A Parallel Genetic Algorithms for treating Multi-Objective Optimization Problems

Dr. Bayoumi M.A.H*
El-Feky E.Z**

Abstract

Its known that the parallelization of Genetic Algorithms in the worst case, make it faster without any change in the solution. Almost all of the real world problems has more than one objective, so it must to concentrate on the Multi-Objective Genetic Algorithms "*MOGAs*" and to parallelize it. The most known *MOGAs* could be run in parallel environment, but this paper tries to exploit the parallelization in order to develop a new *MOGAs* algorithm which named "*PMOGA*". The *PMOGA* determines a number of populations equals to the number of objective functions in the problem. Each population uses its own objective function to assign the fitness, Adjusting the parameters of the migration, migration interval and migration rate, plays the main role in forming the pareto front, this algorithm uses the restricted mating and the reinitialization to over come the problem of the diversity.

Keywords: Parallel Genetic Algorithms, Multi-Objective Optimization, Multi-Objective Genetic Algorithms.

---

*Lecturer., Department of OR&DSS, Faculty of Computers and Information, Cairo University, Tharwat 5 St. , 12613 Orman, Dokki, Giza, Egypt.
E-mail : bayoumi2000@operamail.com
** Department of OR&DSS, Faculty of Computers and Information, Cairo University, Tharwat 5 St. , 12613 Orman, Dokki, Giza, Egypt.
E-mail : ehab_zaky@hotmail.com

1. Introduction

Many real-world design or decision making problems involve simultaneous optimization of multiple objectives. In such problems, there exist a set of solutions which are superior to the rest of solutions. These solutions are known as Pareto-optimal or nondominated solutions [1,9].

In the past few years, Parallel Genetic Algorithms (*PGAs*) have been used to solve difficult problems. Huge problems need a bigger population and this translates directly into higher computational costs. The basic motivation behind many early studies of *PGAs* was to reduce the processing time needed to reach an acceptable solution. This was accomplished by the implementation of *GAs* on different parallel architectures. In addition, it was noted that in some cases the *PGAs* found better solutions than comparably sized serial *GAs*.

The plan of this paper is organized as follows, In section **2** a theoretical background is introduced, section **3** the problem under treatment is formulated, section **4** presents the proposed algorithm, illustrative examples and comparison are stated at section **5** and finally, section **6** states the conclusion.

2. Theoretical Background

2.1 Multi-Objective Genetic Algorithms.

Multi-objective optimization problems give rise to a set of Pareto-optimal solutions, none of which can be said to be better than other in all objectives. In any interesting multi-objective optimization problem, there exists a number of such solutions which are of interest to designers and practitioners. Since no one solution is better than any other solution in the Pareto-optimal set, it is also a goal in a multi-objective optimization to find as many such Pareto-optimal solutions as possible. Unlike most classical search and optimization problems, GAs work with a population of solutions and thus are likely (and unique) candidates for finding multiple Pareto-optimal solutions simultaneously [6,15,10,13,16] .

There are two tasks that are achieved in a *MOGA*:

1. Convergence to the Pareto-optimal set, and
2. Maintenance of diversity among solutions of the Pareto-optimal set.

GAs with a suitable modification in their operators have worked well to treat many multi-objective optimization problems with respect to above two tasks. Most *MOGAs* work with the concept of domination. In the following the brief history of *MOGA* is stated.

History of *MOGA*:

1. ***Pioneers Before 1990:*** *VEGA* [14], Fourman [7], *ESVO* [12] and others researchers has developed their search in Objective-wise selection and proof of principle.
2. ***Classics Until 1995:*** *MOGA* [6], *NPGA* [10], *NSGA* [15] and others researchers has developed their search in Pareto-based selection, Niching and Visual comparisons.

3. ***Elitists Until 2000:*** *SPEA* [16], *PAES*, *PESA* [11], *NSGA II* [3] and others researchers has developed their search in Archiving, elitism and Quantitative performance metrics.
4. ***Fine Tunning Until Now :*** *SPEAII* [5] and others researchers has improved their search techniques, Run time analysis has performed.

2.2 Parallel Genetic Algorithms.

Genetic Algorithms (*GAs*) are efficient search methods based on principles of natural selection and genetics. They are being applied successfully to find acceptable solutions to problems in business, engineering, and science[8]. *GAs* are generally able to find good solutions in reasonable amounts of time, but as they are applied to larger and harder problems there is an increase in the time required to find adequate solutions. As a consequence of this there have been multiple efforts to make *GAs* faster, and one of the most promising choices is to use parallel implementations.

