Multi-level selection genetic algorithm applied to CEC '09 test instances

Przemyslaw A. Grudniewski
Fluid Structure Interactions Group
University of Southampton
Southampton, United Kingdom
pag1c15@soton.ac.uk

Adam J. Sobey
Fluid Structure Interactions Group
University of Southampton
Southampton, United Kingdom
ajs502@soton.ac.uk

***Abstract*— Genetic algorithms (GAs) are population-based optimisation tools inspired by evolution and natural selection. They are applied in many areas of engineering and industry, on increasingly complex problems. To improve the performance, the new algorithms have a tendency to be derived from sophisticated mathematical and computational mechanisms, where many biological and evolutionary advances have been neglected. One such mechanism is multi-level selection theory which has been proposed as being necessary for evolution. Previously, an algorithm developed using this theory as its inspiration has shown promising performance on simple test problems. It proposes the addition of a collective reproduction mechanism alongside the standard individual one. Here the algorithm, Multi-Level Selection Genetic Algorithm (MLSGA), is benchmarked on more sophisticated test instances from CEC '09 and compared to the final rankings. In this instance a simple genetic algorithm is used at the individual level. The developed algorithm cannot compete with top algorithms on complex unconstrained problems, however it shows interesting results and behaviour, and better performance on constrained test functions. The approach provides promise for further investigation, especially in integrating state-of-the-art individual reproduction methods to improve the performance and improving the novel collective mechanism.**

Keywords*— Evolutionary theory, evolutionary computation, genetic algorithms, multi-level selection, multi-objective optimization.

I. INTRODUCTION

A number of evolutionary algorithms have been developed since the original work of Holland in the 1960's [1]-[19]. Each of them introduced novel mechanisms and improvements, with many of the leading algorithms benchmarked on complex multi-objective optimisation problems as part of CEC'09 [20]. Current trends are towards more complex algorithms based on mathematical and computational concepts, and advanced evolutionary-based concepts are often pushed aside. Evolutionary theory has made much progress since the original GA was developed and the authors feel that exploration of these modern evolutionary concepts can lead to an increase in performance of the modern genetic algorithms. One of them is multi-level selection originally proposed by Sober and Wilson [21].

In nature individuals often group together, into packs or tribes, in order to cooperate and increase their chances of survival. During the life cycle, collectives are competing with each other, and similar to individuals, fitter groups survive and create offspring, while those with a lower fitness produce fewer or no offspring. This process can be compared to insect colonies, or human tribes, competing for food sources, where fitter colonies are able to collect enough to sustain their population and the less fit die from starvation [22]. According to Sober and Wilson [21] and Okasha [22] development of many skills of organisms such as, social skills and altruism are difficult to explain if selection only occurs at one level, as these traits are usually disadvantageous to the individual. A good example is self-sacrifice of the individual for the "greater good" of the group, reducing the chances of the individual passing on its genetic material. Multi-level selection (MLS) theory states that fitness can be defined, and thus selection and competition can occur, separately on more than one level. In multi-level selection theory two main variants are proposed MLS1 and MLS2. In multilevel selection 1, both collective and individual levels of selections depend on a single evolutionary function and thus the fitness of the collective is simply the aggregate of the fitness of individuals inside of it. This means that both collective and each individual contribute to the improvement of this single function. In MLS2 the fitness function differs on each level and therefore the selection is independent; which generates an emergent selection property. Thus, it is possible that the evolution of collectives will be dependent on different key aspects than the evolution of individuals, leading to competition between them [22].

There have already been attempts to incorporate the levels of selection theory into evolutionary computation, however in these attempts key aspects have been ignored. Lenaert et al. [23] presented an algorithm with intradermic group selection. However, no direct selection method, with separate fitness, can be found at the higher level, which is required for a true multi-level selection approach. Akbari et al. [24][25] proposed an algorithm with three separate between-group dynamics: colonization, regrouping and migration. The colonization module partially reproduces MLS1, and after each generation the worst collective is eliminated with one of the remaining collectives reproducing and creating offspring. However, in this algorithm, only the best individual in the collective is considered to represent the fitness for the whole

This work has been sponsored by Lloyd Register Foundation

978-1-5090-4601-0/17/$31.00 ©2017 IEEE

collective, and the collective with one strong individual can survive, when other stronger in average, but with weaker the best solution will extinct, which results in the loss of many good solutions. In addition, only one collective takes part in the reproduction step, which decreases the possible gene pool. Wu and Banzhaf [26] introduced a genetic algorithm using a many level hierarchical model. In this algorithm reproduction occurs separately on group level and on the individual level. However, in this approach, groups from different levels can interact with each other during the reproduction step and basic concepts of MLS, such as separate selection mechanisms on each level, are not obviously introduced.

