International Journal of Advanced Trends in Computer Applications
www.ijatca.com

Demystification of Many-Objective Swarm Optimization Problem using Controlling Dominance Area of Solutions and Shift-Based Density Estimator

[1]Imhanlahimi R. E, [2]Kaba Y, [3]John-Otumu A. M, [4]Imhanlahimi G. O
[1]Department of Computer Science, Ambrose Alli University, Ekpoma, Nigeria
[2]myTaxi, Berlin Germany
[3]Training, Research & Innovations Unit, Directorate of ICT, Ambrose Alli University, Ekpoma, Nigeria
[4]Professional and Applied Continuous Education, University of Winnipeg, Canada
[1]rebeccaimhans@gmail.com, [2]y_kabs@yahoo.com, [3]macgregor.otumu@gmail.com,
[4]gloryimhanlahimi@gmail.com

Abstract: *Evolutionary algorithms have been used to solve multi-objective optimization problems of two or three objectives giving results that both converge to the optimal front and are diverse along that front. However, once the number of objectives rises to four and above, the Pareto dominance concept that most of these algorithms are based on loses selection pressure, recombination operations are ineffective as individuals in populations are not close in the problem space and the evaluation of performance measures such as hyper-volume becomes computationally expensive. To overcome the loss in selection pressure, recent evolutionary algorithms adopt a number of techniques which include modifying the dominance relation to increase selection pressure as done with €-dominance, α-dominance and dominance area control, using a secondary selection mechanism such as the shift-based density estimator, grid-based fitness metric or knee-point driven analysis to filter individuals and decomposing the so-called many objective problems into sub-problems to simultaneously solve. The controlling dominance area of solutions, CDAS technique has been found to be effective in generating solutions that converge to the optimal front but leads to a loss in the diversity of the generated solutions. In this paper, we propose an extension to the speed-constrained particle swarm optimizer that uses a relaxed form of dominance, CDAS and the shift-based density estimator as secondary selection mechanism to increase diversity of solutions obtained. Also, the particles in the swarm will have an archive of personal bests from which the selection will be done using the weighted sum approach. Finally, the performance of the algorithm will be compared with state of the art algorithms such as NSGA-III and MOEA/DD using the DTLZ benchmark functions.*

Keywords: Particle SwarmOptimizer, Many-Objective, CDAS, SDE, Intelligence.

I. INTRODUCTION

Optimization decisions are made in all aspects of nature. In genetics, recombination is done as a search for desirable characteristics while the collaborative behavior of ants, bees, and schools of fishes in avoiding prey and finding food is a form of path optimization. Among humans, optimization can be a business decision between maximizing profits and minimizing risks, an engineering design problem to efficiently utilize scarce materials without compromising safety or in making personal decisions to derive the maximum utility from fixed resources. In all of these cases, decisions must be made in the context of all the conflicting but feasible solutions. In the case of a business-man aiming to achieve maximum profit, the level of risk must be acceptable and reasonable. Many practical real-life problems require the “balancing” or optimizing of three or more usually conflicting objectives. These decisions can be easily made if all the possible scenarios can be enumerated giving the decision maker the ability to make an educated choice. However, this is only possible in trivial cases. It is more common that the size of problem space makes it impossible to enumerate all the different combinations of objectives. Optimization is therefore essentially a search through a space of feasible solutions to obtain the maximum or minimum value depending on the nature of problem being solved or decision being made. Multi-objective optimization involves more than one objective and typically a set of results is returned to give the decision maker a range of choices with different trade-offs. Evolutionary algorithms are well-suited to solving multi-objective problems as they are population-based, make no assumptions about the problem domain, can search large population spaces and return a set or population of results. Evolutionary techniques include genetic algorithm, differential evolution, evolutionary strategies and swarm optimization. As their names suggest, evolutionary algorithms borrow techniques from nature in searching through large multi-dimensional spaces. Evolution is a selective process that consistently and gradually refines a population by ensuring that properties of fitter members of the population are passed on the next generation. In nature, the ability of a species to survive in an environment of intense competition for scarce resources is dependent on how well it passes the characteristics needed for survival to the next generation. Particle swarm optimization is an evolutionary technique that mimics the emergent group intelligence exhibited by a swarm of birds, a school of fishes or a group of individual particles acting locally. Taking a swarm of birds as an example, no single individual has a complete picture 2 of the environment of the swarm, but by interacting and reacting to events locally, a group intelligence is emergent. This can be observed in the search for food and breeding locations and in the avoidance of prey. Particle swarm optimization is a population-based algorithm that updates the velocity of each particle in the swarm depending on the best position found by that particle and the global best position of the swarm. It is employed to solve multi-objective problems of four or more objectives.