The *PGAs* are classified into four types[4]: Global *PGAs*, Coarse-Grained *PGAs*, Fine-Grained *PGAs*, and Hybrid *PGAs*. In the first type there is only one population, but the evaluation of individuals and the genetic operators are parallelized explicitly (see figure 2.1). In the second type the population of the *GAs* is divided into multiple subpopulations or demes that evolve isolated from each other most of the time, but exchange individuals occasionally, this exchange of individuals is called migration (see figure 2.2). In the third type the population is partitioned into a large number of very small sub-populations (see figure 2.3). In the last type the Coarse-Grained *PGAs* hybrid with any of three types (see figure 2.4).

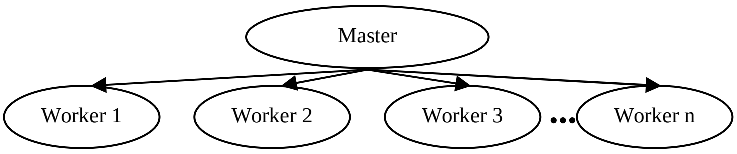


Figure 2.1: A Schematic of a Global PGAs

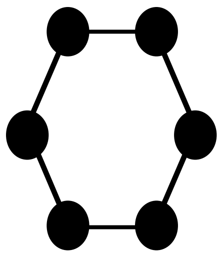


Figure 2.2: A Schematic of a Coarse-grained PGAs

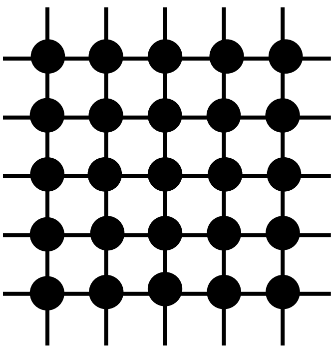


Figure2. 3: A schematic of a fine-grained PGAs.

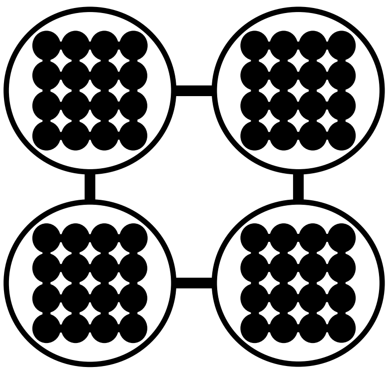


Figure2.4: Hybrid Parallel between coarse-grained and fine-grained PGAs

3. Problem Formulation.

A multi-objective problem has a number of conflicting objective functions, which are to be optimized simultaneously “at the same time”.

Most real world optimization problems are naturally posed as multi-objective optimization problems, However, due to the complexities involved in solving multi-objective optimization problems and due to the lack of suitable and efficient techniques. In traditional techniques, they have been transformed and solved through scalarization approaches, we have assumed that all objectives being minimized simultaneously.

The presence of multiple, usually conflicting, objectives implies that in such problems there is rarely a single solution, which is optimal according to every objective.

Mathematically, A multi-objective problem can be stated as follows:

$$\begin{aligned} &\min F(x) = \{ f_1(x), f_2(x), ..., f_m(x) \} \\ &\quad s.t. \quad x \in M(x) \end{aligned} \tag{3.1}$$

where

$f_i(x): R^n \rightarrow R, i = 1, ..., m.$

$x = (x_1, x_2, ..., x_n)$: *Vector of Decision Variables*,

$M(x)$: *is feasible region*.

m: *number of objective functions.*

4. Parallel Multi-Objective Genetic Algorithms "*PMOGA*".

In order to treat *MOGA* problems, it must face tow problems, the first one is to converge to the Pareto-optimal set. This could be achieved by determining both fitness assignment and selection method, and the second is maintaining the diversity among solutions of the Pareto-optimal set (see 2.1). The suggested *PMOGA*, which uses PGAs, determines a number of populations equals to the number of objective functions in the problem. Each population uses its own objective function to assign the fitness, so it directs its individuals to solve its objective function. In this way, the resulted individuals from all populations will be crowded in different zones in the objective space, so the *PMOGA* uses the migration to move some individuals between the populations. These individuals which is bad for the destination population, mates with the original individuals creating diverse ones. Adjusting two parameters of migration (migration interval and migration rate), plays the main role in forming the pareto front. Also, the proposed algorithm uses the restricted mating and the reinitialization to over come the problem of diversity.

