

The Parallel Evaluation Strategy Approach to Solving Optimization Problems

Effat Jalaeian Zaferani¹⁺ and Mohammad Teshnehlab

Intelligent system Lab, Electrical & Computer Eng. faculty of K. N. Toosi University of Technology

Abstract. Cost functions that involve complex simulations are becoming more common for engineering design problems. Such cost functions are very CPU intensive and their optimization by exact algorithms need complex computations. There are some approaches in the literature to decrease the overall computational time. It needs some experience to know the suitable evaluation algorithm for a defined problem. However, finding the optimum point in a short search time needs an intelligent search algorithm. In this article, a new solution approach is proposed to achieve near global optimal point in a reasonable time for complex cost functions. The proposed approach has been implemented for some test functions. The results show that this approach has better performance comparing to other existing solution algorithms such as Genetic Algorithm, Differential Evolution, Evolution Strategy, and Particle Swarm Optimization.

Keywords: Genetic Algorithm, Differential Evolution, Evolution Strategy and Particle Swarm Optimization.

1. Introduction

Within nearly 20 years of research it has been demonstrated that the search process of natural evolution can yield very robust search algorithms on a computer. Several of these evolutionary algorithms are nowadays broadly accepted, the most important ones being the Genetic Algorithm (GA) by Holland[1], Evolution Strategy (ES) by Rechenberg and Achweffel[2], Differential Evolution (DE) by Storn[3], Particle Swarm Optimization (PSO) by Eberhart and Kennedy.

In the next section we describe some types of optimization algorithm with advantages and disadvantages. In section 3, the proposed approach is introduced. In section 4, implementation of some benchmarks is defined and finally in section 5, the conclusion is expressed.

2. Background

2.1 A Quick Introduction: Genetic Algorithm

The most popular technique in evolutionary computation research has been the genetic algorithm. The reproduction operators used are *crossover* and *mutation*.

GA has following advantages: It can quickly scan a vast solution set it works by its own internal rules. This is very useful for complex or loosely defined problems. While the great advantage of GAs is the fact that they find a solution through evolution; this is also the biggest disadvantage. Evolution is inductive. It evolves *away* from bad circumstances. This can cause a species to evolve into an evolutionary dead end[4,5].

⁺ Corresponding author. Tel.: +98 21 88462174-7; fax: +98 21 88462066.
E-mail address: Jalaeian@ee.kntu.ac.ir.

2.2 A quick introduction: Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a biologically inspired computational search and optimization method based on the social behaviors of birds flocking or fish schooling. It has no overlapping and mutation calculation. The search can be carried out by the speed of the particle. The speed of the researching is very fast. On the other hand, disadvantages of the basic particle swarm optimization algorithm are the method easily suffers from the partial optimism, which causes the less exact at the regulation of its speed and the direction. Then the method cannot work out the problems of scattering and optimization and the method cannot work out the problems of non-coordinate system, such as the solution to the energy field and the moving rules of the particles in the energy field [6].

2.3 A Quick Introduction: Evolution Strategy

This algorithm was a simple mutation-selection scheme. A general version of the algorithm, known as the $(\mu+\lambda)$ evolution strategy. In this strategy, μ parents produce λ offspring. In succeeding generations only the best μ of the λ offspring and μ parents are allowed to survive until the next generation. The selection of parents to form offspring is less constrained than it is in genetic algorithms and genetic programming. For instance, due to the nature of the representation, it is easy to average vectors from many individuals to form a single offspring [7].

2.4 A Quick Introduction: Differential Evolution

This algorithm uses mutation operation as a search mechanism and selection operation to direct the search toward the prospective regions in the search space.

DE algorithm is a heuristic approach mainly having three advantages; Finding the true global minimum regardless of the initial parameter values, fast convergence, and using few control parameters. In many studies it is observe that the convergence speed of the DE is significantly better than the EAs presented. Therefore, the DE algorithm seems to be a promising approach for engineering optimization problems. On the other hand adjusting parameter is really important for this algorithm to converge [8].