Evolutionary techniques have been effective in solving multi-objective problems of two or three objectives. However, when the number of objectives rises to four and above, selection pressure is lost as the Pareto dominance relation that most of the evolutionary algorithms are dependent on as a primary classifier of solutions is ineffective in selecting appropriate parents in genetic algorithms or leaders in swarm optimizers. Many approaches have been proposed to mitigate this, some of which includes a redefinition of the Pareto dominance relation, integrating a secondary classifier (usually a density estimator) to increase selection pressure or completely discarding the Pareto dominance relation. This project proposes a particle swarm optimizer that uses not only a modified form of dominance and but also a secondary classifier optimized for multi-objective problems of four or more objectives.

The objective of this work is to extend existing techniques in swarm optimization to solve multi-objective problems of more than four objectives consistently giving a result set that is converged, well-distributed and diverse along the true Pareto front or the approximate front if the true front is unknown. The swarm optimizer is implemented, experimental tests are run to compare with state of the art algorithms and the results are statistical compared to determine the effectiveness of the proposed optimizer. Secondly, a thorough analysis of the optimizer is undertaken to determine the effect of using parameterization using Saltelli sampling method.

II. LITERATURE REVIEW

This section contains a description of the latest techniques in the field of multi-objective optimization using evolutionary techniques reviewed.

Taxonomy of approaches that have been used to solve many objective optimization problems is given by [1]. The many objective optimization models can be grouped into five: a dominance relation modification model, secondary convergence metric model, performance indicator model, decomposition model and non-Pareto relations models.

The dominance relation modification model are algorithms that modify the existing Pareto dominance technique by using a relaxed form of dominance to solve the loss in selection pressure due to an increase in the number of objectives.

Techniques adopting this model include the use of a relaxed form of dominance called the Controlling Dominance Area of Solutions [2]. An update of Controlling Dominance Area of Solutions (CDAS) in the adoption of a self-controlling dominance area of solutions which attempts to solve the problem of user-defined parameter tuning in the CDAS method [3], the use of CDAS in two particle swarm optimizers in a technique referred to as CDAS-SMPSO [4], the addition of fuzzy logic to determine a new fitness evaluation technique called fuzzy Pareto domination relation in [5, 6], the use of a preference ordering [7] and a novel velocity update and fitness evaluation technique in NMPSO [8].

The secondary convergence metric model uses a second criterion to rank solutions since the number of non-dominated solutions increases as a function of the number of objectives. In this model, the Pareto relation is not modified but selection pressure is maintained by adding an additional convergence metric. Some of the convergence metrics used is the knee-point driven evolutionary algorithm [9], grid-based fitness metric strategy [10] and the modification of [11] to use a substitute distance assignment instead of crowd distancing assignment in [12].

The third model does not use any dominance concept but uses a performance indicator during the run of the algorithm to determine good solutions. The evaluation of performance indicators for multi-objective problem is expensive and this can increase the complexity of this model. Algorithms that use a performance indicator to determine solutions that better fit the diversity and convergence criteria include the indicator based selection in multi-objective search [13], HypE, an algorithm that uses Monte Carlo simulation to calculate approximate hyper-volume values for many objective problems which can be used to monotonically rank solutions [14] and a multiple indicator based algorithm that attempts to use stochastic measures to mitigate the effect of biases in single indicators [15].

Decomposition involves mathematically breaking down the many-objective problems into sub-problems and simultaneously solving these sub-problems. This technique was first implemented in [16]. Decomposition method has also been combined in some cases with the dominance relation to solve many objective problems as done in the reference-based sorting, NSGA-III [17, 18, 19].

