Title: A new approach to particle swarm optimization algorithm

Author: Ireneusz Gościniak

Citation style: Gościniak Ireneusz. (2015). A new approach to particle swarm optimization algorithm. "Expert Systems with Applications" (Vol. 42, iss. 2 (2015), s. 844-854), doi 10.1016/j.eswa.2014.07.034

[postprint]

A New Approach to Particle Swarm Optimization Algorithm

Ireneusz Gosciniak*

Institute of Computer Science, University of Silesia, Będzińska 39, 41-200 Sosnowiec, Poland

Abstract

Particularly interesting group consists of algorithms that implement co-evolution or co-operation in natural environments, giving much more powerful implementations. The main aim is to obtain the algorithm which operation is not influenced by the environment. An unusual look at optimization algorithms made it possible to develop a new algorithm and its metaphors define for two groups of algorithms. These studies concern the particle swarm optimization algorithm as a model of predator and prey. New properties of the algorithm resulting from the co-operation mechanism that determines the operation of algorithm and significantly reduces environmental influence have been shown. Definitions of functions of behaviour scenarios give new feature of the algorithm. This feature allows self controlling the optimization process. This approach can be successfully used in computer games. Properties of the new algorithm make it worth of interest, practical application and further research on its development. This study can be also an inspiration to search other solutions that implementing co-operation or co-evolution.

Keywords: Co-evolutionary systems, PSO algorithm, predator-prey algorithm, immune algorithm, optimization method, games, artificial intelligence, entropy, multifractal analysis.

*Corresponding author

Email address: `ireneusz.gosciniak@gmail.com` (Ireneusz Gosciniak)

1. Introduction

Observations of systems of living organisms are inspiration for the creation of modern computational techniques. Evolution algorithms are like metaphors of biological organisms, adopting from them terminology and mechanisms of operation as well. Adaption mechanisms borrowed from biology decide on the distribution of individuals in the environment. These operators perform the functions responsible for the exploration of environment and the exploitation of areas of local extrema. Adaptation mechanisms make these algorithms more efficient than a completely random search of solution space. Creating an artificial system as a metaphor or set of metaphors connected with the functioning of living organisms removes associated with it restrictions. Unfortunately, for such a system greater limits associated with its implementation are imposed. From the No Free Lunch Theorem [1] one can result that there is no universal optimization algorithm for all classes of tasks. This is a consequence of relation between the behaviour of algorithm and the problem being solved. However, it gives the inspiration to create new solutions and conducts investigations on the behaviour of the algorithm and its suitability for solving the problems of particular class. In most cases it leads to the attempts to increase the calculation efficiency by modifying the existing algorithms. Particularly interesting group consists of algorithms that implement co-evolution in natural environments because the NFL theorem cannot be applied to them. Evolutionary algorithms differ from stochastic algorithms in very efficient adaptive mechanism for searching the solution space. That is why stochastic algorithms require greater number of iterations in the optimization process but are less likely to stop the optimization process in the local optimum. The usefulness of the algorithm is determined by the rules that are well-developed for stochastic algorithms. However, defining metaphors of natural environments for the algorithms – that is, *de facto*, the creation of completely new algorithms is not trivial.