In the following, the steps of *PMOGA* are stated. (see Appendix A).

PMOGA-Steps:

Input: N *(population size)*

T *(maximum number of generations)*

P_c *(crossover probability)*

P_m *(mutation rate)*

O *(Number of objectives)*

M_o *(Probability of Migration Occurrence)*

M_r *(Migration Ratio)*

Output: A *(nondominated set)*

Step 1 : ***Initialization:*** *Set* $P_0^0 = \phi$ *and* $t = 0$.

For $i = 1,\ldots,$ ***N* do**

a) Choose $i \in I$ *according to some probability distribution.*

b) Set $P_0^0 = P_0^0 + \{i\}$.

For $i = 1,\ldots,$ ***O* do**

$Set\, P_0^i = P_0^0$.

Step 2 : ***Fitness assignment:***

For $i = 1,\ldots,$ ***O* do**

For $j = 1,\ldots,$ ***N* do**

Evaluate $f_i(x_j)$ *according to the given objective functions.*

Step 3 : ***Reproduction:***

For $i = 1,\ldots,$ ***O* do**

a) $P_t^i{}' = Select(P_t^i)$.

b) $P_t^i{}'' = CrossOver(P_t^i{}', P_c)$.

c) $P_t^i{}''' = Mutate(P_t^i{}'', P_m)$.

Step 4 : ***Migration:***

Determine the number of individual to migrate N_m *from* M_r

For $i = 1,\ldots,$ ***O* do**

a) Select N_m *Individual to be Migrated from* $P_t^i{}'''$.

b) **For** $j = 1,\ldots,$ ***O* do**

$P_t^i{}'''' = Mutate(P_t^i{}''', M_o)$.

Step 5 : ***Elitism:***

$A = NonDominate(P_t^i{}'''')$

Step 6 : ***Re Initialization:***

$(P_t^i{}''''') = Re\ Inetialize(P_t^i{}'''')$

Step 7 : ***Termination:***

For $i = 1,\ldots,$ ***O* do**

$P_{t+1}^i = P_t^i{}'''''$

$t = t+1$

if $(t \geq T)$ ***then*** *Stop with output A* ***else*** *go to Step 2*

5. Illustrative Examples.

There are six test Problems have been constructed by Deb in 1998 to check how much the convexity, non-convexity, discreteness, non-uniformity, multi-modality and deception affects the solutions of the MOGA algorithms. In order to check the ability of the new algorithm in solve *MODM* problems, these six problems were solved by the suggested *PMOGA*, the results were compared with other techniques. All of these problems have only two objective functions to investigate the simplest case first. In the author's opinion, two-dimensional problems already reflect essential aspects of *MOPs*. Each of the test problems that defined below is structured in the same manner and consists itself of three functions f_1, g, h [2].

$$\underset{x \in R^n}{Minimize} \quad t(X) = (f_1(x_1), f_2(X))$$

$$where \qquad f_2(X) = g(x_2,...,x_n) \bullet h_1(f_1(x_1), g(x_2,...,x_n)), \quad (5.1)$$
$$X = (x_1,...,x_n)$$

The function f_1 is a function of the first decision variable only, g is a function of the remaining $n-1$ variables, and h is a function of values of f_1, g . The test functions differ in these three functions as well as in the number of variables n and in the values that the variables may take.

The test problem number 5 describes a deceptive problem and distinguishes itself from the other test problems in that x_i represents a binary string as equation (4.2). So it requires Binary representation, and it excluded from the comparison.

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2,...,x_n) &= 1 + 9 \bullet (\sum_{i=2}^{n} x_i)/(n-1) \\ h(f_1, g) &= 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1) \\ n = 30, \quad & g = 1, \quad x_i \in [0,1] \end{aligned} \qquad (5.2)$$

5.1 Parameter Settings:

Independent of the algorithm and the test function, each simulation run was carried out using the following parameters:

- Number of generations T : 250
- Population size N : 100
- Crossover rate pc (one-point) : 0.8
- Mutation rate pm (per bit) : 0.01
- Niche radius σ_{share} : 0.4886

In the *PMOGA* the population is divided into two subpopulations (as two objective functions) each of size 50, so the whole population size equals 100.