[20] extended the NSGA-III to provide better convergence by adopting decomposition based fitness evaluation.

Techniques have been developed that use an entirely different form of ranking solutions which are not Pareto based. Examples of this include the use of ranking dominance which aggregates the performance of solution per performance indicator [15], a new archiving method used in Ideal Point Guided MOPSO [21] and the use of a leadership selection scheme, NWSum[22] to handle many objective problems in I-MOPSO [21].

[1]found that PSOs such as CDAS-SMPSO scaled very well as the number of objectives increased likely as “a result of reduced crossover operator effectiveness due to the immense search space and higher levels of epistasis encountered in large-scale MaOPs.” Also, the use of dominance and decomposition especially in MOEA/DD [23] gave promising results.

[1] Concluded that PSOs using the CDAS dominance scheme consistently produced good results as the number of objectives increased. This result is consistent with that found in [21]. However, subsequent studies found that the CDAS favors convergence over diversity which resulted in the modified S-CDAS. Hence, it is expected that CDAS combined with a crowding technique that promotes diversity such as the shift-based density estimator; SDE [24] will give a competitive performance if it incorporated with the SMPSO algorithm.

III. METHODOLOGY

This research work proposes a combination of CDAS, SDE and WSum as a more efficient means of resolving many-objectives problems. The CDAS acts as a fitness evaluation function to decide which particles of the swarm are saved to the archive, and when the archive is full, a secondary metric to filter solutions will be the shift-based density estimation (SDE) to ensure diversity in the leaders selected to lead as suggested by [24].

The leadership selection scheme of WSum for global and personal leadership selections will be incorporated into the proposed technique and the efficiency evaluated to determine if the increase in the quality of solutions produced justify the extra-computational costs.

The CDAS-SDE Particle Swarm Algorithm i.e. CSPSO algorithm extends the speed constrained particle swarm optimization algorithm to enhance the algorithm’s performance on many-objective optimization problems.

The Controlling Dominance Area of Solutions (CDAS)

The CDAS is used to increase the dominance area and hence selection pressure of Pareto-based algorithms. The use of CDAS was found to produce solutions that converge to the true Pareto front but may lead to deterioration in the diversity of swarm.

```
Input-> particle/vector X, userDefinedParameter
Output-> array of modified fitness values, Y
Begin
objectiveCount = getNumberOfObjectives(X)
vectorSum = 0;
fitnessValue = array[objectiveCount]
foreach objective i
vectorSum += (X.getObjectiveValue(i) *
X.getObjectiveValue(i))
vectorNorm = √vectorSum
phiAngle = PI * userDefinedParameter
foreach objective i
omegaAngle = Arccosine
(X.getObjectiveValue(i)/vectorNorm)
fitnessValue[i] = (vectorNorm * Sine(phiAngle +
omegaAngle))/Sine(phiAngle)
return fitnessValue
End
```

Figure 3.1: CDAS Pseudo-code Listing

The Shift-based Density Estimator (SDE)
The loss in selection pressure due to an increase in the number of objectives can be solved by modifying the dominance relation or using a secondary fitness evaluation scheme. Most density, crowding and nearest neighbor fitness evaluation mechanism are focused on the diversity of solutions.

However, the shift-based density estimator (SDE) was developed to provide a secondary fitness evaluation method to produce a well converged and diverse set of particles. To achieve this, SDE considers both the sparsity of the neighborhood of a solution and the convergence of that solution. It ensures that poorly-converged particles are shifted to denser regions hence increasing their chances of elimination.

```
Begin
Input -> Population: P, Size of population: N
Ouput -> NULL
foreach particle i in the population P
foreach objective j
shiftedList = List of size (N-1)
foreach individual q where q ε P &i != q
q′(j) = (q(j) <i(j)) ? i(j) : q(j)
add q′ to shiftedList
foreach individual in shiftedList, r′
dist(i, r′) = euclidean distance between i& r
D(i, P) = D(dist(i, q′(1)), dist(i, q′(2)), ..., dist(i,
q′(N−1)))
i.SHIFT_BASED_DENSITY_ESTIMATOR = D(i, P)
End
```

