

Article

## Entropy Diversity in Multi-Objective Particle Swarm Optimization

Eduardo J. Solteiro Pires <sup>1,\*</sup>, José A. Tenreiro Machado <sup>2</sup> and Paulo B. de Moura Oliveira <sup>1</sup>

<sup>1</sup> INESC TEC—INESC Technology and Science (formerly INESC Porto, UTAD pole), Escola de Ciências e Tecnologia, Universidade de Trás-os-Montes e Alto Douro, 5000–811 Vila Real, Portugal; E-Mail: oliveira@utad.pt

<sup>2</sup> ISEP—Institute of Engineering, Polytechnic of Porto, Department of Electrical Engineering, Rua Dr. António Bernadino de Almeida, 4200–072 Porto, Portugal; E-Mail: jtm@isep.ipp.pt

\* Author to whom correspondence should be addressed; E-Mail: epires@utad.pt; Tel.: +351-259-350356; Fax: +351-259-350480.

Received: 30 August 2013; in revised form: 30 November 2013 / Accepted: 3 December 2013 / Published: 10 December 2013

---

**Abstract:** Multi-objective particle swarm optimization (MOPSO) is a search algorithm based on social behavior. Most of the existing multi-objective particle swarm optimization schemes are based on Pareto optimality and aim to obtain a representative non-dominated Pareto front for a given problem. Several approaches have been proposed to study the convergence and performance of the algorithm, particularly by accessing the final results. In the present paper, a different approach is proposed, by using Shannon entropy to analyze the MOPSO dynamics along the algorithm execution. The results indicate that Shannon entropy can be used as an indicator of diversity and convergence for MOPSO problems.

**Keywords:** multi-objective particle swarm optimization; Shannon entropy; diversity

---

### 1. Introduction

Particle swarm optimization (PSO) is a metaheuristic algorithm based on social species behavior. PSO is a popular method that has been used successfully to solve a myriad of search and optimization problems [1]. The PSO is inspired in the behavior of bird flocking or fish schooling [2]. Each bird or fish is represented by a particle with two components, namely by its position and velocity. A set of particles forms the swarm that evolves during several iterations giving rise to a powerful optimization method.

The simplicity and success of the PSO led the algorithm to be employed in problems where more than one optimization criterion is considered. Many techniques, such as those inspired in genetic algorithms (GA) [3,4], have been developed to find a set of non-dominated solutions belonging to the Pareto optimal front. Since the multi-objective particle swarm optimization (MOPSO) proposal [5], the algorithm has been used in a wide range of applications [1,6]. Moreover, a considerable number of variants of refined MOPSO were developed in order to improve the algorithm performance, e.g., [7].

In single objective problem the performance of the algorithms can be easily evaluated by comparing the values obtained by each one. Moreover, when the performance over time is required the evolution of the best fitness value of the population is normally used. Advanced studies can be accomplished by means of the dynamic analysis [8,9] of the evolution. Many indexes were introduced to measure the performance of multi-objective algorithms according to the solution set produced by them [10–12]. In those cases, when is difficult to identify the best algorithm, nonparametric statistical tests are crucial [13,14].

Shannon entropy has been applied in several fields, such as communications, economics, sociology, and biology among others, but in evolutionary computation it has not been fully explored. For an example of research work in this area we can refer to Galaviz-Casas [15], which studies the entropy reduction during the GA selection at the chromosome level. Masisi *et al.* [16] used the (Renyi and Shannon) entropy to measure the structural diversity of classifiers based in neural networks. The measuring index is obtained by evaluating the parameter differences and the GA optimizes the accuracy of 21 classifiers ensemble. Myers and Hancock [17] predict the behavior of GA formulating appropriate parameter values. They suggested the population Shannon entropy for run-time performance measurement, and applied the technique to labeling problems. Shannon entropy provides useful information about the algorithm state. The entropy is measured in the parameter space. It was shown that populations with entropy smaller than a given threshold become saturated and the population diversity disappears. Shapiro and Bennett [18,19] adopted the maximum entropy method to find out equations describing the GA dynamics. Kita *et al.* [20] proposed a multi-objective genetic algorithm (MOGA) based on a thermodynamical GA. They used entropy and temperature concepts in the selection operator.

Farhang-Mehr and Azarm [21] formulated an entropy based MOGA inspired by the statistical theory of gases, which can be advantageous in improving the solution coverage and uniformity along the front. Indeed, in a enclosed environment, when an ideal gas undergoes an expansion, the molecules move randomly, archiving a homogeneous and uniform equilibrium stated with maximum entropy. This phenomenon occurs regardless of the geometry of the closed environment.

Qin *et al.* [22] presented an entropy based strategy for maintaining diversity. The method maintains the non-dominated number of solutions by deleting those with the worst distribution, one by one, using the entropy based strategy. Wang *et al.* [23] developed an entropy-based performance metric. They pointed out several advantages, namely that (i) the computational effort increases linearly with the solution number, (ii) the metric qualifies the combination of uniformity and coverage of Pareto set and (iii) it determines when the evolution has reached maturity.

LinLin and Yunfang [24] proposed a diversity metric based on entropy to measure the performance of multi-objective problems. They not only show when the algorithm can be stopped, but also compare the performance of some multi-objective algorithms. The entropy is evaluated from the solution density

of a grid space. These researchers compare a set of MOGA algorithms performance with different optimization functions.

In spite of having MOPSO used in a wide range of applications, there are a limited number of studies about its dynamics and how particles self-organize across the Pareto front. In this paper the dynamic and self-organization of particles along MOPSO algorithm iterations is analyzed. The study considers several optimization functions and different population sizes using the Shannon entropy for evaluating MOPSO performance.