As using parallel *GAs*, there are different parameters used in the algorithm which are:

- Migration rate : 0.05
- Migration interval : 0.3 – 0.8

5.2 *Test Problem # 1.*

The first problem $\boldsymbol{t_1}$ has a convex Pareto-optimal front as shown in figure(5.1):

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2,...,x_n) &= 1+9 \bullet (\sum_{i=2}^{n} x_i)/(n-1) \\ h(f_1,g) &= 1-\sqrt{f_1/g} \\ n=30, \quad & g=1, \quad x_i \in [0,1] \end{aligned} \qquad (5.3)$$

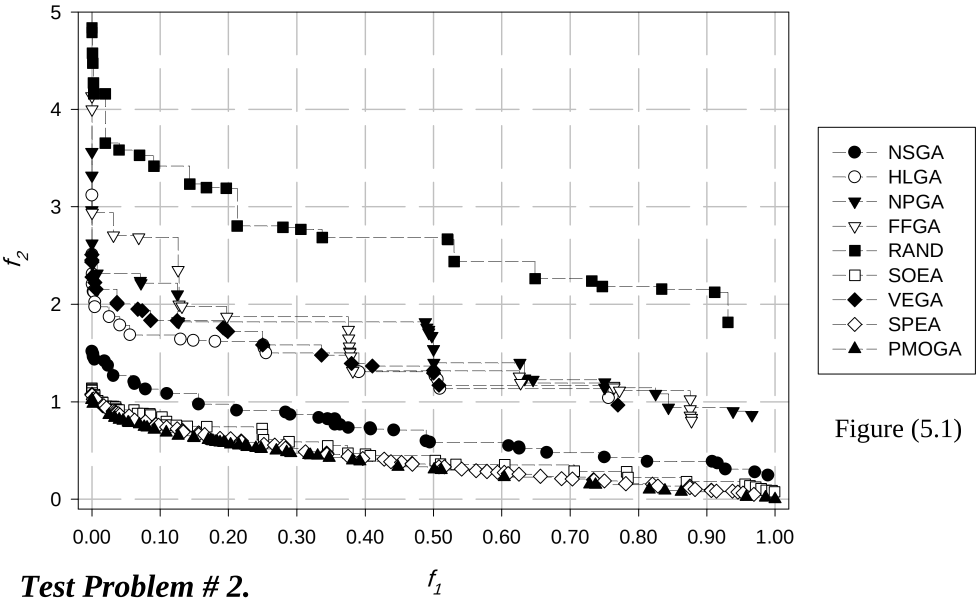


Figure (5.1)

5.3 *Test Problem # 2.*

The second problem $\boldsymbol{t_2}$ has a non-convex counterpart for $\boldsymbol{t_1}$ as shown in figure(5.2):

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2,...,x_n) &= 1+9 \bullet (\sum_{i=2}^{n} x_i)/(n-1) \\ h(f_1,g) &= 1-(f_1/g)^2 \\ n=30, \quad & g=1, \quad x_i \in [0,1] \end{aligned} \qquad (5.4)$$