In this paper both MLS1 and MLS2 variants of MLSGA are benchmarked on a number of complex two-objectives constrained and unconstrained test instances, taken from CEC'09, in order to evaluate their performance on more demanding functions.

II. Methodology

MLSGA splits the whole population into subgroups, collectives, which operate on the same search space, and introduces the additional level of selection. After each generation, the worst collectives, defined by low fitness, are eliminated and replaced by an equal number of new groups, generated from the remaining population. The size of the new collectives is equal to the size of the old ones in order to maintain a constant size for the whole population. Two variants of multi-level selection (MLS) are defined in MLSGA. In MLS1 the fitness function is defined as the same at both levels, collective and individual, and is the average of the f1 and f2 objective functions (1). In MLS2 both levels have separate fitness definitions; the f1 objective function is assigned to individual level (2) and f2 is assigned to collective level (3). The assignment of the fitness function at each level, individual or collective, has a large impact on the results as the both selection levels have a different selection pressure; where previous research implies that the individual level is stronger. In order to continue to investigate the sensitivity of the results to the way in which the objective is defined, MLS2R is defined here as the opposite of MLS2. Table I demonstrates the specific mechanisms used in this paper.

$$Fitness = \frac{f_1(x)+f_2(x)}{2} \quad (1)$$

$$Individual\ Fitness = f_1(x) \quad (2)$$

$$Collective\ Fitness = f_2(x) \quad (3)$$

TABLE I. MLSGA PARAMETERS

Step	Parameter	Value
1. Initialisation	Type	Random
	Encoding	Real values 16 decimals
	Pop. Size	600
2. Classification	Method	SVM
	N. Collectives	6
3. Individual Reproduction		
3.1 Fitness Evaluation	Type	MLS1 or MLS2
3.2 Selection	Type	Roulette wheel
3.3 Mating	Crossover type	Real variable SBX
	Crossover rate	0.7
	Mutation type	Polynomial
	Mutation rate	0.08
3.4 Elitism	Rate	0.1
4. Collective Reproduction		
4.1 Fitness Evaluation	Type	MLS1 or MLS2 or MLS2R
4.2 Elimination	Number of eliminated collectives	2
4.3 Replacement	Number of new collectives	2
5.Termination	Criterion	Reaching 500 generation

TABLE II. UNCONSTRAINED TEST CASES

Test case	Objective functions
UF1 [28]	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} y_j^2$ $f_2 = 1-\sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} y_j^2$ $y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2,\dots,n$ For all UF functions: $J_1 = \{j \mid j\ is\ odd\ and\ 2\le j\le n\}$ $J_2 = \{j \mid j\ is\ even\ and\ 2\le j\le n\}$ $Search\ space\ is\ [0,1]\times[-1,1]^{n-1}$
UF2 [28]	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} y_j^2$ $f_2 = 1-\sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} y_j^2$ $y_j = \begin{cases} x_j - h(t)cos\left(6\pi x_1 + \frac{j\pi}{n}\right) j\in J_1 \\ x_j - h(t)sin\left(6\pi x_1 + \frac{j\pi}{n}\right) j\in J_1 \end{cases}$ $h(t) = 0.3x_1^2cos\left(24\pi x_1 + \frac{4t\pi}{n}\right) + 0.6x_1$ $Search\ space\ is\ [0,1]\times[-1,1]^{n-1}$
UF3 [28]	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\left(4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$ $f_2 = 1-\sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\left(4\sum_{j\in J_2} y_j^2 - 2\prod_{j\in J_2} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$ $y_j = x_j - x_1^{0.5\left(1+\frac{3(j-2)}{n-2}\right)}, j = 2,\dots,n$ $Search\ space\ is\ [0,1]^n$
UF4 [28]	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} h(y_j)$

Test case	Objective functions
	$f_2 = 1 - x_1^2 + \frac{2}{\lvert J_2 \rvert} \sum_{j \in J_2} h(y_j)$ $y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, \dots, n$ $h(t) = \frac{\lvert t \rvert}{1 + e^{2\lvert t \rvert}}$ $Search\ space\ is\ [0,1] \times [-2,2]^{n-1}$
UF5 [28]	$f_1 = x_1 + \left(\frac{1}{2N} + \varepsilon\right) \lvert sin(2N\pi x_1) \rvert + \frac{2}{\lvert J_1 \rvert} \sum_{j \in J_1} h(y_j)$ $f_2 = 1 - x_1 + \left(\frac{1}{2N} + \varepsilon\right) \lvert sin(2N\pi x_1) \rvert + \frac{2}{\lvert J_2 \rvert} \sum_{j \in J_2} h(y_j)$ $y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, \dots, n$ $h(t) = 2t^2 - cos(4\pi t) + 1$ $Search\ space\ is\ [0,1] \times [-1,1]^{n-1}$
UF7 [28]	$f_1 = \sqrt[5]{x_1} + \frac{2}{\lvert J_1 \rvert} \sum_{j \in J_1} y_j^2$ $f_2 = 1 - \sqrt[5]{x_1} + \frac{2}{\lvert J_2 \rvert} \sum_{j \in J_2} y_j^2$ $y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, \dots, n$ $Search\ space\ is\ [0,1] \times [-1,1]^{n-1}$