Bearing these ideas in mind, the remaining of the paper is organized as follows. Section 2 describes the MOPSO adopted in the experiments. Section 3 presents several concepts related with entropy. Section 4 addresses five functions that are used to study the dynamic evolution of MOPSO using entropy. Finally, Section 5 outlines the main conclusions and discusses future work.

## 2. Multiobjective Particle Swarm Optimization

The PSO algorithm is based on a series of biological mechanisms, particularly in the social behavior of animal groups [2]. PSO consists of particles movement guided by the most promising particle and the best location visited by each particle. The fact that particles work with stochastic operators and several potential solutions, provides PSO the ability to escape from local optima and to maintain a population with diversity. Moreover, the ability to work with a population of solutions, introduces a global horizon and a wider search variety, making possible a more comprehensive assessment of the search space in each iteration. These characteristics ensure a high ability to find the global optimum in problems that have multiple local optima.

Most real world applications have more than a single objective to be optimized, and therefore, several techniques were proposed to solve those problems. Due to these reasons, in the last years many of the approaches and principles that were explored in different types of evolutionary algorithms have been adapted to the MOPSO [5].

Multi-objective optimization problem solving aims to find an acceptable set of solutions, in contrast with uni-objective problems where there is only one solution (except in cases where uni-objective functions have more than one global optimum). Solutions in multi-objective optimization problems intend to achieve a compromise between different criteria, enabling the existence of several optimal solutions. It is common to use the concept of dominance to compare the various solutions of the population. The final set of solutions may be represented graphically by one or more fronts.

Algorithm 1 illustrates a standard MOPSO algorithm. After the swarm initialization, several loops are performed in order to increase the quality of both the population and the archive. In iteration loop  $t$ , each particle in the population selects a particle guide from the archive  $A(t)$ . Based on the guide and personal best, each particle moves using simple PSO formulas. At the end of each loop (Line 12) the archive  $A(t + 1)$  is updated by selecting the non-dominant solutions among the population,  $P(t)$ , and the archive  $A(t)$ . When the non-dominant solution number is greater than the size of the archive, the solutions with best diversity and extension are selected. The process comes to an end, usually after a certain number of iterations.

---

**Algorithm 1:** The Structure of a standard MOPSO Algorithm
 

---

```

1:  $t = 0$ 
2: Random initialization of  $P(t)$ 
3: Evaluate  $P(t)$ 
4:  $A(t)$  = Selection of non-dominated solutions
5: while the process do
6:   for Each particle do
7:     Select  $p_g$ 
8:     Change position
9:     Evaluate particle
10:    Update  $p$ 
11:   end for
12:    $A(t)$  = Selection( $P(t) \cup A(t)$ )
13:    $t = t + 1$ 
14: end while
15: Get results from  $A$ 

```

---

### 3. Entropy

Many entropy interpretations have been suggested over the years. The best known are *disorder*, *mixing*, *chaos*, *spreading*, *freedom* and *information* [25]. The first description of entropy was proposed by Boltzmann to describe systems that evolve from ordered to disordered states. *Spreading* was used by Guggenheim to indicate the diffusion of a energy system from a smaller to a larger volume. Lewis stated that, in a spontaneous expansion gas in an isolated system, *information* regarding particles locations decreases while, the missing information or, *uncertainty* increases.

Shannon [26] developed the information theory to quantify the information loss in the transmission of a given message. The study was carried out in a communication channel and Shannon focused in physical and statistical constraints that limit the message transmission. Moreover, the measure does not addresses, in this way, the meaning of the message. Shannon defined  $H$  as a measure of information, choice and uncertainty:

$$H(X) = -K \sum_{x \in X} p_i(x) \log p_i(x) \quad (1)$$

The parameter  $K$  is a positive constant, often set to value 1, and is used to express  $H$  in an unit of measure. Equation (1) considers a discrete random variable  $x \in X$  characterized by the probability distribution  $p(x)$ .

Shannon entropy can be easily extended to multivariate random variables. For two random variables  $(x, y) \in (X, Y)$  entropy is defined as:

$$H(X, Y) = -K \sum_{x \in X} \sum_{y \in Y} p_i(x, y) \log p_i(x, y) \quad (2)$$

### 4. Simulations Results

This section presents five functions to be optimized with 2 and 3 objectives, involving the use of entropy during the optimization process. The optimization functions  $F_1$  to  $F_4$ , defined by Equations (3) to (6), are known as Z2, Z3, DTLZ4 and DTLZ2 [27,28], respectively, and  $F_5$  is known as UF8, from CEC 2009 special session competition [29].