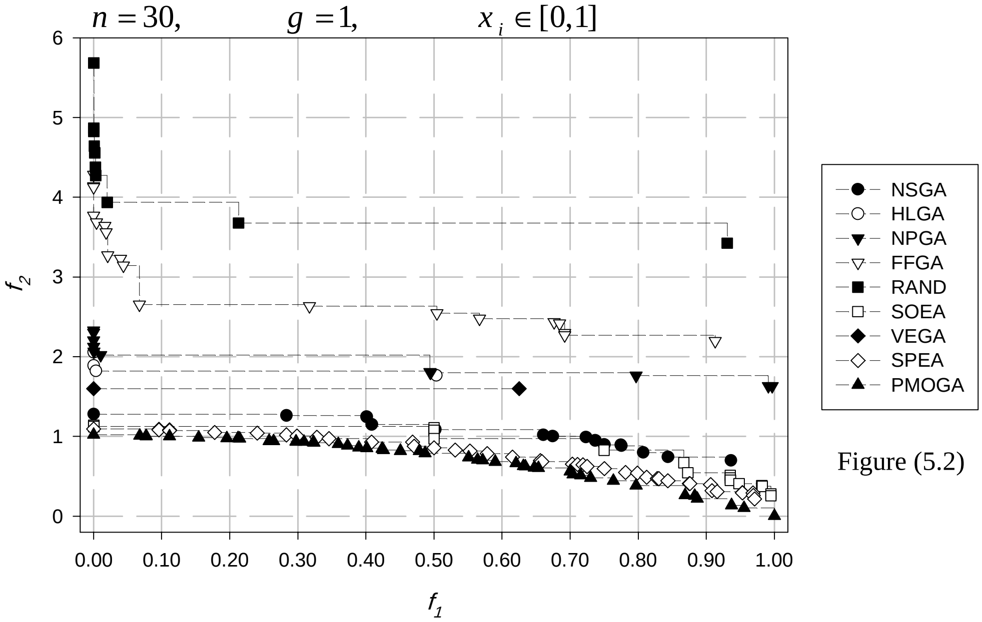


Figure (5.2)

5.4 *Test Problem # 3.*

The third problem t_3 represent the discreteness features: its Pareto-optimal front consist of several non-contiguous convex parts as shown in figure(5.3):

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2,\ldots,x_n) &= 1+9\bullet(\sum_{i=2}^{n} x_i)/(n-1) \\ h(f_1,g) &= 1-\sqrt{f_1/g}-(f_1/g)\sin(10\pi f_1) \\ n=30, \quad & g=1, \quad x_i\in[0,1] \end{aligned} \tag{5.5}$$

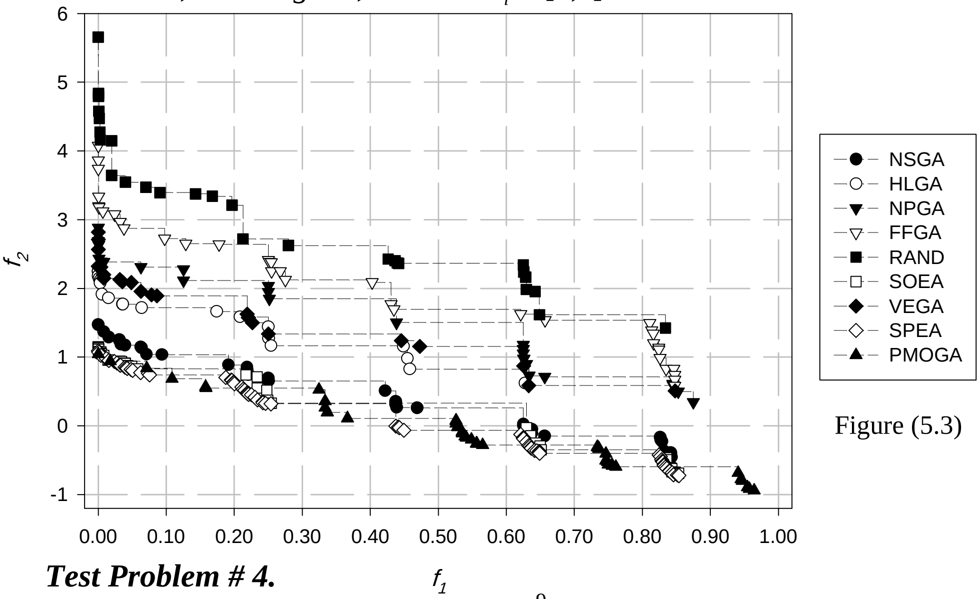


Figure (5.3)

5.5 *Test Problem # 4.*

The fourth problem t_4 contains 21^9 local Pareto-optimal sets and therefore test for the algorithm ability to deal with multimodality as shown in figure(5.4):

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2,\ldots,x_n) &= 1+10(n-1)\bullet(\sum_{i=2}^{n} x_i^2-10\cos(4\pi x_i)) \\ h(f_1,g) &= 1-\sqrt{f_1/g} \\ n=10, \quad & g=1, \quad x_1\in[0,1] \quad x_2,\ldots,x_n\in[-5,5] \end{aligned} \tag{5.6}$$

