# Dynamic Fitness Inheritance Proportion For Multi-Objective Particle Swarm Optimization

Margarita Reyes-Sierra and Carlos A. Coello Coello
CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Ingeniería Eléctrica, Sección Computación
Av. IPN No. 2508, Col. San Pedro Zacatenco, México D.F. 07360, México
mreyes@computacion.cs.cinvestav.mx,ccoello@cs.cinvestav.mx

## ABSTRACT

In this paper, we propose a dynamic mechanism to vary the probability by which fitness inheritance is applied throughout the run of a multi-objective particle swarm optimizer, in order to obtain a greater reduction in computational cost (than the obtained with a fixed probability), without dramatically affecting the quality of the results. The results obtained show that it is possible to reduce the computational cost by 32% without affecting the quality of the obtained Pareto front.

**Categories and Subject Descriptors:** I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search — *Heuristic Methods*; G.1.6 [**Numerical Analysis**]: Optimization.

**General Terms:** Algorithms, Performance.

**Keywords:** Fitness Inheritance, Multi-Objective Optimization, Particle Swarm Optimization.

## 1. INTRODUCTION

In fitness inheritance, originally proposed by Smith et al. [3], the fitness value of an offspring is obtained from the fitness values of its parents, with certain probability called *inheritance proportion* ($p_i$). This parameter ($p_i$) has to be fixed by the user and its value determines how much the computational cost is going to be reduced. We propose a mechanism to adapt the value of the inheritance proportion in a dynamic way throughout the run, in order to analyze how much can we reduce the computational cost without dramatically deteriorating the quality of the obtained results. The proposed approach is incorporated into a Multi-Objective Particle Swarm Optimization (MOPSO) algorithm previously proposed in [1], and tested using four well-known multi-objective test functions.

## 2. DESCRIPTION OF OUR APPROACH

The MOPSO algorithm used in this work uses Pareto dominance and a crowding factor to select and also to filter the list of available leaders. Also, this approach uses different mutation operators which act on different subdivisions of the swarm, and incorporates the $\epsilon$-dominance concept to fix the size of the set of final solutions produced by the algorithm. Figure 1 shows the pseudocode of this MOPSO algorithm (the symbol (⇒) indicates the line in which the concept of fitness inheritance is incorporated).

```
Begin
    Initialize swarm. Initialize leaders.
    Send leaders to ε-archive
    crowding(leaders), g = 0
    While g < gmax
        For each particle
            Select leader. Flight. Mutation.
⇒           If(p_i) Inherit Else Evaluation.
            Update pbest.
        EndFor
        Update leaders, Send leaders to ε-archive
        crowding(leaders), g++
    EndWhile
    Report results in ε-archive
End
```

**Figure 1: Pseudocode of the MOPSO algorithm.**

We use a fitness inheritance technique that calculates the new position of a particle in the objective space using the formula[1]: $\tilde{\mathbf{f}}_{\mathbf{i}}(t) = \tilde{\mathbf{f}}_{\mathbf{i}}(t-1) + \tilde{\mathbf{vf}}_{\mathbf{i}}(t)$, $\tilde{\mathbf{vf}}_{\mathbf{i}}(t) = C_1 r_1(\tilde{\mathbf{f}}_{\mathbf{pbest_i}} - \tilde{\mathbf{f}}_{\mathbf{i}}(t)) + C_2 r_2(\tilde{\mathbf{f}}_{\mathbf{gbest_i}} - \tilde{\mathbf{f}}_{\mathbf{i}}(t))$, where $\tilde{\mathbf{f}}_{\mathbf{i}}$, $\tilde{\mathbf{f}}_{\mathbf{pbest_i}}$ and $\tilde{\mathbf{f}}_{\mathbf{gbest_i}}$ are the values of the objective function $i$ for the current particle, its *pbest* and *gbest*, respectively.

Based on previous experiments, we concluded that the most important improvement throughout one run of our MOPSO approach takes place during the first quarter of the total of generations. Thus, we propose to set the value of the parameter $p_i$ dynamically with respect to the current generation number, in such a way that we increase the use of fitness inheritance throughout the evolutionary process. We propose six different functions (that we will call *adaptive functions*) to adapt the value of the inheritance proportion. Let *gen* be the number of the current generation and *Gmax* the total number of generations. Figure 2 presents a plot of the six adaptive functions. For each case $p_i = f(x)$.

## 3. RESULTS

We performed 30 runs using functions ZDT1, ZDT2, ZDT3 and ZDT4 [4]. The parameters used were 200 particles, 100 generations and 100 points in the final Pareto Front. We implemented two unary measures of performance: Suc-

[1]This technique was found to be the best among the techniques studied in [2].