$$F_1 = \begin{cases} f_1(X) & = x_1 \\ g(X) & = 1 + 9 \sum_{i=2}^m \frac{x_i}{m-1} \\ h(f_1, g) & = 1 - \left(\frac{f_1}{g}\right)^2 \\ f_2(X) & = g(X)h(f_1, g) \end{cases} \tag{3}$$

$$F_2 = \begin{cases} f_1(X) & = x_1 \\ g(X) & = 1 + \frac{9}{m} \sum_{i=1}^m x_i \\ f_2(X) & = g(X) - \sqrt{g(X)}x_1 - 10x_1 \sin \pi x_1 \end{cases} \tag{4}$$

$$F_3 = \begin{cases} f_1(X) & = [1 + g(X)] \cos(x_1^\alpha \pi/2) \cos(x_2^\alpha \pi/2) \\ f_2(X) & = [1 + g(X)] \cos(x_1^\alpha \pi/2) \sin(x_2^\alpha \pi/2) \\ f_3(X) & = [1 + g(X)] \sin(x_1^\alpha \pi/2) \\ g(X) & = 1 + 9 \sum_{i=3}^m (x_i^\alpha - 0.5)^2 \end{cases} \tag{5}$$

$$F_4 = \begin{cases} f_1(X) & = [1 + g(X)] \cos(x_1 \pi/2) \cos(x_2 \pi/2) \\ f_2(X) & = [1 + g(X)] \cos(x_1 \pi/2) \sin(x_2 \pi/2) \\ f_3(X) & = [1 + g(X)] \sin(x_1 \pi/2) \\ g(X) & = 1 + 9 \sum_{i=3}^m (x_i - 0.5)^2 \end{cases} \tag{6}$$

$$F_5 = \begin{cases} f_1(X) & = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{|J_1|} \sum_{j \in J_1} \left(x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{m})\right)^2 \\ f_2(X) & = \cos(0.5x_1\pi) \sin(0.5x_2\pi) + \frac{2}{|J_2|} \sum_{j \in J_2} \left(x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{m})\right)^2 \\ f_3(X) & = \sin(0.5x_1\pi) + \frac{2}{|J_3|} \sum_{j \in J_3} \left(x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{m})\right)^2 \\ J_1 & = \{j|3 \leq j \leq m, \text{ and } j - 1 \text{ is a multiplication of } 3\} \\ J_2 & = \{j|3 \leq j \leq m, \text{ and } j - 2 \text{ is a multiplication of } 3\} \\ J_3 & = \{j|3 \leq j \leq m, \text{ and } j \text{ is a multiplication of } 3\} \end{cases} \tag{7}$$

These functions are to be optimized using a MOPSO with a constant inertia coefficient  $w = 0.7$  and acceleration coefficients  $\phi_1 = 0.8$  and  $\phi_2 = 0.8$ . The experiments adopt  $t = 1000$  iterations and the archive has a size of 50 particles. Furthermore, the number of particles is maintained constant during each experiment and its value is predefined at the begging of each execution.

To evaluate the Shannon entropy the objective space is divided into cells forming a grid. In the case of 2 objectives, the grid is divided into 1024 cells,  $n_{f_1} \times n_{f_2} = 32 \times 32$ , where  $n_{f_i}$  is the number of cells in objective  $i$ . On the other hand, when 3 objectives are considered the grid is divided in 1000 cells, so

that  $n_{f_1} \times n_{f_2} \times n_{f_3} = 10 \times 10 \times 10$ . The size in each dimension is divided according to the maximum and minimum values obtained during the experiments. Therefore, the size  $s_i$  of dimension  $i$  is given by:

$$s_i = \frac{f_i^{\max} - f_i^{\min}}{n_{f_i}} \tag{8}$$

The Shannon entropy is evaluated by means of the expressions:

$$H_2(O) = \sum_i^{n_{f_1}} \sum_j^{n_{f_2}} \frac{n_{ij}}{N} \log \frac{n_{ij}}{N} \tag{9}$$

$$H_3(O) = \sum_i^{n_{f_1}} \sum_j^{n_{f_2}} \sum_k^{n_{f_3}} \frac{n_{ijk}}{N} \log \frac{n_{ijk}}{N} \tag{10}$$

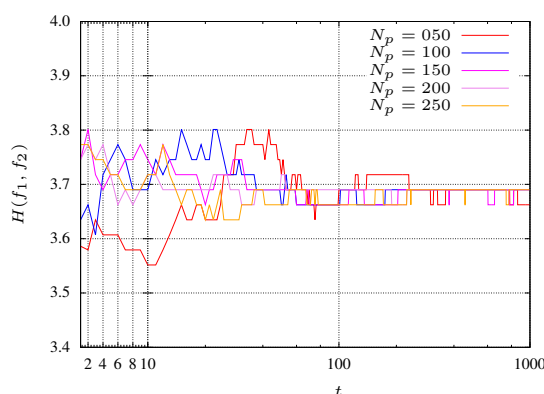
where  $n_{ijk}$  is the number of solutions in the range of cell with indexes  $ijk$ .

The dynamical analysis considers only the elements of the archive  $A(t)$  and, therefore, the Shannon entropy is evaluated using that set of particles.

#### 4.1. Results of $F_1$ Optimization

The first optimization function to be considered is  $F_1$ , with 2 objectives, represented in Equation (3). For measuring the entropy, Equation (9) is adopted (*i.e.*,  $H_2$ ). The results depicted in Figure 1 illustrate several experiments with different population sizes  $N_p = \{50, 100, 150, 200, 250\}$ . The number of parameters is maintained constant, namely with value  $m = 30$ .

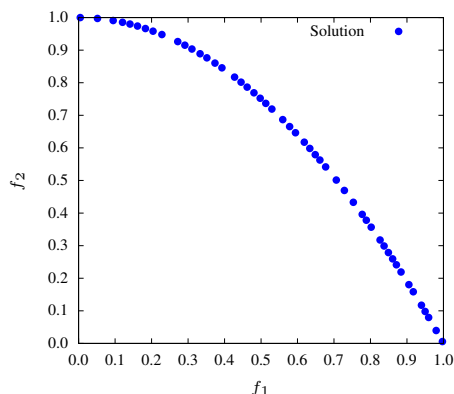
**Figure 1.** Entropy  $H(f_1, f_2)$  during the MOPSO evolution for  $F_1$  function.