New algorithm must therefore be searched in the group of algorithms that implement co-evolution (cooperation) in natural environments basing on rules developed for stochastic algorithms. The main aim is to obtain an algorithm on which operation the environment would have a very small impact. This feature allows controlling the optimization process and not only tuning the algorithm to the problem being solved as it is nowadays. Many problems are treated as unchangeable – they are represented by the stationary environment. However, the change in resources, tasks or other elements of the system results in the changes of problem from stationary into non-stationary – these problems are represented by the non-stationary environment. The majority of the algorithms used in non-stationary environments are adapted from algorithms applied in the stationary environments. The presented algorithm has been designed to use it in the non-stationary environments. Thus, the article presents the situation which is opposite to the mostly discussed. An unusual look at the optimization algorithms made it possible to develop the new algorithm and define its metaphors in two groups of algorithms. This algorithms can be used to describe artificial life. The resulting algorithms are effective optimization algorithms and the proposed approach introduces new features in their operation. The new algorithm and its metaphors in the group of immune algorithms and particle swarm optimization algorithms are presented in the article. Functions of behaviour scenarios are defined in the particle swarm optimization algorithm. New properties of the algorithm resulting from the co-operation mechanism have been shown. It determines the algorithm behaviour and reduces the environmental impact. This is the original and unpublished achievement of work. Immune algorithm is not widely discussed because research results were partially presented in [2] and the results of research carried out on its development require a new study. Modern PSO algorithms of high-efficiency should be classified as the hybrid algorithms. This proposed algorithm is presented rather as a base one – the base for future modifications. It was compared with different algorithms, also older (there are the references to the older literature) because presented there solutions can be considered as a base form, which is subject to further improvements. On the basis of the description it is being easy to notice the necessity to make modifications that will create a hybrid algorithm of high efficiency.

2. The comparison of selected algorithms

The analysis of algorithms operation becomes more complicated. Modifications affect many aspects of algorithm's operation. There are lots of terms that are closely dependent on each other. Exploration and exploitation of the solution space are contradictory goals. It becomes extremely difficult to maintain an appropriate balance between exploration and exploitation of the solution space during the work of an algorithm. Convergence realises the exploitation and reduces the diversity of the population. Reducing of the population diversity causes the loss of information on the solution space – the memory loss. Concentrated individuals form a cluster. The excessive closeness of particles does not increase the information on the solution space. So in this place it should be reminded that the aim of this algorithm operation is to search for solution space to designate the global extreme or set of local extremes. There is also the effect of modifications on the algorithm operation. Frequently used modification is a mutation – but it may have the character improving the exploration or exploitation. Co-evolutionary systems are the most interesting. The co-evoluton can be different in each group of algorithms and the functions creating co-evolution can be diffrent in each cooprating system There is also the group of modifications basing on applying of other methods known from mathematics – implementing as a local method. Presentation of the structure of base algorithm will be preceded by the discussion on the selected algorithms. This discussion is very general, and it has only been used to introduce existing solutions. However, this would help to understand the concept of the new algorithm.

During dislocation, population forms a compact group of individuals that exploits one part of the solution area; however, exploration is carried out by the movement of population. Phases of movement can be distinguished – in each phase the dislocation effectiveness of a population is not the same. Swarm adapts to the environment during subsequent iterations. The swarm leader represents the position of the best adaptation (the best solution). The assignment of swarm particle neighbours is performed usually once at the beginning of calculations – it makes the designation of the best adapted neighbour easier. The change in behaviour of particles swarm is a function of changes in the leader behaviour. There are many modifications of the above-mentioned PSO algorithm – the majority of them can be found in [3], [4], [5] and [6]. The behavior of the PSO algorithm depends on the internal weights. The exploration or exploitation nature of the algorithm work depends on the inertia weight. Appropriate change in this coefficient during the algorithm work will have a significant impact on the efficiency of its work. Linear decrease in the weight factors was proposed in the work [7]. In the paper [8] the inertia weight is reducing in the course of the algorithm work. In [9] the decrease in inertia weight using fuzzy methods was proposed. In paper [10] is proposed an improved self-adaptive particle swarm optimization algorithm (ISAPSO). In the process algorithm of evolution parameters are changed dynamically: cognitive and social learning rates parameters. It allows to maintain the diversity of the population. Control strategy has a random character which permits to take into account the various constraints of solved the problem. In [11] a local neighbourhood version is used – only the behaviour of neighbouring particles is taken into account. Keeping diversity of populations during the algorithm work is also important for PSO algorithms. The diversity of the population increases the chances of local extreme leaving. For keeping diversity of population PSO algorithm with self-organized criticality was introduced in [12]. In order to achieve a greater variety of particles the "critical value" is created when two particles are too close to each other. Negative entropy was used in [13] to discourage premature convergence.

