Improving Constrained Glider Trajectories for Ocean Eddy Border Sampling within Extended Mission Planning Time

Aleš Zamuda
Faculty of Electrical Engineering and Computer Science,
University of Maribor
Smetanova ul. 17, 2000 Maribor, Slovenia
Email: ales.zamuda@um.si

José Daniel Hernández Sosa, Leonhard Adler
Institute of Intelligent Systems and Numerical Applications in Engineering,
University of Las Palmas de Gran Canaria
Campus de Tafira, 35017 Las Palmas de Gran Canaria, Spain
Email: dhernandez@iusiani.ulpgc.es, leonhard.adler101@alu.ulpgc.es

***Abstract*—This paper extends the performance assessment of an underwater glider path planning approach recently proposed for constrained sub-mesoscale eddy border sampling conditions, for situations benefiting from extended mission planning time. The aim of addressing such situations is to improve the glider vehicle capabilities through improving its off-board controller, which computes an improved trajectory for the eddy sampling task, compared to the usual rather shorter planning time.**

The improvement in robustness for the controller for several scenarios in this global trajectory optimization is also analyzed, together with comparison to shorter planning time for this autonomous vehicle and environmental data sampling type. As shown through results, the approach is able to provide several useful and non-intuitive solutions, improving in helpful ways. The trajectories for sub-mesoscale eddy sampling are thereby improved, in a way that might be useful for possible machine controller pondering or auto-piloting at open sea, when piloting user feedback is not available or even amidst the consecutive interruptions of user-intensive planning instructions.

Managing complexity under limited resources and designing vessel navigation schedule plan under uncertain conditions within such extended mission planning time, therefore improves the mission quality as well. By optimizing trajectories with differential evolution and then visualizing them, we provide human-machine interaction for rapid knowledge discovery, data mining, and presentation of possibly large space satellite captured data sets (Big Data) analysis and exploitation.

I. Introduction

This paper extends the performance assessment of an underwater glider path planning (UGPP) approach [1], recently proposed for constrained sub-mesoscale eddy border sampling conditions, utilizing evolutionary optimization.

An underwater glider is an autonomous underwater vehicle (AUV), a nautical robotic vessel, operating on and below the sea's surface. The glider propels itself forward on a path by producing small changes in its buoyancy and transforming the resultant vertical motion to horizontal displacement by means of the interaction between the vehicle control surfaces and the water column. The vehicle trajectory is formed by the concatenation of successive underwater segments or stints. Each stint, in turn, consists of a series of saw-tooth profiles (yo-yos) between two target maximum and minimum depths (typically 1000 and 10 meters), followed by a final emersion. Once at sea surface, satellite or wireless communication with the control center takes place, during which the glider can fix its position, send data, and receive new commands. [1]

In the next section, the related work and background are presented, then the description of the executed sub-mesoscale eddy sampling UGPP within extended mission planning time is defined in Section 3. In Section 4, experiments are presented. Section 5 conveys the conclusions.

II. Related Work and Background

In this section, the differential evolution optimization algorithm and underwater glider path planning related work and background are presented, especially focusing on constraints optimization and sub-mesoscale eddy sampling trajectories.

A. Differential Evolution and Optimization

Differential Evolution (DE) was introduced by Storn and Price [2] using a floating-point encoding evolutionary algorithm [3] for global optimization over continuous spaces. Its main performance advantages over other evolutionary algorithms [4], [5], [6] lie in floating-point encoding and a good combination of evolutionary operators, the mutation step size adaptation, and the elitist selection. DE has been modified and extended several times with new versions proposed [7] and the performances of different DE variant instance algorithms have been widely studied and compared with other evolutionary algorithms, also at various competitions of major scientific conferences, where DE has won several evolutionary algorithm competitions [8].

DE is also used for constraint optimization and there are several variants which entered competitions held at the IEEE Congress on Evolutionary Computation (CEC) [9], [10]. A surrogate matrix DE was recently applied in scheduling hydro and thermal power systems production [11]. DE was also used for the reconstruction of procedural tree models [12] within the EcoMod ecosystem rendering framework [13], which is an biogeography-like algorithm for virtual ecosystem afforestation; in numerical optimization, a biogeography-like

978-1-5090-0623-6/16/$31.00

algorithm was also applied to a grid map discretized robot path planning [14]. Civicioglu [15] used DE for transforming geocentric Cartesian coordinates to geodetic coordinates. DE has also been used for learning and intelligence accumulation approaches optimization, like [16], [17]. DE has also been applied in the robotics and autonomous systems class of applied soft computing [18]. Joshi et al. [19] used DE to fuse multi-sensor data in building intelligent robotic systems. Robotic motion planning and navigation was also addressed by DE by Aydin and Temeltas [20] and Chakraborty et al. [21]. Neri and Mininno [22] applied DE to Cartesian robot control by introducing memetic operators and compact representation. Chen et al. [23] used DE to design satellite orbit for prioritized multiple targets.

The basic DE [2] has a main evolution loop within which successive generations compute new populations of vectors. During one generation g, for each vector $\mathbf{x}_i$, $\forall i \in \{1, 2, ..., NP\}$ in the current population, DE employs evolutionary operators, namely mutation, crossover, and selection, to produce a trial vector (offspring) and to select one of the vectors with the best fitness value. NP denotes population size and $g \in \{1, 2, ..., G\}$, the current generation number. The looping over this population for G times, yields the total maximum number of function evaluations (MAXFES) utilized in one evolutionary optimization run.

B. Constrained UGPP

The UGPP was already addressed for mesoscale eddy sampling in [18], where several computational intelligence algorithms for optimization were compared and showing, that 1) DE was suitable for UGPP optimization and 2) suggesting especially well performing DE for UGPP. In that approach, some interesting trajectories nearby islands were presented, and the land area had already created some sort of constraints for adhering, i.e. the approach allowed for certain land constraints. However, the simulator used in that approach has simply set glider velocity to zero when colliding, so it remained in that position until the simulation stopped. This also always made the evolved path feasible within that approach. By excluding any enhanced constraints handling in the algorithm, the collision trajectories would then have gradually become discarded because they would stop far from the target point; an early marginal collision (a hard limit) could prevent some really good trajectories being evaluated. Also, that approach would not allow the defining of special corridor areas on the sea to adhere to.