In MLSGA the Pareto front is stored externally, and is used for illustrative purposes and to calculate the Inverted Generational Distance (IGD), but is not integral to the reproduction mechanisms. The biggest advantage of the approach is the fact that the mechanism allows for the use of any standard genetic algorithm mechanism at the individual level, though a simple mechanism is selected for early benchmarking. IGD is the performance measure function of Pareto front, which shows the average distance between all points in the true Pareto front and the closest solution from the achieved set and is calculated by (4);

$$IGD(A, P^*) = \frac{\sum_{v \in P^*} d(v,A)}{\lvert P^* \rvert}; \qquad (4)$$

where P∗ is a set of uniformly distributed points along the Pareto front, in the objective space, A is an approximate set to the Pareto front and d(v, A) is the minimum Euclidean distance between v and the points in A.

The test cases used are the unconstrained UF and constrained CF functions taken from the CEC ’09 competition [28] and are shown in the Table II and Table III. The results are compared with other algorithms following the CEC ’09 competition rules [20], based on the calculated IGD values.

TABLE III. CONSTRAINED TEST CASES

Test case	Objective functions
CF1 [28]	$f_1 = x_1 + \frac{2}{\lvert J_1 \rvert} \sum_{j \in J_1} \left(x_j - x_1^{0.5\left(1 + \frac{3(j-2)}{n-2}\right)}\right)^2$ $f_2 = 1 - x_1 + \frac{2}{\lvert J_2 \rvert} \sum_{j \in J_2} \left(x_j - x_1^{0.5\left(1 + \frac{3(j-2)}{n-2}\right)}\right)^2$ For all CF functions: $J_1 = \{j \mid j\ is\ odd\ and\ 2 \le j \le n\}$ $J_2 = \{j \mid j\ is\ even\ and\ 2 \le j \le n\}$ Constrain: $f_1 + f_2 - a\lvert \sin[N\pi(f_1 - f_2 + 1)] \rvert - 1 \ge 0$ $Search\ space\ is\ [0,1]^n$
CF2 [28]	$f_1 = x_1 + \frac{2}{\lvert J_1 \rvert} \sum_{j \in J_1} \left(x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2 \rvert} \sum_{j \in J_2} \left(x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2$ Constrain: $\frac{t}{1 + e^{4\lvert t \rvert}} \ge 0$ $t = \sqrt{f_1} + f_2 - a \sin[N\pi(\sqrt{f_1} - f_2 + 1)] - 1$ $Search\ space\ is\ [0,1] \times [-1,1]^{n-1}$
CF3 [28]	$f_1 = x_1 + \frac{2}{\lvert J_1 \rvert}\left(4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} cos\left(\frac{20 y_j \pi}{\sqrt{j}}\right) + 2\right)$ $f_2 = 1 - x_1^2 + \frac{2}{\lvert J_2 \rvert}\left(4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} cos\left(\frac{20 y_j \pi}{\sqrt{j}}\right) + 2\right)$ $y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, \dots, n$ Constrain: $f_1^2 + f_2 - a \sin[N\pi(f_1^2 - f_2 + 1)] - 1 \ge 0$ $Search\ space\ is\ [0,1] \times [-2,2]^{n-1}$
CF4 [28]	$f_1 = x_1 + \sum_{j \in J_1} h_j(y_j)$ $f_2 = 1 - x_1 + \sum_{j \in J_2} h_j(y_j)$ $y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, \dots, n$ $h_2(t) = \begin{cases} \lvert t \rvert & if\ t < \frac{3}{2}\left(1 - \frac{\sqrt{2}}{2}\right) \\ 0.125 + (t-1)^2 & otherwise \end{cases}$ and $h_j(t) = t^2$ Constrain: $\frac{t}{1 + e^{4\lvert t \rvert}} \ge 0$ $t = x_2 - sin\left(6\pi x_1 + \frac{2\pi}{n}\right) - 0.5x_1 + 0.25$ $Search\ space\ is\ [0,1] \times [-2,2]^{n-1}$
CF5 [28]	$f_1 = x_1 + \sum_{j \in J_1} h_j(y_j)$