The neighborhood of other particles has the impact on the behavior of particles in the swarm. The neighborhood's analysis is the basis for separation of species or the use of multi-swarm. In [14] a dynamically changing neighborhood was used. The influence of neighbors, which depends on the fitness function and the position in relation to particles, is presented in the study [15]. To achieve it, in [16] some collision-avoiding mechanisms were applied, the individuals moving was used in the study [12], whereas mutation was applied in the paper [17]. It should be here mentioned that operators typical of genetic algorithms, such as mutation, crossover or selection were used in PSO [18]. In [19] the introduction of an additional repellor was suggested. It influences the swarm behaviour by directing it into the areas of the environment which have better adaptation.

The use of multi-swarm allows to maintain the diversity of the particles. Algorithms creating clusters are particularly noteworthy. There are very interesting groups of algorithms.

In multi-swarm and clusters creating systems there are key issues such as: how to define a promising area in the solution space and how to implement motion of the particles in the direction of various sub-areas, how to determine the required number of sub-swarms or clusters, and how to generate the sub-swarms or clusters.

In [20] NbestPSO algorithm was proposed, which is designed to locate many solutions. Particle's neighborhood in NbestPSO algorithm is defined as the closest particles in the population. The best neighborhood is determined on the base of the average distance of the nearest particles .

In [21] NichePSO algorithm was proposed – the main swarm can create sub-swarm when the niche is identified. The criterion for a sub-swarm creation is the lack of significant changes in subsequent iterations, while the sub-swarm can absorb particles or other sub-swarms in dependance on the distance. In [22] the adaptive niching PSO algorithm (ANPSO) is proposed; the radius of the species is here determined on the base of population statistics.

PSO algorithm basing on species (SPSO) is presented in [23, 24]. This algorithm dynamically adjusts the number and size of swarms through the creation of ordered list of particles sorted according to their fitness and grouping particles of a particular species. In every generation, SPSO aims to identify many spores of species into the swarm. The particles within a radius of the spore are assigned to the same species.

In the improved version of SPSO the mechanism to remove duplicates of species particles was introduced [25]. Another improved version of SPSO using regression (rSPSO) was also introduced [26]. In [27] the use of k-means clustering algorithm grouping particles in the pre-defined number of clusters was proposed. In order to obtain clusters stability the algorithm iterations are executed three times. To determine the number of clusters in the "k-means PSO" algorithm in [28] particle distribution generated by a combination of several probabilistic distributions is proposed – each cluster corresponds to other distribution. Then, finding the optimal number of clusters is equivalent to finding the best-fit models. This algorithm gives better results for the problems described in the stationary environments than SPSO algorithms and ANPSO.

Co-evolution of swarms (CESO) was proposed in [29]. In the CESO two swarms, one of which uses the differential evolution (CDE) [30], and the second one model of the PSO, co-operate with each other. The swarm, which uses the CDE is responsible for the diversity while a swarm of PSO tracks the global optimum.

In Clustering PSO (CPSO) [31] every particle learns - knows its own, historically the best position and the best position of the nearest neighbour, which affects the motion of the particles. Using hierarchical clustering method the whole swarm in CPSO can be divided into smaller sub-swarms. It enables the adaptive detection of sub-areas. In the CPSO, hierarchical clustering method is realized by two steps: rough grouping and clusters refining. The strategy for the best global particle was also introduced in CPSO [31].

In [32] some simplifications in comparison to the original CPSO were applied: the training process has been removed, hierarchical clustering method is simplified to only one phase.