3. Proposed Method

This algorithm is a multi-population approach for some existing solution algorithms such as Genetic Algorithm, Differential Evolution, Evolution Strategy and Particle Swarm Optimization. Each population is located in a processor (island) which has its independent evolution process. In order to promote cooperation among the islands, the searching procedure starts from its own initial population and, for some generations, some part of the population is explored.

In this new approach some driving forces in the islands (GA, DE, ES, and PSO) cause to some interchanges among the Islands. The driving force in this way is *periodic migration* between the subpopulation islands, increasing local diversity.

In this article, three periodic migrations are proposed as follows:

- 1) Elitism: According to the size of population, the best chromosomes are selected and replaced with old population. The population for islands is the same in each iteration.
- 2) Merge: According to the random number $a, a \in (0,1)$, the $(a * \text{size of population})$ of the best chromosomes are selected and merged with $((a - 1) * \text{size of population})$ of old population in each island. It is clear in this periodic migration each island has different new population.
- 3) Lambda: According to two random numbers $\in (0,1)$ and $\lambda \in (0,1)$, μ is emigration and λ is immigration, the random numbers of chromosome based on μ and λ of each island can immigrate and emigrate to another random island.

The flowchart of this method is shown in Fig. 1.

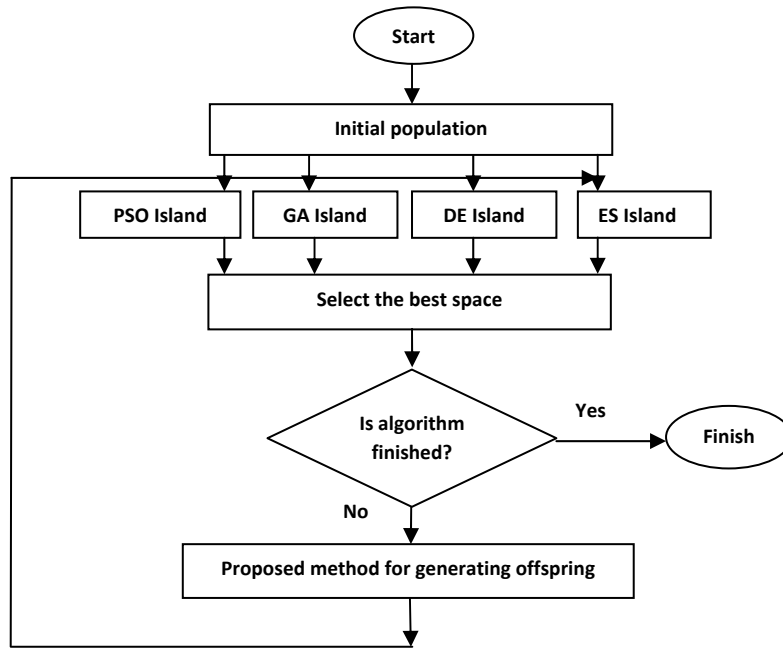


Fig. 1: The flowchart of the proposed method

4. Results

Results of the proposed method are compared to single known methods such as GA, DE, ES and PSO for three test functions which have been reported in the following tables. Test functions have been introduced in the Table 1 and Fig. 2, 3, 4.