In [1], a new recent suitable configuration of a DE optimization algorithm for the constrained UGPP challenge was then presented, where a new approach to UGPP for sub-mesoscale eddy sampling was proposed as well. The approach focused in the sampling of a corridor defined along the eddy border, an area of main interest for the characterization of the structure, and introduced a specific fitness function based on the angular extent of the vehicle trajectory. The suggested configured variant of the optimization algorithm combined the mechanisms of self-adaptation [4], [6], population size reduction [24], [25], and ϵ-constraint handling using ϵ level adjustment [26]. This configuration was a DE for UGPP as the main contribution of that paper, enabling automatic corridor-constrained UGPP. Also, as the jDE algorithm [4] was usually mostly reported to perform best when using a randomly chosen population vector during mutation (known as 'rand' DE mutation [2]), the paper showed that on a small number of function evaluations like UGPP, it was confirmed that when the best population vector within mutation is used (known as 'best' DE mutation [2]), the jDE algorithm performed better than when applying its most widely used 'rand' mutation operation. Also, as proposed and demonstrated by the recent paper by Zamuda and Brest [6], a contribution was testing and confirming that a different than the default value of $\tau = 0.1$ for control parameters randomization frequency was suitable on the benchmark used, such as a value of $\tau = 0.5$ in that case (and an unsuitable setting would make the algorithm rank worse). Further contributions were that, different aspects of that new DE configuration for constraint optimization were studied in that paper for the constrained UGPP challenge, utilizing a newly contributed prepared benchmark set, comprised of 28 different specialized scenarios. The DE configurations were tested over a benchmark set on 51 independent runs for each DE configuration aspect, then per-scenario and aggregated statistical performance differences were reported, including different constraint handling definition strategies, DE mutation strategies configuration, and population sizing parameterizations. Utterly, the proposed approach contributed to improve the robotic vehicle capabilities, giving support to better autonomous operation levels. The DE optimizes the glider path as encoded in a DE population vector $\mathbf{x}_i$, consisting of a set of bearings, where the fitness function computation measures the maximum angular extent the trajectory samples:

$$f(\mathbf{x}_i) = 2\pi - \left(\max_{k=1}^{K}(\alpha_k^{\text{cum}}) - \min_{k=1}^{K}(\alpha_k^{\text{cum}}) \right), \qquad (1)$$

where the α_k^{cum} are glider's k positions angular extents around an eddy center. Constraints handling on glider locations keeps the trajectory path inside the sampled eddy border area [1]. The complete framework of the adopted algorithm and the specifics of the test scenarios are further defined in [1].

III. Sub-mesoscale Eddy Sampling UGPP within Extended Mission Planning Time

In this paper, we further contribute to the improvement of the controlled robotic vehicle capabilities, through allowing the mission planning time to be extended. By doing so, we increase the number of fitness evaluations allowed during optimization, i.e. extending interaction to non near real-time.

The fitness and constraints definition for corridor area sampling utilized in this paper are used as defined in the paper [1] (the $\text{NP}_{\text{dyn}}\epsilon_{\text{sum}}$jDE algorithm) and we change only the allowed MAXFES in the algorithm, in order to check, by how much (and if, at all) the approach can be improved. By doing so, we put the originally chosen MAXFES of $6 * 2^{10}$ from [1] into a broader perspective of a mission planner and noting, when a

Table I
THE OPTIMIZATION ALGORITHM INSTANCES UTILIZED FOR CONSTRAINED UGPP PERFORMANCE ASSESSMENT.

Algorithm	Origin, variant	Modifications tested	Total MAXFES
A4 and A2	ad hoc (the $NP_{dyn}\epsilon_{sum}$jDE/best/1/bin, [1])	2 new variant settings of A	$\times 2$ and $\times 4$
A, B, C, D, E, F	[1]	6 variants	$\times 1$
SaDE	[27], [1]	1 variant (SaDE-C)	$\times 1$
JADE	[28], [1]	1 variant (JADE-C)	$\times 1$
EPSDE	[29], [1]	1 variant (EPSDE-C)	$\times 1$
CoDE	[30], [1]	1 variant (CoDE-C)	$\times 1$
CLPSO	[31], [1]	1 variant (CLPSO-C)	$\times 1$
CMAES	[32], [1]	1 variant (CMAES-C)	$\times 1$
ECHT-DE	[33]	1 original	$\times 1$
DMS-PSO	[34]	1 original	$\times 1$
ABC	[35], [1]	1 variant (ABC-C)	$\times 1$
MABC	[36], [1]	1 original, 1 variant (MABCe)	$\times 1$

MAXFES adjustments might be useful. Namely, the MAXFES parameter was not yet studied for this eddy sampling approach algorithm.

Changing MAXFES however changes the behavior of the population size dynamics (resizing at different generations) and the adaptation of the constraints handling (after generation 50, strict feasibility is required in selection). Also, as the population size is reduced at later stages, this brings the constraints level adjustment mechanism used in the algorithm to larger population sizes, possibly leaving the later smaller populations in full control of strict feasibility/in-feasibility selection without ϵ constraints violation tolerance. Although expecting an improvement of the results, it is due to these inherent population structural effects when changing MAXFES, necessary to test if the algorithm would nonetheless still perform even as well as when MAXFES is changed, as it might possibly be too much dependent on the orchestrated parameters and methods. In other respects, the architecture of the approach stays same in this paper, in order to make the analysis of the MAXFES parameter impact on algorithm orchestration most consistent.

In this paper, the MAXFES is doubled (denoted as MAXFES$\times 2$) and quadrupled (denoted as MAXFES$\times 4$). In the following section, we report the effects and reflect on the results obtained with such configuration of the algorithm $NP_{dyn}\epsilon_{sum}$jDE, comparing it also to several other algorithms under MAXFES$\times 1$ condition as merely a reference ground for considering the doubling and quadrupling of the original MAXFES. The algorithms assessed in this paper (based on [1]) are listed in Table II: the A2 uses MAXFES$\times 2$ and A4 uses MAXFES$\times 4$, while others for reference use MAXFES$\times 1$.