Figure 3.2: SDE Pseudo-code Listing

The Weighted Sum Approach (WSum)
To select global or local leaders from an archive for a particle p, the WSum approach assigns more weight to the objectives that p is already good at. The sum is calculated for each particle in the archive using the expression in equation 1 below:

$$F = \sum_j \frac{f_j(x_i)}{\sum_k f_k(x_i)} f_j(p_i) \quad (1)$$

Where fj (xi) is the jth fitness value of particle i, pi is the particle whose leader is to be selected. The leader with the smallest weighted sum is selected for the personal best while the reverse is done for the global leaders.

To ensure diversity, the leader with the smallest weighted sum value is selected. To promote convergence the leader with the highest weight is selected. In the CDAS-SDE algorithms, diversity is promoted on the local level and convergence on the global scale, so global best is selected using the highest weighted sum value and local leaders are selected using the smallest weighted sum value.

```
Begin
Input-> Population of particles: swarm, Particle:
particle
Output-> Population with
WSUM_FITNESS_ATTRIBUTE set
foreach individual in swarm:
weightedAggregate = 0
objectiveSum = 0
foreach objective i
objectiveSum += individual.getObjectiveValue(i)
foreach objective j
weightedAggregate +=
((individual.getObjectiveValue(j)/objectiveSum) *
particle.getObjectiveValue(j)
individual.WSUM_FITNESS_ATTRIBUTE =
weightedAggregate
end
```

Figure 3.3: WSum Pseudo-code Listing

The Complete CSPSO Algorithm including leadership scheme
Step 1: The swarm is initialized with random positions. A personal best archive is attached to each particle and initialized with the current position of that particle

Step 2: Non-dominated particles using CDAS as a classifier are added to the archive.

Step 3: Speed is computed. For a particle x, a local leader is selected from the $x's$personal archive using WSum, while it's global leader is selected using the WSum with three randomly selected leaders from the archive as candidate leaders.

Step 4: Positions are updated using the speed and personal archive of each particle is also updated if a better position is found.

Step 5: Polynomial mutation is probabilistically applied.

Step 6: Leader's archive is updated using the CDAS. If archive is full and a non-dominated solution is found, the secondary fitness evaluation of SDE is used to determine if the new solution should replace a particle in the archive or be discarded

Step 7: If the maximum number of evaluations is not exceeded: steps (iii) – (vii) are repeated.

Step 8: return leaders archive.

Figure 3.4: Complete CSPSO Algorithm including leadership scheme

IV. RESULTS AND DISCUSSION

The aggregate column is the value of a performance measure derived from a combination of the approximate results set from all the seeds. The indifferent column in the Figures signifies statistically similar or different median results. This indifferent column is of importance as it shows that the addition of a deliberate local and global leadership selection scheme has no significant effect on the performance of the algorithm. This is evidenced by the observation that CSPSO and CDASPSO have statistically different performances in only ≈4% of the observed cases (2 of 45). The increased complexity and computational cost of applying the WSum selection scheme does not result in a commiserate increase in the quality of results produced.

With the value of Si=0.25, each particle in the swarm is moved through an angle 45⁰ to each of the objectives leading to a significant increase in the area of dominance and hence selection pressure. This results in very early convergence of the CDAS-SDE algorithms and a loss of diversity despite the secondary mechanism of shift-based density estimator. This is illustrated in the Figures 4.1 and 4.2.

