Success History Applied to Expert System for Underwater Glider Path Planning using Differential Evolution

Aleš Zamuda

Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova ul. 17, 2000 Maribor, Slovenia

José Daniel Hernández Sosa

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

Abstract

This paper presents an application of a recently well performing evolutionary algorithm for continuous numerical optimization, Success-History Based Adaptive Differential Evolution Algorithm (SHADE) including Linear population size reduction (L-SHADE), to an expert system for Underwater Glider Path Planning (UGPP). The proposed algorithm is compared to other similar algorithms and also to results from literature. The motivation of this work is to provide an alternative to the current glider mission control systems, that are based mostly on multidisciplinary human-expert teams from robotic and oceanographic areas. Initially configured as a decision-support expert system, the natural evolution of the tool is targeting higher autonomy levels.

To assess the performance of the applied optimizers, the test functions for UGPP are utilized as defined in literature, which simulate real-life oceanic mission scenarios. Based on these test functions, in this paper, the performance of the proposed application of L-SHADE to UGPP is aggregated using statistical analyis.

The depicted fitness convergence graphs, final obtained fitness plots, trajectories drawn, and per-scenario analysis show that the new proposed algorithm yields stable and competitive output trajectories. Over the set of benchmark missions, the newly obtained results with a configured L-SHADE outperforms existing literature results in UGPP and ranks best over the compared algorithms. Moreover, some additional previously applied algorithms have been reconfigured to yield improved performance. Thereby, this new application of evolutionary algorithms to UGPP contributes significantly to the capacity of the decision-makers, when they use the improved UGPP expert system yielding better trajectories.

Keywords:
Differential Evolution, Linear Population Size Reduction, Success-history based parameter adaptation, L-SHADE, Underwater Glider Path Planning

Email addresses: ales.zamuda@um.si (Aleš Zamuda), dhernandez@iusiani.ulpgc.es (José Daniel Hernández Sosa)

1. Introduction

This paper presents Underwater Glider Path Planning (UGPP) complex mission scenarios' optimization with a recently well performing evolutionary algorithm for continuous numerical optimization, Success-History Based Adaptive Differential Evolution Algorithm (SHADE in (Tanabe & Fukunaga, 2013)) including linear population size reduction (L-SHADE in (Tanabe & Fukunaga, 2014)). Nowadays, the glider missions operate under a manual control basis. Human experts from Robotics and Oceanographic fields analyse the environmental conditions and the mission objectives to decide about which commands should be sent to the vehicle. The introduction of an expert system is of great interest here, both from tactical and strategic perspectives, liberating human experts from this low-level, time consuming task.

The real-world implementation of UGPP over a dynamic and changing environment including deep oceans requires complex mission planning under very high uncertainties (Thompson et al., 2010). Such mission is also influenced to a large extent by the remote sensing for forecasting weather models' outcomes used to predict spatial currents in deep sea, further limiting the available time for accurate run-time decisions by the pilot who needs to re-test several possible mission scenarios in a short time, usually a few minutes. All these factors represent a clear application opportunity for an expert system, configured initially as a decision-support tool. As more complex missions are introduced, the natural evolution of such a tool is to gain relevance, targeting higher autonomy levels in operation.

UGPP and Underwater Autonomous Vehicle (UAV) applications in (Lermusiaux et al., 2016), coupled with stochastic optimization like evolutionary optimization, are a recently well accepted challenges, with, e.g., the best papers awarded at the Genetic and Evolutionary Computation Conference (GECCO) 2016 Real World Applications (RWA) Track (Ellefsen et al., 2016) or modern evolutionary optimization studies like the first one in 2014 (Zamuda & Hernández Sosa, 2014) and other applications (Zhao et al., 2016; Cashmore et al., 2017; Zadeh et al., 2017).

In the 2017 study, (Lermusiaux et al., 2017), the intelligent systems context of UGPP is well defined, together with future prospects for UGPP, saying that "*Ocean scientists have dreamed of and recently started to realize an ocean observing revolution with autonomous observing platforms and sensors*", and listing evolutionary algorithms under concepts and results in reachability and path planning components of the Marine Science of Autonomy, supporting the capabilities and benefits of expert observing systems listed in the paper. Furthermore, in that paper, (Lermusiaux et al., 2017) it is also recognized in that context, first that "*computational cost grows exponentially for classic deterministic optimization*", and second, also that "*solution is not guaranteed in evolutionary algorithms*"; so we have to work on these recognitions, apply evolutionary algorithms to overcome the computational cost of classic deterministic optimization (as already argued in the paper (Zamuda & Hernández Sosa, 2014)), and assess the outcomes of the applied evolutionary algorithms to evaluate their guarantees empirically, since theoretical insight for these complex algorithms has only recently been catching up, as explained in the review paper (Opara & Arabas, 2018). Therefore, over the modern UGPP mission planning task like (Zamuda & Hernández Sosa, 2014), it is important to incorporate and assess state-of-the-art quality optimizer algorithms to push the boundaries and capabilities of autonomous glider missions, like the first modern evolutionary optimization empirical comparative study (Zamuda & Hernández Sosa, 2014),

which introduced the most advanced evolutionary algorithms, especially versions of Differential Evolution, and advanced the constraints scenarios study (Zamuda et al., 2016a) which has also introduced novel versions of evolutionary algorithms, and has set more open challenges for the future of this emerging topic in expert systems for autonomous glider missions. The new L-SHADE results, which evaluate the new application in UGPP, are reported in this new paper, outperforming existing results as well as configuring better some already applied algorithms for UGPP.

In UGPP, an optimization algorithm considers the ocean currents model predictions, vessel dynamics, and limited communication, to yield potential way-points for the vessel. Based on the most probable scenario for path optimization, this is specially useful in oceanographic engineering for short-term opportunistic missions where no reactive control is possible. Therefore, by improving the expert system and its optimization algorithm component, this can contribute to incrementing the capacity of the vehicles to take intelligent decisions, fostering autonomy.

Related work is presented in the next section. The proposed approach with L-SHADE application is presented in Section 3. Results are reported in Section 4. Conclusions with future work prospects are then given in Section 5.

2. Related Work

In this section, we present the optimization algorithms, using emphasis on Differential Evolution terminology and, specifically, L-SHADE details; then, the insight to the current acchievements in UGPP involving expert systems is presented, including UGPP with evolutionary algorithms.

2.1. Optimization and L-SHADE

L-SHADE is an extension of the basic Differential Evolution (DE), as a floating-point encoding evolutionary algorithm (Eiben & Smith, 2003) for global optimization over continuous spaces. There exist several recent surveys and insights with the DE algorithm's base name (Neri & Tirronen, 2010; Das & Suganthan, 2011; Das et al., 2011; Zamuda & Brest, 2015; Das et al., 2016; Piotrowski, 2017) and its metaphors (Sörensen, 2015; Boussaïd et al., 2013; Zhou et al., 2011) stemming from the progress on computational mechanisms, mainly from the branches of the DE, as well as applications (Zamuda et al., 2011; Zamuda & Brest, 2014; Glotić & Zamuda, 1 March 2015; Parouha & Das, 2016; Mlakar et al., 2017; Baig et al., 2017; Das et al., 2016).

The basic DE (Storn & Price, 1997) consists of an evolutionary loop, within which new population D-dimensional population vectors $\mathbf{x}_i$, $\forall i \in \{1, 2, ..., NP\}$ are evolved. During each generation step number $g \in \{1, 2, ..., G\}$, computational operators like mutation, crossover, and selection on the population are performed, until a termination criterion is satisfied, like a fixed number of maximum fitness evaluations (MAX_FES). L-SHADE extends DE with population size reduction (Tanabe & Fukunaga, 2014) that was already introduced to DE in continuous optimization (Brest & Maučec, 2008) and real-world industry challenges (Zamuda & Brest, 2012); and a well performing parameter control mechanism, named Success History (SH) (Tanabe & Fukunaga, 2014). L-SHADE and its variants have won several recent evolutionary benchmarking competitions (Tanabe & Fukunaga, 2013, 2014). Therefore, a detailed L-SHADE explanation follows in the next paragraphs.

The L-SHADE uses two storages of vectors to be applied during the optimization process. First is the main population, which is used for evolution of vectors from one generation to the next generation. The second storage is the auxiliary archive, which is used as an additional source of the evolutionary process, allowing for an increased number of stored successful vectors, or perhaps, also for later, during evolution after storage, when these are replaced in the main population. For this auxiliary archive, the L-SHADE defines a truncation mechanism, so that the size of this archive does not exceed the value of archive size $|A|$, which is usually set to the size of the main population. When this archive is empty, it is filled with new entries at the end, and when the limiting size is exceeded, the individual ranked last after sorting the archive vectors by fitness is removed.

In L-SHADE, the mathematical operation termed shortly as mutation, creates a mutant vector $\mathbf{v}_{i,g+1}$ for each corresponding current main population vector using a version called 'current-to-$p$best/1' mutation (introduced in JADE (Zhang & Sanderson, 2009)):

$$\mathbf{v}_{i,g+1} = \mathbf{x}_{i,g} + F_i(\mathbf{x}_{p_{\text{best}},g} - \mathbf{x}_{i,g}) + F_i(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g}), \tag{1}$$

where the indexes r_1, r_2, and r_3 are generated randomly with uniform distribution, and are mutually different integers generated within the range $\{1, 2, ..., NP\}$, and they are also different from index i. F_i is an amplification factor of the difference vector for this individual at index i. The first term in the mutation operators defined above is a base vector. Following, the difference between two chosen vectors denotes a difference vector which, after multiplication with F_i, is known as an amplified difference vector. There are two difference vectors in this mutation. While the second one is among two randomly chosen main population members, the first is more complex, because it includes exploitation greediness, which is controlled by a factor of portion $p \in [0, 1]$. A chosen $\mathbf{x}_{p_{\text{best}},g}$ in this mutation denotes a uniform randomly selected individual among the p portion of best ranking vectors sorted from unifying sets of: 1) current main population of size NP and 2) auxiliary archive population that stores successful past trial vectors from generations up to g.

After mutation in L-SHADE, the mutant vector $\mathbf{v}_{i,g+1}$ is recombined using binary crossover with the target vector $\mathbf{x}_{i,g}$ to create a trial vector $\mathbf{u}_{i,g+1} = \{u_{i,1,g+1}, u_{i,2,g+1}, ..., u_{i,D,g+1}\}$:

$$u_{i,j,g+1} = \begin{cases} v_{i,j,g+1} & \text{if } rand(0,1) \leq CR_i \text{ or } j = j_{\text{rand}} \\ x_{i,j,g} & \text{otherwise} \end{cases}, \tag{2}$$

where $j \in \{1, 2, ..., D\}$ denotes the j-th search parameter of D-dimensional search space, $rand(0, 1) \in [0, 1]$ represents a uniformly distributed random number, and j_{rand} denotes a uniform randomly chosen index of the search parameter, which is always exchanged to prevent cloning of target vectors. CR_i denotes the crossover rate (Zaharie, 2009; Zamuda & Brest, 2015).

The mechanism for computing F_i and CR_i in L-SHADE is Success History (SH) (Tanabe & Fukunaga, 2014), which was inspired after mechanisms from algorithms like SPDE (Abbass, 2002), jDE (Brest et al., 2006), and JADE (Zhang & Sanderson, 2009). The SH mechanism computes F_i and CR_i by choosing index $r_i \in [1, H]$ randomly for each new generated trial vector, then applying randomization over the value at this index:

$$F_i = randc_i\left(M_{\text{F},r_i}, 0.1\right), \tag{3}$$

$$CR_i = randn_i\left(M_{\mathrm{CR},r_i}, 0.1\right), \tag{4}$$

which use control parameters memory archives M_{CR} and M_{F} of size $H = |M_{\mathrm{CR}}| = |M_{\mathrm{F}}|$ to store successful offspring control parameters. The memory is initialized as $M_{\mathrm{CR},k} = M_{\mathrm{F},k} = 0.5$, $\forall k \in \{1, 2, ..., H\}$. The $randn_i(\mu, \sigma^2)$ and $randc_i(\mu, \sigma^2)$ are random generator functions for normal or Cauchy distribution with mean μ and variance σ^2, respectively.

After applying mutation and crossover, the selection operator (in minimization) propagates the fittest individual (Darwin, 1859) in the new generation for the main population:

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g+1} & \text{if } f(\mathbf{u}_{i,g+1}) \leq f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{otherwise} \end{cases}, \tag{5}$$

then the possibly successful propagated new vector is also stored in the auxiliary archive using the L-SHADE archiving. After this, the Linear Population Size Reduction (LPSR) mechanism execution trigger is checked against the case, if the new population size differs from the previous value:

$$NP_{g+1} = NP_{\min} + \left\lfloor \frac{NFE}{MAX\_FES}(NP_{\mathrm{init}} - NP_{\min}) + 0.5 \right\rfloor, \tag{6}$$

when the worst individuals are removed from the main population, and also from the auxiliary archive. NFE denotes the current number of fitness evaluations and L-SHADE (Tanabe & Fukunaga, 2014) uses $NP_{\mathrm{init}} = 18D$, $NP_{\min}$=4, archive size equal to main population size, $p = 0.11$, and $H = 5$.