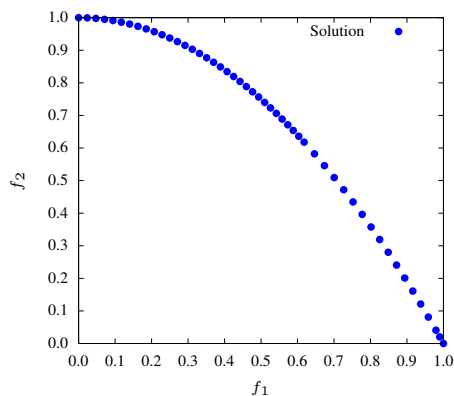
In Figure 1 it is verified that, in general, entropy has a value that hardly varies over the MOPSO execution. At the beginning, outside the transient, the entropy measure is  $H_2 \approx 3.7$ . This transient tends to dissipate as the PSO converges and the particles become organized. Additionally, from Figure 1 it can be seen that the archive size does not influence the PSO convergence rate. Indeed, MOPSO is an algorithm very popular to find optimal Pareto fronts in multi-objective problems, particularly with two objectives.

Figures 2, 3 and 4 show that during all the evolution process the MOPSO presents a good diversity. Therefore, it is expected that entropy has a small variation throughout iterations. Moreover, after generation 90, entropy presents minor variations revealing the convergence of the algorithm.

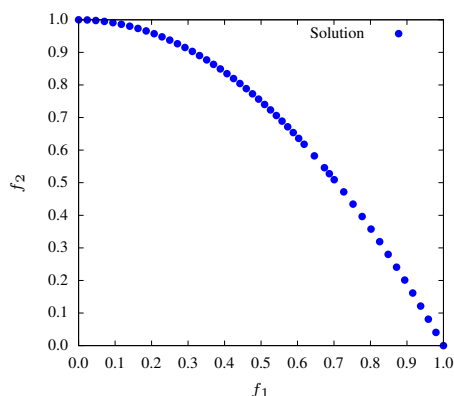
**Figure 2.** Non-dominated solutions at iteration  $t = 1$  for  $F_1$  function and  $N_p = 150$ .



**Figure 3.** Non-dominated solutions at iteration  $t = 90$  for  $F_1$  function and  $N_p = 150$ .



**Figure 4.** Non-dominated solutions at iteration  $t = 1000$  for  $F_1$  function and  $N_p = 150$ .

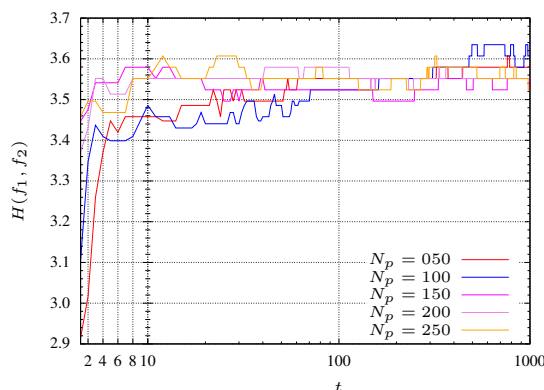


4.2. Results of  $F_2$  Optimization

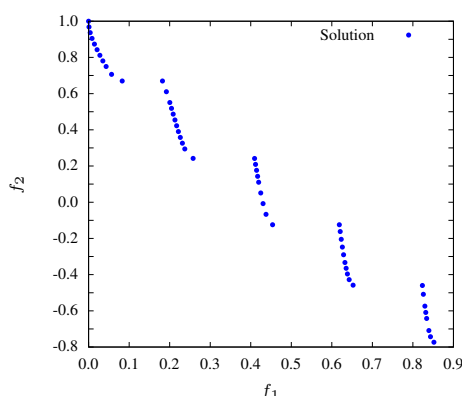
Figure 5 illustrates the entropy evolution during the optimization of  $F_2$ . This function includes 2 objectives and leads to a discontinuous Pareto front represented in Figure 6. The experiments were executed with the same population sizes as for  $F_1$ . It was verified that experiments with a low number

of population solutions have a poor (low) initial entropy, revealing a nonuniform front solution at early iterations.

**Figure 5.** Entropy  $H(f_1, f_2)$  during the MPSO evolution for  $F_2$  function.



**Figure 6.** Non-dominated solutions at iteration  $t = 1000$  for  $F_2$ .



### 4.3. Results of $F_3$ Optimization

In the case of the optimization of function  $F_3$  in Equation (5), three objectives are considered. The entropy evolution is plotted in Figure 7 for  $N_p = \{100, 150, 200, 250, 300, 350, 400\}$ . Moreover, is considered  $m = 12$  and  $\alpha = 100$ .

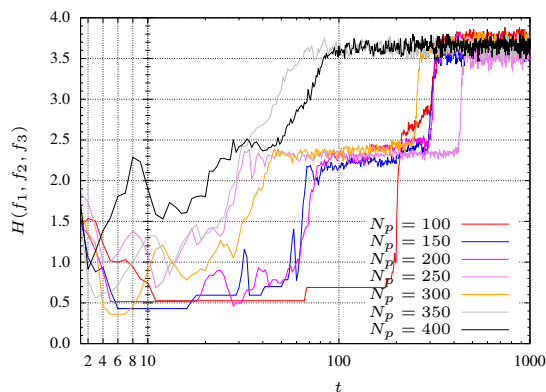
For experiments with a small population size, the convergence of the algorithm reveals some problems. Indeed, for populations with  $N_p = 50$  particles the algorithm does not converge to the Pareto optimal front. With  $N_p = 100$  particles the algorithm takes some time to start converging. This behavior is shown in Figure 7 where pools with many particles (*i.e.*, 350 and 400 particles) reach faster the maximum entropy. In other words, a maximum entropy corresponds to a maximum diversity.