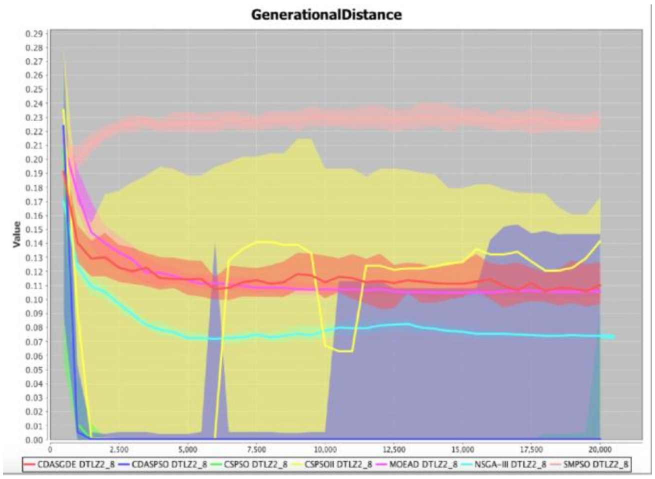

Figure 4.1: S = 0.25, Problem is DTLZ2

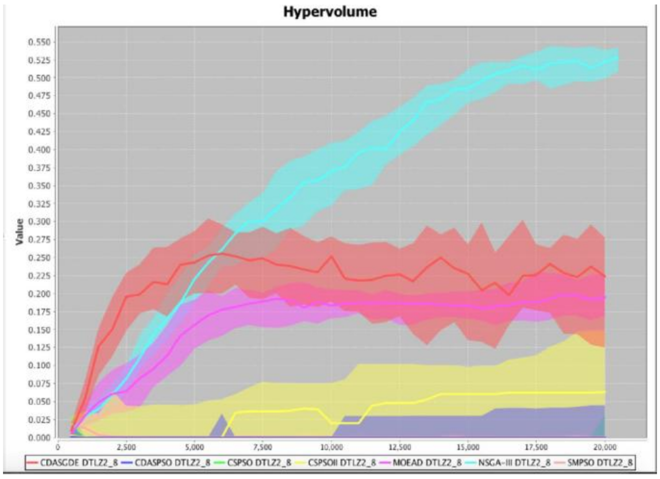

Figure 4.2: S = 0.25, Problem is DTLZ2

The CDAS-SDE algorithms perform bests in the additive epsilon indicator and generational distance performance measures, with *CDASPSO* converging to the optimal values for generational distance within 2,500 function evaluations but have a comparatively worse performance in hyper-volume to NSGA-III and MOEAD.

Similar results are observed for *DTLZ3* and *DTLZ4* with the CDAS-based algorithms struggling with *DTLZ4* but performing better in *DTLZ3.* When the value of Si=0.35, the angle through which particles are translated is reduced in comparison to Si= 0.25.

This again results in a fast convergence for all the CDAS-SDE algorithms resulting in comparatively better values for generational distance and additive epsilon indicator.

This is illustrated in figures 4.3 and 4.4.

Note from figure 4.3 that CDASPSO and CSPSO have attained almost 0 generational distance values at roughly 1000 function evaluations. MOEAD and NSGA-III reach this value at close to 20,000 function evaluations. A similar observation is made from the generational distance chart. From figure 4.4, the CDAS-based algorithms have similar results in hyper-volume to NSGA-III and MOEAD. DTLZ4 pose a more difficult challenge for the CDAS-SDE algorithms as they perform poorly in the hyper-volume in comparison with NSGA-III and MOEAD.

At Si=0.45, the shift-based density estimator plays a more dominant role as the area of dominance of solutions is contracted. This results in a better spread of solutions but also slower convergence towards the true Pareto front. The hyper-volume chart shown in figure 4.5 illustrates this.