2.2. UGPP Expert Systems and Optimization

UGPP missions are also influenced to a large extent by weather models yielding false predictions or equipment failures, which then requires re-planning of missions when detected. Such events interfere with the available time of UGPP pilots, who need to make updated plans based on new scenarios' simulations and predictions, sometimes within minutes. (Sangrà et al., 2007; Crawford et al., 2005). The manual re-planning scheme is valid for long-term ocean crossing missions, but for short duration trajectories or multiple-vehicle coordination, a different approach is required. The sampling of dynamic ocean structures (fronts, eddies, upwellings), for example, cannot be tackled using this control, due to the severe glider speed limitations. The UGPP research goal is to increase the autonomy level of this kind of vehicles (Zamuda et al., 2016a). Mission planning robustness to prediction, communication, and other failures thereby benefits glider operational capabilities (Lermusiaux et al., 2016; Moura et al., 2010; Eichhorn, 2015).

An expert system outline for UGPP is seen in Figure 1. Such system includes automatization of the AUV path control through the piloter delivering targets and the evolutionary system providing intelligent optimization based on environmental sensors and models and vehicle sensors, connected through limited communication lines. The challenge of path control is nontrivial, as a 4D, spatio-temporally varying field over which to glide needs to be taken into account (Zamuda et al., 2016a). Having the potential of fully autonomous operation, the usual control scheme of ocean gliders does not, however exploit these capacities too much and relies mainly on a human-in-the-loop (Hamann et al., 2016) approach, in part also because of the expensive and fragile equipment involved. The evolutionary system therefore

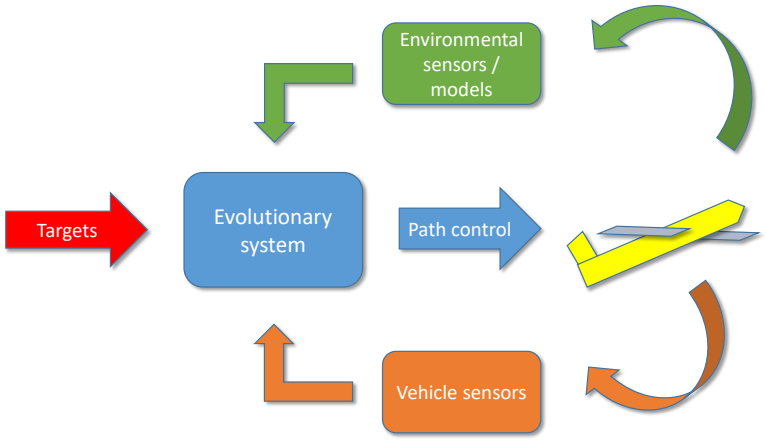

Figure 1: Schematic depiction of a UGPP expert system.

becomes an integral part of an expert system advising the team of human pilots that manage one or more AUVs (Zamuda & Hernández Sosa, 2014). Considering the above explanation of an UGPP expert system, different solutions to the UGPP and AUVs are proposed in the literature:

- Nonlinear Trajectory Generation (NTG) on a Lagrangian Coherent Structures (LCS) model to generate near-optimal routes for gliders within dynamic environments was proposed in (Inanc et al., 2004).
- UGPP trajectory considering depth constraints was designed in (Yoon, 2014).
- Genetic Algorithms (GA) to produce suitable paths in the presence of strong currents while trying to minimize energy consumption was presented in (Alvarez et al., 2004).
- Optimal sampling strategies for coordination of glider fleets, in (Leonard et al., 2010).
- Line formation by AUV swarms was addressed in (Sousselier et al., 2015).
- A multi-objective GA was also applied to autonomous underwater vehicles for sewage outfall plume dispersion observations, which considered two objectives, i.e. the maximum number of water samples besides the total travel distance minimization, in (Moura et al., 2010).
- In the particular case of eddies, the complexity of the path planning scenario is aggravated by the high spatio-temporal variabilities of these structures and their specific sampling requirements in (Hátún et al., 2007).
- A* search algorithm was applied to find optimal paths over a set of eddies with variable scale and dynamics in (Garau et al., 2005).
- An iterative optimization method based on the Regional Ocean Modeling System (ROMS) predictions for generating optimal tracking and sampling trajectories for evolving ocean processes was proposed in (Smith et al., 2010). Their scheme includes near real-time data assimilation, and has been tested both in simulation and real field experiments.

- Zig-zag sampling enhancement and vehicle adaptively following such trajectory was presented in (Kularatne et al., 2015).
- Design of controllable glider wings was addressed in (Wang et al., 2017).
- Path planning within time varying scenarios was also presented in (Eichhorn, 2015).
- Optimization based on a Nelder-Mead algorithm in (Cabrera Gámez et al., 2013) (the `fmisearch` Matlab implementation (Lagarias et al., 1998)) or genetic algorithms (GA) in (Hernández Sosa et al., 2013) were used in early UGPP approaches.
- Level set theory was applied to some path planning problems in the presence of dynamic flows including obstacle avoidance, in (Lolla et al., 2014), focusing on multi-vehicle coordination applications.
- GA to evolve rendezvous trajectories for multiple gliders with minimal energy consumption over all participating vehicles was applied in (Cao et al., 2017).
- Growing popularity deep learning (LeCun et al., 2015) schemes are hard to implement in this problem, due to the difficulty to obtain a representative dataset of glider operation samples, and these techniques are currently reserved mainly only for good visibility unmanned scenarios like airborne robots or space robots (Zhang et al., 2017).
- A UGPP system using DE was developed and analyzed in (Zamuda & Hernández Sosa, 2014, 2015; Zamuda et al., 2016a,b). The selection of the right path planning algorithm version for the expert system is of main importance when dealing with complex and potentially risky situations. During recently developed real-missions, for example, strong currents were present in the vehicle recovery area near the shoreline. Intensive simulation and optimization procedures are required to provide as much evidences as possible for the decision making. Slight differences in the proposed path could result in significant trajectory alterations.

In the approach (Zamuda & Hernández Sosa, 2014), a path based on the local kinematic simulation of an underwater glider is evolved, by considering the daily and hourly sea currents predictions. The path is represented by control points where the glider is expected to resurface for connection with a space communication satellite and to receive further navigational instructions. Such an Evolutionary Elgorithm (EA) application (UGPP DE), the UGPP trajectory optimization, is even more interesting when the current field is not homogeneous and bearings are non-trivial. For specifics about the fitness evaluation and the UGPP approach with DE, the reader is advised to follow the full length descriptions in (Zamuda & Hernández Sosa, 2014), and we merely re-iterate fitness evaluation steps here. To evaluate a DE trial $u_{i,j,g+1}$, its components are used as follows:

$$b_0 = \text{bearing}(\mathbf{p}_0, \mathbf{p}_{\text{target}}), \tag{7}$$

$$b_1 = b_0 + u_{i,1,g+1}, \tag{8}$$

$$\mathbf{p}_1 = \text{simulate\_stint}(\mathbf{p}_0, b_1, \text{map}), \tag{9}$$

where b_1 is the new bearing for the stint to be simulated and obtain fitness, by calculating geometrical distance from the last evolved trajectory point $\mathbf{p}_D$ and the target point $\mathbf{p}_{\text{target}}$, using D stints ($2 \leq j \leq D$), which are followed by drift due to each resurfacing:

$$b_j = b_{j-1} + u_{i,j,g}, \tag{10}$$

$$\mathbf{p}'_{j-1} = \text{simulate\_drift}(\mathbf{p}_{j-1}, \text{map}), \tag{11}$$

$$\{\mathbf{p}_j\} = \{\mathbf{p}_{j-1}\} \cup \text{simulate\_stint}(\mathbf{p}'_{j-1}, b_j, \text{map}), \tag{12}$$

$$f(\mathbf{u}_{i,g}) = \parallel \mathbf{p}_D - \mathbf{p}_{\text{target}} \parallel, \tag{13}$$

where functions bearing, simulate_stint, simulate_drift, together with the map parameter, implement the UGPP simulation to generate a glider trajectory for evaluation.

3. Proposed Application

The L-SHADE algorithm is applied in this paper to the approach for UGPP, and its performance assessed over 12 UGPP benchmark scenarios from (Zamuda & Hernández Sosa, 2014). These scenarios cover diverse ocean conditions in the Canary Island sea, characterized by the presence of highly dynamic ocean structures. As the goal of the path planning is to drive the vehicle from different starting points as close as possible to the corresponding targets, the initial-final points have also been selected according to the benchmark set in (Zamuda & Hernández Sosa, 2014).

The novelty in this paper is described with the following contributions:

1. Application of L-SHADE to UGPP benchmark scenarios previously defined in (Zamuda & Hernández Sosa, 2014) – see in Figure 2, lines 14–23 including UGPP simulation,
2. Significant improvements of the attained results on this UGPP benchamark scenarios, compared to (Zamuda & Hernández Sosa, 2014),
3. Configuration of the L-SHADE LPSR mechanism for UGPP: The initial population for L-SHADE is proposed to be set at 60 here, this is 5 times the number of bearings (it was originally 18 times in (Tanabe & Fukunaga, 2014)), $D = 12$, therefore we denote the new algorithm in this paper as L-SHADE$_5$,
4. Comparison study with similar optimization algorithms (these are listed in Table 1) that have previously been compared on UGPP, and
5. As L-SHADE$_5$ includes population size changes, several other algorithms (based on those previously applied to UGPP) are now presented with changing population sizes during an evolutionary run, comparing their population sizing performances.

The pseudocode of the configured algorithm L-SHADE$_5$ is shown in Figure 2. Requirements to run the algorithm are the inputs as variable parameters (supplied differently for specific mission scenarios), the constant parameters for L-SHADE. The algorithm yields a list of bearing angles. The optimization starts by initializing the evolved population and L-SHADE parameter memory and archive storages (lines 1 and 2). Then the algorithm cycles over the generation loop (line 3) and enters the iteration loop for each population vector (line 4). With the exception of NP_{init} value, all these are as defined by L-SHADE. The lines 5–13 are computation of a trial vector using control parameters adaptation, mutation, and crossover. Lines 14–23 evaluate the trial vector. Lines 24 to 34 update the L-SHADE population, archive, and memory storages. In the end, the best obtained DE vector is returned, containing the values of angles for each evolved bearing (line 36).

Figure 2: Algorithm L-SHADE$_5$: using success history control parameters and LPSR on UGPP with DE.

Require: Variable parameters — $\mathbf{p}_0$ (current glider location), map (MyOcean IBI), $\mathbf{p}_{\text{target}}$ (mission virtual target point), MAX_FES (maximum number of FES allocated), constants — p and $|A|$ (L-SHADE constants for JADE (Zhang & Sanderson, 2009) parameters: mutation greediness $p = 0.11$, archive size $|A|$ matching NP), $NP_{\min} = 4$ and $NP_{\text{init}} = 5D = 60$ (L-SHADE constants for LPSR minimum and maximum DE population sizes), H (L-SHADE constants for setting memory size parameter in storing successful history for control parameters F_i and CR_i).

Ensure: $\mathbf{x}$ – list of instructions (D incremental bearing angles) for glider navigation,

uniform randomly generate DE initial population $\mathbf{x}_{i,0}$, $\forall i \in \{1, 2, ..., NP_{\text{init}}\}$; $NP_0 = NP_{\text{init}}$;
initialize L-SHADE storages: $A = \emptyset$, $\forall k \in \{1, 2, ..., H\}$: $M_{\text{CR},k} = M_{\text{F},k} = 0.5$;
for DE generation loop $g = 1$ to G, $NFE < MAX\_FES$ **do**
  for DE iteration loop i (all individuals $\mathbf{x}_{i,g}$ in population) **do**
    DE new trial vector $\mathbf{x}_{i,g}$ computation:
      uniform randomly generate $r_i \in [1, H]$;
      $F_i = randc_i\,(M_{\text{F},r_i}, 0.1)$;
      $CR_i = randn_i\,(M_{\text{CR},r_i}, 0.1)$;
      $\mathbf{P} = \{\forall k \in [1, NP_g] : \mathbf{x}_{k,g}\} \cup \mathbf{A}$;
      $p_{\text{best}} = \lfloor p|P| + 0.5 \rfloor$;
      choose $\mathbf{x}_{p_{\text{best}},g}$ at p_{best} from $\mathbf{P}$, ranked by fitness;
      $\mathbf{v}_{i,g+1} = \mathbf{x}_{i,g} + F_i(\mathbf{x}_{p_{\text{best}},g} - \mathbf{x}_{i,g}) + F_i(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g})$;
      $\forall j \in \{1, .., D\}$: $u_{i,j,g+1} = \begin{cases} v_{i,j,g+1} & \text{if } rand(0,1) \le CR_{i,g+1} \text{ or } j = j_{\text{rand}} \\ x_{i,j,g} & \text{otherwise} \end{cases}$;
    DE fitness evaluation (UGPP simulation):
      $b_0 = \text{bearing}(\mathbf{p}_0, \mathbf{p}_{\text{target}})$;
      $b_1 = b_0 + u_{i,1,g+1}$;
      $\mathbf{p}_1 = \text{simulate\_stint}(\mathbf{p}_0, b_1, \text{map})$;
      for $\forall j \in \mathbb{N} \mid 2 \le j \le D$:
        $b_j = b_{j-1} + u_{i,j,g}$;
        $\mathbf{p}'_{j-1} = \text{simulate\_drift}(\mathbf{p}_{j-1}, \text{map})$;
        $\{\mathbf{p}_j\} = \{\mathbf{p}_{j-1}\} \cup \text{simulate\_stint}(\mathbf{p}'_{j-1}, b_j, \text{map})$;
      end for
      $f(\mathbf{u}_{i,g}) = \| \mathbf{p}_D - \mathbf{p}_{\text{target}} \|$;
    $\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g+1} & \text{if } f(\mathbf{u}_{i,g+1}) \le f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{otherwise} \end{cases}$;
    if $f(\mathbf{u}_{i,g+1}) < f(\mathbf{x}_{i,g})$:
      replace worst individual in $\mathbf{A}$ with $\mathbf{u}_{i,g+1}$ (or add it when $|\mathbf{A}| < NP_g$);
      save propagated F_i and CR_i for updating parameter memories $\mathbf{M}_{\text{F}}$, $\mathbf{M}_{\text{CR}}$;
    end if
  end for
  update memories $\mathbf{M}_{\text{F}}$ and $\mathbf{M}_{\text{CR}}$ with propagated parameters;
  $NP_{g+1} = NP_{\min} + \left\lfloor \frac{NFE}{MAX\_FES}(NP_{\text{init}} - NP_{\min}) + 0.5 \right\rfloor$; // checks LPSR
  if $NP_{g+1} < NP_g$ **then**
    for population and archive: reduce sizes of both to NP_{g+1} by removing their worst individuals;
  end if
end for
return the best individual obtained among $\mathbf{x}_{i,G}$;

4. Results

In this section, different algorithms are analyzed, besides the newly suggested L-SHADE$_5$ algorithm, including different population sizes and population sizing strategies. We report the results in the format as specified in (Zamuda & Hernández Sosa, 2014), so that the results are directly comparable numerically and graphically to the basic measurements in (Zamuda & Hernández Sosa, 2014). Therefore, the maximum number of fitness evaluations was set at 2,048, number of bearings at $D = 12$, number of scenarios was 12, while the number of independent runs was 10, and scenario settings (ocean currents and glider points) were kept the same as in (Zamuda & Hernández Sosa, 2014). The other optimization algorithm instances, utilized for performance comparisons, are listed in Table 1, together with citations to literature sources.