Managing the UGPP complexity under limited (time and data) resources and designing vessel navigation schedule plan under uncertain conditions (currents, prediction errors, human factor) within such extended mission planning time, improves the mission quality as follows from the experiments below, where each algorithm was run with 51 independent runs at 28 different constrained benchmark scenarios.

IV. EXPERIMENTS

First, by visualizing the evolved trajectories, we provide an interface for human-machine interaction to enable rapid knowledge discovery (showing promising trajectories), data mining (displaying trajectories and nearby ocean structures on-screen), and presentation of possibly large space satellite captured data sets (Big Data) analysis and exploitation (we are using live forecast ocean data from the sea current maps provided from MyOcean IBI service (http://myodata.puertos.es/), which include a high resolution Regional Ocean Modeling System, ROMS). The median trajectories obtained from the 51 runs of each of the A, A2, and A4 algorithms, are displayed in Figure 1. As can be seen, for some of the scenarios, the trajectories are improved, but for some the improvements at median trajectories plots are not as visible. When using the interactive software, the trajectories can be further studied, such as checking vicinities of island bodies, sea depths, and seeking the currents time map in order to gain a better introspection of the suggested solution. In the scenarios namely, the currents are changing through time and we only draw the initial currents state in Figure 1 backgrounds. An example un-intuitive trajectory solution suggested is namely seen in e.g. scenario E3_N (first from left in second row from top), where the glider is scheduled to waste some time in a way before heading towards the eddy sampling circular route, due to the later more suitable currents dynamics.

Although the trajectories in Figure 1 already display some promising performance for A4, the convergence graphs as seen in Figure 2 more clearly show the performance differences for these algorithms. The MAXFES$\times 4$ (A4) line converges better than A2 and A in all convergence graphs and the algorithm A2 is also very good compared to the algorithm A.

Further, to make statistical hypotheses testing, a t-test over the obtained final values is used for these three MAXFES configurations. In Table II, the t-tests show that the algorithm A4 significantly outperforms A on almost all scenarios (on scenario 16, it is only slightly better w.r.t. minimum, median, maximum, and average value). A Friedman statistic ranking as

Table II

OBTAINED FITNESS WITH 3 $NP_{DYN}\epsilon_{SUM}$JDE ALGORITHM VARIANTS ON 28 TEST SCENARIOS IN 51 INDEPENDENT RUNS. THE SYMBOL † SIGNIFIES WHEN USING t-TEST ($\alpha = 0.05$, t-STATISTIC WITH $p = 0.95$ AT N=51 IS 1.6753) THAT AN ALGORITHM PERFORMANCE IS BETTER THAN THE VARIANT A PERFORMANCE ON THE SCENARIO TESTED (THE TEST NEVER SIGNIFIED THE OPPOSITE, I.E., THAT ANY CASE WOULD BE WORSE, RESPECTIVELY).