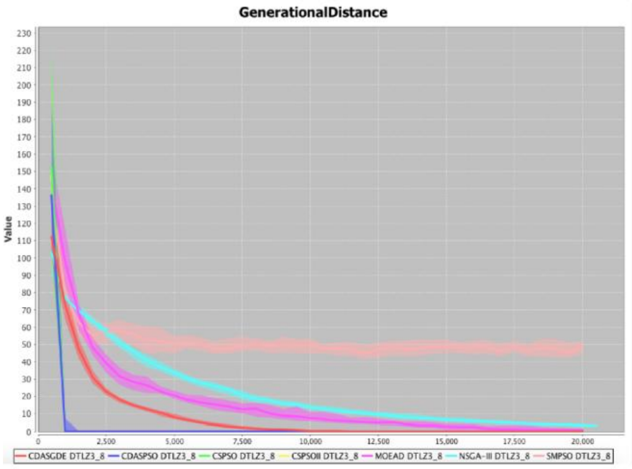

Figure 4.3: S = 0.35, Problem is DTLZ3

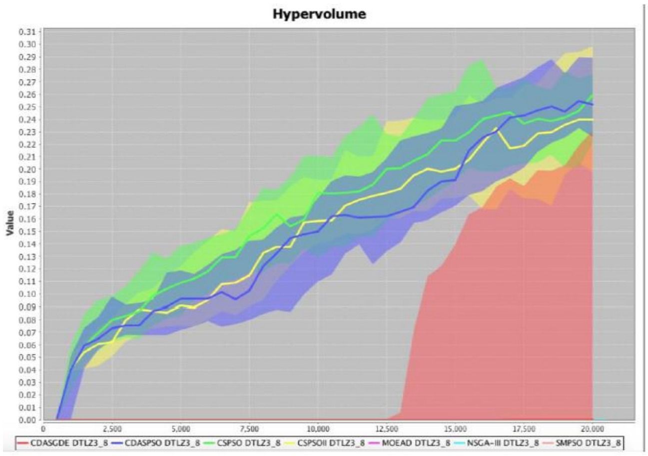

Figure 4.4: S = 0.35, Problem is DTLZ3

When Si = 0.50, which is the same as the conventional Pareto dominance, selection pressure is reduced as the number of objectives under consideration is 8. From the chart in figure 4.6, this results in a similar convergence profile as NSGA-III and MOEAD.

When Si = 0.65, the CDAS-based algorithms perform poorly for both convergence and diversity of solutions as shown in the figures 4.7 and 4.8.

In all of these cases, SMPSO was found to be ineffective in finding a well-converged or distributed solution set further confirming the loss in selection pressure due to Pareto dominance in many-objective optimization problems.