In Figure 7 three search phases are denoted by SP1, SP2 and SP3. The SP1 phase corresponds to an initial transient where the particles are spread for all over the search space with a low entropy. For the experiment with  $N_p = 300$ , phase SP1 corresponds to the first 30 iterations (see Figures 8 and 9). The second phase, SP2, occurs between iterations 40 and 200, where the particles search the  $f_1 \times f_3$  plane, finding mainly a 2-dimensional front (Figures 10 and 11). Finally, in the SP3 phase (*e.g.*, steady state)



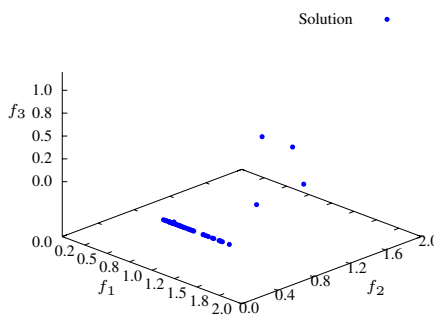
the algorithm approaches the maximum entropy. In this phase, particles move in the entire front and are organized in order to give a representative front with good diversity (see Figures 12 and 13).

**Figure 7.** Entropy  $H(f_1, f_2, f_3)$  during the MPSO evolution for  $F_3$  function.

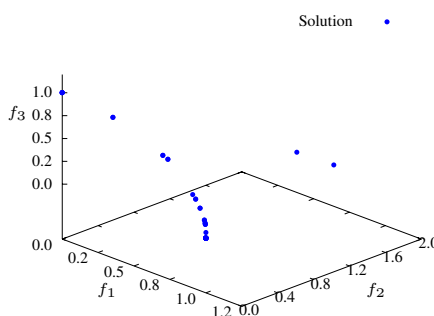


For experiments considering populations with more particles, these phases are not so clearly defined. This effect is due to the large number of particles that allows the algorithm to perform a more comprehensive search. In other words, the MOPSO stores more representative space points helping, in this way, the searching procedure.

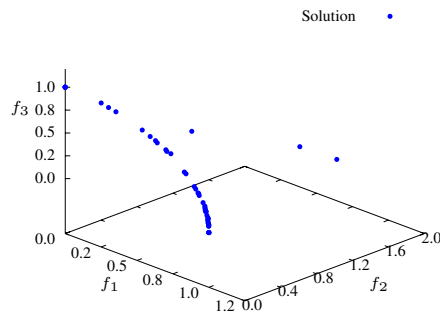
**Figure 8.** Non-dominated solutions at iteration  $t = 1$  for  $F_3$ .



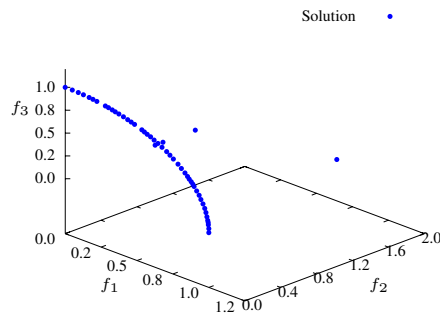
**Figure 9.** Non-dominated solutions at iteration  $t = 30$  for  $F_3$ .



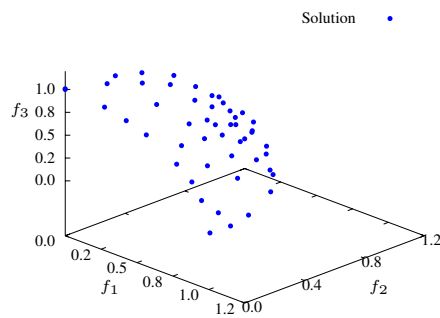
**Figure 10.** Non-dominated solutions at iteration  $t = 40$  for  $F_3$ .



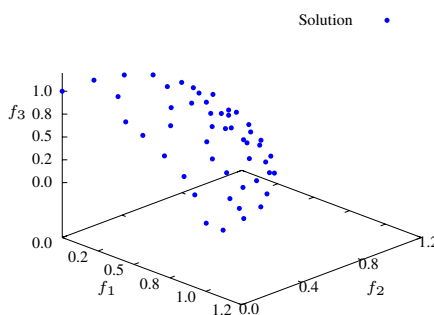
**Figure 11.** Non-dominated solutions at iteration  $t = 200$  for  $F_3$ .



**Figure 12.** Non-dominated solutions at iteration  $t = 300$  for  $F_3$ .



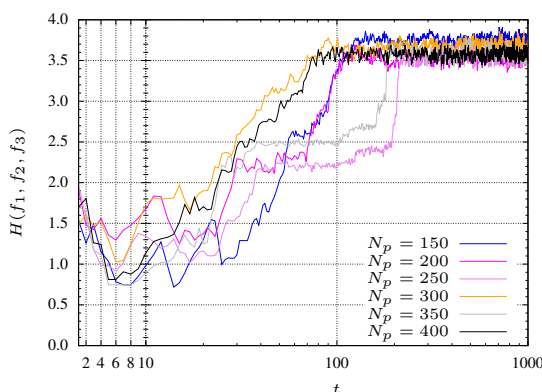
**Figure 13.** Non-dominated solutions at iteration  $t = 1000$  for  $F_3$ .



