Searching Under Multi-Evolutionary Pressures

Hussein A. Abbass[†] and Kalyanmoy Deb[‡]

† Artificial Life and Adaptive Robotics (A.L.A.R.) Lab, School of Computer Science,
University of New South Wales, Australian Defence Force Academy Campus,
Canberra, Australia. {h.abbass@adfa.edu.au}
‡ Mechanical Engineering Department, Indian Institute of Technology, Kanpur,
Kanpur, PIN 208 016, India. {deb@iitk.ac.in}

Abstract. A number of authors made the claim that a multiobjective approach preserves genetic diversity better than a single objective approach. Sofar, none of these claims presented a thorough analysis to the effect of multiobjective approaches. In this paper, we provide such analysis and show that a multiobjective approach does preserve reproductive diversity. We make our case by comparing a pareto multiobjective approach against a single objective approach for solving single objective global optimization problems in the absence of mutation. We show that the fitness landscape is different in both cases and the multiobjective approach scales faster and produces better solutions than the single objective approach.

Keywords: artificial evolution; diversity; multiobjective optimization.

1 Introduction

In biological systems, natural selection is carried out on multi-traits that are usually in conflict. In real life animal breeding, a farmer selects those animals to cull or mate based on a multi-trait selection index. Yet, in applying artificial evolutionary systems to solve real life problems, researchers deviate from the biological phenomena claiming that a biological characteristic may not be the idle choice for a computational problem. Although this assertion contains some truth, it is somehow misused. An example for this is the use of very high mutation rates in evolutionary optimization algorithms to overcome the premature convergence problem due to loss of genetic diversity. As such, it seems that using very high mutation rates is becoming a common practice that sometimes makes us wonder whether the process is evolutionary or merely a random search? Mutation rates in biological systems such as human is extremely low (varies between 1 in 10,000 and 1 in 1,000,000 genes within a gamete). The largest mutation rate is 1 in 10,000 and is associated with neurofibromatosis type I, a cancer of the nervous system. A natural question which arises here is: why premature convergence did not occur in biological systems? Is it because of Niching, Speciation, or something else?

Another similar problem exists in genetic programming, where some papers use large populations in the order of 500-1000 individuals and the artificial evolution is carried out only for a few generations 30-50 to overcome premature convergence. A debate here is whether 30-50 generations are enough for evolution to find its way in a population size of 1000 and in very large search spaces. An obvious question is whether the final result is actually because of evolutionary forces or because of luck and random factors.

In an attempt to address some of these questions, we present a multiobjective evolutionary approach for solving single-objective global optimization problems. Our objective is not to invent a better method, although we will see that the proposed method is performing well. Rather, we will show that searching under multiple evolutionary pressures alone, in the absence of mutation, maintains a considerable amount of genetic diversity evidenced by the performance of the proposed approach on three global optimization hard functions. We will also show that the proposed approach exhibits better scalability. Our analysis will also cover a comparison to the fitness landscape as well as the evolutionary dynamics for multiple and single evolutionary pressures.

The research questions that this paper attempts to answer some of them are:

1. Is multiobjectivity a force that balances the high selection pressure in evolutionary computations?
2. Is multiobjectivity a diversity preserving and/or promoting mechanism?
3. How does the evolutionary dynamics compares between multiple and single selection pressures?
4. How does the fitness landscape compares between multiple and single selection pressures?
5. Should the analysis for neutral paths be within a multi-selection pressure framework in the absence of mutation?

In [7] a multiobjective approach is used for maintaining diversity in genetic programming. The second objective in this approach was an explicit diversity measure; therefore it is difficult to judge whether the good results are because of the multiobjective component (that evolution is searching in two dimensional objective space) or because the explicit inclusion of a diversity objective. In [4], a multiobjective approach is used once more for genetic programming. The second objective was to minimize the program size, which is an implicit diversity preserving mechanism. However, as the authors compare a number of different techniques, it is difficult to isolate the contribution of the multiobjective component from other evolutionary operators in the evolutionary method.

Knowles and Corne [8] also proposed the use of multiobjective optimization to solve single objective problems. They have shown that the multiobjective approach reduces the number of local optimum. However, their approach requires either decomposing the original objective or the decision space. Apart from the fact that the objective of this paper is different, the proposed approach in this paper does not require decomposing neither the objective nor the decision space.