Table 1: The different optimization algorithm base instances utilized for UGPP.

Algorithm	Proposed
L-SHADE	(Tanabe & Fukunaga, 2014)
jDE/best/1/bin	(Zamuda & Hernández Sosa, 2014)
jDE/rand/1/bin	(Brest et al., 2006)
DynNP jDE/best/1/bin	(Zamuda et al., 2016a)
DynNP jDE/rand/1/bin	(Brest & Maučec, 2008)
CLPSO	(Liang et al., 2006)
SaDE	(Qin et al., 2009)
JADE	(Zhang & Sanderson, 2009)
EPSDE	(Mallipeddi et al., 2011)
CoDE	(Wang et al., 2011)
CMAES	(Hansen & Ostermeier, 2001)

4.1. Per-scenario Competitiveness Analysis

First, let us compare the fitness value convergence graphs for the L-SHADE$_5$ and the algorithms using fixed population size from (Zamuda & Hernández Sosa, 2014). We set the population sizes at 100 for the fixed population size algorithms, the same as in (Zamuda & Hernández Sosa, 2014) (the different algorithm labels: best/1/bin, rand/1/bin, CLPSO, SaDE, JADE, EPSDE, CoDE, CMAES). When adding population size changes during evolution, we renamed the algorithms by adding a subscript to their name, so that those compared algorithms show information on using population sizes different than the default $NP = 100$. These are defined for L-SHADE, or for those with population size reduction from DynNP jDE (Brest & Maučec, 2008; Zamuda et al., 2016a). The numbers split by "-" after NP_{init} for some of the algorithms, therefore denote changing population size NP: number of population size reductions and NP_{min}, respectively, as applied through the mechanism presented in (Zamuda & Brest, 2012) and previously used in UGPP (Zamuda & Brest, 2015).

As seen in Figures 3 and 4, all L-SHADE convergences compared to the other algorithms from Table 1, as well as to algorithms from (Zamuda & Hernández Sosa, 2014), are competitive. The performance of L-SHADE is more interesting, when comparing the final obtained fitness values on average over independent runs, which is seen in Figure 5. Figure 5 shows more indicatively, that the L-SHADE variants all attain fitness mostly among the relatively

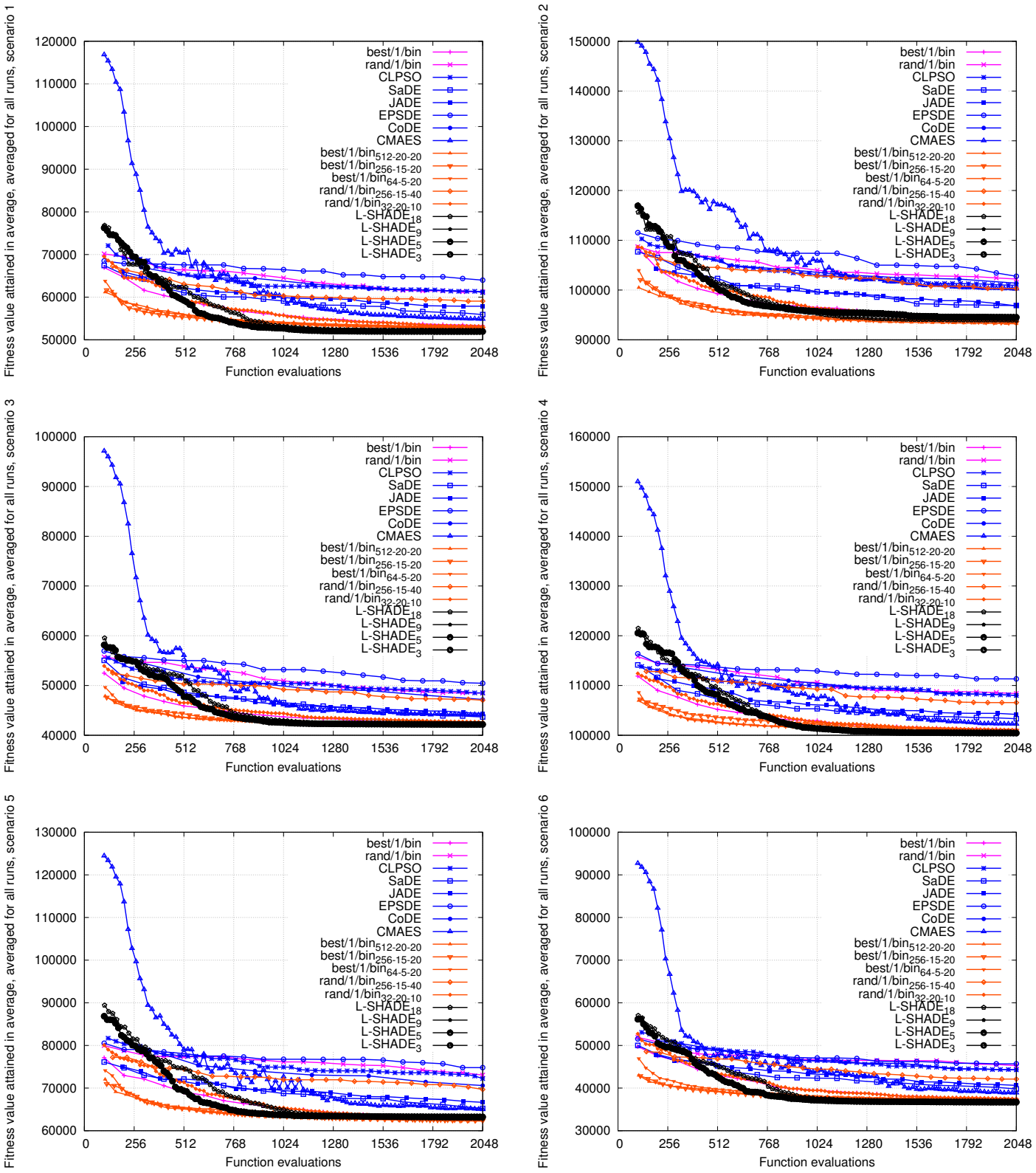

Figure 3: Convergence graphs of average attained fitness for test scenarios 1–6 for the main algorithms.

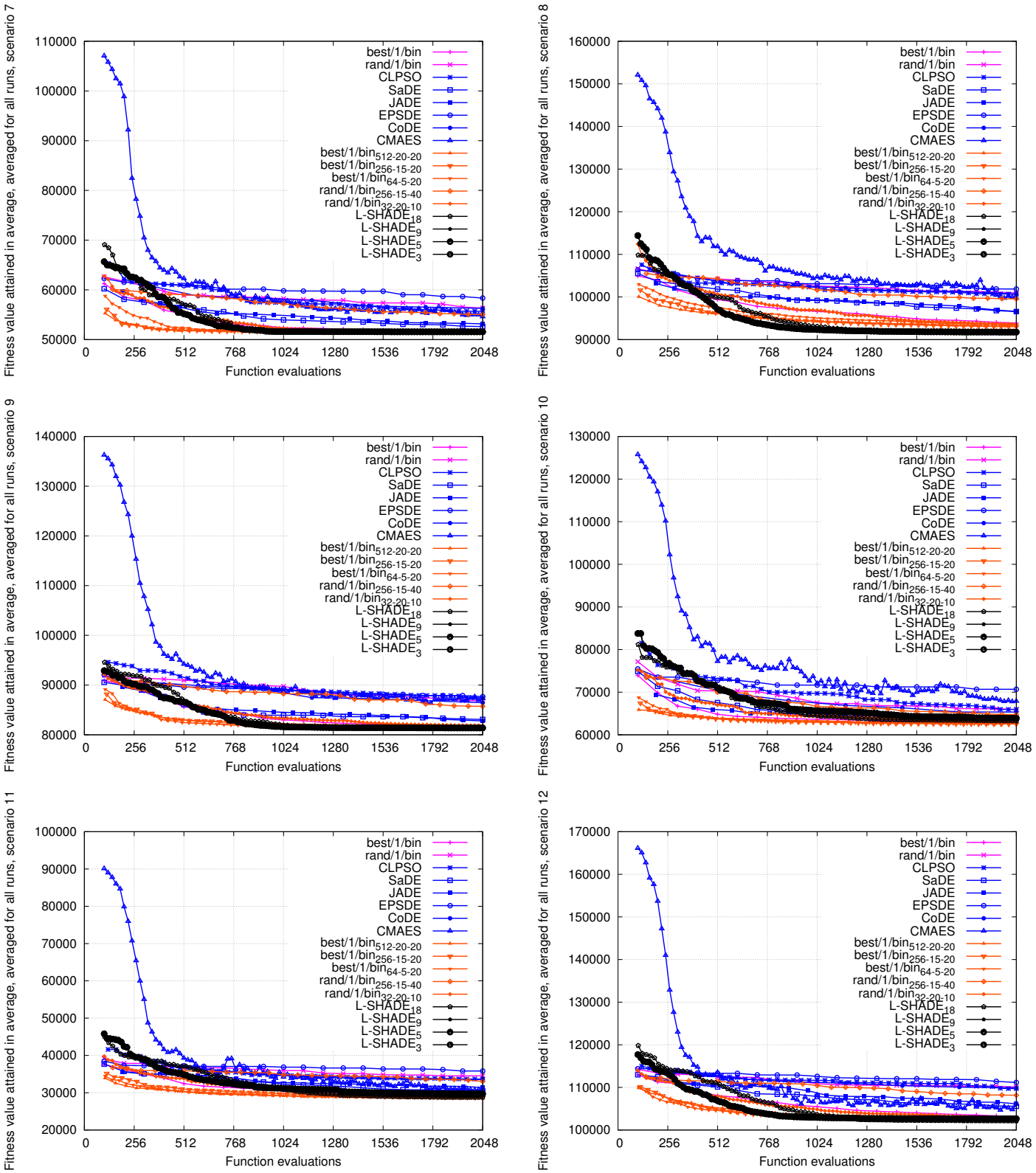

Figure 4: Convergence graphs of average attained fitness for test scenarios 7–12 for the main algorithms.

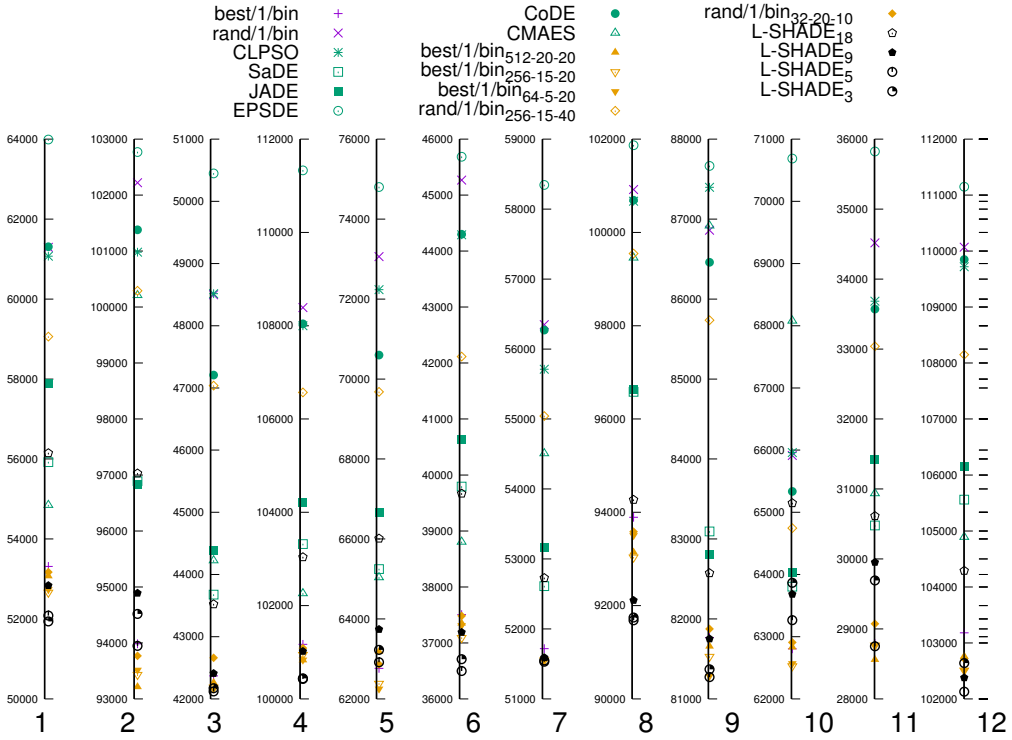

Figure 5: Final fitness average over independent runs on 12 test scenarios – results from previously reported algorithms for UGPP, compared to the proposed algorithm.