4.4. Results of  $F_4$  Optimization

The results for the optimization function  $F_4$  are depicted in Figure 14. The function has 3 objectives and the Pareto front is similar to the one for function  $F_3$ . It can be observed that optimization with a larger number of solutions presents a regular convergence, as was verified for  $F_3$ .

**Figure 14.** Entropy  $H(f_1, f_2, f_3)$  during the MPSO evolution for  $F_4$  function.

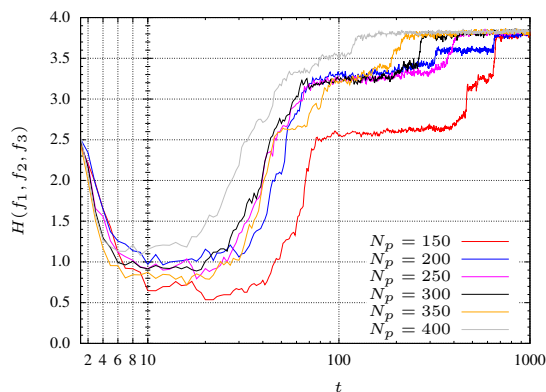


4.5. Results of the  $F_3$ ,  $F_4$  and  $F_5$  Medians

MOPSO is a stochastic algorithm and each time it is executed is obtained a different convergence path for the best particle. In this line of thought, for each test group, 22 distinct simulations were performed and the median taken as representing the entropy evolution. This section presents the entropy evolution for 12 cases test set, 6 for  $F_3$  and 6 for  $F_4$ , with population sizes of  $N_p = \{150, 200, \dots, 350, 400\}$  particles.

Figures 15 and 16 show evolution of the median of the 6 cases test sets for  $F_3$  and  $F_4$ , respectively. It can be seen that the larger the population size, the faster the convergence of the algorithm in finding an uniform spreading covering the Pareto front. The only exception is the case of  $N_p = 200$  particles and  $F_4$  function, that leads to a faster convergence than the case with  $N_p = 250$  particles.

**Figure 15.** Median entropy  $H(f_1, f_2, f_3)$  during the MPSO evolution for  $F_3$  function.



**Figure 16.** Median entropy  $H(f_1, f_2, f_3)$  during the MPSO evolution for  $F_4$  function.

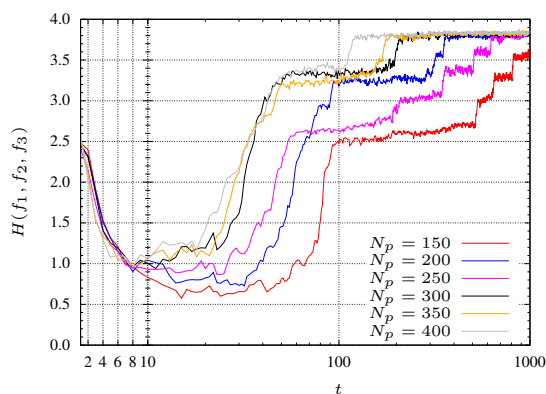
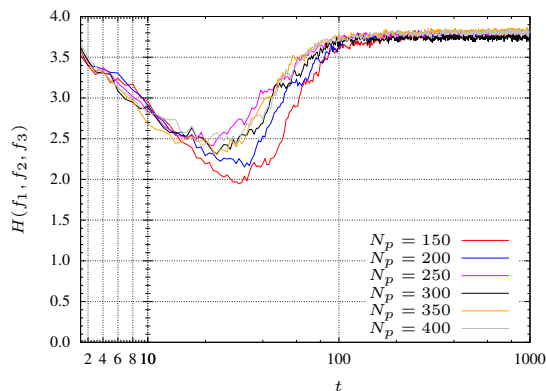


Figure 17 presents the evolution of the median of 6 cases test sets for  $F_5$ . It can also be observed that population size affects the diversity, and consequently the space exploration, of the algorithm at early iterations.

**Figure 17.** Median entropy  $H(f_1, f_2, f_3)$  during the MPSO evolution for  $F_5$  function.



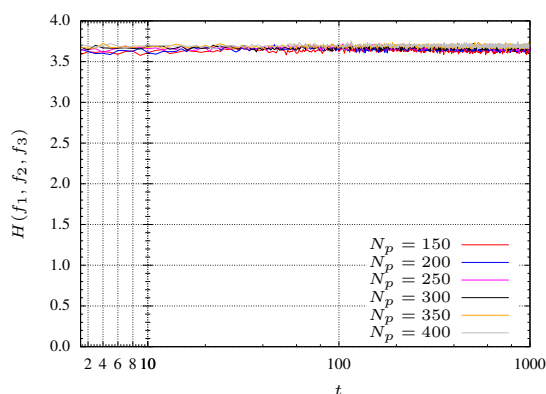
At initial iterations, it is natural to observe a entropy peak because the particles are scattered throughout the objective space, and it is difficult to find near particles among the others. In these stages spread was not maximum (entropy) because the distribution is not uniform.