In our experiments, we avoided using mutation neither comparing our results against traditional evolutionary algorithms with mutation operator. To this end, it is important that we differentiate between a diversity promoting mechanism and a diversity preserving or maintaining mechanism. The former, such as the use of mutation, is a continuous source of variations to the population. The latter, does not add variations to the population, but slows down the selection pressure and maintains the diversity which already exists in the population.

Diversity preserving mechanisms include low selection pressure, large populations and as we present in this paper, multi–evolutionary pressures. In the absence of a diversity promoting mechanism, the only source of diversity is the initial population. In other words, these methods slow down the side–effect of the selection pressure but they do not add new variations. By using a good diversity preserving mechanism, we may not need high mutation.

In this paper, we present a first attempt to scrutinize the effect of a multiobjective approach on the fitness landscape of a problem as well as its contribution to the diversity in the population. The rest of this paper is divided as follows: in Section 2, the methods are introduced followed by the experimental setup and results in Sections 3 and 4 respectively. Conclusions are drawn in Section 5.

2 Methods

A variation of the Self–adaptive Pareto–frontier Differential Evolution (SPDE) algorithm [1] is used. We present this variation here followed by a similar algorithm using a *Breeder Genetic Algorithm* (BGA) [9, 10] selection strategy for single objective evolutionary optimization.

2.1 The multiobjective algorithm: The SPDE Algorithm

SPDE is a variation of PDE [2, 3] where both crossover and mutation rates self–adapt. In the current paper, we do not allow for mutation. The crossover rate is inherited from the parents in the same way crossover is undertaken for the decision variables. Here, a child is generated from three parents as follows

$$x^{child} \leftarrow x^{\alpha_1} + F \times (x^{\alpha_2} - x^{\alpha_3}) \tag{1}$$

where, $x^{\alpha_1}, x^{\alpha_2}, x^{\alpha_3}$ are three different genotypes and F is a step.

If the maximum number of non–dominated solutions in a generation is greater than the user specified maximum, the following nearest neighbor distance function is adopted:

$$D(x) = \frac{(min||x - x^i|| + min||x - x^j||)}{2}, \tag{2}$$

where $x \neq x^i \neq x^j$. That is, the nearest neighbor distance is the average Euclidean distance between the closest two points. The non–dominated solution with the smallest neighbor distance is removed from the population until the total number of non–dominated solutions is retained to the user specified maximum.

If some variables in the child fall outside their range, a repair rule is used. The rule is simply to multiply the variable by a random number from a uniform distribution in the range 0–1 till the variable retains its original bound. A generic version of the SPDE algorithm used in this paper follows:

1. Create a random initial population of potential solutions. Each variable is assigned a random value according to a a Gaussian distribution $N(\mu, \sigma)$, where μ is the mean of the variable's range and σ is usually taken to be one sixth of the variable possible range.
2. Repeat
   (a) Evaluate the individuals in the population and label those who are non–dominated.
   (b) If the number of non–dominated individuals in the population is less than 3 repeat the following until the number of non–dominated individuals in the population is greater than or equal to 3.
      i. Find a non–dominated solution among those who are not labelled.
      ii. Label the solution as non–dominated.
   (c) If the number of non–dominated individuals in the population is greater than the allowed maximum, apply the neighborhood distance function (Equation 2) until the number of non–dominated individuals in the population is less than the allowed maximum.
   (d) Delete all dominated solutions from the population.
   (e) Repeat
      i. Select at random an individual as the main parent α_1, and two individuals, α_2 and α_3 as supporting parents.
      ii. Select at random a variable j.
      iii. **Crossover rate:** Let the crossover rate for the child be

$$x_c^{child} \leftarrow x_c^{\alpha_1} + F \times (x_c^{\alpha_2} - x_c^{\alpha_3}) \tag{3}$$

      If the crossover rate is negative, make it positive. If the crossover rate is greater than 1, let it be $0.5 * (x_c^{child} - 1)$.

        iv. **Crossover:** For each variable i
            With some probability $Uniform(0,1) > x_c^{child}$ or if $i = j$, do

$$x_i^{child} \leftarrow x_i^{\alpha_1} + F \times (x_i^{\alpha_2} - x_i^{\alpha_3}) \quad (4)$$

            otherwise

$$x_i^{child} \leftarrow x_i^{\alpha_1} \quad (5)$$

            where each variable i in the main parent, $x_i^{\alpha_1}$, is perturbed by adding to it a ratio, $F \in Gaussian(0,1)$, of the difference between the two values of this variable in the two supporting parents. At least one variable must be changed.
        v. If the child dominates the main parent, place the child into the population.
    (f) Until the population size is M
3. Until the maximum number of objective evaluations allowed is reached.