Alg.	Traj.	Best	Median	Worst	Average	Std. dev.
A4	T1	1.7310e+00	2.0423e+00	2.7773e+00	**2.0394e+00**†	2.7032e-01
A4	T2	1.6173e+00	1.8681e+00	2.3670e+00	**1.8699e+00**†	1.8713e-01
A4	T3	8.7239e-01	1.2334e+00	2.4284e+00	**1.3194e+00**†	3.9730e-01
A4	T4	6.7767e-01	1.2217e+00	2.1737e+00	**1.2174e+00**†	3.1223e-01
A4	T5	1.6675e+00	4.3395e+00	5.4486e+00	**4.0394e+00**†	9.6190e-01
A4	T6	2.1139e+00	2.6156e+00	3.1311e+00	**2.5966e+00**†	2.2792e-01
A4	T7	2.1710e+00	2.6060e+00	3.4012e+00	**2.6745e+00**†	3.1907e-01
A4	T8	2.4354e+00	3.2803e+00	4.3233e+00	**3.2948e+00**†	6.3141e-01
A4	T9	2.0639e+00	2.3877e+00	3.0314e+00	**2.4154e+00**†	2.5229e-01
A4	T10	3.8205e+00	4.0318e+00	4.2297e+00	**4.0100e+00**†	1.0496e-01
A4	T11	2.1555e+00	2.5206e+00	2.9411e+00	**2.4924e+00**†	1.9522e-01
A4	T12	2.0162e+00	2.9085e+00	4.6841e+00	**2.9092e+00**†	6.4217e-01
A4	T13	1.0141e+00	1.6191e+00	2.9224e+00	**1.6374e+00**†	4.9358e-01
A4	T14	1.2460e+00	1.6533e+00	2.6466e+00	**1.7101e+00**†	3.5561e-01
A4	T15	-1.1337e+00	-4.7556e-01	8.6688e-01	**-3.6538e-01**†	5.3256e-01
A4	T16	-9.4381e-01	-1.6001e-01	1.1149e+00	-1.6405e-01	5.4784e-01
A4	T17	4.7905e+00	5.4849e+00	5.6613e+00	**5.4792e+00**†	1.0577e-01
A4	T18	5.2439e+00	5.3050e+00	5.5326e+00	**5.3231e+00**†	6.1733e-02
A4	T19	3.4134e+00	3.4637e+00	3.7021e+00	**3.4821e+00**†	6.4535e-02
A4	T20	2.7527e+00	3.8928e+00	4.7468e+00	**3.9105e+00**†	5.3432e-01
A4	T21	1.9886e+00	2.4408e+00	3.2405e+00	**2.3902e+00**†	3.2714e-01
A4	T22	2.8851e+00	3.2290e+00	3.6638e+00	**3.2491e+00**†	1.9568e-01
A4	T23	2.5774e+00	2.8920e+00	3.7203e+00	**2.9018e+00**†	2.7435e-01
A4	T24	3.2961e+00	3.3741e+00	4.2029e+00	**3.4463e+00**†	1.8641e-01
A4	T25	-1.0123e+00	-2.2315e-01	5.2119e-01	**-2.6716e-01**†	4.5015e-01
A4	T26	-8.4029e-01	-2.1112e-01	7.5723e-01	**-2.2001e-01**†	3.1235e-01
A4	T27	1.9533e+00	2.3502e+00	3.0452e+00	**2.4074e+00**†	2.9217e-01
A4	T28	2.5537e+00	3.1605e+00	4.3231e+00	**3.2041e+00**†	3.5716e-01
A2	T1	1.7240e+00	2.0883e+00	2.8101e+00	**2.0746e+00**†	2.5967e-01
A2	T2	1.6113e+00	1.8639e+00	2.3582e+00	**1.8501e+00**†	2.0046e-01
A2	T3	8.7560e-01	1.2964e+00	2.7944e+00	1.4246e+00	5.1383e-01
A2	T4	7.1934e-01	1.2548e+00	2.1388e+00	**1.2392e+00**†	2.9541e-01
A2	T5	1.8283e+00	4.5851e+00	5.4388e+00	4.1684e+00	9.9225e-01
A2	T6	2.1863e+00	2.6546e+00	3.5933e+00	**2.6403e+00**†	2.6789e-01
A2	T7	2.1276e+00	2.7383e+00	3.6176e+00	**2.7726e+00**†	3.4683e-01
A2	T8	2.4218e+00	3.1197e+00	4.3547e+00	**3.2303e+00**†	6.3215e-01
A2	T9	2.0665e+00	2.5128e+00	3.0972e+00	**2.5065e+00**†	2.4603e-01
A2	T10	3.8128e+00	4.0446e+00	4.3388e+00	4.0406e+00	1.1347e-01
A2	T11	2.1098e+00	2.5469e+00	3.0257e+00	**2.5154e+00**†	2.0579e-01
A2	T12	2.0627e+00	2.9905e+00	5.1881e+00	**3.1609e+00**†	7.4347e-01
A2	T13	1.0182e+00	1.6387e+00	2.6710e+00	1.6851e+00	4.9441e-01
A2	T14	1.2311e+00	1.6693e+00	2.7283e+00	**1.7303e+00**†	3.7422e-01
A2	T15	-1.0722e+00	-3.6948e-01	1.9535e+00	**-3.0253e-01**†	5.1522e-01
A2	T16	-9.8626e-01	-2.5268e-01	1.7986e+00	-1.7669e-01	6.3734e-01
A2	T17	4.7974e+00	5.4961e+00	5.6835e+00	5.4883e+00	1.3265e-01
A2	T18	5.2495e+00	5.3261e+00	5.4963e+00	5.3368e+00	6.1141e-02
A2	T19	3.4121e+00	3.4760e+00	4.0201e+00	3.5037e+00	1.0997e-01
A2	T20	2.7896e+00	4.0211e+00	5.0333e+00	**3.9535e+00**†	5.5707e-01
A2	T21	1.9797e+00	2.4480e+00	3.6514e+00	2.4703e+00	3.9068e-01
A2	T22	2.8835e+00	3.2735e+00	3.9395e+00	3.3050e+00	2.2993e-01
A2	T23	2.5730e+00	2.9022e+00	4.2148e+00	2.9898e+00	3.8438e-01
A2	T24	3.2994e+00	3.4699e+00	4.2141e+00	3.5990e+00	2.8439e-01
A2	T25	-1.0104e+00	-2.5751e-01	1.0816e+00	-1.8373e-01	4.8233e-01
A2	T26	-7.6970e-01	-1.4825e-01	7.9006e-01	**-1.1556e-01**†	3.4661e-01
A2	T27	1.9678e+00	2.4676e+00	2.9974e+00	**2.4698e+00**†	2.7013e-01
A2	T28	2.5094e+00	3.1596e+00	4.5422e+00	**3.2688e+00**†	4.7348e-01
A	T1	1.7674e+00	2.1261e+00	2.7025e+00	2.1720e+00	2.2618e-01
A	T2	1.6244e+00	1.9029e+00	2.5204e+00	1.9357e+00	2.0154e-01
A	T3	8.7040e-01	1.2661e+00	3.2448e+00	1.4334e+00	5.4547e-01
A	T4	7.6096e-01	1.3169e+00	2.1555e+00	1.3544e+00	2.8607e-01
A	T5	1.8334e+00	4.4143e+00	5.5626e+00	4.2990e+00	8.7211e-01
A	T6	2.2994e+00	2.7011e+00	3.5593e+00	2.7372e+00	2.4504e-01
A	T7	2.1728e+00	2.8470e+00	4.2371e+00	2.8762e+00	3.8242e-01
A	T8	2.4986e+00	3.5445e+00	4.5309e+00	3.4625e+00	5.9638e-01
A	T9	2.0603e+00	2.5971e+00	3.4156e+00	2.5869e+00	3.0648e-01
A	T10	3.8490e+00	4.0427e+00	4.4186e+00	4.0637e+00	1.3508e-01
A	T11	2.2573e+00	2.5725e+00	3.1102e+00	2.5945e+00	2.0809e-01
A	T12	2.0341e+00	3.3748e+00	5.4085e+00	3.3425e+00	7.6138e-01
A	T13	1.0342e+00	1.6717e+00	2.8566e+00	1.7831e+00	4.4453e-01
A	T14	1.3470e+00	1.8350e+00	2.6524e+00	1.8817e+00	3.4718e-01
A	T15	-9.9174e-01	-1.2333e-01	1.1312e+00	-1.2713e-01	4.4563e-01
A	T16	-9.5301e-01	-1.0099e-01	1.2718e+00	-4.9759e-02	5.0626e-01
A	T17	5.4327e+00	5.4979e+00	5.6372e+00	5.5000e+00	3.2532e-02
A	T18	5.2436e+00	5.3339e+00	5.4936e+00	5.3491e+00	6.4232e-02
A	T19	3.4227e+00	3.4893e+00	4.0338e+00	3.5218e+00	1.0592e-01
A	T20	2.9836e+00	4.1957e+00	4.8505e+00	4.0834e+00	4.8735e-01
A	T21	2.0046e+00	2.5730e+00	3.3584e+00	2.5370e+00	3.7704e-01
A	T22	2.8986e+00	3.2932e+00	3.7677e+00	3.2986e+00	2.0007e-01
A	T23	2.5925e+00	2.9183e+00	3.9335e+00	3.0016e+00	3.0719e-01
A	T24	3.3187e+00	3.5745e+00	4.5536e+00	3.6497e+00	2.8390e-01
A	T25	-8.1672e-01	-1.3410e-01	7.6318e-01	-9.1121e-02	3.7906e-01
A	T26	-8.0083e-01	-1.0429e-02	1.3713e+00	2.3025e-02	3.7411e-01
A	T27	2.0053e+00	2.5412e+00	3.3050e+00	2.5418e+00	3.0787e-01
A	T28	2.7872e+00	3.2677e+00	4.4462e+00	3.3826e+00	3.9488e-01