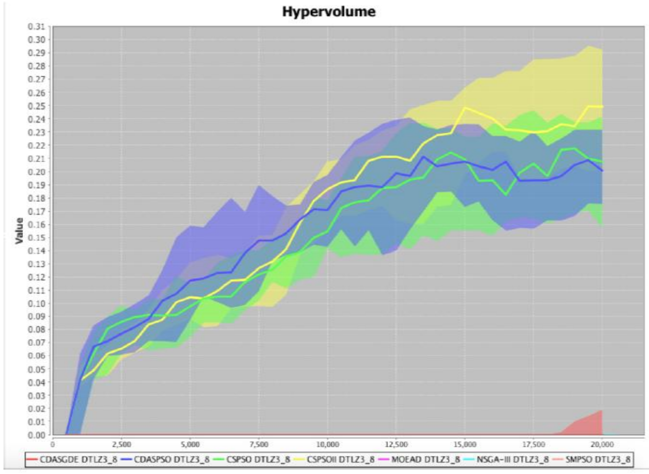

Figure 4.5: S = 0.45, Problem is DTLZ3

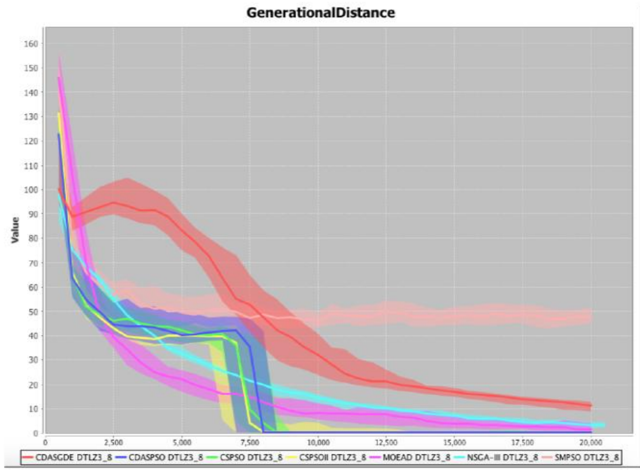

Figure 4.6: S = 0.50, Problem is DTLZ3

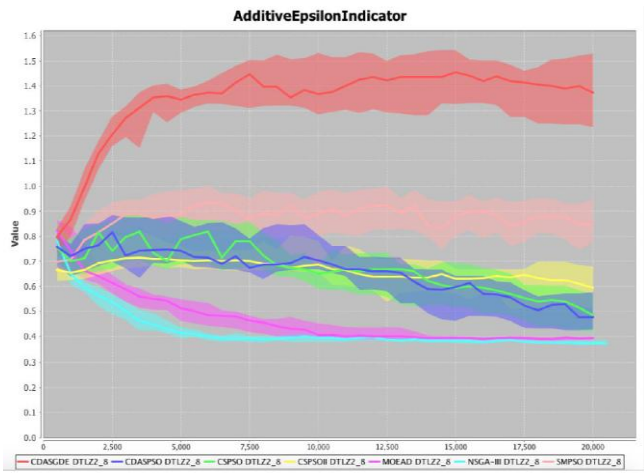

Figure 4.7: S = 0.65, Problem is DTLZ2

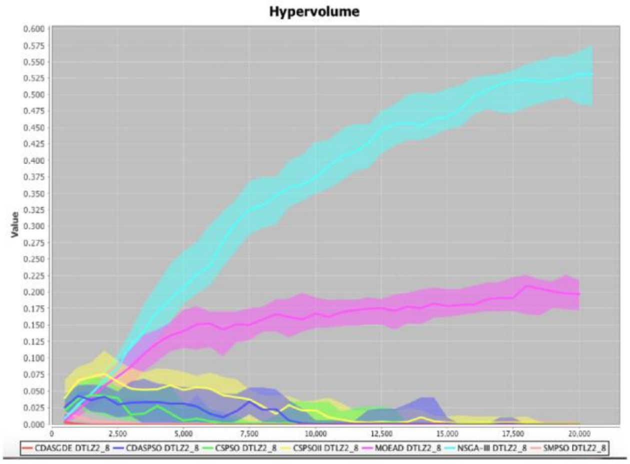

Figure 4.8: S = 0.65, Problem is DTLZ2

V. CONCLUSION

In this research work shows that a combination of CDAS and SDE with a swarm optimizer is competitive with state of the art algorithms in solving multi-objective problems of more than four objectives. Also, the addition of a deliberate leadership selection strategy at the local and global level while increasing the amount of computation required during each flight did not result in a corresponding level of increase in the quality of results produced. The diagnostic tests showed that CDASPSO provided very quick convergence in cases where an appropriate Si value was selected and that Si = 0.45 produced the best results in terms of convergence and diversity.

While most of the CDAS-based algorithms could find good results in the DTLZ2 and DTLZ3, they consistently performed worse than NSGA-III and MOEAD in terms of hyper-volume on DTLZ4.

Further research is needed to investigate this. Secondly, the fast convergence of the CDAS-based algorithms at lower Si values can be capitalized upon by promoting diversity in the swarm after a certain percentage of the maximum number of function evaluations has been exceeded. This can be done by applying a scheme that uses a performance indicator to detect convergence in the swarm and increment the Si value as swarm converges to promote diversity.

References

[1] Maltese, J., Ombuki-Berman, B. M., & Engelbrecht, A. P. (2016). A Scalability Study of Many-Objective Optimization Algorithms. IEEE.

[2] Sato, H., Aguirre, H. E., & Tanaka, K. (2007). Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs. In H. Sato, H. E. Aguirre, & K. Tanaka, Evolutionary Multi-Criterion Optimization (pp. 5-20). Berlin Heidelberg: Springer.

[3] Hiroyuki, S., Aguirre, H. E., & Tanaka, K. (2010). Self-Controlling Dominance Area of Solutions in Evolutionary Many-Objective Optimization. In K. Deb, A. Bhattacharya, N. Chakraborti, P. Chakroborty, S. Das, J. Dutta, . . . K. C. Tan, Simulated Evolution and Learning (pp. 455-465). Berlin Heidelberg: Springer.

[4] de Carvalho, A. B., &Pozo, A. (2012). Measuring the Convergence and Diversity of CDAS Multi-Objective Particle Swarm Optimization Algorithms: A Study of Many-Objective Problems. International Conference on Hybrid Artificial Intelligence Systems (pp. 43–51). Brazil: Neuro-Computing

[5] He, Z., Yen, G. G., & Zhang, J. (2014). Fuzzy-Based Pareto Optimality for Many-Objective Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation, 269 - 285.