Sofar, we did not mention what the second objective is for the multi–pressure case. In designing our investigation into this problem, we were faced with the question of how can we transform a single objective unconstrained global optimization problem into a multiobjective one? A number of measures which may preserve diversity explicitly and are relative to each population is:

1. Entropy of the population (*ie.* Shannon's entropy term)
2. Mean variance of variables in the population
3. Information content (as introduced in Section 4.4)

However, we need to find a natural objective which may preserve diversity implicitly but not a direct measure for diversity. Also, we need to find a measure on the level of the individual rather than the population level. Assuming a global minimization problem, a number of options include:

1. Maximizing the reverse of the original objective.
2. Maximizing/minimizing a random value assigned to each chromosome as its second fitness.
3. Maximizing the age of the chromosome; analogue to reality where the age reflects the experience.

From a feasibility study to investigate the three options, the first option was the worst in terms of computational time because there was a large number of pareto–optimal solutions at each generation. Therefore, the neighborhood function needs to be applied a large number of times to cut down the number of pareto–optimal solutions to the user predefined number. The second was performing well but the third gave the best performance in terms of the quality of solutions and processing time. In this paper, we report the results obtained by the third option of maximizing the age.

The age of a chromosome is a fixed value assigned to the chromosome at the time it is created. For the first generation, the year of birth index starts with 1. Each time a chromosome is generated in the initial population, the year of birth index is incremented with 1. By the time the initial population is initialized, the year of birth ranges from 1 to the population size. In subsequent generations, the year of birth is increased by one; that is, all solutions to be generated in a subsequent generation will share the same year of birth, which is population size + generation number.

A possible explanation for the good performance of the age over other objective functions is, the age allows old solutions (presumably not so good) to be maintained in the population, thereby maintaining diversity and slowing down the selection pressure.

2.2 The single objective algorithm

A similar algorithm to SPDE is used with two variations; the first is instead of maintaining at most P pareto–optimal solutions - which we do not have any more - we maintain the top P best solutions similar to the BGA strategy. The second variation is we dropped the condition of placing the child into the population if it dominates the parent. We simply place the child into the population. If we place the child into the population only when it is better than the parent, we got very bad results because it increased the evolutionary pressures.

3 Test suite and experimental setup

We tested the performance using three global optimization problems; these are: De Jong, Griewangk, and Rastrigin functions [5, 6]. The three functions are given below.

1. De Jong's F1 function:

$$f(x) = \sum x_i^2, i = 1 \ldots n, -5.12 \leq x_i \leq 5.12$$

   De Jong's F1 function is a unimodal function with a unique minimum at $x_1 = x_2 = 0$. It should be simple for a genetic algorithm as well as any hill–climbing algorithm with a proper neighborhood operator.

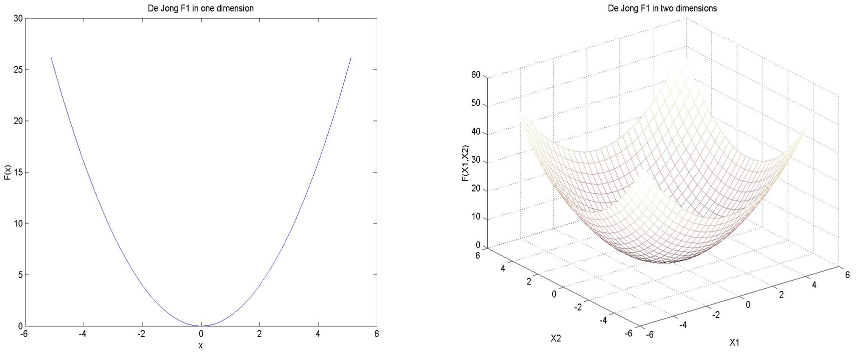

Fig. 1. De Jong function in one and two dimensions.