Test case	Objective functions
	$f_2 = 1 - x_1 + \sum_{j\in J_2} h_j(y_j)$ $y_j = \begin{cases} x_j - 0.8x_1 cos\left(6\pi x_1 + \frac{j\pi}{n}\right) j \in J_1 \\ x_j - 0.8x_1 sin\left(6\pi x_1 + \frac{j\pi}{n}\right) j \in J_1 \end{cases}$ $h_2(t) = \begin{cases} \lvert t \rvert \quad if\ t < \frac{3}{2}\left(1 - \frac{\sqrt{2}}{2}\right) \\ 0.125 + (t-1)^2 \quad otherwise \end{cases}$ and $h_j(t) = 2t^2 - \cos(4\pi t) + 1$ Constrain: $x_2 - 0.8x_1 sin\left(6\pi x_1 + \frac{2\pi}{n}\right) - 0.5x_1 + 0.25 \ge 0$ $Search\ space\ is\ [0,1] \times [-2,2]^{n-1}$
CF6 [28]	$f_1 = x_1 + \sum_{j\in J_1} y_j^2$ $f_2 = (1-x_1)^2 + \sum_{j\in J_2} y_j^2$ $y_j = \begin{cases} x_j - 0.8x_1 cos\left(6\pi x_1 + \frac{j\pi}{n}\right) j \in J_1 \\ x_j - 0.8x_1 sin\left(6\pi x_1 + \frac{j\pi}{n}\right) j \in J_1 \end{cases}$ Constrains: $x_2 - 0.8x_1 sin\left(6\pi x_1 + \frac{2\pi}{n}\right) - sign(k_1)\sqrt{\lvert k_1 \rvert} \ge 0$ $x_4 - 0.8x_1 sin\left(6\pi x_1 + \frac{4\pi}{n}\right) - sign(k_2)\sqrt{\lvert k_2 \rvert} \ge 0$ $k_1 = 0.5(1-x_1) - (1-x_1)^2$ $k_2 = 0.25\sqrt{1-x_1} - 0.5(1-x_1)$ $Search\ space\ is\ [0,1] \times [-2,2]^{n-1}$
CF7 [28]	$f_1 = x_1 + \sum_{j\in J_1} h_j(y_j)$ $f_2 = (1-x_1)^2 + \sum_{j\in J_2} h_j(y_j)$ $y_j = \begin{cases} x_j - cos\left(6\pi x_1 + \frac{j\pi}{n}\right) j \in J_1 \\ x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right) j \in J_1 \end{cases}$ $h_2(t) = h_4(t) = t^2$ and $h_j(t) = 2t^2 - \cos(4\pi t) + 1$ Constrains: $x_2 - sin\left(6\pi x_1 + \frac{2\pi}{n}\right) - sign(k_1)\sqrt{\lvert k_1 \rvert} \ge 0$ $x_4 - sin\left(6\pi x_1 + \frac{4\pi}{n}\right) - sign(k_2)\sqrt{\lvert k_2 \rvert} \ge 0$ $k_1 = 0.5(1-x_1) - (1-x_1)^2$ $k_2 = 0.25\sqrt{1-x_1} - 0.5(1-x_1)$ $Search\ space\ is\ [0,1] \times [-2,2]^{n-1}$

III. Results

The benchmarking is performed for different unconstrained two-objective functions, UF1-UF5 and UF7 shown in Table II, with 30 variables each and for constrained problems, CF1-5 with 1 constraint and CF6-7 with 2 constraints shown in Table III, with 10 variables each. For each function 30 separate runs have been performed, the stopping criteria is 300,000 function evaluations for each run, and the 100 best solutions are considered for the purpose of the IGD calculation; the average IGD value is presented in order to follow CEC'09 rules [20].

The results for UF1-5 and 7 are found and the Pareto fronts are illustrated for the UF1, Fig 1, and for the UF2 function Fig 2. UF1 and UF2 functions are chosen as they provide results representative of the worst and the best cases for the unconstrained functions.

For UF1 function the Pareto fronts show that all the variants were able to reach the true Pareto front but were unable to find a range of values. MLS2 and MLS2R found a wider variety of points along the true Pareto front than MLS1 although most of them were far from it.

Better performance can be observed on the UF2 problem, Fig. 2. In this benchmark all variants, were able to reach the front and to develop a diverse range of points. However, even in the best case of MLS2R, only half of the front is explored.