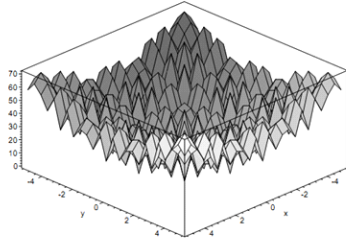


Fig. 2: An over view of Restiring function in 2D

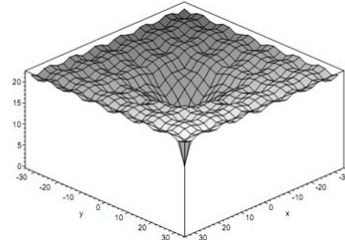


Fig. 3: An over view of Griewangk function in 2D

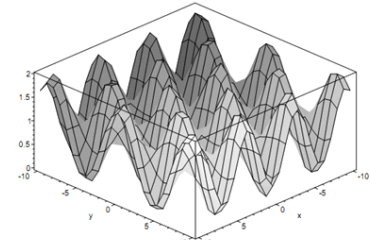


Fig. 4: An over view of Ackley function in 2D

Table 1: Test functions

| | |
|----------------------------|---|
| Rastrigin function: | $f(x) = 10n + \sum_{i=1}^n x_i^2 - 10\cos(2\pi x_i)$ $-5.12 < x_i < 5.12$ $\text{if } x_i = 0 \ i = 1, \dots, n \ f(x) = 0$ |
| Griewangk function: | $f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ $-600 < x_i < 600$ $\text{if } x_i = 0 \ i = 1, \dots, n \ f(x) = 0$ |
| Ackley function: | $f(x) = -a \exp\left(-b \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(cx_i)\right) + a + \exp(1)$ $a=20, b=0.2, c=2\pi$ $-32.768 < x_i < 32.768$ $\text{if } x_i = 0 \ i = 1, \dots, n \ f(x) = 0$ |

Here, three characters are briefly compared: Iteration that methods converge (*iteration*), the optimum point (*Opt. point*) and the total time that the method converges to optimum point (*Opt. Time*) (Table 2-4).

According to these tables, cost functions are converged to the best optimum point in less iteration and more reasonable time.

Table 2: The results of proposed method comparing to single island methods for Rastrigin test function

| Proposed Method: Rastrigin function Population size=35 Max generation=500 | | | | | | |
|--|------------------|-------------------|---------------------|------------------|-------------------|---------------------|
| Num. gen=10 | | | | Num. gen=20 | | |
| | <i>Iteration</i> | <i>Opt. Point</i> | <i>Opt. Time[s]</i> | <i>Iteration</i> | <i>Opt. Point</i> | <i>Opt. Time[s]</i> |
| Lambda | 93 | 0.1128 | 4.6716 | 77 | 0.0198 | 4.0015 |
| Elitism | 189 | 0.7806 | 9.3030 | 250 | 0.8225 | 12.7683 |
| Merge | 154 | 0.2233 | 7.5813 | 252 | 0.7369 | 12.8005 |
| Single Island Method: Rastrigin function Population size=35 Max generation=500 | | | | | | |
| Num. gen=10 | | | | Num. gen=20 | | |
| | <i>Iteration</i> | <i>Opt. Point</i> | <i>Opt. Time[s]</i> | <i>Iteration</i> | <i>Opt. Point</i> | <i>Opt. Time[s]</i> |
| GA | Max | 50 | 0.5698 | Max | 50 | 0.5192 |
| DE | 455 | 1.1207 | 10.0805 | Max | 1.4359 | 11.4035 |
| ES | Max | 23.7136 | 12.5639 | Max | 22.3972 | 12.8821 |
| PSO | 459 | 4.1992 | 0.1357 | Max | 6.2152 | 0.1460 |

Table 3: The results of proposed method comparing to single island methods for Griewangk test function