2. Griewangk's function:

$$f(x) = \sum \frac{x_i^2}{4000} - \prod \cos \frac{x_i}{\sqrt{i}} + 1, i = 1 \ldots n, -600 \leq x_i \leq 600$$

In this function, the local optima are above the parabola level, which is produced by the summation term. The more we increase the search range, the flatter the function. This function is supposed to be difficult for a genetic algorithm because the product term causes the variables to be highly interdependent.

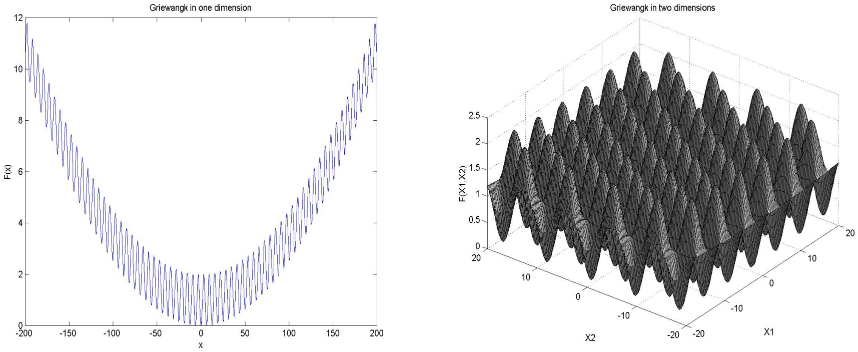

Fig. 2. Griewangk function in one and two dimensions.

3. Rastrigin's function:

$$f(x) = A \times n + \sum (x_i^2 - A \times \cos(2 \times \pi \times x_i)), i = 1 \ldots n, -5.12 \leq x_i \leq 5.12$$

This function is based on De Jong's F1 with the addition of the cosine term which creates a large number of local minima. The amplitude of the function's surface is controlled by the parameter A. When A is 10 (as in our experiments) the domain is highly multimodal. The local minima are located at a rectangular grid with size 1. With increasing distance to the global minimum the fitness values of local minima become larger.

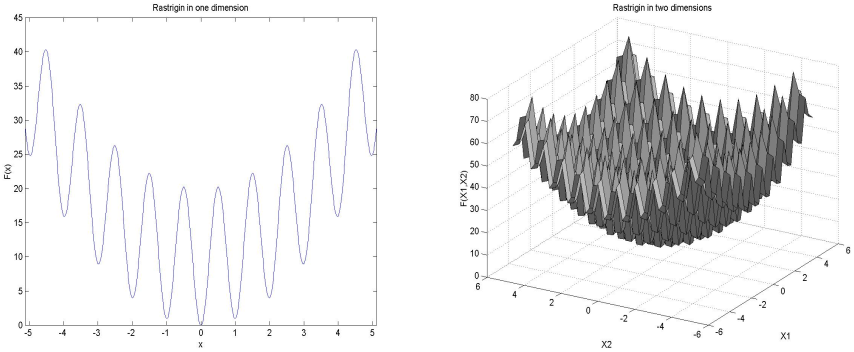

Fig. 3. Rastrigin function in one and two dimensions.

In the rest of this paper, we will refer to the single and multiobjective algorithms as algorithm1 and algrotihm2 respectively. The objective of our experiments is to analyze the differences between searching under single and multiple evolutionary pressures. We would also like to find if multiobjectivity preserves reproductive diversity. Therefore, in all of our experiments, we use the same initial populations for both algortihm1 and algorithm2. Mutation is not used in any experiment to avoid any source of diversity.

Each problem is run with 10, 20, 30, 40, and 50 variables to measure the scalability of each algorithm. A population size of 100 is used and the step size F is 0.1. The maximum number of pareto–optimal solutions for algorithm2 and the number of solutions cloned for algorithm1 are 50. The maximum number of objective evaluations is 1 million and the evolutionary run is terminated once this maximum is reached. Each experiment is run 10 times and the same ten seeds were used for all experiments.