4.6. Results of the  $F_3$ ,  $F_4$  and  $F_5$  Medians for MSPSO Algorithm

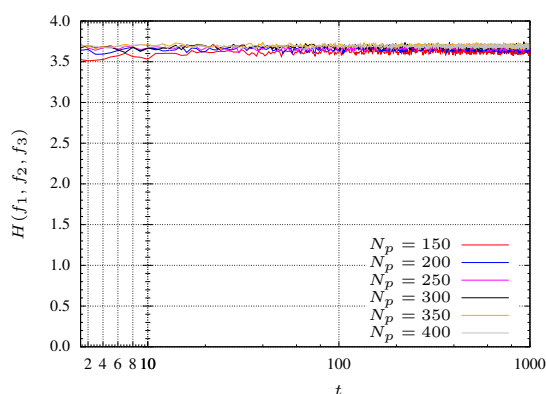
In this section, the functions  $F_3$ ,  $F_4$  and  $F_5$  are optimized using the Spread-constrained Multi-Objective PSO (SMPSO) [30]. The algorithm was downloaded from jMetal website [31]. The algorithm was slightly modified in order to save the archive solutions during the algorithm evolution. This SMPSO has the particularity of producing new effective particles positions in cases where the velocity becomes too high and uses a polynomial mutation as a turbulence factor.

Figures 18 and 19 present the evolution of the median of the 12 cases test sets, 6 for  $F_3$  and 6 for  $F_4$ , with population sizes of  $N_p = \{150, 200, \dots, 350, 400\}$  particles. From these functions it can be seen that the algorithm maintains a good spread during its entire evolution, even in initial iterations. This is due to the polynomial mutation and velocity effect.

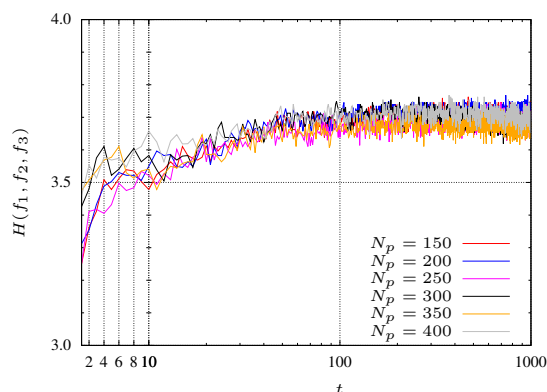
**Figure 18.** Median entropy  $H(f_1, f_2, f_3)$  during the MSPSO evolution for  $F_3$  function.



**Figure 19.** Median entropy  $H(f_1, f_2, f_3)$  during the MSPSO evolution for  $F_4$  function.



When optimizing  $F_5$  (Figure 20) it can be observed a low entropy at early iterations that increases over the iterations, meaning that the spread also improves throughout the execution of the algorithm.

**Figure 20.** Median entropy  $H(f_1, f_2, f_3)$  during the MSPSO evolution for  $F_5$  function.

The MSPSO maintains a good diversity during the search process. This phenomena, does not occur in the standard MOPSO used previously, where diversity decreases in initial stages of the algorithm.

## 5. Conclusions and Future Work

This paper addressed the application of the entropy concept for representing the evolution behavior of a MOPSO. One interpretation of entropy is to express the spreading of a system energy from a small to a large state. This work adopted this idea and transposed it to measure the diversity during the evolution of a multi-objective problem. Moreover, this measure is able to capture the convergence rate of the algorithm. The optimization of four functions was carried-out. According to the entropy index, the  $F_1$  and  $F_2$  functions, with two objectives, are easily and early reached, independently of the number of population particles. When three objectives are considered, for functions  $F_3$ ,  $F_4$  and  $F_5$ , the number of population particles has an important role during the algorithm search. It was verified that entropy can be used to measure the convergence and the MOPSO diversity during the algorithm evolution. On the other hand, when using the MSPSO algorithm an high entropy / diversity was observed during the entire evolution. Therefore, the deployed entropy based index can be used to compare the solution diversity during evolution among different algorithms.

In conclusion, the entropy diversity can be used as a evolution index for multi-objective algorithms in the same way the best swarm element is used in single objective optimization problems.

Future work will be devoted to incorporating entropy based indexes to evaluate the swarm diversity in the algorithm run time. Indeed, as the analysis results presented in this paper confirm, swarm diversity can be evaluated along evolution time, and if the entropy index is lower than a problem function specific threshold, then measures can be adopted to reverse the diversity population decrease. The use of online entropy based indexes can also be applied to the swarm population as well as to the non-dominated archive. This will allow evaluate both populations diversity in order to prevent the MOPSO premature convergence. The former research lines are currently under research and their results will be submitted for another publication soon.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Reyes-Sierra, M.; Coello, C.A.C. Multi-Objective particle swarm optimizers: A survey of the state-of-the-art. *Int. J. Comput. Intell. Res.* **2006**, *2*, 287–308.
2. Kennedy, J.; Eberhart, R.C. Particle Swarm Optimization. In Proceedings of the 1995 IEEE International Conference on Neural Networks, Piscataway, NJ, USA, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
3. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley: Boston, MA, USA, 1989.
4. Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*; Wiley: Hoboken, NJ, USA, 2001.
5. Coello Coello, C.A.; Lechuga, M. MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization. In Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02), Honolulu, HI, USA, 12–17 May 2002; Volume 2, pp. 1051–1056.
6. Zhou, A.; Qu, B.Y.; Li, H.; Zhao, S.Z.; Suganthan, P.N.; Zhang, Q. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm Evol. Comput.* **2011**, *1*, 32–49.
7. Zhao, S.Z.; Suganthan, P.N. Two-lbests based multi-objective particle swarm optimizer. *Eng. Optim.* **2011**, *43*, 1–17.
8. Solteiro Pires, E.J.; Tenreiro Machado, J.A.; de Moura Oliveira, P.B. Dynamical modelling of a genetic algorithm. *Signal Process.* **2006**, *86*, 2760–2770.
9. Solteiro Pires, E.J.; Tenreiro Machado, J.A.; de Moura Oliveira, P.B.; Cunha, J.B.; Mendes, L. Particle swarm optimization with fractional-order velocity. *Nonlinear Dyn.* **2010**, *61*, 295–301.
10. Schott, J.R. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master Thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, MA, USA, 1995.
11. Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II; In *Parallel Problem Solving from Nature-PPSN VI*, Proceeding of 6th International Conference, Paris, France, 18–20 September 2000; Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J., Schwefel, H.P., Eds.; Springer: Berlin, Germany, 2000; Volume 1917, pp. 849–858.
12. Okabe, T.; Jin, Y.; Sendhoff, B. A Critical Survey of Performance Indices for Multi-Objective Optimisation. In Proceedings of the 2003 Congress on Evolutionary Computation, Canberra, Australia, 8–12 December 2003; Volume 2, pp. 878–885.
13. Derrac, J.; Garca, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18.