[6] Farina, M., & Amato, P. (2004). A Fuzzy Definition of "Optimality" for Many-Criteria Optimization Problems. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 315 - 326.

[7] di Pierro, F., Khu, S.-T., & Savic, D. A. (2007). A Investigation on Preference Order Ranking Scheme for Multiobjective Evolutionary Optimization. IEEE Transactions on Evolutionary Computation, 17-45.

[8] Qiuzhen, L., Liu, S., Zhu, Q., Tang, C., Song, R., Chen, J., Zhang, J. (2016). Particle Swarm Optimization with a Balanceable Fitness Estimation for Many-objective Optimization Problems. IEEE Transactions on Evolutionary Computation, 1-1.

[9] Zhang, X., Tian, Y., &Jin, Y. (2014). A Knee Point-Driven Evolutionary Algorithm for Many-Objective Optimization. IEEE Transactions on Evolutionary Computation, 761 - 776.

[10] Li, M., Zheng, J., Shen, R., Li, K., & Yuan, Q. (2010). A Grid-Based Fitness Strategy for Evolutionary Many-Objective Optimization. Proceedings of the 12th annual conference on Genetic and evolutionary computation (pp. 463-470). Oregon: ACM.

[11] Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. Lecture Notes in Computer Science (pp. 849-858). Paris: Springer.

[12] Köppen, M., & Yoshida, K. (2007). Substitute Distance Assignments in NSGA-II for Handling Many-objective Optimization Problems. In M. Köppen, & K. Yoshida, Evolutionary Multi-Criterion Optimization (pp. 727-741). Berlin Heidelberg: Springer.

[13] Zitzler, E., & Künzli, S. (2004). Indicator-Based Selection in Multiobjective Search. In X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, H.P. Schwefel, Parallel Problem Solving from Nature - PPSN VIII (pp. 832-842). Berlin Heidelberg: Springer.

[14] Bader, J., & Zitzler, E. (2011). HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. Evolutionary Computation, 45-76.

[15] Li, B., Tang, K., Li, J., & Yao, X. (2016). Stochastic Ranking Algorithm for Many-Objective Optimization Based on Multiple Indicators. *IEEE Transactions on Evolutionary Computation*, 924-938

[16] Zhang, Q., & Li, H. (2007). MOEA/D: A Multi objective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation, 712 - 731.

[17] Deb, K., & Jain, H. (2014). An Evolutionary Many Objective Optimization Algorithm Using Reference Point-Based Non-Dominated Sorting Approach. IEEE Transactions on Evolutionary Computation, 577-601.
[18] Jain, H., & Deb, K. (2014). An Evolutionary Many-Objective Optimization Algorithm using Reference-Point Based Non-Dominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. IEEE Transactions on Evolutionary Computation, 602-622.
[19] Li, K., Deb, K., Qingfu, Z., & Kwong, S. (2014). An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. IEEE Transactions on Evolutionary Computation (pp. 694 - 716). IEEE.
[20] Yuan, Y., Xu, H., Wang, B., & Yao, X. (2016). A New Dominance Relation-Based Evolutionary Algorithm for Many-Objective Optimization. IEEE Transactions on Evolutionary Computation, 16-37.
[21] Britto, A., & Pozo, A. (2012). I-MOPSO: A Suitable PSO Algorithm for Many-Objective Optimization. Brazilian Symposium on Neural Networks (pp. 166-171). IEEE.
[22] Padhye, N., Branke, J., & Mostaghim, S. (2009). Empirical comparison of MOPSO methods - Guide selection and diversity preservation. Evolutionary Computation (pp. 2516-2523). IEEE.
[23] Li, K., Deb, K., Qingfu, Z., & Kwong, S. (2014). An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. IEEE Transactions on Evolutionary Computation (pp. 694 - 716). IEEE.
[24] Li, M., Yang, S., & Liu, X. (2013). Shift-Based Density Estimation for Pareto-Based Algorithms in Many-Objective Optimization. IEEE Transactions on Evolutionary Computation (pp. 348 - 365). IEEE.