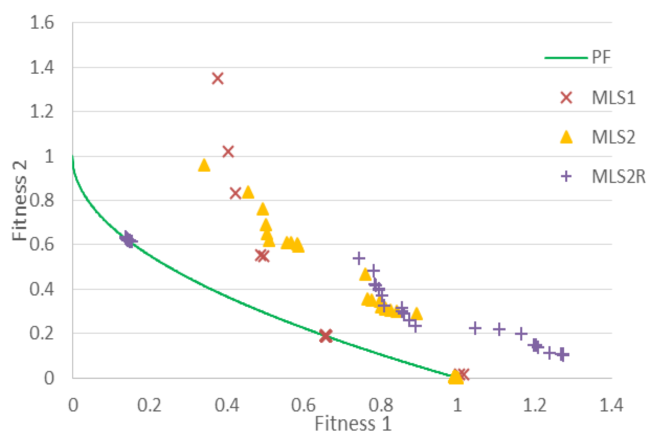

Fig. 1. Pareto front of MLSGA variants on UF1 function

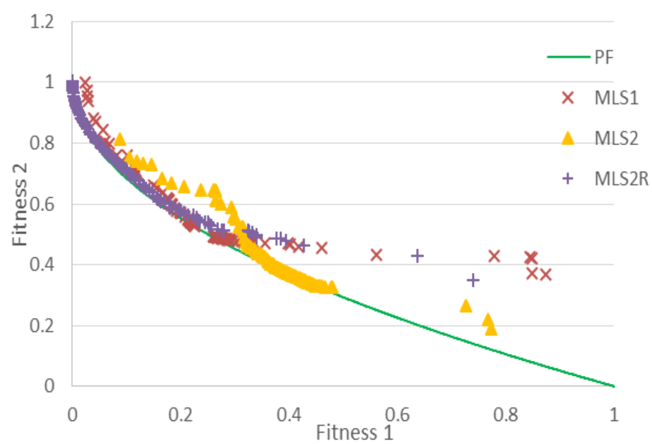

Fig. 2. Pareto front of MLSGA variants on UF2 function

The process is repeated for the constrained CF1-7 functions. CF1 Fig 3, and CF3, Fig 4, are chosen as they illustrate the worst and the best cases for the constrained test set.

The obtained results show that all MLSGA variants perform similarly; they were all able to reach the true Pareto front, and find a diverse range of points. However, the algorithm lacks accuracy for the border values of Pareto front, and concentrates its search on the middle.

The worst performance is observed on the CF3 test instance, Fig 4, where MLSGA was not able to reach the front, and only gets close to it in the best case. For this function, as for the other CF functions, all variants perform similarly.

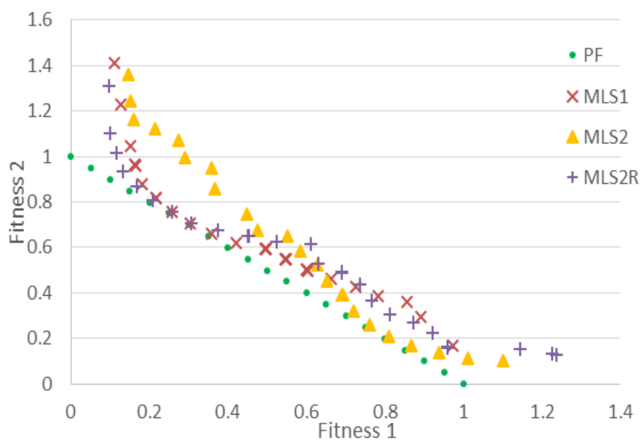

Fig. 3. Pareto front of MLSGA variants on CF1 function

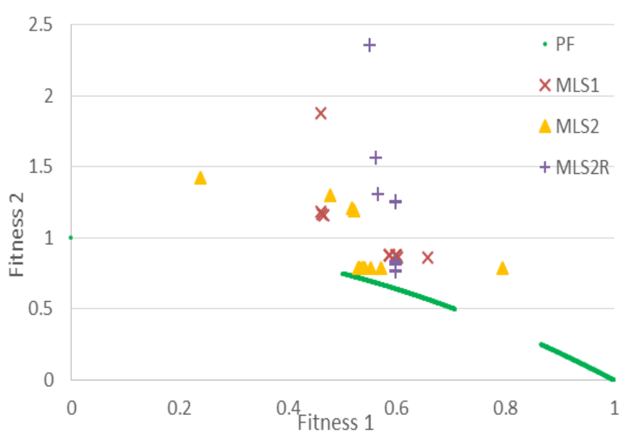

Fig. 4. Pareto front of MLSGA variants on CF3 function

The results of the three variants of MLSGA are compared to each other and to the other algorithms from CEC '09 [20]. The rankings are presented for unconstrained test cases in Table IV for UF1-3, in Table V for UF 4,5 and 7, and for constrained functions CF1-4, Table VI, and CF5-7, Table VII.