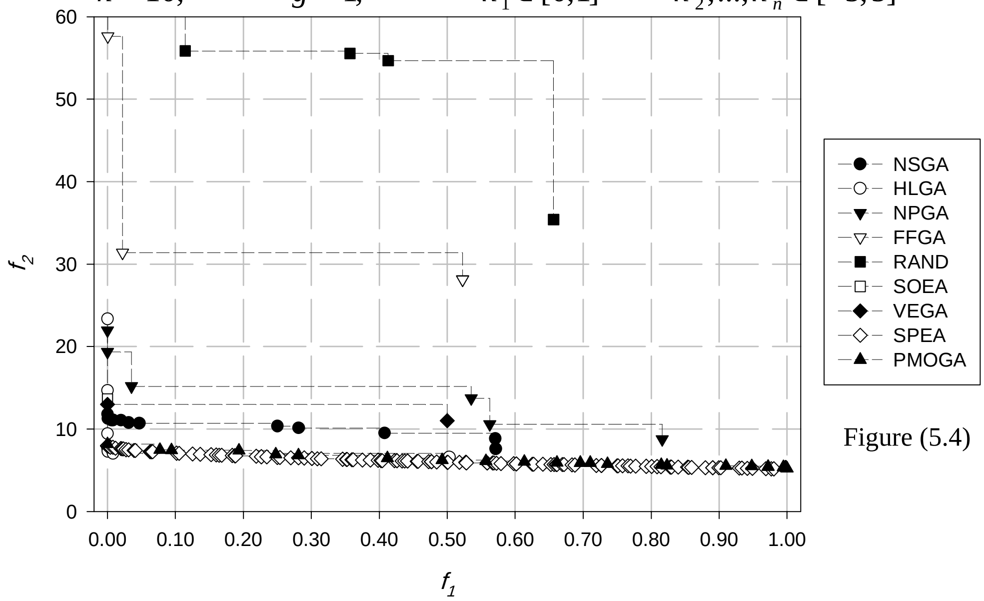


Figure (5.4)

5.6 *Test Problem # 6.*

The six problem t6 includes two difficulties caused by the non-uniformity of the objective space: Firstly, the Pareto-optimal solutions are non-uniformly distributed along the global Pareto front (the front is biased for solutions for which f1.x1/ is near one); secondly, the density of the solutions is least near the Pareto-optimal front and highest away from the front as shown in figure(5.5):

$$
\begin{aligned}
f_1(x_1) &= 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\
g(x_2,\dots,x_n) &= 1 + 9 \bullet \left(\left(\sum_{i=2}^{n} x_i\right)/(n-1) \right)^{0.25} \\
h(f_1, g) &= 1 - (f_1/g)^2 \\
n = 10, \quad & g = 1, \quad x_i \in [0,1]
\end{aligned}
\tag{5.7}
$$

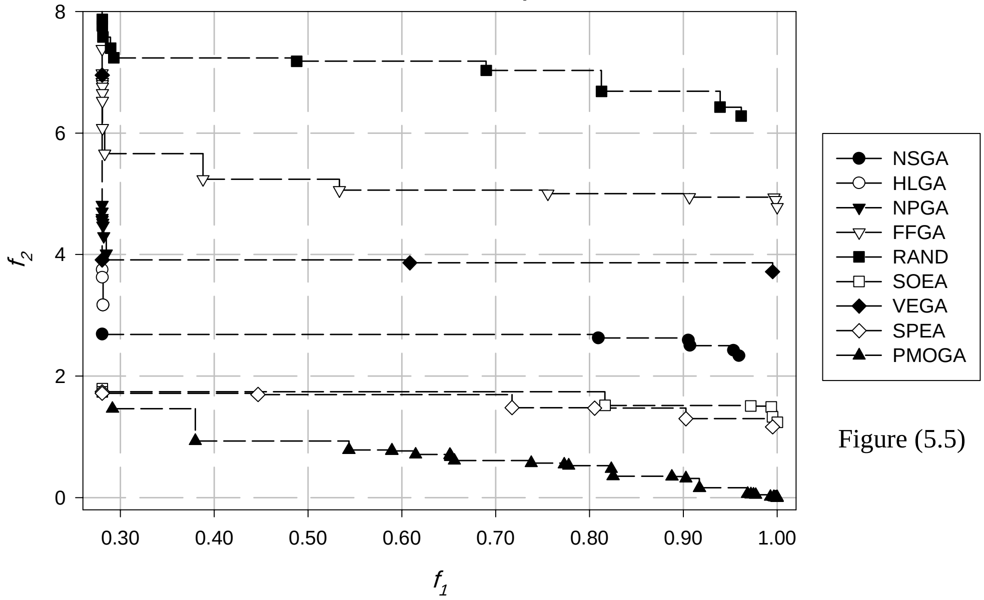


Figure (5.5)

5.7 *Key Results.*

- The suggested test problems provide sufficient complexity to compare multi-objective optimizers. With all functions, differences in performance could be observed among the algorithms under consideration. Regarding features of the particular problem, multimodality and non-uniformly distributed global Pareto front seem to cause the most difficulty for evolutionary approaches.
- In 1st and 2nd test problems, the *PMOGA* achieves slight better pareto front.
- In 3rd test problem, the resulted pareto front dominates others at some points even it is dominated by them at other points.
- In 4th test problem, the *PMOGA* achieves slight worse pareto front.
- In 6th test problem, the *PMOGA* achieves wide significantly better pareto front.