non-parametric test over all scenarios joined, shows through the post-hoc tests that the algorithms A4 and A2 outperform algorithm A also in general, as seen from Table III. This also rejects the hypothesis that when MAXFES was changed, the algorithm A would be too much dependent on the orchestrated parameters and methods and would loose its performance due to its inner structural properties. Therefore, although this might seem trivial, but due to the complexities explained in the previous section, now we can confirm that the algorithm A was configured robustly to allow improving its performance by using more MAXFES (applying it, when the mission planning time permits doing so).

The amount of improvement when increasing MAXFES can in one way be presented based on the data from Table II. On average, the angular extent results improvements from MAXFES$\times$1 to MAXFES$\times$2 is by roughly 0.0892, and to MAXFES$\times$4, 0.1479, i.e. by extending the planning time by $\times$2 or $\times$4, we obtained trajectories sampling 5.11° or 8.47° additional eddy circular border space, respectively. Also, as much as the variant A4 outperforms the variant A on almost all scenarios, it also outperforms the variant A2 roughly almost as much. The average obtained relative percentual improvement per scenario was thereby 2.57% for the doubled calculation time and 4.41% for the calculation quadrupled time. Although this shows that our approach gives already fairly good results on the original calculation time, we show that by extending the calculation (scheduling) time, the results can even on average indeed be improved even further.

A similar improvement against more algorithms utilizing the base MAXFES (see Table I), is seen in Table IV, confirmed by Friedman rankings and post-hoc procedures rejecting the null-hypotheses. As can be seen from the statistical procedures reported, the algorithm A4 also outperforms all other algorithms, which is as expected from the previous analyses on data from Tables II and III. Improving trajectories for sub-mesoscale eddy sampling within extended mission planning time using the utilized approach therefore yields even better capabilities for the underwater glider robotic vehicle.

V. CONCLUSION

This paper presented extending the performance assessment of an approach for the challenge regarding corridor-constrained underwater glider path planning (UGPP). In the utilized approach, the feasible vehicle trajectories were implicitly confined to a corridor area around the border of an ocean sub-mesoscale eddy, a zone of main interest from the point of view of the structure characterization. Sampling of the maximum angular extent of a feasible eddy border was optimized using differential evolution and constraints handling. With extending the allowed mission planning time, we have utilized a greater number of function evaluations for the UGPP simulations and thereby gaining better UGPP trajectories. The 28 test scenarios inhibiting different eddy borders were utilized as a benchmark set for this comparison. Using these, a statistical assessment of the approach was reported, together with comparisons between the normally