*GECCO'06,* July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

**Table 1: Obtained results for all the test functions and all the adaptive functions.**

| **Function ZDT1** | | no-inherit | nonlinear1 | nonlinear2 | nonlinear3 | linear | nonlinear4 | nonlinear5 |
|---|---|---|---|---|---|---|---|---|
| **SCC** | mean | 87 | 84 | 74 | 71 | 68 | 53 | 21 |
| | st. dev. | 12.5 | 12.6 | 21 | 18.6 | 22.7 | 21.6 | 13.5 |
| **IGD** | mean | 0.00096 | 0.00096 | 0.00103 | 0.00313 | 0.00280 | 0.00388 | 0.00838 |
| | st. dev. | 0.00003 | 0.00003 | 0.00024 | 0.00890 | 0.00693 | 0.00979 | 0.01803 |
| **evaluations** | | 20200 | 16306 | 13640 | 10295 | 10303 | 6966 | 4319 |
| **savings** | | 0% | 19.3% | 32.5% | 49% | 49% | 65.5% | 78.6% |
| **Function ZDT2** | | no-inherit | nonlinear1 | nonlinear2 | nonlinear3 | linear | nonlinear4 | nonlinear5 |
| **SCC** | mean | 92 | 93 | 89 | 83 | 84 | 69 | 45 |
| | st. dev. | 12.9 | 6.1 | 12.2 | 21.7 | 22.9 | 26.6 | 34.2 |
| **IGD** | mean | 0.00067 | 0.00066 | 0.00067 | 0.00092 | 0.00078 | 0.00516 | 0.00378 |
| | st. dev. | 0.00006 | 0.00002 | 0.00004 | 0.00100 | 0.00053 | 0.01390 | 0.00904 |
| **evaluations** | | 20200 | 16304 | 13641 | 10295 | 10298 | 6968 | 4316 |
| **savings** | | 0% | 19.3% | 32.5% | 49% | 49% | 65.5% | 78.6% |
| **Function ZDT3** | | no-inherit | nonlinear1 | nonlinear2 | nonlinear3 | linear | nonlinear4 | nonlinear5 |
| **SCC** | mean | 76 | 73 | 72 | 53 | 59 | 37 | 16 |
| | st. dev. | 12.7 | 11.6 | 15.9 | 21.5 | 16.2 | 18 | 12.6 |
| **IGD** | mean | 0.00090 | 0.00101 | 0.00200 | 0.00742 | 0.00232 | 0.01085 | 0.01779 |
| | st. dev. | 0.00014 | 0.00027 | 0.00322 | 0.01375 | 0.00490 | 0.01371 | 0.01473 |
| **evaluations** | | 20200 | 16312 | 13622 | 10290 | 10304 | 6966 | 4336 |
| **savings** | | 0% | 19.2% | 32.6% | 49.1% | 49% | 65.5% | 78.5% |
| **Function ZDT4** | | no-inherit | nonlinear1 | nonlinear2 | nonlinear3 | linear | nonlinear4 | nonlinear5 |
| **SCC** | mean | 96 | 94 | 93 | 89 | 90 | 77 | 47 |
| | st. dev. | 4.8 | 6.6 | 6.0 | 12.6 | 14.2 | 18.1 | 22.6 |
| **IGD** | mean | 0.00096 | 0.00096 | 0.00096 | 0.00095 | 0.00100 | 0.00098 | 0.00124 |
| | st. dev. | 0.00002 | 0.00002 | 0.00002 | 0.00003 | 0.00003 | 0.00008 | 0.00032 |
| **evaluations** | | 20200 | 16287 | 13626 | 10315 | 10304 | 6958 | 4315 |
| **savings** | | 0% | 19.4% | 32.5% | 48.9% | 49% | 65.6% | 78.6% |

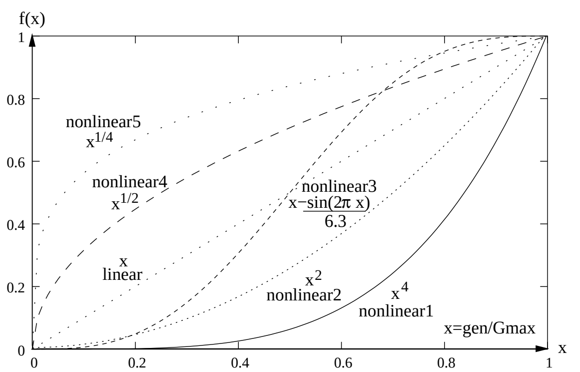


**Figure 2: Plot of the six adaptive functions proposed.**

cess Counting (SCC) and Inverted Generational Distance (IGD)[2]. Table 1 presents the results obtained.

As we can see in Table 1, it is possible to save even a 32% of evaluations (using adaptive functions nonlinear1 and nonlinear2) without significantly affecting the quality of the obtained solutions. Also, the quality of the results when having savings of 49% of the evaluations (using adaptive functions linear and nonlinear3), is very acceptable. Only in the cases in which savings of 65% and 78% of the evaluations are obtained (using adaptive functions nonlinear4 and nonlinear5), the corresponding results are of relatively low quality.

[2]The SCC measure indicates the number of elements of the Pareto front obtained, that belong to the true Pareto front of the problem. The IGD measure indicates how far is the true Pareto front from the obtained Pareto front.

# 4. CONCLUSIONS AND FUTURE WORK

We have proposed a mechanism to adapt the value of the inheritance proportion in a dynamic way, throughout the evolutionary process. ¿From the obtained results, we conclude that only when savings of more than a 50% of the total number of evaluations are obtained, the quality on the results is significantly affected. However, even in those cases, the approach with inheritance still provides very good approximations of the true Pareto front. In this way, if an application is very expensive to evaluate and we are interested only on a few optimal solutions, the proposed approach may be a suitable choice. As part of our future work, we plan to test the proposed approaches on different test functions and also on different evolutionary algorithms.

**Acknowledgments.** The first author acknowledges support from CONACyT for granting her a scholarship to pursue graduate studies at CINVESTAV-IPN. The second author acknowledges support from CONACyT project number 42435-Y.

# 5. REFERENCES


[1] M. Reyes-Sierra and C. A. Coello Coello. Improving PSO-based multi-objective optimization using crowding, mutation and ε-dominance. In *Third International Conference on Evolutionary Multi-Criterion Optimization*, pages 505–519. LNCS 3410, Springer-Verlag, 2005.

[2] M. Reyes-Sierra and C. A. Coello Coello. A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization. In *Congress on Evolutionary Computation*, pages 65–72. IEEE Service Center, 2005.

[3] R. E. Smith, B. A. Dike, and S. A. Stegmann. Fitness inheritance in genetic algorithms. In *Proceedings of the 1995 ACM Symposium on Applied Computing*, pages 345–350. ACM Press, 1995.

[4] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.