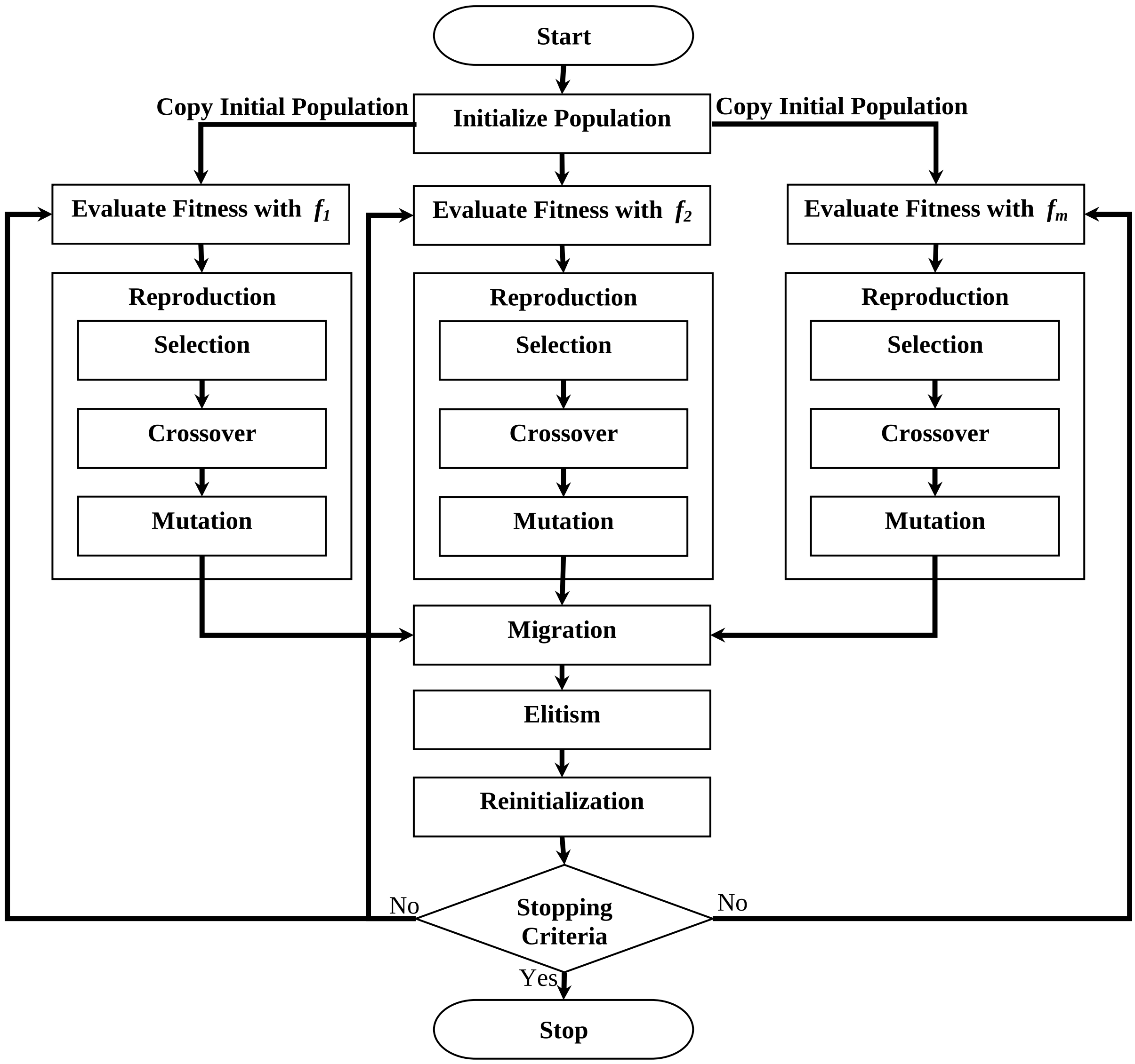
6. Conclusions.