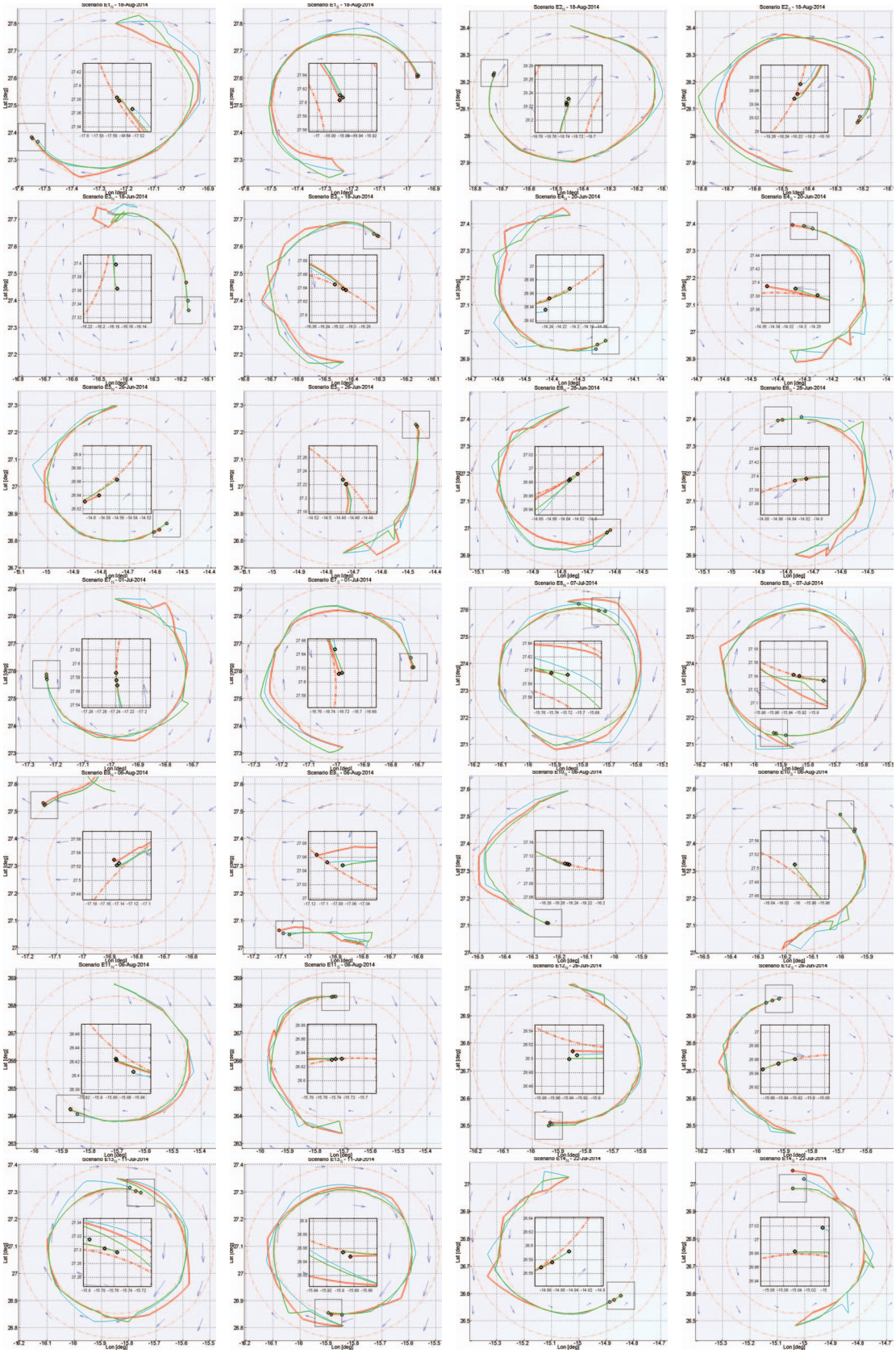

Figure 1. Trajectories obtained using MAXFES×1 (light blue), MAXFES×2 (red), and MAXFES×4 (green).

Table III
ON FRIEDMAN STATISTIC AT $\alpha = 0.05$, THE POST-HOC PROCEDURES REJECT HYPOTHESES OF SAME PERFORMANCE, I.E. THE BASE ALGORITHM A IS OUTPERFORMED (REVERSED RANKS) BY EACH ENHANCED ALGORITHM: BONFERRONI-DUNN'S AT p-VALUE ≤ 0.025, HOLM/HOCHBERG'S (H/H), ROM'S, AND LI'S AT p-VALUE ≤ 0.05, AND HOMMEL'S REJECTS ALL.

i	algorithm	$z = (R_0 - R_i)/SE$	p	H/H/H	Holland	Rom	Finner	Li
2	A4	16.9	8.93e-64	0.025	0.0253	0.025	0.0253	0.0526
1	A2	10	1.13e-23	0.05	0.05	0.05	0.05	0.05

Table IV
ON FRIEDMAN STATISTIC AT $\alpha = 0.05$, THE POST-HOC PROCEDURES REJECT HYPOTHESES OF SAME PERFORMANCE, I.E. THE BEST ALGORITHM A4 OUTPERFORMS EACH OTHER ALGORITHM: BONFERRONI-DUNN'S AT p-VALUE ≤ 0.00125, HOLM/HOCHBERG'S (H/H), ROM'S, AND LI'S AT p-VALUE ≤ 0.05, AND HOMMEL'S REJECTS ALL.

i	algorithm	$z = (R_0 - R_i)/SE$	p	H/H/H	Holland	Rom	Finner	Li
1	A2	3.57	0.000356	0.05	0.05	0.05	0.05	0.05
2	A	8.42	3.88e-17	0.025	0.0253	0.025	0.0488	0.0526
3	NP=30: DMS-PSO	12.8	1.41e-37	0.0167	0.017	0.0167	0.0476	0.0526
4	NP=30: JADE-C	13.3	4.46e-40	0.0125	0.0127	0.0131	0.0463	0.0526
5	C	14.3	1.12e-46	0.01	0.0102	0.0105	0.0451	0.0526
6	NP=30: SaDE-C	17.2	4.16e-66	0.00833	0.00851	0.00876	0.0439	0.0526
7	NP=50: DMS-PSO	18.1	4.33e-73	0.00714	0.0073	0.00751	0.0427	0.0526
8	NP=30: MABCe	20.6	7.64e-94	0.00625	0.00639	0.00657	0.0414	0.0526
9	NP=30: MABC	20.7	3.44e-95	0.00556	0.00568	0.00584	0.0402	0.0526
10	NP=100: DMS-PSO	21.4	8.07e-102	0.005	0.00512	0.00526	0.039	0.0526
11	NP=50: MABCe	21.4	5.39e-102	0.00455	0.00465	0.00478	0.0377	0.0526
12	NP=50: MABC	21.7	2.88e-104	0.00417	0.00427	0.00438	0.0365	0.0526
13	NP=50: JADE-C	23.1	1.73e-118	0.00385	0.00394	0.00405	0.0353	0.0526
14	D	23.3	1.77e-120	0.00357	0.00366	0.00376	0.034	0.0526
15	E	25.3	2.3e-141	0.00333	0.00341	0.00351	0.0328	0.0526
16	NP=50: SaDE-C	25.3	1.1e-141	0.00313	0.0032	0.00329	0.0315	0.0526
17	NP=100: MABCe	26	4.71e-149	0.00294	0.00301	0.00309	0.0303	0.0526
18	NP=100: MABC	26.1	2.22e-150	0.00278	0.00285	0.00292	0.0291	0.0526
19	NP=30: ECHT-DE	27.2	2.77e-163	0.00263	0.0027	0.00277	0.0278	0.0526
20	NP=100: CMAES-C	27.2	2.66e-163	0.0025	0.00256	0.00263	0.0266	0.0526
21	NP=50: CMAES-C	28.1	1.99e-173	0.00238	0.00244	0.0025	0.0253	0.0526
22	NP=30: CMAES-C	28.1	1.99e-173	0.00227	0.00233	0.00239	0.0241	0.0526
23	NP=30: CoDE-C	32.8	9.61e-236	0.00217	0.00223	0.00229	0.0228	0.0526
24	NP=30: CLPSO-C	32.9	2.78e-237	0.00208	0.00213	0.00219	0.0216	0.0526
25	NP=100: JADE-C	32.9	8.96e-238	0.002	0.00205	0.0021	0.0203	0.0526
26	NP=100: SaDE-C	33.5	1.58e-245	0.00192	0.00197	0.00202	0.0191	0.0526
27	NP=30: ABC-C	33.7	1.56e-249	0.00185	0.0019	0.00195	0.0178	0.0526
28	NP=50: ABC-C	34.5	8.87e-261	0.00179	0.00183	0.00188	0.0165	0.0526
29	NP=50: ECHT-DE	35.9	5.24e-283	0.00172	0.00177	0.00181	0.0153	0.0526
30	NP=50: CLPSO-C	38.1	0	0.00167	0.00171	0.00175	0.014	0.0526
31	NP=50: CoDE-C	38.3	0	0.00161	0.00165	0.0017	0.0127	0.0526
32	NP=30: EPSDE-C	38.3	0	0.00156	0.0016	0.00164	0.0115	0.0526
33	NP=100: ABC-C	39.1	0	0.00152	0.00155	0.00159	0.0102	0.0526
34	F	41.3	0	0.00147	0.00151	0.00155	0.00894	0.0526
35	B	41.4	0	0.00143	0.00146	0.0015	0.00766	0.0526
36	NP=100: CLPSO-C	41.6	0	0.00139	0.00142	0.00146	0.00639	0.0526
37	NP=100: CoDE-C	41.9	0	0.00135	0.00139	0.00142	0.00512	0.0526
38	NP=100: ECHT-DE	43	0	0.00132	0.00135	0.00138	0.00384	0.0526
39	NP=50: EPSDE-C	45.5	0	0.00128	0.00131	0.00135	0.00256	0.0526
40	NP=100: EPSDE-C	50.5	0	0.00125	0.00128	0.00132	0.00128	0.0526