4 Results

In Figure 4, the average and standard deviation of the best solutions found in the ten runs are reported for the three problems with the different number of variables. It is clear from the figure that the quality of solutions obtained by algorithm2 is much better than those obtained by algorithm1. The scalability of algorithm2 is notably better than algorithm1. The figure demonstrates that searching under multiple evolutionary pressures produced better solutions than searching under single evolutionary pressures.

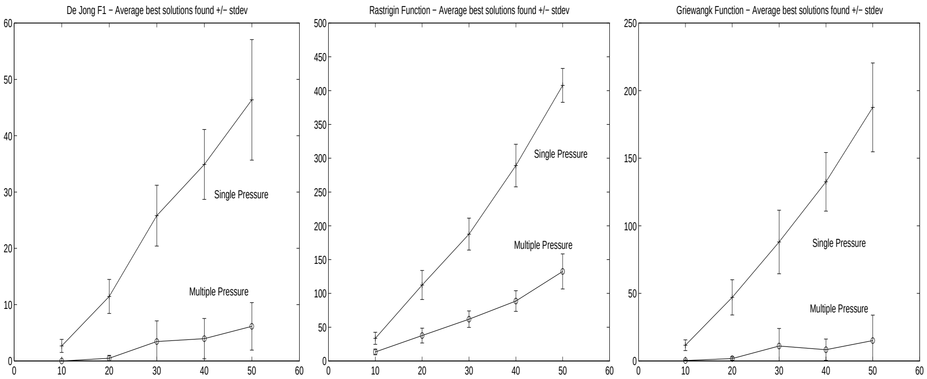

Fig. 4. The average best solution found in ten different runs for the three differen problems with dimensions of 10, 20, 30, 40, and 50.

To scrutinize the effect of multiobjective evolutionary search and contrast it with the single–pressure evolutionary search, we will present a detailed analysis in the following sections using the Griewangk function. The results are consistent on the other problems.

4.1 Convergence

Figure 5 presents the best and average fitness over 300 generations for both algorithm1 and algorithm2. Premature convergence is very clear for algorithm1, where the best and average fitness stagnated very early in the search. The very bad solution quality is also clear from the figure. For algorithm2, we can see that some runs reached the global optimum. It is also apparent that the average fitness is not decaying over time, but fluctuating; therefore, this entails that the algorithm is exploring the search space better caused by the multiobjective pareto selection method. Convergence in algorithm2 occurred between 150 and

250 generations.

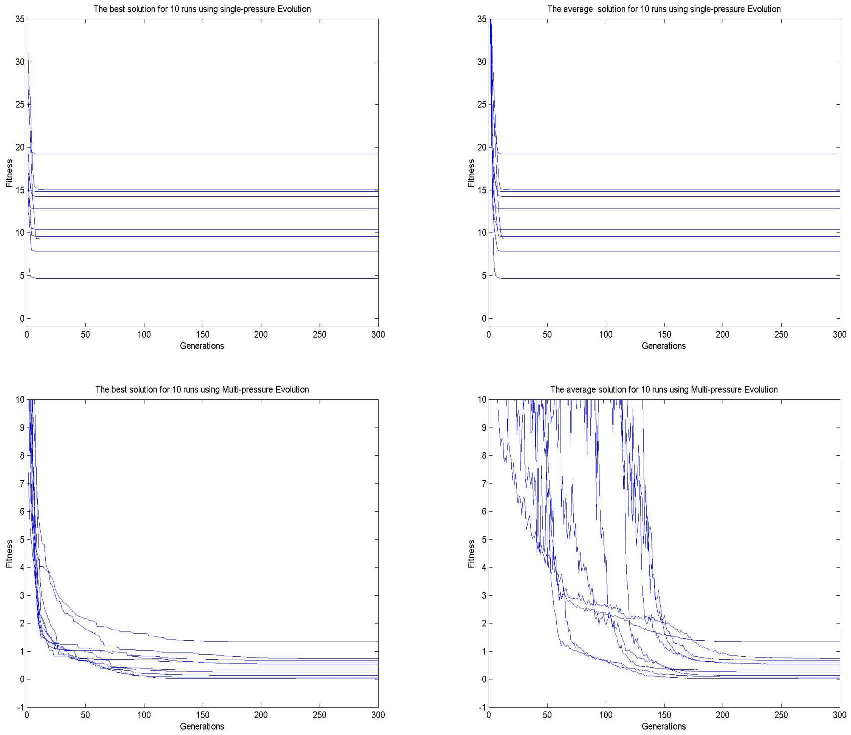

Fig. 5. The best (on left) and average (on right) solutions found in ten different runs for the Griewangk's problem with 10 variables for single (top graphs) and multiple (bottom graphs) pressures.