In this paper, a Parallel Genetic Algorithm technique known as "*PMOGA*" is introduced in order to treat the multi-objective optimization problems. The introduced algorithm determines a number of populations equals to the number of objective functions in the problem. Each population uses its own objective function to assign the fitness. Adjusting the parameters of migration, migration interval and migration rate, plays the main role in forming the pareto front, this algorithm uses the restricted mating and the reinitialization to over come the problem of the diversity.

There are comparison carried out with other 8 optimizers in this area through five main test problems, this comparisons lead us that the *PMOGA* was superior in some situations and was same in the others.

Appendix A

Flow chart of the *PMOGA*



References:

[1] **Chankong, V., and Haimes, Y. Y.** *"Multiobjective decision making theory and methodology"*. New York: North-Holland. (1983).

[2**] Deb, K.** *"Multi-objective genetic algorithms: Problem difficulties and construction of test functions*". Technical Report No. CI-49/98, Department of Computer Science/XI, University of Dortmund, Germany. (1998).

[3] **Deb, K., Agarwal, S., Pratap, A., and Meyarivan, T.** *"A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization*: *NSGA-II"*.. Proceedings of the Parallel Problem Solving from Nature, pp. 849-858. Springer. (2000).

[4] **E. Cantú-Paz,** *"A survey of parallel genetic algorithms*" Calculateurs Parallèles, Reseaux et Systems Repartis, vol. 10, no. 2, pp. 141-171,1998.

[5] **E.Zitzler,M.Laumanns,and L.Thiele.** *"SPEA2:Improving the strength pareto evolutionary algorithm for multi-objective optimization"*. In K.Giannakoglou, D.Tsahalis,J. Periaux,K. Papailiou, and T.Fogarty, editors, Evolutionary Methods for Design, Optimization, and Control. pages 19-26,Barcelona,Spain,2002.

[6**] Fonseca, C. M. and P. J. Fleming,** *"Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization"*. In S. Forrest (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, California, pp. 416-423. Morgan Kaufmann. (1993).

[7]**Fourman, M. P.** *"Compaction of symbolic layout using genetic algorithms"*. In J. J. Grefenstette (Ed.), Proceedings of an International Conference on Genetic Algorithms and Their Applications, Pittsburgh, PA, pp. 141-153. sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI). (1985).

[8] **Goldberg, D. E.** *"Genetic and evolutionary algorithms come of age"*. Communications of the ACM , 37 (3), 113-119. (1994).

[9] **Hans, A. E.** *"Multicriteria optimization for highly accurate systems. Multicriteria Optimization in Engineering and Sciences"*, W. Stadler(Ed.), Mathematical concepts and methods in science and engineering, 19, 309-352. New York Plenum press. (1988).

[10] **Horn, J. and Nafploitis, N., and Goldberg, D. E.** *"A niched Pareto genetic algorithm for multi-objective optimization"*. Proceedings of the First IEEE Conference on Evolutionary Computation. 82-87. (1994).

[11] **Knowles, J. and D. Corne.** *"Assessing the performance of the pareto archived evolution strategy"*. In A. S. Wu (Ed.), Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO'99), Work-shop Program, pp. 123-124. (1999b).

[12] **Kursawe, F.** *"A variant of evolution strategies for vector optimization"*. In H.-P. Schwefel and R. M¨ anner (Eds.), Parallel Problem Solving from Nature — Proc. 1st Workshop PPSN, Berlin, pp. 193-197. Springer. (1991).

[13] **Laumanns, M., Rudolph, G., and Schwefel, H.-P.** *"A spatial predator-prey approach to multi-objective optimization"*, A preliminary study. Proceedings of the Parallel Problem Solving from Nature, V. 241-249. (1998).

[14]**Schaffer, J. D.** *"Multiple objective optimization with vector evaluated genetic algorithms"*. In J. J. Grefenstette (Ed.), Proceedings of an International Conference on Genetic Algorithms and Their Applications, Pitts-burgh, PA, pp. 93-100. sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI). (1985).

[15] **Srinivas, N. and Deb, K.** “*Multi-Objective function optimization using non-dominated sorting genetic algorithms*”, Evolutionary Computation, 2(3): 221-248. (1994).
[16] **Zitzler, E. and Thiele, L.** “*Multiobjective optimization using evolutionary algorithms, A comparative case study*”. Parallel Problem Solving from Nature, V, 292-301. (1998).