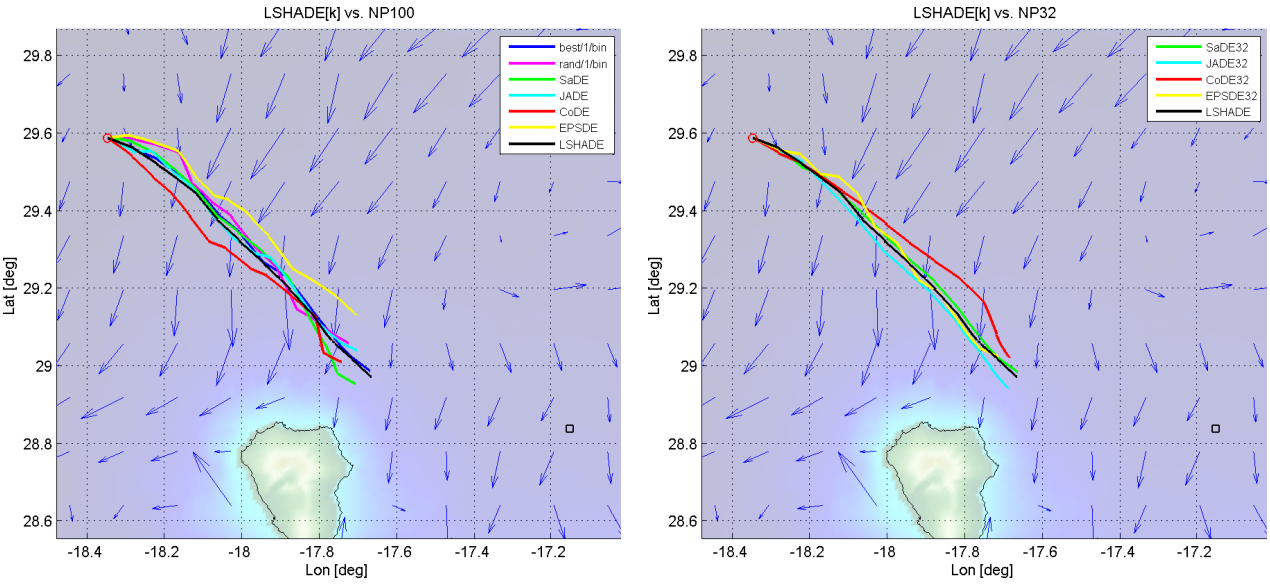

Figure 6: Phenotype path images obtained using sets of algorithms, on scenario 1 (median performing run final trajectory drawn) – left: L-SHADE$_5$ against algorithms with $NP = 100$, right: L-SHADE$_5$ against algorithms with $NP = 32$ that were improved over their corresponding $NP = 100$ variants (see labels for particular algorithm).

lowest (best) values in all scenarios. To display the meaningful differences among trajectories obtained even more evidently, Figure 6 renders the obtained trajectories for example scenario (scenario 1), comparing all algorithms with $NP = 100$ (left) and $NP = 32$ (right, only those are shown with improvement over corresponding original configuration version, as will be discussed in the next subsection) against L-SHADE$_5$, using the final obtained trajectory in their corresponding median performing run. Figures 7 and 8 show in more detail the differences among all 10 independent runs for particular algorithms: it can be seen that L-SHADE$_5$ yields trajectories that are most stable (i.e. the yielded bearing angles' dispersion seems smallest) and also that its trajectories are closest to the target point (which is drawn using a black rectangle).

To evaluate reported results numerically, the statistical analysis of some of these results is reported in Tables 3 and 2, which measure more clearly that L-SHADE$_5$ outperforms several of these algorithms in many scenarios at $\alpha = 0.05$: the † signifies that L-SHADE$_5$ is outperforming the compared algoritm significantly, and ‡ signifies that L-SHADE$_5$ is outperformed. On each scenario, the L-SHADE$_5$ outperformed all competing algorithms reported with $NP = 100$ from previous work (Zamuda & Hernández Sosa, 2014) (Table 3). For almost all scenarios it also outperformed the other L-SHADE configurations tested and the compared algorithms that are based on changing population size for UGPP (Zamuda et al., 2016a) in Table 2: 10 out of 72 cases were where it was outperformed, but still L-SHADE$_5$ was better than each algorithm here at least twice as many times as L-SHADE$_5$ was worse. It outperformed the default L-SHADE$_{18}$ in all cases. Therefore, more population sizing comparisons and aggregative statistics are reported in the following subsection.

4.2. Aggregated Performance Statistics for Algorithms

For comparison as a non-parametric test over the algorithms in Table 4, different DE and evolutionary algorithms with even more different initial population sizes (NP_{init}) are compared, using codes at `http://sci2s.ugr.es/keel/multipleTest.zip` for Friedman ranking (Demšar, 2006). The ranking (lower rank value denotes better algorithm performance) displays an aggregative statistic of algorithms' relative performances, at $\alpha = 0.05$ confidence and Friedman statistic value 4776.1. Again, subscripts denote population sizing, otherwise it was set at 100. In the Table, only the most interesting combinations are reported, as for others, the performance was worse than the proposed L-SHADE$_5$.

Table 5 then reports post-hoc hypotheses on these Friedman statistics, where the post-hoc procedures reject hypotheses of the same performance, i.e. the best algorithm L-SHADE$_5$ outperforms some other algorithm at p_{val}-values: Bonferroni-Dunn's, $\leq$0.000909, Holm's, $\leq$0.00714, Hochberg's, $\leq$0.00625, Hommel's, $\leq$0.00714, Holland's, $\leq$0.0073, Rom's, $\leq$0.00657, Finner's, $\leq$0.0447, and Li's, $\leq$0.0288, respectively.

As can be said on the overall benchmark performance in Table 5, the most post-hoc tests confirm that the proposed L-SHADE$_5$ outperforms the algorithms after rank 7 in all versions of JADE, CMAES, CoDE, and EPSDE, and some versions of DynNP jDE and SaDE.

When initial population for algorithms at a different value that at 100 in Table 5, the ranking results improvement for SaDE$_{32}$ with $NP = 32$ is high compared to SaDE with $NP = 100$. Similar can be observed for JADE, CoDE, and EPSDE, but not for CLPSO or CMAES, which gives another insight in performance comparison and configuration for these algorithms for UGPP.

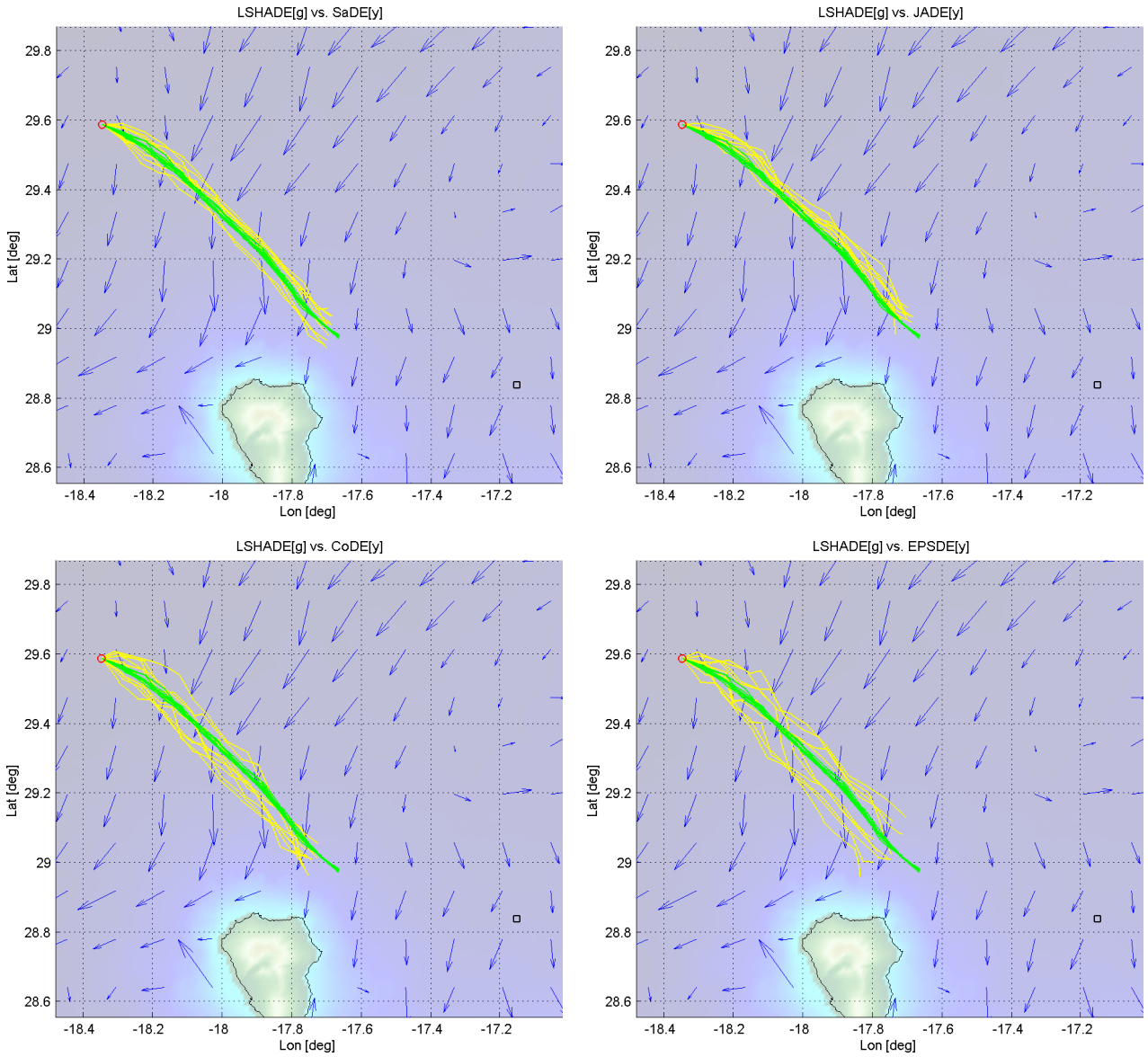

Figure 7: Phenotype path images obtained using selected algorithms, on scenario 1 (all 10 runs final trajectories drawn) – green: L-SHADE$_5$, yellow: compared algorithm (see particular image label).

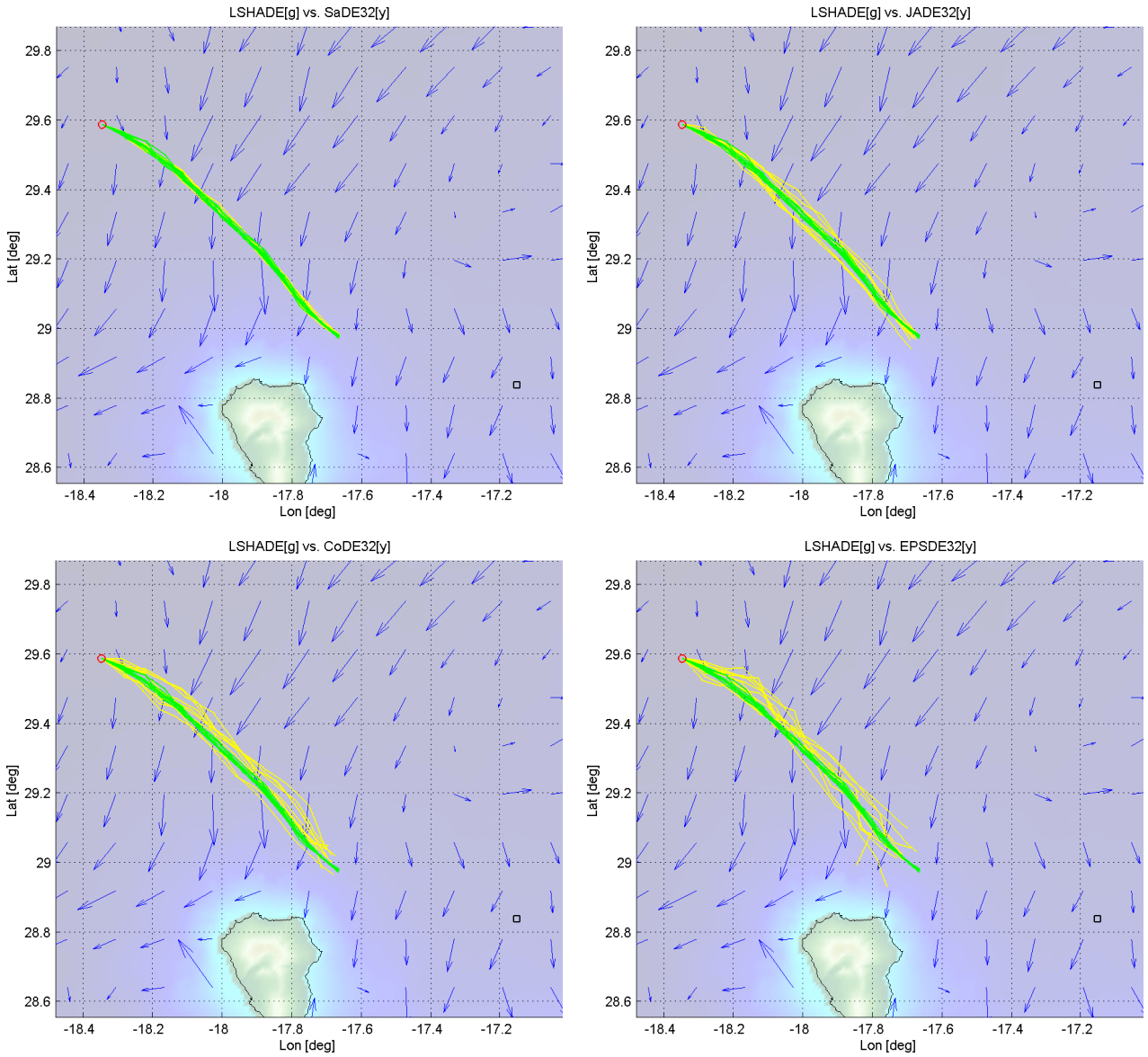

Figure 8: Phenotype path images obtained using selected algorithms, on scenario 1 (all 10 runs final trajectories drawn) – green: L-SHADE$_5$, yellow: compared algorithm (see particular image label).