4.2 Population trajectories and Diversity

In Figure 6, we present the progress of genotype diversity in the evolution. We define genotype diversity as the average distance between all genotypes in a population and the population centroid. We also present the population trajectories represented by the change of the population centroid overtime.

We measured the average correlation coefficient between the diversity and the movement of the population over the ten runs. For the multiple pressure algorithm, the correlation was 0.79 for the first 200 generations and 0.78 for the single pressure algorithm over the first 50 generations. This may entail that the

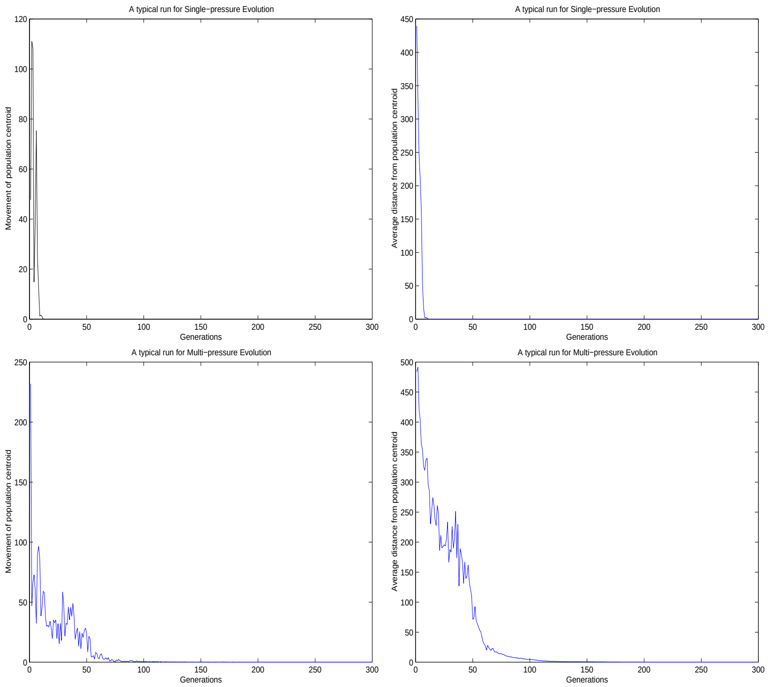

Fig. 6. The population trajectories (on left) and diversity measure (on right) for a typical run using the Griewangk's function for single (top graphs) and multiple (bottom graphs) pressures.

more the population shifts in the space, the more diversity it maintains.

The previous conclusion may not be entirely true. The population movement - measured by the change of the population centroid - can simply take place by deleting or injecting an outlier which will also affect the diversity measure. Therefore, we measured the average correlation coefficient between the population diversity and the best and worst solution for each algorithm. For algorithm1, the population diversity was highly correlated with the best and worst solutions (0.95 for each case). For algorithm2, however, the population diversity was highly correlated with worst solution (0.88) and less with the best (0.72). These findings can also be seen in Figure 7, which depicts a typical run. It is clear that the average fitness is closer to the worst solution than to the best for algorithm2. However, since the worst solution is not fluctuating and is monotonically decreasing, it cannot cause the oscillation that we see with the diversity and centroid's movement.