TABLE IV. MLSGA IN CEC'09 RANKING ON TWO-OBJECTIVE UNCONSTRAINED UF1-3 PROBLEMS

Rank	Name/Average IGD		
	UF1	***UF2***	***UF3***
1	MOEA/D 0.00435	MTS 0.00615	MOEA/D 0.00742
2	GDE3 0.00534	MOEA/D-GM 0.0064	LiuLi 0.01497
3	MOEA/D-GM 0.0062	DMOEA-DD 0.00679	DMOEA-DD 0.03337
4	MTS 0.00646	MOEA/D 0.00679	MOEA/D-GM 0.049
5	LiuLi 0.00785	OW-MOSaDE 0.0081	MTS 0.0531
6	DMOEA-DD 0.01038	GDE3 0.01195	Clustering MOEA 0.0549
7	NSGA II LS 0.01153	LiuLi 0.0123	AMGA 0.06998
8	OW-MOSaDE 0.0122	NSGA II LS 0.01237	DECMOSA-SQP 0.0935
9	Clustering MOEA 0.0299	AMGA 0.01623	MOEP 0.099
10	AMGA 0.03588	MOEP 0.0189	OW-MOSaDE 0.103
11	MOEP 0.0596	Clustering MOEA 0.0228	NSGA II LS 0.10603
12	DECMOSA-SQP 0.07702	DECMOSA-SQP 0.02834	GDE3 0.10639
13	OMOEA II 0.08564	OMOEA II 0.03057	OMOEA II 0.27141
14	MLS2R 0.205	MLS2 0.105	MLS1 0.401
15	MLS1 0.213	MLS1 0.156	MLS2R 0.438
16	MLS2 0.223	MLS2R 0.176	MLS2 0.447

TABLE V. MLSGA IN CEC'09 RANKING ON TWO-OBJECTIVE UNCONSTRAINED UF4-7 PROBLEMS

Rank	Name/Average IGD		
	UF4	***UF5***	***UF7***
1	MTS 0.02356	MTS 0.01489	MOEA/D 0.00587
2	GDE3 0.0265	GDE3 0.03928	LiuLi 0.0073
3	DECMOSA-SQP 0.03392	AMGA 0.09405	MOEA/D-GM 0.0076
4	AMGA 0.04062	LiuLi 0.16186	DMOEA-DD 0.01032
5	DMOEA-DD	DECMOSA-SQP	MOEP

Rank	Name/Average IGD		
	UF4	*UF5*	*UF7*
	0.04268	0.16713	0.0197
6	MOEP 0.0427	OMOEA II 0.1692	NSGA II LS 0.02132
7	LiuLi 0.0435	MOEA/D 0.18071	Clustering MOEA 0.0223
8	OMOEA II 0.04624	MOEP 0.2245	DECMOSA-SQP 0.02416
9	MOEA/D-GM 0.0476	Clustering MOEA 0.2473	GDE3 0.02522
10	OW-MOSaDE 0.0513	DMOEA-DD 0.31454	OMOEA II 0.03354
11	NSGA II LS 0.0584	OW-MOSaDE 0.4303	MTS 0.04079
12	Clustering MOEA 0.0585	NSGA II LS 0.5657	AMGA 0.05707
13	MOEA/D 0.06385	MLS2 0.696	OW-MOSaDE 0.0585
14	MLS2R 0.0881	MLS1 0.733	MLS1 0.398
15	MLS1 0.0882	MLS2R 0.767	MLS2 0.410
16	MLS2 0.103	MOEA/D-GM 1.7919	MLS2R 0.412

The unconstrained rankings show that all the MLSGA variants are outperformed by all other algorithms on almost all of the functions, except UF5 where MOEA/D-GM is the worst; additionaly in this case the MLS2 variant is not far from the NSGA II LS. Interesingly all MLS varaints show similar performance in all tests, and it's hard to distinguish which one is better, which is in contradiction with previous research on the simpler ZDT functions, where on these simpler functions MLS2R variant strongly outperforms the others and MLS1 provides poor diversity of points.

For the constrained ranking, the results are more promising. In this case MLSGA variants are outperformed by all other algorithms only on the CF1 and CF4 benchmark. However, MLS2R is better than both DECMOSA and MOEA/D-GM on CF3 and CF5; MLS1 outperforms these algorithms on CF5 and CF6. Again all the MLSGA variants have similar performance, but MLS1 is slightly better, which is surprising in comparison to the unconstrained ZDT functions, where the MLS1 algorithm lacks the diversity of points to get a low IGD value.

The detailed Pareto front results show that MLSGA is unable to properly find the whole Pareto front, however it is able to reach it and explore it partially, providing promising results at this stage, especially for constrained problems. The current algorithm only implements simple genetic algorithm mechanics at the individual level and the collective level mechanics are at an early stage of development but is already outperforming some leading algorithms on a range of state-of-the-art problems.