| Proposed Method: Griewangk function Population size=35 Max generation=500 | | | | | | |
|--|------------------|-------------------|---------------------|------------------|-------------------|---------------------|
| Num. gen=10 | | | | Num. gen=20 | | |
| | <i>Iteration</i> | <i>Opt. Point</i> | <i>Opt. Time[s]</i> | <i>Iteration</i> | <i>Opt. Point</i> | <i>Opt. Time[s]</i> |
| Lambda | 52 | 0.0549 | 2.6978 | 42 | 0.0788 | 2.2289 |
| Elitism | 53 | 0.0532 | 2.6886 | 102 | 0.0434 | 5.2969 |
| Merge | 44 | 0.0584 | 2.2110 | 83 | 0.0869 | 4.3479 |
| Single Island Method: Griewangk function Population size=35 Max generation=500 | | | | | | |
| Num. gen=10 | | | | Num. gen=20 | | |
| | <i>Iteration</i> | <i>Opt. Point</i> | <i>Opt. Time[s]</i> | <i>Iteration</i> | <i>Opt. Point</i> | <i>Opt. Time[s]</i> |
| GA | Max | 16.6988 | 0.4102 | 163 | 18.4777 | 0.3013 |
| DE | 128 | 0.0949 | 2.8758 | 175 | 0.1073 | 4.0352 |
| ES | Max | 0.9001 | 12.9936 | Max | 0.7955 | 13.2228 |
| PSO | 247 | 0.1081 | 0.1022 | 285 | 0.0990 | 0.1253 |

Table 4: The results of proposed method comparing to single island methods Ackley test function

| Proposed Method: Ackley function Population size=35 Max generation=500 | | | | | | |
|---|------------------|-------------------|---------------------|------------------|-------------------|---------------------|
| Num. gen=10 | | | | Num. gen=20 | | |
| | <i>Iteration</i> | <i>Opt. Point</i> | <i>Opt. Time[s]</i> | <i>Iteration</i> | <i>Opt. Point</i> | <i>Opt. Time[s]</i> |
| Lambda | 127 | 0.0000 | 6.4721 | 310 | 0.0003 | 16.3826 |
| Elitism | 138 | 0.0000 | 6.8737 | 295 | 0.0097 | 15.4299 |
| Merge | 81 | 0.0000 | 4.0060 | 278 | 0.0040 | 14.5472 |
| Single Island Method: Ackley function Population size=35 Max generation=500 | | | | | | |
| Num. gen=10 | | | | Num. gen=20 | | |
| | <i>Iteration</i> | <i>Opt. Point</i> | <i>Opt. Time[s]</i> | <i>Iteration</i> | <i>Opt. Point</i> | <i>Opt. Time[s]</i> |
| GA | 189 | 0.0000 | 0.3380 | 335 | 10.4273 | 0.5431 |
| DE | Max | 20.0118 | 11.1204 | Max | 20.0059 | 11.6243 |
| ES | Max | 20.1816 | 12.7693 | Max | 20.0535 | 13.2086 |
| PSO | Max | 20.0634 | 0.1642 | Max | 20.0567 | 0.1841 |

5. Conclusions

In this paper we discussed a parallel evolution strategy approach to find the optimum point in a short search time. The results show that this method is more effective for complex cost functions with many local minimum and maximum points. Thus, the solutions obtained are superior when the cost function is more

complex. The proposed method uses the advantages of four acceptable algorithms and avoids disadvantages to participate in optimization ways.

6. Acknowledgments

Authors are much grateful to Dr. Mahdi Bashiri, a faculty member of Shahed University, for his guidance and enthusiasm.

7. References

- [1] D. E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*, 1989.
- [2] I. Rechenberg. *Cybernetic Solution Path of an Experimental Problem*. Technical report, 1965.
- [3] R. Storn. Differential Evolution Design of an IIR-Filter. *IEEE International Conference on Evolutionary Computation*, pp. 268–273, 1996.
- [4] S. N. Sivanandam and S. N. Deepa. *Introduction to Genetic Algorithms*. Springer, pages 2-3, 2007.
- [5] Information on <http://subsimple.com/genealgo.asp>.
- [6] Q. Bai. Analysis of Particle Swarm Optimization Algorithm. *Computer and Information Science*. Vol. 3, No. 1, 2010.
- [7] R. L. Haupt and S. E. Haupt. *Practical Genetic Algorithms*. Second edition, Page 199, 2004.
- [8] D. Karaboga and S. Okdem. A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm. Vol.12, No.1, 2004.