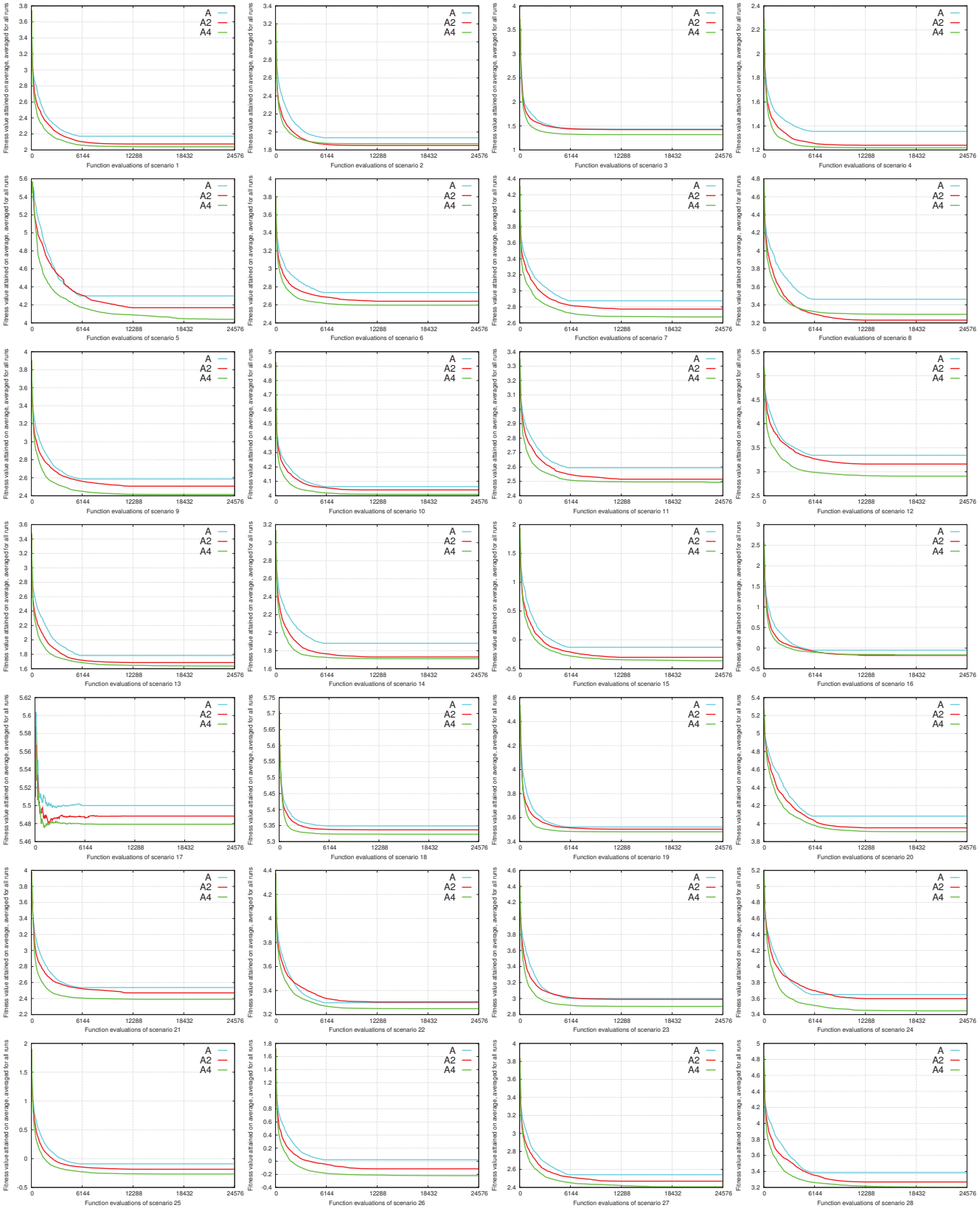

Figure 2. Convergence graphs of average attained fitness for test scenarios 1–28. In those scenarios (scenario 17), where the line is jittery, the algorithm in that part is still trying to find a feasible trajectory.

operating short time interval planning and the extended time planning. With extending the planning time, we show that improved trajectories can be obtained and that the differences are significant on several scenarios. Also, we confirmed that the utilized approach was configured robustly for improving its performance through more MAXFES in the algorithm.