Table 2: Obtained fitness with L-SHADE$_5$ and t-test comparisons with other algorithms on 12 scenarios.

Algorithm #	Scenario	Minimum	Median	Maximum	Average	Std. dev.
L-SHADE$_5$	T1	5.1617e+04	5.1854e+04	5.3084e+04	5.2083e+04	5.4742e+02
L-SHADE$_5$	T2	9.3023e+04	9.3687e+04	9.5817e+04	9.3948e+04	9.7431e+02
L-SHADE$_5$	T3	4.1874e+04	4.2058e+04	4.2848e+04	4.2120e+04	2.7070e+02
L-SHADE$_5$	T4	1.0016e+05	1.0045e+05	1.0083e+05	1.0043e+05	2.0395e+02
L-SHADE$_5$	T5	6.2204e+04	6.2637e+04	6.3908e+04	6.2917e+04	6.7038e+02
L-SHADE$_5$	T6	3.6222e+04	3.6423e+04	3.7226e+04	3.6498e+04	2.9590e+02
L-SHADE$_5$	T7	5.1484e+04	5.1538e+04	5.1574e+04	5.1532e+04	2.9726e+01
L-SHADE$_5$	T8	9.1233e+04	9.1685e+04	9.2116e+04	9.1684e+04	2.8596e+02
L-SHADE$_5$	T9	8.0872e+04	8.1268e+04	8.1629e+04	8.1273e+04	1.9728e+02
L-SHADE$_5$	T10	6.2485e+04	6.3042e+04	6.5683e+04	6.3267e+04	9.1905e+02
L-SHADE$_5$	T11	2.8155e+04	2.8589e+04	3.0551e+04	2.8753e+04	6.6710e+02
L-SHADE$_5$	T12	1.0172e+05	1.0204e+05	1.0314e+05	1.0213e+05	4.3015e+02
L-SHADE$_{18}$	T1	5.3808e+04	5.5668e+04	6.0550e+04	*5.6146e+04*‡	2.2169e+03
L-SHADE$_{18}$	T2	9.6051e+04	9.6978e+04	9.8803e+04	*9.7029e+04*‡	8.5721e+02
L-SHADE$_{18}$	T3	4.2906e+04	4.3359e+04	4.5823e+04	*4.3523e+04*‡	8.5129e+02
L-SHADE$_{18}$	T4	1.0135e+05	1.0313e+05	1.0593e+05	*1.0304e+05*‡	1.3841e+03
L-SHADE$_{18}$	T5	6.4259e+04	6.6344e+04	6.9575e+04	*6.6012e+04*‡	1.5108e+03
L-SHADE$_{18}$	T6	3.8045e+04	3.9440e+04	4.2078e+04	*3.9669e+04*‡	1.2561e+03
L-SHADE$_{18}$	T7	5.1628e+04	5.2251e+04	5.6006e+04	*5.2728e+04*‡	1.3677e+03
L-SHADE$_{18}$	T8	9.2644e+04	9.3813e+04	9.7611e+04	*9.4268e+04*‡	1.4696e+03
L-SHADE$_{18}$	T9	8.1780e+04	8.2440e+04	8.3764e+04	*8.2574e+04*‡	6.7935e+02
L-SHADE$_{18}$	T10	6.3507e+04	6.4911e+04	6.8082e+04	*6.5147e+04*‡	1.5691e+03
L-SHADE$_{18}$	T11	2.8993e+04	3.0790e+04	3.2320e+04	*3.0614e+04*‡	1.1329e+03
L-SHADE$_{18}$	T12	1.0320e+05	1.0432e+05	1.0624e+05	*1.0429e+05*‡	8.9543e+02
L-SHADE$_9$	T1	5.1787e+04	5.2368e+04	5.4790e+04	*5.2838e+04*‡	1.0774e+03
L-SHADE$_9$	T2	9.3436e+04	9.5318e+04	9.5902e+04	*9.4890e+04*‡	9.0426e+02
L-SHADE$_9$	T3	4.2202e+04	4.2320e+04	4.3080e+04	*4.2410e+04*‡	2.7356e+02
L-SHADE$_9$	T4	1.0021e+05	1.0102e+05	1.0202e+05	*1.0102e+05*‡	5.1986e+02
L-SHADE$_9$	T5	6.2435e+04	6.3798e+04	6.5895e+04	*6.3739e+04*‡	1.0809e+03
L-SHADE$_9$	T6	3.6607e+04	3.7201e+04	3.8075e+04	*3.7192e+04*‡	4.5789e+02
L-SHADE$_9$	T7	5.1504e+04	5.1575e+04	5.1897e+04	*5.1592e+04*‡	1.1462e+02
L-SHADE$_9$	T8	9.1369e+04	9.2228e+04	9.3103e+04	*9.2115e+04*‡	5.5955e+02
L-SHADE$_9$	T9	8.1255e+04	8.1589e+04	8.2511e+04	*8.1752e+04*‡	4.1417e+02
L-SHADE$_9$	T10	6.2931e+04	6.3571e+04	6.4702e+04	*6.3684e+04*‡	5.3175e+02
L-SHADE$_9$	T11	2.8006e+04	2.9920e+04	3.4163e+04	*2.9953e+04*‡	1.7946e+03
L-SHADE$_9$	T12	1.0178e+05	1.0240e+05	1.0316e+05	*1.0238e+05*‡	3.7907e+02
L-SHADE$_3$	T1	5.1545e+04	5.1883e+04	5.2636e+04	**5.1937e+04**†	3.3148e+02
L-SHADE$_3$	T2	9.2950e+04	9.3732e+04	1.0263e+05	*9.4518e+04*‡	2.9276e+03
L-SHADE$_3$	T3	4.1941e+04	4.2180e+04	4.2441e+04	4.2173e+04	1.6166e+02
L-SHADE$_3$	T4	1.0017e+05	1.0041e+05	1.0078e+05	1.0044e+05	2.0045e+02
L-SHADE$_3$	T5	6.2133e+04	6.3199e+04	6.4434e+04	*6.3225e+04*‡	7.6912e+02
L-SHADE$_3$	T6	3.6286e+04	3.6601e+04	3.7880e+04	*3.6709e+04*‡	4.4540e+02
L-SHADE$_3$	T7	5.1506e+04	5.1549e+04	5.1662e+04	*5.1550e+04*‡	4.3998e+01
L-SHADE$_3$	T8	9.1286e+04	9.1686e+04	9.2717e+04	9.1743e+04	4.4713e+02
L-SHADE$_3$	T9	8.0945e+04	8.1440e+04	8.1784e+04	*8.1370e+04*‡	2.5567e+02
L-SHADE$_3$	T10	6.2557e+04	6.3184e+04	6.8395e+04	*6.3866e+04*‡	1.8132e+03
L-SHADE$_3$	T11	2.8015e+04	2.9607e+04	3.2927e+04	*2.9692e+04*‡	1.5128e+03
L-SHADE$_3$	T12	1.0181e+05	1.0212e+05	1.0731e+05	*1.0264e+05*‡	1.6559e+03
best/1/bin$_{512-20-20}$	T1	5.2400e+04	5.2735e+04	5.4622e+04	*5.3070e+04*‡	7.0771e+02
best/1/bin$_{512-20-20}$	T2	9.2780e+04	9.3188e+04	9.3968e+04	**9.3214e+04**†	3.8284e+02
best/1/bin$_{512-20-20}$	T3	4.2095e+04	4.2290e+04	4.2466e+04	*4.2259e+04*‡	1.3440e+02
best/1/bin$_{512-20-20}$	T4	1.0043e+05	1.0074e+05	1.0172e+05	*1.0083e+05*‡	4.1239e+02
best/1/bin$_{512-20-20}$	T5	6.2032e+04	6.3218e+04	6.4577e+04	*6.3180e+04*‡	9.4324e+02
best/1/bin$_{512-20-20}$	T6	3.6804e+04	3.7457e+04	3.8050e+04	*3.7350e+04*‡	4.0086e+02
best/1/bin$_{512-20-20}$	T7	5.1491e+04	5.1563e+04	5.1784e+04	*5.1569e+04*‡	8.0100e+01
best/1/bin$_{512-20-20}$	T8	9.2529e+04	9.3022e+04	9.4041e+04	*9.3140e+04*‡	5.1479e+02
best/1/bin$_{512-20-20}$	T9	8.1141e+04	8.1614e+04	8.2593e+04	*8.1657e+04*‡	4.6893e+02
best/1/bin$_{512-20-20}$	T10	6.2363e+04	6.2727e+04	6.3519e+04	**6.2830e+04**†	3.5982e+02
best/1/bin$_{512-20-20}$	T11	2.8087e+04	2.8562e+04	2.9198e+04	**2.8559e+04**†	3.5099e+02
best/1/bin$_{512-20-20}$	T12	1.0203e+05	1.0239e+05	1.0479e+05	*1.0276e+05*‡	9.5257e+02
best/1/bin$_{256-15-20}$	T1	5.2273e+04	5.2697e+04	5.3155e+04	*5.2665e+04*‡	3.3036e+02
best/1/bin$_{256-15-20}$	T2	9.2362e+04	9.3749e+04	9.4815e+04	**9.3434e+04**†	9.4603e+02
best/1/bin$_{256-15-20}$	T3	4.1966e+04	4.2126e+04	4.2623e+04	*4.2185e+04*‡	2.0237e+02
best/1/bin$_{256-15-20}$	T4	1.0048e+05	1.0097e+05	1.0156e+05	*1.0095e+05*‡	4.1746e+02
best/1/bin$_{256-15-20}$	T5	6.1941e+04	6.2181e+04	6.3805e+04	**6.2387e+04**†	5.8488e+02
best/1/bin$_{256-15-20}$	T6	3.6802e+04	3.7087e+04	3.7561e+04	*3.7094e+04*‡	2.2014e+02
best/1/bin$_{256-15-20}$	T7	5.1501e+04	5.1542e+04	5.1689e+04	*5.1553e+04*‡	5.4434e+01
best/1/bin$_{256-15-20}$	T8	9.2368e+04	9.2817e+04	9.4012e+04	*9.3036e+04*‡	5.8131e+02
best/1/bin$_{256-15-20}$	T9	8.0866e+04	8.1482e+04	8.2545e+04	*8.1529e+04*‡	5.1202e+02
best/1/bin$_{256-15-20}$	T10	6.2143e+04	6.2542e+04	6.2901e+04	**6.2530e+04**†	2.6719e+02
best/1/bin$_{256-15-20}$	T11	2.8227e+04	2.8859e+04	2.9512e+04	2.8767e+04	4.4143e+02
best/1/bin$_{256-15-20}$	T12	1.0202e+05	1.0241e+05	1.0327e+05	*1.0251e+05*‡	3.9674e+02
best/1/bin$_{64-5-20}$	T1	5.2339e+04	5.2754e+04	5.3204e+04	*5.2741e+04*‡	2.6141e+02
best/1/bin$_{64-5-20}$	T2	9.2540e+04	9.3328e+04	9.5926e+04	**9.3518e+04**†	1.0054e+03
best/1/bin$_{64-5-20}$	T3	4.1971e+04	4.2181e+04	4.2399e+04	4.2164e+04	1.2693e+02
best/1/bin$_{64-5-20}$	T4	1.0045e+05	1.0067e+05	1.0162e+05	*1.0084e+05*‡	4.1677e+02
best/1/bin$_{64-5-20}$	T5	6.1881e+04	6.2236e+04	6.3155e+04	**6.2255e+04**†	4.0004e+02
best/1/bin$_{64-5-20}$	T6	3.6768e+04	3.7106e+04	3.8431e+04	*3.7315e+04*‡	5.5685e+02
best/1/bin$_{64-5-20}$	T7	5.1490e+04	5.1566e+04	5.1676e+04	*5.1558e+04*‡	6.0421e+01
best/1/bin$_{64-5-20}$	T8	9.2685e+04	9.3431e+04	9.5238e+04	*9.3507e+04*‡	8.5588e+02
best/1/bin$_{64-5-20}$	T9	8.1140e+04	8.1233e+04	8.1801e+04	8.1287e+04	2.0847e+02
best/1/bin$_{64-5-20}$	T10	6.2026e+04	6.2557e+04	6.3348e+04	**6.2572e+04**†	4.3162e+02
best/1/bin$_{64-5-20}$	T11	2.8007e+04	2.8726e+04	2.9526e+04	2.8769e+04	4.6938e+02
best/1/bin$_{64-5-20}$	T12	1.0200e+05	1.0231e+05	1.0359e+05	*1.0251e+05*‡	5.0376e+02

Table 3: Obtained fitness compared (L-SHADE$_5$ t-test) with some algorithms on 12 test scenarios.