TABLE VI. MLSGA IN CEC'09 RANKING ON TWO-OBJECTIVE CONSTRAINED CF1-4 PROBLEMS

Rank	Name/Average IGD			
	CF1	*CF2*	*CF3*	*CF4*
1	LiuLi 0.00085	DMOEADD 0.0021	DMOEADD 0.056305	DMOEADD 0.00699
2	NSGAIILS 0.00692	LiuLi 0.0042	MTS 0.10446	GDE3 0.00799
3	MEOADGM 0.0108	MEOADGM 0.008	GDE3 0.127506	MTS 0.01109
4	DMOEADD 0.01131	NSGAIILS 0.01183	LiuLi 0.182905	LiuLi 0.01423
5	MTS 0.01918	GDE3 0.01597	NSGAIILS 0.23994	NSGAIILS 0.01576
6	GDE3 0.0294	MTS 0.02677	MLS2R 0.435	MEOADGM 0.0707
7	MLS2R 0.0872	DECMOSA 0.0946	MEOADGM 0.5134	DECMOSA 0.15265
8	MLS1 0.0933	MLS1 0.160	MLS1 0.522	MLS1 0.332
9	MLS2 0.0996	MLS2R 0.164	MLS2 0.633	MLS2R 0.348
10	DECMOSA 0.10773	MLS2 0.1829	DECMOSA 1000000	MLS2 0.387

TABLE VII. MLSGA IN CEC'09 RANKING ON TWO-OBJECTIVE CONSTRAINED CF5-7PROBLEMS

Rank	Name/Average IGD		
	CF5	*CF6*	*UF7*
1	DMOEADD 0.01577	LiuLi 0.013948	DMOEADD 0.01905
2	MTS 0.02077	DMOEADD 0.01502	MTS 0.02469
3	GDE3 0.06799	MTS 0.01616	GDE3 0.04169
4	LiuLi 0.10973	NSGAIILS 0.02013	LiuLi 0.10446
5	NSGAIILS 0.1842	GDE3 0.06199	NSGAIILS 0.23345
6	MLS1 0.390	MLS1 0.141	DECMOSA 0.26049
7	MLS2R 0.393	DECMOSA 0.14782	MLS2R 0.522116
8	DECMOSA 0.41275	MEOADGM 0.2071	MEOADGM 0.5356
9	MLS2 0.426	MLS2 0.502	MLS1 0.538741
10	MEOADGM 0.5446	MLS2R 2.746	MLS2 0.571981

IV. Discussion

The results show that MLSGA provides promising results on complex two-objective problems, representing the current state-of-art. On all the test cases, each MLSGA variant has a similar performance and is able to reach the true Pareto front, in some cases with a good diversity of points. This increases the performance of the classic genetic algorithm, on which the current version of MLSGA is based, to compete with current state-of-the-art genetic algorithms. Lack of a substantial difference between MLS2 and MLS2R was predictable as all UF and CF test cases have similar complexity for both f_1 and f_2. This is different from the ZDT functions tested on previously where performance between the variants showed a large difference, but these functions are less well balanced between objectives. More interesting is the similarity in performance between MLS1 and MLS2 variants, which also differs from the previous results. The authors suggest that the reason behind this is that the UF test problems are too complex for the current, simple version of MLSGA and both variants are struggling in finding the proper solutions. MLS1 results have previously shown poor diversity of solutions whereas MLS2 finds a diverse range of solutions once the front is found. In these cases, the front is more difficult to find so the MLS2 variant is not able to create a spread of results, showing similar performance to MLS1.

MLSGA is based on simple principles and thus further work is necessary. Possible methods to improve the performance are to develop the new collective evolutionary mechanism, which is underdeveloped compared to individual level mechanisms, and employ better performing methods than the simple classic GA mechanisms. The authors suggest to incorporate top individual mechanisms, such as MOEA/D, NSGAII and DMOEA-DD into MLSGA for even greater gains in performance. Other approaches could be implementation of local search methods such as MTS into collective reproduction mechanisms, and the introduction of different MLS types and fitness definitions into each collective, where some collectives in one run can utilise MLS1 and others MLS2 or MLS2R, for wider area searches. Additionally, the authors suggest that the MLSGA approach may be more successful on other kind of problems including: multi-level functions, noisy and dynamic problems, where the split in fitness function and population may lead to increased performance.

V. Conclusion

There are a number of different approaches to improve performance of genetic algorithms. This paper implements a novel bio-inspired mechanism, based on modern theories of evolution, multi-level selection. A simple genetic algorithm is used at the individual level and an initial, novel, collective level mechanism is implemented; these are benchmarked on CEC'09 test problems and compared with current state-of-art competitors. Interestingly all variants are able to reach the true Pareto front and show similar performance to each other. The developed algorithm needs further improvements, however the current mechanisms are simple and it is proposed to introduce more complex individual reproduction from the current state-of-art to improve performance.

References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, 2002

[2] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," pp. 95–100, 2001.

[3] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," The 1st international Conference on Genetic Algorithms. pp. 93–100, 1985.