14. Solteiro Pires, E.J.; de Moura Oliveira, P.B.; Tenreiro Machado, J.A. Multi-objective MaxiMin Sorting Scheme. In Proceedings of Third Conference on Evolutionary Multi-Criterion Optimization—EMO 2005, Guanajuato, México, 9–11 March 2005; Lecture Notes in Computer Science, Volume 3410; Springer-Verlag: Berlin and Heidelberg, Germany, 2005; pp. 165–175.
15. Galaviz-Casas, J. Selection Analysis in Genetic Algorithms. In *Progress in Artificial Intelligence—IBERAMIA 98*, Proceedings of 6th Ibero-American Conference on Artificial Intelligence, Lisbon, Portugal, 5–9 October 1998; Coelho, H., Ed.; Lecture Notes in Computer Science, Volume 1484; Springer: Berlin and Heidelberg, Germany, 1998; pp. 283–292.
16. Masisi, L.; Nelwamondo, V.; Marwala, T. The Use of Entropy to Measure Structural Diversity. In Proceedings of IEEE International Conference on Computational Cybernetics (ICCC 2008), Atlanta, GA, USA, 27–29 November 2008; pp. 41–45.
17. Myers, R.; Hancock, E.R. Genetic algorithms for ambiguous labelling problems. *Pattern Recognit.* **2000**, *33*, 685–704.
18. Shapiro, J.L.; Prügel-Bennett, A. Maximum Entropy Analysis of Genetic Algorithm Operators. In *Evolutionary Computing*; Lecture Notes in Computer Science, Volume 993; Springer-Verlag: Berlin and Heidelberg, Germany, 1995; pp. 14–24.
19. Shapiro, J.; Prügel-Bennett, A.; Rattray, M. A Statistical Mechanical Formulation of the Dynamics of Genetic Algorithms. In *Evolutionary Computing*; Lecture Notes in Computer Science, Volume 865; Springer-Verlag: Berlin and Heidelberg, Germany, 1994; pp. 17–27.
20. Kita, H.; Yabumoto, Y.; Mori, N.; Nishikawa, Y. Multi-Objective Optimization by Means of the Thermodynamical Genetic Algorithm. In *Parallel Problem Solving from Nature—PPSN IV*, Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, Berlin, Germany, 22–26 September 1996; pp. 504–512.
21. Farhang-Mehr, A.; Azarm, S. Entropy-based multi-objective genetic algorithm for design optimization. *Struct. Multidiscipl. Optim.* **2002**, *24*, 351–361.
22. Qin, Y.; Ji, J.; Liu, C. An entropy-based multiobjective evolutionary algorithm with an enhanced elite mechanism. *Appl. Comput. Intell. Soft Comput.* **2012**, *2012*, No. 17.
23. Wang, L.; Chen, Y.; Tang, Y.; Sun, F. The Entropy Metric in Diversity of Multiobjective Evolutionary Algorithms. In Proceedings of 2011 International Conference of Soft Computing and Pattern Recognition (SoCPar), Dalian, China, 14–16 October 2011; pp. 217–221.
24. Wang, L.; Chen, Y. Diversity based on entropy: A novel evaluation criterion in multi-objective optimization algorithm. *Int. J. Intell. Syst. Appl.* **2012**, *4*, 113–124.
25. Ben-Naim, A. *Entropy and the Second Law: Interpretation and Misinterpretations*; World Scientific Publishing Company: Singapore, Singapore, 2012.
26. Shannon, C.E. A mathematical theory of communication. Available online: <http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf> (accessed on 4 December 2014).
27. Zitzler, E.; Deb, K.; Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.* **2000**, *8*, 173–195.
28. Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E. Scalable Multi-Objective Optimization Test Problems. In Proceedings of the 2002 Congress on Evolutionary Computation (CEC2002), Honolulu, HI, USA, 12–17 May 2002; pp. 825–830.



29. Zhang, Q.; Zhou, A.; Zhao, S.; Suganthan, P.N.; Liu, W.; Tiwari, S. *Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition*; Technical Report CES-487; Nanyang Technological University: Singapore, Singapore, 2008.
30. Nebro, A.J.; Durillo, J.J.; García-Nieto, J.; Coello Coello, C.A.; Luna, F.; Alba, E. SMPSO: A New PSO-based Metaheuristic for Multi-objective Optimization. Presented in IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MCDM 2009), Nashville, TN, USA, 30 March–2 April 2009; pp. 66–73.
31. Home page of jMetal. Available online: <http://jmetal.sourceforge.net> (accessed on 4 December 2013).

© 2013 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).