Algorithm #	Scenario	Minimum	Median	Maximum	Average	Std. dev.
rand/1/bin	T1	5.8831e+04	6.1018e+04	6.4258e+04	*6.1294e+04*‡	1.7347e+03
rand/1/bin	T2	9.8231e+04	1.0270e+05	1.0511e+05	*1.0222e+05*‡	1.8090e+03
rand/1/bin	T3	4.5862e+04	4.8961e+04	5.0495e+04	*4.8498e+04*‡	1.5316e+03
rand/1/bin	T4	1.0651e+05	1.0795e+05	1.1140e+05	*1.0839e+05*‡	1.7144e+03
rand/1/bin	T5	7.0657e+04	7.3050e+04	7.7186e+04	*7.3061e+04*‡	1.9739e+03
rand/1/bin	T6	4.2263e+04	4.5323e+04	4.7401e+04	*4.5267e+04*‡	1.5267e+03
rand/1/bin	T7	5.4904e+04	5.6425e+04	5.7245e+04	*5.6349e+04*‡	7.2850e+02
rand/1/bin	T8	9.9129e+04	1.0141e+05	1.0256e+05	*1.0092e+05*‡	1.2345e+03
rand/1/bin	T9	8.3833e+04	8.7101e+04	8.9164e+04	*8.6861e+04*‡	1.7859e+03
rand/1/bin	T10	6.3967e+04	6.5943e+04	6.9026e+04	*6.5915e+04*‡	1.7161e+03
rand/1/bin	T11	3.2699e+04	3.5012e+04	3.5457e+04	*3.4517e+04*‡	9.0154e+02
rand/1/bin	T12	1.0814e+05	1.0994e+05	1.1227e+05	*1.1007e+05*‡	1.2225e+03
CLPSO	T1	5.7232e+04	6.2109e+04	6.4419e+04	*6.1071e+04*‡	2.6920e+03
CLPSO	T2	9.6935e+04	1.0157e+05	1.0365e+05	*1.0098e+05*‡	1.9939e+03
CLPSO	T3	4.6473e+04	4.7836e+04	5.1775e+04	*4.8515e+04*‡	1.5989e+03
CLPSO	T4	1.0484e+05	1.0836e+05	1.1079e+05	*1.0800e+05*‡	1.9187e+03
CLPSO	T5	6.8757e+04	7.2466e+04	7.7497e+04	*7.2236e+04*‡	2.5705e+03
CLPSO	T6	4.1681e+04	4.4102e+04	4.6482e+04	*4.4290e+04*‡	1.4631e+03
CLPSO	T7	5.3109e+04	5.5461e+04	5.8097e+04	*5.5711e+04*‡	1.5913e+03
CLPSO	T8	9.7071e+04	1.0079e+05	1.0385e+05	*1.0067e+05*‡	2.0795e+03
CLPSO	T9	8.5010e+04	8.7456e+04	8.9887e+04	*8.7397e+04*‡	1.3641e+03
CLPSO	T10	6.4016e+04	6.6144e+04	6.7642e+04	*6.5960e+04*‡	1.0831e+03
CLPSO	T11	3.1801e+04	3.3585e+04	3.7017e+04	*3.3684e+04*‡	1.4485e+03
CLPSO	T12	1.0745e+05	1.0963e+05	1.1420e+05	*1.0972e+05*‡	2.3920e+03
SaDE	T1	5.4947e+04	5.5725e+04	5.8389e+04	*5.5921e+04*‡	1.0151e+03
SaDE	T2	9.5087e+04	9.7193e+04	9.8731e+04	*9.6889e+04*‡	1.2208e+03
SaDE	T3	4.2774e+04	4.3591e+04	4.4935e+04	*4.3676e+04*‡	6.3626e+02
SaDE	T4	1.0234e+05	1.0315e+05	1.0455e+05	*1.0332e+05*‡	7.8232e+02
SaDE	T5	6.3888e+04	6.5289e+04	6.7138e+04	*6.5247e+04*‡	9.6098e+02
SaDE	T6	3.8324e+04	3.9772e+04	4.1079e+04	*3.9794e+04*‡	7.9325e+02
SaDE	T7	5.1941e+04	5.2682e+04	5.3980e+04	*5.2612e+04*‡	6.9953e+02
SaDE	T8	9.3978e+04	9.6990e+04	9.8012e+04	*9.6581e+04*‡	1.1997e+03
SaDE	T9	8.1819e+04	8.3310e+04	8.4457e+04	*8.3095e+04*‡	8.5266e+02
SaDE	T10	6.3057e+04	6.3945e+04	6.4641e+04	*6.3804e+04*‡	5.8987e+02
SaDE	T11	2.9328e+04	3.0610e+04	3.1974e+04	*3.0479e+04*‡	9.6880e+02
SaDE	T12	1.0379e+05	1.0592e+05	1.0673e+05	*1.0556e+05*‡	8.5235e+02
JADE	T1	5.5987e+04	5.8393e+04	5.9150e+04	*5.7878e+04*‡	1.1077e+03
JADE	T2	9.5806e+04	9.6537e+04	9.8382e+04	*9.6828e+04*‡	8.8847e+02
JADE	T3	4.3881e+04	4.4259e+04	4.5367e+04	*4.4379e+04*‡	5.0169e+02
JADE	T4	1.0298e+05	1.0408e+05	1.0581e+05	*1.0420e+05*‡	8.5028e+02
JADE	T5	6.5259e+04	6.6603e+04	6.8610e+04	*6.6665e+04*‡	1.1297e+03
JADE	T6	3.9359e+04	4.0427e+04	4.2217e+04	*4.0637e+04*‡	9.6144e+02
JADE	T7	5.1943e+04	5.3136e+04	5.4355e+04	*5.3162e+04*‡	7.2664e+02
JADE	T8	9.5308e+04	9.6926e+04	9.8139e+04	*9.6630e+04*‡	8.3405e+02
JADE	T9	8.2269e+04	8.2581e+04	8.4552e+04	*8.2802e+04*‡	7.2030e+02
JADE	T10	6.3681e+04	6.4120e+04	6.4356e+04	*6.4033e+04*‡	2.2212e+02
JADE	T11	3.0104e+04	3.1279e+04	3.2946e+04	*3.1424e+04*‡	7.8868e+02
JADE	T12	1.0453e+05	1.0649e+05	1.0749e+05	*1.0615e+05*‡	9.7355e+02
EPSDE	T1	6.0905e+04	6.5217e+04	6.8258e+04	*6.3989e+04*‡	2.8396e+03
EPSDE	T2	9.7320e+04	1.0333e+05	1.0543e+05	*1.0277e+05*‡	2.2329e+03
EPSDE	T3	4.7843e+04	5.0883e+04	5.3432e+04	*5.0448e+04*‡	2.1938e+03
EPSDE	T4	1.0739e+05	1.1202e+05	1.1314e+05	*1.1133e+05*‡	1.6578e+03
EPSDE	T5	6.9069e+04	7.4531e+04	7.9716e+04	*7.4802e+04*‡	3.4634e+03
EPSDE	T6	4.2304e+04	4.5866e+04	4.7694e+04	*4.5684e+04*‡	1.8281e+03
EPSDE	T7	5.5255e+04	5.8490e+04	6.0669e+04	*5.8345e+04*‡	1.9964e+03
EPSDE	T8	9.6857e+04	1.0226e+05	1.0494e+05	*1.0187e+05*‡	2.3831e+03
EPSDE	T9	8.3116e+04	8.7369e+04	9.0239e+04	*8.7665e+04*‡	2.1060e+03
EPSDE	T10	6.4028e+04	7.0737e+04	7.6787e+04	*7.0688e+04*‡	4.0223e+03
EPSDE	T11	3.2014e+04	3.5967e+04	3.7392e+04	*3.5823e+04*‡	1.6716e+03
EPSDE	T12	1.0657e+05	1.1192e+05	1.1402e+05	*1.1115e+05*‡	2.5062e+03
CoDE	T1	5.9325e+04	6.0919e+04	6.3147e+04	*6.1310e+04*‡	1.2180e+03
CoDE	T2	9.8561e+04	1.0162e+05	1.0439e+05	*1.0138e+05*‡	1.5753e+03
CoDE	T3	4.5395e+04	4.7075e+04	5.0039e+04	*4.7206e+04*‡	1.4416e+03
CoDE	T4	1.0569e+05	1.0857e+05	1.1008e+05	*1.0804e+05*‡	1.6739e+03
CoDE	T5	6.6428e+04	7.1226e+04	7.1961e+04	*7.0599e+04*‡	1.6482e+03
CoDE	T6	4.2389e+04	4.4381e+04	4.6385e+04	*4.4301e+04*‡	1.2695e+03
CoDE	T7	5.4594e+04	5.6798e+04	5.8300e+04	*5.6271e+04*‡	1.2234e+03
CoDE	T8	9.9512e+04	1.0107e+05	1.0152e+05	*1.0069e+05*‡	7.2547e+02
CoDE	T9	8.4345e+04	8.5851e+04	9.0992e+04	*8.6459e+04*‡	2.1479e+03
CoDE	T10	6.4150e+04	6.5314e+04	6.6253e+04	*6.5335e+04*‡	7.3808e+02
CoDE	T11	3.1772e+04	3.3348e+04	3.5563e+04	*3.3570e+04*‡	1.2003e+03
CoDE	T12	1.0895e+05	1.1017e+05	1.1088e+05	*1.0985e+05*‡	6.8247e+02
CMAES	T1	5.2250e+04	5.2591e+04	7.5940e+04	*5.4845e+04*‡	7.4149e+03
CMAES	T2	9.2667e+04	9.3414e+04	1.1655e+05	*1.0021e+05*‡	9.6166e+03
CMAES	T3	4.1969e+04	4.2052e+04	6.3968e+04	*4.4223e+04*‡	6.9378e+03
CMAES	T4	1.0036e+05	1.0050e+05	1.1698e+05	*1.0226e+05*‡	5.1846e+03
CMAES	T5	6.1743e+04	6.3118e+04	8.5491e+04	*6.5035e+04*‡	7.2589e+03
CMAES	T6	3.6878e+04	3.7059e+04	5.3065e+04	*3.8803e+04*‡	5.0435e+03
CMAES	T7	5.1501e+04	5.1607e+04	6.6530e+04	*5.4507e+04*‡	6.1116e+03
CMAES	T8	9.2469e+04	9.3005e+04	1.4588e+05	*9.9456e+04*‡	1.6580e+04
CMAES	T9	8.0903e+04	8.1468e+04	1.0408e+05	*8.6917e+04*‡	9.6706e+03
CMAES	T10	6.2405e+04	6.3051e+04	9.1605e+04	*6.8078e+04*‡	9.2774e+03
CMAES	T11	2.8185e+04	2.8577e+04	4.2253e+04	*3.0935e+04*‡	5.0722e+03
CMAES	T12	1.0202e+05	1.0216e+05	1.2844e+05	*1.0489e+05*‡	8.2861e+03

Table 4: Average Rankings of the algorithms.

Algorithm	Ranking
L-SHADE$_5$	6.9
SaDE$_{32}$	8.5
L-SHADE$_3$	8.8
best/1/bin$_{64-5-20}$	8.9
best/1/bin$_{256-15-20}$	9.1
best/1/bin$_{64}$	9.7
best/1/bin$_{32}$	9.8
best/1/bin$_{512-20-20}$	10.3
L-SHADE$_9$	12.7
best/1/bin	12.9
rand/1/bin$_{32-20-10}$	12.9
best/1/bin$_{128}$	14.6
JADE$_{32}$	15.2
CMAES	16.6
CMAES$_{128}$	16.6
CMAES$_{256}$	16.6
CMAES$_{32}$	16.6
CMAES$_{512}$	16.6
CMAES$_{64}$	16.6
SaDE$_{64}$	19.4
best/1/bin$_{256}$	20.4
JADE$_{64}$	23.0
L-SHADE$_{18}$	23.7
SaDE	23.9
rand/1/bin$_{32}$	24.7
best/1/bin$_{512}$	25.3
CoDE$_{32}$	25.4
SaDE$_{128}$	26.1
JADE	26.6
JADE$_{128}$	28.2
JADE$_{256}$	32.4
SaDE$_{256}$	34.6
rand/1/bin$_{256-15-40}$	35.4
CoDE$_{64}$	35.6
EPSDE$_{32}$	36.4
SaDE$_{512}$	36.5
JADE$_{512}$	36.6
rand/1/bin$_{64}$	36.6
CoDE	40.7
CLPSO	41.6
CLPSO$_{128}$	41.6
CLPSO$_{32}$	41.6
CLPSO$_{512}$	41.6
CLPSO$_{64}$	41.6
rand/1/bin	43.1
CoDE$_{128}$	43.2
CoDE$_{256}$	44.2
rand/1/bin$_{128}$	44.2
EPSDE$_{64}$	45.9
CoDE$_{512}$	47.0
EPSDE$_{512}$	47.1
rand/1/bin$_{512}$	47.1
rand/1/bin$_{256}$	47.6
EPSDE	48.1
EPSDE$_{128}$	49.4
EPSDE$_{256}$	49.6

Table 5: Post-hoc procedures over Friedman statistics (reference best ranking R_0 algorithm is L-SHADE$_5$).

