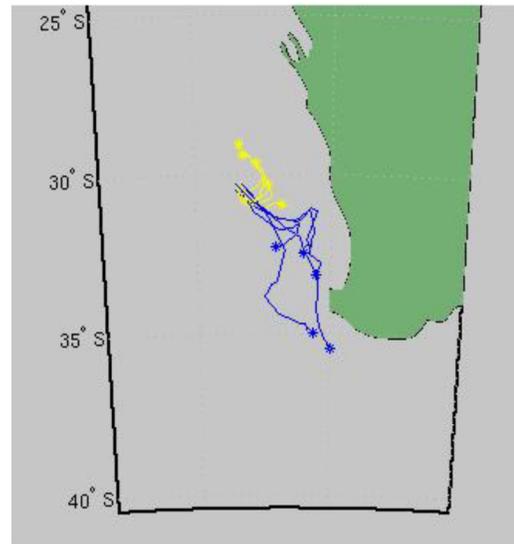
The two plots depict the cost function values for both goal configurations: the first maximized coverage, and the second was intended to send the floats to a specific target location. Our algorithms achieved dramatically improved configurations with respect to coverage maximization, generating utilities at an average of twice or three times the value of the baseline method, as displayed in figure 3a. The box plots of figure 3a show the median and quartiles of the data, while the whiskers indicate extrema. Red plus signs show extreme outliers.

The target location cost function yielded less dramatic results. While our algorithms still produced improved path plans on average, the cost margin between the baseline and algorithms was smaller, as displayed in 3b. The smaller magnitude of the improvements is an effect of the structure of our cost function, rather than the performance of the algorithms themselves. Graphically, the performance for motion toward a target location is comparable to that for the coverage maximization objective.

A case study of one specific area just west of Australia illustrates qualitatively the advantage to the algorithmic methods. We simulated five floats over 30 days (with a timestep length of two days) using two depth profiles: those generated by the greedy algorithm and the naïve Argo approach. The floats were initially deployed in a single location, then allowed to follow depth profiles according to the current Argo method (yellow) or our algorithmic depth profiles (blue). Figure 4a shows that our path plans were more effective in maximizing coverage. The utility of the blue configuration is twice that of the yellow, which would map onto figure 3 at a 100 percent difference. Figure 4b represents the same case study run with the target location cost function, and the algorithmic paths still performed better than the current Argo methods. The difference in cost between the algorithms modeled here corresponds to only a 60 percent difference, quantitatively smaller than the dispersion example. In practice, though, both trials show that algorithmic path planning offers significant advantages.



**Fig. 4a** Case study of 5 floats, just west of Australia. Yellow was used for the trajectories generated by the baseline method, and blue was used to show trajectories developed by the greedy algorithm.



**Fig. 4b** Case study in figure 4 redone to direct the floats to a target location southeast of their deployment location. Blue represents the greedy algorithm, an improvement over the naïve Argo approach, shown in yellow.

The simulations run on forecast and hindcast data showed promising reliability. The discrepancies between expected (based on forecast data) and actual (based on real-time currents) locations were generally low. Figure 5 gives the average location variation for each algorithm after a 30 day simulation.

Algorithm	Latitude	Longitude
	Displacement	Displacement
Constant Depth	0.304	0.234
Greedy	0.36	0.226

Fig 5. details the average discrepancy in final locations of floats based on differences between forecast and actual ocean current data.

Further tests are necessary to fully explore the reliability of forecast data for path planning.

## VII. CONCLUSION

The majority of trials show that path-planning of some sort is advantageous to the operation of submersible floats. Neither path planning algorithm clearly dominated the other. Any ability to control and direct floats' movements in a more directed manner will be valuable to the Argo operation or similar future deployments.

Throughout the trials, the greedy algorithm performed equally well or better than the non-myopic approaches; this adds great practicality to path planning. Since the greedy algorithm only requires current forecasts for the immediate future, the small amount of prediction data made available by HYCOM is completely sufficient. Operators can generate discrete segments of depth profiles continuously and direct the floats as desired using the forecasts available to us at a given time. This process proves much more feasible than the constant-depth model, which requires current velocities for every time point in the simulation in advance.

Further research in this area should explore correlations between algorithms' performances and qualities of the simulation location. The attempt at mapping utility to physical aspects of deployment locations (distance from shore, etc.) included in this paper demands further study. Also, there may be benefits to a more probabilistic approach that permits uncertainty over current predictions. This would offer a path toward extrapolating for long-range forecasts.

## ACKNOWLEDGMENT

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology. Copyright 2011 California Institute of Technology. All Rights Reserved. US Government Support Acknowledged.

## REFERENCES

- [1] J. Gould, D. Roemmich, S. Wijffels, H. Freeland, M. Ignaszewsky, X. Jianping, S. Pouliquen, Y. Desaubies, U. Send, K. Radhakrishnan, K. Takeuchi, K. Kim, M. Danchenkov, P. Sutton, B. King, B. Owens and S. Riser "Argo profiling floats bring new era of in situ ocean observations", EOS, vol. 85, pp. 179 2004.
- [2] [http://www.argo.ucsd.edu/How\\_Argo\\_Floats.html](http://www.argo.ucsd.edu/How_Argo_Floats.html)
- [3] <http://www.hycom.org/>
- [4] <http://www.hycom.org/dataserver/glb-analysis/expt-90pt8>
- [5] Krause, A., and Guestrin, C. 2005c. *Near-optimal Value of Information in Graphical Models*.
- [6] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," IEEE Trans. Syst., Man, Cybern., vol. 22, no. 2, pp. 224-241, 1992.
- [7] D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi. An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments. In AIAA 21st Digital Avionics Systems Conference, Irvine, CA, October 2003.
- [8] C. Petres and P. Patron "Path planning for unmanned underwater vehicles", Proc. IJCAI Workshop Planning, Learning in A Priori Unknown or Dynamic Domains, pp. 47 2005.
- [9] A. Alvarez, A. Caiti and R. Onken "Evolutionary path planning for autonomous underwater vehicles in a variable ocean", IEEE J. Ocean. Eng., vol. 29, pp. 418 2004.
- [10] C. W. Warren, "A technique for autonomous underwater vehicle route planning", IEEE J. Ocean. Eng., vol. 15, no. 3, pp. 199 - 204, 1990.
- [11] Soullignac, M.; Talilibert, P.; and Rueher, M. 2009. Timeminimal path planning in dynamic current fields. Proceedings of the IEE
- [12] D. Thompson, S. Chien, M. Arrott, A. Balasuriya, Y. Chao, P. Li, M. Meisinger, S. Petillo, O. Schofield. Mission Planning in a Dynamic Ocean Sensorweb, International Conference on Planning and Scheduling (ICAPS) SPARK Applications Workshop, 2009.
- [13] David R. Thompson, Steve Chien, Arjuna Balasuriya, Stephanie Petillo, Yi Chao, Peggy Li, Bronwyn Cahill, Julia Levin, Michael Meisinger, Matthew Arrott, Oscar Schofield. "Spatiotemporal Path Planning in Strong, Dynamic, Uncertain Currents". Proc. of the IEEE International Conference on Robotics and Automation 2010.
- [14] B. Garau, A. Alvarez, and G. Oliver, "Path planning of autonomous un- derwater vehicles in current fields with complex spatial variability: An A\* approach," in Proc. Int. Conf. Robot. Autom., 2005, pp. 18-22.
- [15] D. Kruger, B. Stolkin, A. Blum, and J. Briganti, "Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments," in Proc. Int. Conf. Robot. Autom., 2007, pp. 4265-4270.