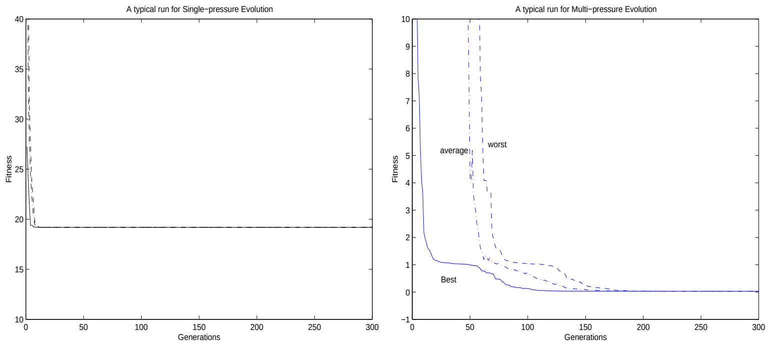

Fig. 7. A typical evolutionary run showing the best, average, and worst fitness for single pressure (left graph), and multiple pressures (right graph).

4.3 Fitness Landscape: fitness histograms

A fitness landscape is defined by the genotype representation, the fitness function, and the operators used to generate solutions in the neighborhood. In Figure 8, we present the fitness histogram for a hill–climbing algorithm, algorithm1, and algorithm2. In hill–climbing, the algorithm starts with a random solution, mutate it by adding a Gaussian random value $N(0, 10)$, if the resultant solution is better than the old one, it replaces it; otherwise a new solution is generated. The average ± standard deviation for the best solutions obtained by the hill–climbing algorithm in 10 runs are 0.550 ± 0.089. The corresponding numbers for algorithm1 and algorithm2 are 11.678 ± 3.978 and 0.365 ± 0.326 respectively. It is clear that algorithm1 was much worse than the hill–climbing algorithm while algorithm2 is better than both.

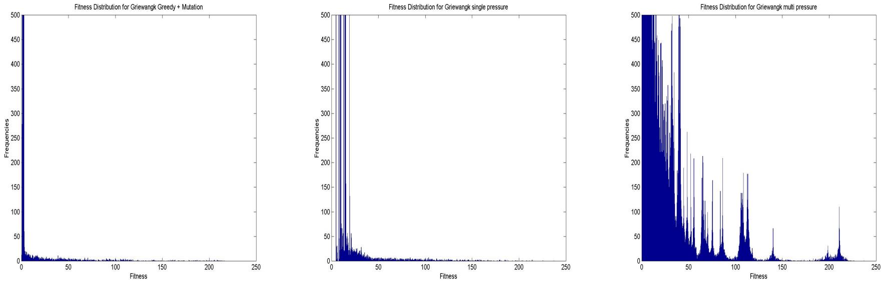

Fig. 8. The fitness histogram for the Griewangk's function using hill climbing (left graph), single pressure (middle graph), and multiple pressures (right graph).

4.4 Fitness Landscape: information contents

A rugged landscape is expected to have large fitness values between neighborhood points while a smooth landscape will have small differences in the fitness values. In this section, we employ [11] information theoretic measures for analyzing the fitness landscape. We use three measures defined in [11]; the information content, the partial information content, and the information stability. The first measure provides an insight into the variety of shapes on, or local neighborhood of, the landscape. The second measure gives an insight into the modality of the path. The third measure captures the upper bound of the magnitude of the landscape optima.

In Table 1, we provide the information contents, partial information contents, and expected number of optima for single pressure and multiple pressures evolutionary algorithms. We considered the evolutionary algorithm as a black box which generates a sequence of solutions. This makes it easy to apply these measures.

Table 1. The information contents and partial information contents for the Griewangk function using single pressure and multiple pressures methods.

Method	ϵ	Information Contents	Partial Information Contents	Expected Number of Optima
Single pressure	0	0.0034	0.0006	3302
	0.10	0.0017	0.0002	1398
	0.50	0.0016	0.0002	1216
	10.0	0.0008	0.0001	450
	90.0	0.0001	0.0	40
	100.0	0.0000	0.0	0
Multiple pressure	0	0.4261	0.2171	1085560
	0.10	0.0629	0.0182	91162
	0.50	0.0592	0.0141	70633
	10.0	0.0321	0.0061	30361
	100.0	0.0051	0.0006	2957
	210.0	0.0002	0.0000	42
	220.0	0.0000	0.0000	0

The expected number of local optima is simply half the partial information content multiplied by the size of the sample. When ϵ was zero, algorithm2 has the highest information content and partial information content.