Rank (i)	Algorithm	$z = (R_0 - R_i)/SE$	p_{val}	H/H/H	Holland	Rom	Finner	Li
(0)	(**L-SHADE$_5$**)							
1	SaDE$_{32}$	0.752	0.452	0.05	0.05	0.05	0.05	0.05
2	L-SHADE$_3$	0.934	0.35	0.025	0.0253	0.025	0.0491	0.0288
3	best/1/bin$_{64-5-20}$	0.974	0.33	0.0167	0.017	0.0167	0.0482	0.0288
4	best/1/bin$_{256-15-20}$	1.05	0.292	0.0125	0.0127	0.0131	0.0473	0.0288
5	best/1/bin$_{64}$	1.37	0.172	0.01	0.0102	0.0105	0.0464	0.0288
6	best/1/bin$_{32}$	1.42	0.157	0.00833	0.00851	0.00876	0.0456	0.0288
7	best/1/bin$_{512-20-20}$	1.63	0.104	**0.00714**	**0.0073**	0.00751	0.0447	0.0288
8	L-SHADE$_9$	2.79	**0.00527**	**0.00625**	**0.00639**	**0.00657**	0.0438	0.0288
9	best/1/bin	2.85	**0.00438**	**0.00556**	**0.00568**	**0.00584**	0.0429	0.0288
10	rand/1/bin$_{32-20-10}$	2.88	**0.00396**	**0.005**	**0.00512**	**0.00526**	0.042	0.0288
11	best/1/bin$_{128}$	3.69	**0.000222**	**0.00455**	**0.00465**	**0.00478**	0.0411	0.0288
12	JADE$_{32}$	3.98	**6.96e-05**	**0.00417**	**0.00427**	**0.00438**	0.0402	0.0288
13	CMAES	4.61	**4.09e-06**	**0.00385**	**0.00394**	**0.00405**	0.0393	0.0288
14	CMAES$_{32}$	4.61	**4.09e-06**	**0.00357**	**0.00366**	**0.00376**	0.0384	0.0288
15	CMAES$_{64}$	4.61	**4.09e-06**	**0.00333**	**0.00341**	**0.00351**	0.0375	0.0288
16	CMAES$_{128}$	4.61	**4.09e-06**	**0.00313**	**0.0032**	**0.00329**	0.0366	0.0288
17	CMAES$_{256}$	4.61	**4.09e-06**	**0.00294**	**0.00301**	**0.00309**	0.0357	0.0288
18	CMAES$_{512}$	4.61	**4.09e-06**	**0.00278**	**0.00285**	**0.00292**	0.0348	0.0288
19	SaDE$_{64}$	5.94	**2.91e-09**	**0.00263**	**0.0027**	**0.00277**	0.0339	0.0288
20	best/1/bin$_{256}$	6.43	**1.26e-10**	**0.0025**	**0.00256**	**0.00263**	0.033	0.0288
21	JADE$_{64}$	7.65	**1.94e-14**	**0.00238**	**0.00244**	**0.0025**	0.0321	0.0288
22	L-SHADE$_{18}$	7.99	**1.3e-15**	**0.00227**	**0.00233**	**0.00239**	0.0312	0.0288
23	SaDE	8.09	**6.18e-16**	**0.00217**	**0.00223**	**0.00229**	0.0303	0.0288
24	rand/1/bin$_{32}$	8.45	**2.82e-17**	**0.00208**	**0.00213**	**0.00219**	0.0294	0.0288
25	best/1/bin$_{512}$	8.75	**2.12e-18**	**0.002**	**0.00205**	**0.0021**	0.0285	0.0288
26	CoDE$_{32}$	8.78	**1.6e-18**	**0.00192**	**0.00197**	**0.00202**	0.0276	0.0288
27	SaDE$_{128}$	9.13	**7.06e-20**	**0.00185**	**0.0019**	**0.00195**	0.0267	0.0288
28	JADE	9.36	**7.96e-21**	**0.00179**	**0.00183**	**0.00188**	0.0258	0.0288
29	JADE$_{128}$	10.1	**4.88e-24**	**0.00172**	**0.00177**	**0.00181**	0.0249	0.0288
30	JADE$_{256}$	12.1	**6.44e-34**	**0.00167**	**0.00171**	**0.00175**	0.024	0.0288
31	SaDE$_{256}$	13.2	**1.01e-39**	**0.00161**	**0.00165**	**0.0017**	0.023	0.0288
32	rand/1/bin$_{256-15-40}$	13.6	**5.62e-42**	**0.00156**	**0.0016**	**0.00164**	0.0221	0.0288
33	CoDE$_{64}$	13.6	**2.63e-42**	**0.00152**	**0.00155**	**0.00159**	0.0212	0.0288
34	EPSDE$_{32}$	14.1	**7.68e-45**	**0.00147**	**0.00151**	**0.00155**	0.0203	0.0288
35	SaDE$_{512}$	14.1	**3.92e-45**	**0.00143**	**0.00146**	**0.0015**	0.0194	0.0288
36	rand/1/bin$_{64}$	14.1	**2.8e-45**	**0.00139**	**0.00142**	**0.00146**	0.0185	0.0288
37	JADE$_{512}$	14.1	**2.06e-45**	**0.00135**	**0.00139**	**0.00142**	0.0176	0.0288
38	CoDE	16	**5.82e-58**	**0.00132**	**0.00135**	**0.00138**	0.0166	0.0288
39	CLPSO	16.5	**3.67e-61**	**0.00128**	**0.00131**	**0.00135**	0.0157	0.0288
40	CLPSO$_{32}$	16.5	**3.67e-61**	**0.00125**	**0.00128**	**0.00132**	0.0148	0.0288
41	CLPSO$_{64}$	16.5	**3.67e-61**	**0.00122**	**0.00125**	**0.00128**	0.0139	0.0288
42	CLPSO$_{128}$	16.5	**3.67e-61**	**0.00119**	**0.00122**	**0.00125**	0.013	0.0288
43	CLPSO$_{512}$	16.5	**3.67e-61**	**0.00116**	**0.00119**	**0.00122**	0.0121	0.0288
44	rand/1/bin	17.2	**2.91e-66**	**0.00114**	**0.00117**	**0.0012**	0.0111	0.0288
45	CoDE$_{128}$	17.3	**7.41e-67**	**0.00111**	**0.00114**	**0.00117**	0.0102	0.0288
46	CoDE$_{256}$	17.7	**2.6e-70**	**0.00109**	**0.00111**	**0.00114**	0.00928	0.0288
47	rand/1/bin$_{128}$	17.7	**2.03e-70**	**0.00106**	**0.00109**	**0.00112**	0.00836	0.0288
48	EPSDE$_{64}$	18.6	**8.13e-77**	**0.00104**	**0.00107**	**0.0011**	0.00743	0.0288
49	CoDE$_{512}$	19.1	**3.81e-81**	**0.00102**	**0.00105**	**0.00107**	**0.00651**	0.0288
50	rand/1/bin$_{512}$	19.1	**2.33e-81**	**0.001**	**0.00103**	**0.00105**	**0.00558**	0.0288
51	EPSDE$_{512}$	19.1	**1.37e-81**	**0.00098**	**0.00101**	**0.00103**	**0.00465**	0.0288
52	rand/1/bin$_{256}$	19.3	**2.22e-83**	**0.000962**	**0.000986**	**0.00101**	**0.00372**	0.0288
53	EPSDE	19.6	**2.24e-85**	**0.000943**	**0.000967**	**0.000993**	**0.00279**	0.0288
54	EPSDE$_{128}$	20.2	**1.23e-90**	**0.000926**	**0.000949**	**0.000974**	**0.00186**	0.0288
55	EPSDE$_{256}$	20.3	**1.02e-91**	**0.000909**	**0.000932**	**0.000956**	**0.000932**	0.0288

For the SaDE with $NP = 32$ in Table 5, the post-hoc procedures are not significant, but on the ranking it is still worse than L-SHADE$_5$. This shows that we have also attained a well performing SaDE version that is competitive with our proposed algorithm L-SHADE$_5$, while the original version with $NP = 100$ (Zamuda & Hernández Sosa, 2014) was seen to be outperformed by L-SHADE$_5$ on all scenarios in Table 3. Specifically, when comparing pairs of new instances of configured algorithms and results reported in literature before, these are also the new improvements that are seen from Table 5: L-SHADE$_5$ improves with ranking over the previously best algorithms best/1/bin (without population resizing) and best/1/bin$_{64-5-20}$ (with population size reduction), SaDE$_{32}$ over SaDE, JADE32 over JADE, CoDE32 over CoDe, EPSDE32 over EPSDE, best/1/bin$_{64\text{-}5\text{-}20}$ over best/1/bin and, rand/1/bin$_{32\text{-}20\text{-}10}$ over rand/1/bin (as population size reduction versus fixed sizing jDE-based approaches).

As is evident from the findings listed in this section, we have shown that our proposed method (L-SHADE$_5$ for UGPP, outlined in Figure 2), clearly yields benefit to the identified future prospects in UGPP (Lermusiaux et al., 2017) as explained in the Introduction, by new empirical assessment, and even much more, by improving on the recognitions of applying evolutionary algorithms which now further overcome the computational cost (not only of classic deterministic methods as argued in (Zamuda & Hernández Sosa, 2014), and, once more, the best state-of-the-art existing evolutionary UGPP algorithms, as demonstrated in Table 5). By improving the computational cost due to new faster convergences (as demonstrated in Figures 3–4) and the improving trajectories qualities for missions, this yields new capabilites which mission planners will utilize.

Specifically explaining the contributions overall, it is that the selection of the right path planning algorithm version for the expert system is of high importance when dealing with complex and potentially risky situations as during recently developed real-missions – e.g., strong currents were present in the vehicle recovery area near the shoreline. Intensive simulation and optimization procedures are required to provide as much evidence as possible for the decision-making, and slight differences in the proposed path could, therefore, result in significant trajectory alterations.

5. Conclusions

This paper proposed underwater glider path planning mission scenarios optimization with a recently well performing evolutionary algorithm for continuous numerical optimization, Success-History Based Adaptive Differential Evolution Algorithm (SHADE) including Linear population size reduction (L-SHADE). An algorithm for path optimization considering the ocean currents model predictions, vessel dynamics, and limited communication, yielded potential way-points for the vessel based on the most probable scenario; this is specially useful in oceanic engineering for short term opportunistic missions where no reactive control is possible.

Also, the applicability of several configured algorithms has been compared to, analysed, and further improved by configuration. Through the depicted fitness convergence graphs, final obtained fitness plots, trajectories drawn, and per-scenario analysis, it was shown that the new proposed algorithm yielded stable and competitive output trajectories. As was further seen from statistical analysis, the proposed algorithm has outperformed all other previous results from literature, and ranked first among the existing and newly configured algorithms for UGPP. Moreover, some configured algorithms were shown to outperform

their previous corresponding versions, improved by changing the population size for UGPP evolution.

As the real-world implementation of underwater glider path planning over dynamic and changing environment in deep ocean waters requires complex mission planning under very high uncertainties, this paper contributes to improvements in the scheduling and planning of real UGPP missions undertaken using such methods. UGGP missions are also influenced to a large extent by the remote sensing for forecasting weather models' outcomes used to predict spatial currents in deep sea, further limiting the available time for accurate run-time decisions by the pilot, who need to re-test several possible mission scenarios in a short time, usually a few minutes. In this respect, performance improvements with L-SHADE over previous UGPP applications would contribute to increased opportunity for mission scenario re-tests or in very hard scenarios, to improve with practical significance upon suggested trajectories during the decision-making process.

We have had the opportunity of testing our system during real ocean missions. Currently, only pilot recommendation mode is allowed, so glider commands need to be validated before sending them to the vehicle. The benefits of the proposal are more evident in highly variable short-term scenarios, were dynamic ocean states need to be evaluated in a few minutes. The quality of the actual final result is obviously conditioned by the forecast accuracy, but this is also a limitation for a human expert based operation.

Future work includes further applications of L-SHADE and combinations of mechanisms for UGPP, as well as adding newly appearing optimization algorithms for UGPP to increase the advisory intelligence of the UGPP expert system and training more user pilots. As already pointed out in our previous reserach, there are many research lines still explored, and to list just a few, these autonomous piloting of glider swarms, varying of bearings definition, multi-objective optimization of glider scenarios, and additional applications for interactive glider mission decision maker expert support, like changing of mission priorities during mission planning.