ACKNOWLEDGMENT

This work was funded by the Slovenian Research Agency under project P2-0041 and the Canary Islands government and FEDER funds under project 2010/62. Part of codes in Matlab for extending the optimization algorithms utilized are provided by Qingfu Zhang at http://dces.essex.ac.uk/staff/qzhang/code/. We thank P.N. Suganthan for sending us the original source codes of the ECHT-DE and DMS-PSO algorithms. We thank Dervis Karaboga for providing the original source codes of the algorithms ABC (Matlab) and MABC (Delphi) at http://mf.erciyes.edu.tr/abc/.

REFERENCES

[1] A. Zamuda, J. D. H. Sosa, and L. Adler, "Constrained Differential Evolution Optimization for Underwater Glider Path Planning in Submesoscale Eddy Sampling," *Applied Soft Computing*, vol. 42, pp. 93–118, 2016.

[2] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[3] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer, 2003.

[4] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[5] F. Neri and V. Tirronen, "Recent Advances in Differential Evolution: A Survey and Experimental Analysis," *Artificial Intelligence Review*, vol. 33, no. 1–2, pp. 61–106, 2010.

[6] A. Zamuda and J. Brest, "Self-adaptive control parameters' randomization frequency and propagations in differential evolution," *Swarm and Evolutionary Computation*, vol. 25, pp. 72–99, 2015, (Special Issue on Recent Advances in Modern Nature-Inspired Algorithms, RAMONA).

[7] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[8] J. Brest, P. Korošec, J. Šilc, A. Zamuda, B. Bošković, and M. S. Maučec, "Differential evolution and differential ant-stigmergy on dynamic optimisation problems," *International Journal of Systems Science*, vol. 44, no. 4, pp. 663–679, 2013.

[9] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. C. Coello, and K. Deb, "Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization," Nanyang Technological University, Singapore, Tech. Rep. Technical Report Report 2006005, 2005.

[10] R. Mallipeddi and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization," Nanyang Technological University, Singapore, Tech. Rep., 2010.

[11] A. Glotić and A. Zamuda, "Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution," *Applied Energy*, vol. 141, pp. 42–56, 1 March 2015.

[12] A. Zamuda and J. Brest, "Vectorized procedural models for animated trees reconstruction using differential evolution," *Information Sciences*, vol. 278, pp. 1–21, 2014.

[13] ——, "Environmental framework to visualize emergent artificial forest ecosystems," *Information Sciences*, vol. 220, pp. 522–540, 2013.

[14] H. Mo and L. Xu, "Research of biogeography particle swarm optimization for robot path planning," *Neurocomputing*, vol. 148, pp. 91–99, 2015.

[15] P. Civicioglu, "Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm," *Computers & Geosciences*, vol. 46, pp. 229–247, 2012.

[16] H. Jiang, Y. Dong, J. Wang, and Y. Li, "Intelligent optimization models based on hard-ridge penalty and RBF for forecasting global solar radiation," *Energy Conversion and Management*, vol. 95, pp. 42–58, 2015.

[17] A. P. Piotrowski, "Differential evolution algorithms applied to neural network training suffer from stagnation," *Applied Soft Computing Journal*, vol. 21, pp. 382–406, 2014.

[18] A. Zamuda and J. D. Hernández Sosa, "Differential Evolution and Underwater Glider Path Planning Applied to the Short-Term Opportunistic Sampling of Dynamic Mesoscale Ocean Structures," *Applied Soft Computing*, vol. 24, pp. 95–108, November 2014.

[19] R. Joshi and A. Sanderson, "Minimal representation multisensor fusion using differential evolution," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 29, no. 1, pp. 1083–4427, 1999.

[20] S. Aydin and H. Temeltas, "Fuzzy-differential evolution algorithm for planning time-optimal trajectories of a unicycle mobile robot on a predefined path," *Advanced Robotics*, vol. 18, no. 7, pp. 725–748, 2004.

[21] J. Chakraborty, A. Konar, L. C. Jain, and U. K. Chakraborty, "Cooperative multi-robot path planning using differential evolution," *Journal of Intelligent and Fuzzy Systems*, vol. 20, no. 1, pp. 13–27, 2009.

[22] F. Neri and E. Mininno, "Memetic compact differential evolution for cartesian robot control," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 54–65, 2010.

[23] Y. Chen, V. Mahalec, Y. Chen, R. He, and X. Liu, "Optimal satellite orbit design for prioritized multiple targets with threshold observation time using self-adaptive differential evolution," *Journal of Aerospace Engineering*, vol. 28, no. 2, 2014.

[24] J. Brest and M. S. Maučec, "Population Size Reduction for the Differential Evolution Algorithm," *Applied Intelligence*, vol. 29, no. 3, pp. 228–247, 2008.

[25] A. Zamuda and J. Brest, "Population Reduction Differential Evolution with Multiple Mutation Strategies in Real World Industry Challenges," in *Swarm and Evolutionary Computation*, ser. Lecture Notes in Computer Science, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, Eds. Springer Berlin / Heidelberg, 2012, pp. 154–161.

[26] A. Zamuda, J. Brest, B. Bošković, and V. Žumer, "Differential Evolution with Self-adaptation and Local Search for Constrained Multiobjective Optimization," in *IEEE Congress on Evolutionary Computation 2009*. IEEE Press, 2009, pp. 195–202.

[27] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[28] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[29] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.

[30] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.

[31] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[32] N. Hansen and A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[33] R. Mallipeddi and P. N. Suganthan, "Ensemble of constraint handling techniques," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 561–579, 2010.

[34] J. J. Liang and P. Suganthan, "Dynamic multi-swarm particle swarm optimizer with a novel constraint-handling mechanism," in *IEEE Congress on Evolutionary Computation 2006*. IEEE, 2006, pp. 9–16.

[35] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[36] D. Karaboga and B. Akay, "A modified artificial bee colony (ABC) algorithm for constrained optimization problems," *Applied Soft Computing*, vol. 11, no. 3, pp. 3021–3031, 2011.