It is worth mentioning that ϵ is a tolerance factor which flattens the landscape. When ϵ is zero, the measure is very sensitive to any variations and when ϵ is high, most of the landscape is flat. Information stability is the value of ϵ where the information content measure in the third column of the table reaches 0. It is clear from the table that the highest information stability is achieved with the multiple pressure algorithm, where ϵ is around 220.

4.5 A fair comparison

It may sound that the previous comparison was not that fair since the single pressure algorithm did not have obvious diversity preserving mechanism. To show that this is not entirely true, we did experiments with the single pressure algorithm while varying the selection pressure. The selection pressure is varied as follows. We used the breeder genetic algorithm strategy in the previous experiments by cloning the best 50 individuals and generating additional 50 children by breeding among them. Here, to reduce the selection pressure, we define a parameter ξ which represents the best ξ individuals. When ξ is 10, we clone 50 individuals from current population. These 50 are formed from the best 10 individuals and the other 40 are chosen at random without replacement. When selecting parents from these 50, we simply select parents at random regardless of their fitness.

Table 2. The results under different selection pressures using the single pressure algorithm.

ξ	Best solution
1	10.365 ± 5.770
2	10.564 ± 4.456
3	9.780 ± 5.759
4	9.880 ± 5.793
5	8.090 ± 2.470
6	9.998 ± 5.151
7	11.353 ± 5.520
8	10.181 ± 3.580
9	9.960 ± 5.234
10	11.528 ± 3.745
20	12.484 ± 4.426
30	12.473 ± 4.441
40	12.218 ± 4.951
50	11.987 ± 5.536

We varied ξ between 1 and 10 in a step of 1 then up to 50 in a step of 10. The results are given in Table 2. Surprisingly, the best result obtained when ξ was 5; that is, select the best five solutions to clone as parents then select the rest of the parents at random. This corresponds to a very low selection pressure, which entails that searching under single pressure is very sensitive to the selection pressure. Recalling from Section 4.3, the multi–pressure algorithm is still much better than the single pressure algorithm.

5 Conclusion

In this paper, we presented a multiobjective approach for solving single objective optimization problems. We have shown that the multiobjective approach preserves genetic diversity better than the single objective approach in the absence of mutation. The former also gave the best results. When analyzing the behavior of both algorithms, we found that the fitness landscapes for both algorithms are different. The multiobjective approach also gave the highest information contents.

References

1. H.A. Abbass. Self-adaptive pareto differential evolution. In *The IEEE Congress on Evolutionary Computation, Honolulu, USA*, pages 831–836. IEEE Press, 2002.
2. Hussein A. Abbass. A memetic pareto evolutionary approach to artificial neural networks. In M. Stumptner, D. Corbett, and M. Brooks, editors, *Proceedings of the 14th Australian Joint Conference on Artificial Intelligence (AI'01)*, pages 1–12, Berlin, 2001. Springer-Verlag.
3. Hussein A. Abbass. An evolutionary artificial neural network approach for breast cancer diagnosis. *Artificial Intelligence in Medicine*, 25(3):265–281, 2002.
4. S. Bleuler, M. Brack, L. Thiele, and E. Zitzler. Multiobjective genetic programming: Reducing bloat using SPEA2. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 536–543. IEEE Press, 2001.
5. K DeJong. *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, 1975.
6. V.S. Gordon and Darrell Whitley. Serial and parallel genetic algorithms as function optimization. In *The 5th International Conference on Genetic Algortihms*, pages 177–183. Morgan Kaufmann, 1993.
7. E. De Jong, R. Watson, and J. Pollack. Reducing bloat and promoting diversity using multi-objective methods. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 11–18, 2001.
8. J. Knowles and D. Corne. Reducing local optima in single-objective problems by multi-objectivization. In *First International Conference on Evolutionary Multi-criterion Optimization (EMO'01)*, pages 269–283. Springer-Verlag, 2001.
9. H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithms: continuous parameter optimization. *Evolutionary Computation*, 1(1):25–49, 1993.
10. H. Mühlenbein and D. Schlierkamp-Voosen. The science of breeding and its application to the breeder genetic algorithm bga. *Evolutionary Computation*, 1(4):335–360, 1994.
11. V.K. Vassilev, T.C. Fogarty, and J.F. Miller. Information characteristics and the structure of landscapes. *Evolutionary Computation*, 8(1):31–61, 2000.