Acknowledgement

The work was supported in part by the Slovenian Research Agency, under Research Programme P2-0041. This work was also supported by the Canary Islands government and FEDER funds under Project 2010/62. This article is also based upon work from COST Action CA15140 'Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice (ImAppNIO) and COST Action IC1406 'High-Performance Modelling and Simulation for Big Data Applications (cHiPSet)', supported by COST (European Cooperation in Science and Technology). The high cost of the real missions would make it difficult to perform extensive experimentation without a big budget, so, in this sense, we want to thank ULPGC's SITMA service and Rutgers University for the valuable support received during the execution of the glider mission. Part of the codes in Matlab for extending the optimization algorithms utilized are provided by Qingfu Zhang at `http://dces.essex.ac.uk/staff/qzhang/code/` and by Ryoji Tanabe at `https://sites.google.com/site/tanaberyoji/home`. The high cost of the real missions would make it difficult to perform extensive experimentation without a big budget, so in this sense we sincerely appreciate the collaboration facilities offered by Pablo Sangrá (PI of the PUMP project) and Rui Caldeira (Head of OOM). We also thank the editors and anonymous reviewers for their valuable time, comments, and suggestions.

References

Abbass, H. A. (2002). The self-adaptive Pareto differential evolution algorithm. In *2002 Congress on Evolutionary Computation, 2002* (pp. 831–836). volume 1.

Alvarez, A., Caiti, A., & Onken, R. (2004). Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE Journal of Oceanic Engineering*, *29*, 418–429.

Baig, M. Z., Aslam, N., Shum, H. P., & Zhang, L. (2017). Differential evolution algorithm as a tool for optimal feature subset selection in motor imagery eeg. *Expert Systems with Applications*, *90*, 184–195.

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, *237*, 82–117.

Brest, J., Greiner, S., Bošković, B., Mernik, M., & Žumer, V. (2006). Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. *IEEE Transactions on Evolutionary Computation*, *10*, 646–657.

Brest, J., & Maučec, M. S. (2008). Population Size Reduction for the Differential Evolution Algorithm. *Applied Intelligence*, *29*, 228–247.

Cabrera Gámez, J., Isern González, J., Hernández Sosa, J. D., Domínguez Brito, A. C., & Fernández Perdomo, E. (2013). Optimization-based weather routing for sailboats. In *Robotic Sailing 2012: Proceedings of the 5th International Robotic Sailing Conference* (pp. 23–33). Springer.

Cao, J., Cao, J., Zeng, Z., Yao, B., & Lian, L. (2017). Toward optimal rendezvous of multiple underwater gliders: 3D path planning with combined sawtooth and spiral motion. *Journal of Intelligent & Robotic Systems*, *85*, 189–206.

Cashmore, M., Fox, M., Long, D., Magazzeni, D., & Ridder, B. (2017). Opportunistic planning in autonomous underwater missions. *IEEE Transactions on Automation Science and Engineering*, . DOI: 10.1109/TASE.2016.2636662.

Crawford, W. R., Brickley, P. J., Peterson, T. D., & Thomas, A. C. (2005). Impact of Haida Eddies on chlorophyll distribution in the Eastern Gulf of Alaska. *Deep Sea Research Part II: Topical Studies in Oceanography*, *52*, 975–989.

Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray.

Das, S., Maity, S., Qu, B.-Y., & Suganthan, P. N. (2011). Real-parameter evolutionary multimodal optimization – A survey of the state-of-the-art. *Swarm and Evolutionary Computation*, *1*, 71–88.

Das, S., Mullick, S. S., & Suganthan, P. (2016). Recent advances in differential evolution – An updated survey. *Swarm and Evolutionary Computation*, *27*, 1–30.

Das, S., & Suganthan, P. N. (2011). Differential Evolution: A Survey of the State-of-the-art. *IEEE Transactions on Evolutionary Computation*, *15*, 4–31.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, *7*, 1–30.

Eiben, A. E., & Smith, J. E. (2003). *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer.

Eichhorn, M. (2015). Optimal routing strategies for autonomous underwater vehicles in time-varying environment. *Robotics and Autonomous Systems*, *67*, 33–43. Advances in Autonomous Underwater Robotics.

Ellefsen, K. O., Lepikson, H. A., & Albiez, J. C. (2016). Planning inspection paths through evolutionary multi-objective optimization. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference* (pp. 893–900). ACM.

Garau, B., Alvarez, A., & Oliver, G. (2005). Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an A* approach. In *2005 IEEE International Conference on Robotics and Automation* (pp. 194–198). IEEE.

Glotić, A., & Zamuda, A. (1 March 2015). Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution. *Applied Energy*, *141*, 42–56.

Hamann, H., Khaluf, Y., Botev, J., Soorati, M. D., Ferrante, E., Kosak, O., Montanier, J.-M., Mostaghim, S., Redpath, R., Timmis, J., Veenstra, F., Wahby, M., & Zamuda, A. (2016). Hybrid Societies: Challenges and Perspectives in the Design of Collective Behavior in Self-organizing Systems. *Frontiers in Robotics and AI*, *3*, 1–8.

Hansen, N., & Ostermeier, A. (2001). Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, *9*, 159–195.

Hátún, H., Eriksen, C. C., & Rhines, P. B. (2007). Buoyant eddies entering the Labrador Sea observed with gliders and altimetry. *Journal of Physical Oceanography*, *37*, 2838–2854.

Hernández Sosa, J. D., Smith, R. N., Fernández Perdomo, E., Isern González, J., Cabrera Gámez, J., Domínguez Brito, A. C., & Prieto Marañón, V. (2013). Glider path-planning for optimal sampling of mesocale eddies. In *Proceedings of the 14th International Conference on Computer Aided System Theory* (pp. 321–325). University of Las Palmas de Gran Canaria, Canary Islands, Spain.

Inanc, T., Shadden, S. C., & Marsden, J. E. (2004). Optimal trajectory generation in ocean flows. In *Proceedings of the American Control Conference* (pp. 674–679). Portland, OR, USA.

Kularatne, D., Smith, R. N., & Hsieh, M. A. (2015). Zig-zag wanderer: Towards adaptive tracking of time-varying coherent structures in the ocean. In *2015 IEEE International Conference on Robotics and Automation* (pp. 3253–3258).

Lagarias, J. C., Reeds, J. A., Wright, M. H., & Wright, P. E. (1998). Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM Journal on Optimization*, *9*, 112–147.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436.

Leonard, N. E., Paley, D. A., Davis, R. E., Fratantoni, D. M., Lekien, F., & Zhang, F. (2010). Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. *Journal of Field Robotics*, *27*, 718–740.

Lermusiaux, P., Subramani, D., Lin, J., Kulkarni, C., Gupta, A., Dutt, A., Lolla, T., Haley, P., Ali, W., Mirabito, C., & Jana, S. (2017). A future for intelligent autonomous ocean observing systems. *Journal of Marine Research*, *75*, 765–813.

Lermusiaux, P. F. J., Lolla, T., Haley Jr., P. J., Yigit, K., Ueckermann, M. P., Sondergaard, T., & Leslie, W. G. (2016). Science of autonomy: Time-optimal path planning and adaptive sampling for swarms of ocean vehicles. In T. Curtin (Ed.), *Autonomous Ocean Vehicles, Subsystems and Control* Handbook of Ocean Engineering chapter 11. (pp. 481–498). Springer.

Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, *10*, 281–295.

Lolla, T., Haley Jr., P. J., & Lermusiaux, P. F. J. (2014). Time-optimal path planning in dynamic flows using level set equations: realistic applications. *Ocean Dynamics*, *64*, 1399–1417.

Mallipeddi, R., Suganthan, P. N., Pan, Q. K., & Tasgetiren, M. F. (2011). Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing*, *11*, 1679–1696.

Mlakar, U., Fister, I., Brest, J., & Potočnik, B. (2017). Multi-objective differential evolution for feature selection in facial expression recognition systems. *Expert Systems with Applications*, *89*, 129–137.

Moura, A., Rijo, R., Silva, P., & Crespo, S. (2010). A multi-objective genetic algorithm applied to autonomous underwater vehicles for sewage outfall plume dispersion observations. *Applied Soft Computing*, *10*, 1119–1126.

Neri, F., & Tirronen, V. (2010). Recent Advances in Differential Evolution: A Survey and Experimental Analysis. *Artificial Intelligence Review*, *33*, 61–106.

Opara, K. R., & Arabas, J. (2018). Differential Evolution: A survey of theoretical analyses. *Swarm and Evolutionary Computation*, . . DOI: https://doi.org/10.1016/j.swevo.2018.06.010.

Parouha, R. P., & Das, K. N. (2016). Dpd: An intelligent parallel hybrid algorithm for economic load dispatch problems with various practical constraints. *Expert Systems with Applications*, *63*, 295–309.

Piotrowski, A. P. (2017). Review of differential evolution population size. *Swarm and Evolutionary Computation*, *32*, 1–24.

Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, *13*, 398–417.

Sangrà, P., Auladell, M., Marrero-Díaz, A., Pelegrí, J. L., Fraile-Nuez, E., Rodríguez-Santana, A., Martín, J. M., Mason, E., & Hernández-Guerra, A. (2007). On the nature of oceanic eddies shed by the island of gran canaria. *Deep Sea Research Part I: Oceanographic Research Papers*, *54*, 687–709.

Smith, R. N., Chao, Y., Li, P. P., Caron, D. A., Jones, B. H., & Sukhatme, G. S. (2010). Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model. *The International Journal of Robotics Research*, *29*, 1475–1497.

Sörensen, K. (2015). Metaheuristicsthe metaphor exposed. *International Transactions in Operational Research*, *22*, 3–18.

Sousselier, T., Dreo, J., & Sevaux, M. (2015). Line formation algorithm in a swarm of reactive robots constrained by underwater environment. *Expert Systems with Applications*, *42*, 5117–5127.

Storn, R., & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, *11*, 341–359.

Tanabe, R., & Fukunaga, A. (2013). Evaluating the performance of SHADE on CEC 2013 benchmark problems. In *2013 IEEE Congress on Evolutionary Computation* (pp. 1952–1959). IEEE.

Tanabe, R., & Fukunaga, A. S. (2014). Improving the search performance of SHADE using linear population size reduction. In *2014 IEEE Congress on Evolutionary Computation* (pp. 1658–1665). IEEE.

Thompson, D. R., Chien, S., Chao, Y., Li, P., Cahill, B., Levin, J., Schofield, O., Balasuriya, A., Petillo, S., Arrott, M., & Meisinger, M. (2010). Spatiotemporal path planning in strong, dynamic, uncertain currents. In *2010 IEEE International Conference on Robotics and Automation* (pp. 4778–4783). IEEE.

Wang, Y., Cai, Z., & Zhang, Q. (2011). Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation*, *15*, 55–66.

Wang, Y., Zhang, Y., Zhang, M., Yang, Z., & Wu, Z. (2017). Design and flight performance of hybrid underwater glider with controllable wings. *International Journal of Advanced Robotic Systems*, *14*, 1729881417703566.

Yoon, S.-M. (2014). *Trajectory design of an underwater glider considering depth constraints*. Ph.D. thesis.

Zadeh, S. M., Powers, D. M., & Sammut, K. (2017). An autonomous reactive architecture for efficient auv mission time management in realistic dynamic ocean environment. *Robotics and Autonomous Systems*, *87*, 81–103.

Zaharie, D. (2009). Influence of crossover on the behavior of Differential Evolution Algorithms. *Applied Soft Computing*, *9*, 1126–1138.

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

Zamuda, A., & Brest, J. (2014). Vectorized procedural models for animated trees reconstruction using differential evolution. *Information Sciences*, *278*, 1–21.

Zamuda, A., & Brest, J. (2015). Self-adaptive control parameters' randomization frequency and propagations in differential evolution. *Swarm and Evolutionary Computation*, *25*, 72–99.

Zamuda, A., Brest, J., Bošković, B., & Žumer, V. (2011). Differential Evolution for Parameterized Procedural Woody Plant Models Reconstruction. *Applied Soft Computing*, *11*, 4904–4912.

Zamuda, A., & Hernández Sosa, J. D. (2014). Differential Evolution and Underwater Glider Path Planning Applied to the Short-Term Opportunistic Sampling of Dynamic Mesoscale Ocean Structures. *Applied Soft Computing*, *24*, 95–108.

Zamuda, A., & Hernández Sosa, J. D. (2015). Underwater glider path planning and population size reduction in differential evolution. In R. Moreno-Daz, F. Pichler, & A. Quesada-Arencibia (Eds.), *Computer Aided Systems Theory – EUROCAST 2015* (pp. 853–860). Springer International Publishing volume 9520 of *Lecture Notes in Computer Science*.

Zamuda, A., Sosa, J. D. H., & Adler, L. (2016a). Constrained Differential Evolution Optimization for Underwater Glider Path Planning in Sub-mesoscale Eddy Sampling. *Applied Soft Computing*, *42*, 93–118.

Zamuda, A., Sosa, J. D. H., & Adler, L. (2016b). Improving Constrained Glider Trajectories for Ocean Eddy Border Sampling within Extended Mission Planning Time. In *2016 IEEE Congress on Evolutionary Computation* (pp. 1727–1734).

Zhang, J., & Sanderson, A. C. (2009). JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, *13*, 945–958.

Zhang, T., Li, Q., Zhang, C.-s., Liang, H.-w., Li, P., Wang, T.-m., Li, S., Zhu, Y.-l., & Wu, C. (2017). Current trends in the development of intelligent unmanned autonomous systems. *Frontiers of Information Technology & Electronic Engineering*, *18*, 68–85.

Zhao, W., Zhou, Y., Yu, J., Zhang, A., & Huang, Y. (2016). Tracking control of underwater gliders in ocean mesoscale eddies observation task. In *OCEANS 2016 MTS/IEEE Monterey* (pp. 1–6). IEEE.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, *1*, 32–49.