[4] D. Corne, J. D. Knowles, and M. Oates, "The Pareto envelope-based selection algorithm for multiobjective optimization," Parallel Probl. Solving from Nature-PPSN VI, no. Mcdm, pp. 839–848, 2000.

[5] N. Srinivas and K. Deb, "Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," Evol. Comput., vol. 2, no. 3, pp. 221–248, 1994.

[6] R. Hinterding, Z. Michalewicz, and T. C. Peachey, "Self-Adaptive Genetic Algorithm for Numeric Functions," Proc. 4th Int. Conf. Parallel Probl. Solving from Nat., vol. 1141, pp. 420–429, 1996.

[7] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," Evol. Comput. IEEE Trans., vol. 3, no. 4, pp. 257–271, 1999.

[8] H. Seade, K. Deb, "U-NSGA-III: A unified evolutionary optimization procedure for single, multiple, and many objectives: proof-of-principle results", 8th International Conference Evolutionary Multi-Criterion Optimisation, EMO 2015, Guimares, Portugal, 2015.

[9] J. D. Knowles and D. W. Corne, "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy," Evol. Comput., vol. 8, no. 2, pp. 149–172, 2000.

[10] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," IEEE Trans. Evol. Comput., vol. 11, no. 6, pp. 712–731, 2007.

[11] M. Liu, X. Zou, C. Yu, and Z. Wu, "Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances," IEEE Congr. Evol. Comput., no. 1, pp. 2913–2918, 2009.

[12] H. L. Liu and X. Li, "The multiobjective evolutionary algorithm based on determined weight and sub-regional search," 2009 IEEE Congr. Evol. Comput. CEC 2009, pp. 1928–1934, 2009.

[13] V. L. Huang, S. Z. Zhao, R. Mallipeddi, and P. N. Suganthan, "Multi-objective optimization using self-adaptive aifferential evolution algorithm," 2009 IEEE Congr. Evol. Comput., pp. 190–194, 2009.

[14] Y. Wang, C. Dang, H. Li, L. Han, and J. Wei, "A clustering multi-objective evolutionary algorithm based on orthogonal and uniform design," 2009 IEEE Congr. Evol. Comput., no. 5, pp. 2927–2933, 2009.

[15] S. Tiwari, G. M. Fadel, P. Koch, and K. Deb, "AMGA : An Archive-based Micro Genetic Algorithm," Gecco, pp. 729–736, 2008.

[16] B. Y. Qu and P. N. Suganthan, "Multi-objective evolutionary programming without non-domination sorting is up to twenty times faster," 2009 IEEE Congr. Evol. Comput. CEC 2009, pp. 2934–2939, 2009.

[17] A. Zamuda, J. Brest, B. Boskovic, and V. Zumer, "Differential evolution with self-adaptation and local search for constrained multiobjective optimization," 2009 IEEE Congr. Evol. Comput. CEC 2009, pp. 195–202, 2009.

[18] S. Gao et al., "An orthogonal multi-objective evolutionary algorithm with lower-dimensional crossover," 2009 IEEE Congr. Evol. Comput. CEC 2009, no. 1, pp. 1959–1964, 2009.

[19] S. Kukkonen and J. Lampinen, "GDE3: the third evolution step of generalized differential evolution," 2005 IEEE Congr. Evol. Comput., vol. 1, pp. 443–450, 2005.

[20] Q. Zhang and P. N. Suganthan, "Final report on CEC ' 09 MOEA Competition," Congress on Evolutionary Computation, pp. 1-11, 2009.

[21] E. Sober, D. S. Wilson, "Unto others: evolution and psychology of unselfish behaviour", Harvard University Press, 1999.

[22] S. Okasha, "Evolution and the Levels of Selection", Clarendon Press, 2006.

[23] T. Lenaerts, A. Defaweux, P. Van Remortel, and B. Manderick, "Modeling Artificial Multi-level Selection," AAAI Technical Report SS-03-02, 2003.

[24] R. Akbari, V. Zeighami, and K. Ziarati, "MLGA: A multilevel cooperative genetic algorithm," Proc. 2010 IEEE 5th Int. Conf. Bio-Inspired Comput. Theor. Appl. BIC-TA 2010, pp. 271–277, 2010.

[25] R. Akbari and K. Ziarati, "A multilevel evolutionary algorithm for optimizing numerical functions," Int. J. Ind. Eng. Comput., vol. 2, no. 2, pp. 419–430, 2011.

[26] S. X. Wu, S. John, W. Banzhaf, I. P. Solving, and C. Methods, "A Hierarchical Cooperative Evolutionary Algorithm," Group, pp. 233–240, 2010.

[27] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," Evol. Comput., vol. 8, no. 2, pp. 173–195, 2000.

[28] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, and W. Liu, "Multiobjective optimization pest instances for the CEC 2009 special session and competition," Tech. Rep., pp. 1–30, 2009.