In [33] co-evolution of two swarms was suggested; one of them optimizes the penalty factors and the second looks for the optimum solution (swarms move in the spaces of optimum solutions for penalty factors for the contrary swarm). Another example of the cooperation system is the predator-prey model [34] which uses the model of games theory implementing war strategy. This approach was used in optimization of fuzzy clustering [35]. Article [36] describes the cooperation (co-evolution), which uses the strategy of sardines being attacked by sea wolfs (orcas). Many common features influencing the work of algorithms results from the above discussion. Mechanisms of exploitation of local extremes and mechanism of exploration of space solutions depend on each other. Solutions that make these mechanisms independent on each other should be searched in cooperating or co-evolutionary systems. The operation of these sub-systems is dependent on each other, but their functions may be different.

And so, the algorithm implements co-operation of two systems designating local extremes – the idea is found in [11] and [19]. The particles of both systems are directed to the same local extremes – differently than in the work [33]. The algorithm implements a game strategy based on the round-up strategy, which seems to be similar to the strategy described in [34]. The main difference in the proposed algorithm is a way of elimination of the particles, which results from the excessive proximity of particles in the same group. Examples of prevention of excessive particles closeness are above described. In many studies e.g. [5], [37] the mutation is considered to be the important coefficient in the progress of the algorithm. The proposed method reduces the range of mutations and causes adaptation based on the location of the particles. A strong mutation was replaced by the creation of random particles instead of eliminated ones.

The algorithm presented below seems to be similar to the algorithm presented in the work [38]. However, it is characterised by different operation mechanics. It is a predator-prey algorithm, that illustrates the example of the behavior of sardine shoals and hunting orcas. Predators are heading in center of shoals of sardines – it looks like dispersing of preys, which escape from predators. This contributes to the fact that the particles avoid of local solution optimum to find the global and optimal solution. Predators play the role of exploatation (they realize convergence), while preys escaping from predators realize exploration of the solution space (they play the role of algorithm diversification). The algorithm implements a scenario in which the nearest predator is elected and it is determined whether the particle being the prey escapes (which depends on the distance between predator and prey and basing on it the escape velocity is calculated). This description indicates significant differences between the algorithm [38] and the algorithm proposed in this paper.

The algorithm tuning is also important. Tuning algorithm and also control the operating parameters of the algorithm is also becoming an important issue. Depends on a compromise between speed and accuracy of the algorithm work - or even find a global solution. Makes it necessary to analyze the behavior of particles, their dynamics and trajectory. In [39] presented a new strategy for analyzing the behavior of the algorithm and new parameters for increasing the efficiency of the algorithm. Due to the mechanism of co-evolution and self-adaptation only large changes in parameters give a significant effect. A tuning of the algorithm is not widely discussed because it describes the behavior of the algorithm – so the importance of parameters becomes obvious.

3. The base algorithm

The new algorithm implements cooperation (in evolutionary systems it is termed co-evolution) of the two systems, hereinafter referred to as elements of sample points and seed points (similarly to stochastic systems). The algorithm description is presented below:

1. Random creation of initial trial's sample points and seed points.
2. Activation of a seed point – random choice of the seed point
3. Reduction of seed points – checking whether other seed points are present in the defined neighbourhood of the active seed point. The active seed point is the best of them. Seed points with poorer adaptation are replaced by the randomly created ones.
4. Activation of the sample points (defining the cluster) – sample points that are present in the neighbourhood of the seed point become active.
5. Reduction of the active sample points – among the active sample points, the elements which are closer than the defined distance are removed. They are replaced by randomly created sample points.
6. Processing – the replacement of active sample points elements and active seed point with the results of their processing according to the predefined rules (the execution of local method). The rule of processing is described below.

7. Iteration from step 2 until reaching the final stop.

Behaviour of the algorithm still requires further clarification. Sample points and seed points of the initial trial are randomly created. Set of sample points is greater than set of seed points. Sample points create a kind of net, nodes of which are moving in the direction of local extrema under the influence of processing methods. The density of sample points in the areas of local extrema will be higher than in the other areas. On the basis of information about active sample points, seed points define their new positions moving also to local extrema. Speed of movement (of both active elements of sample points and seed point) depends on the density of the sample points. The speed is greater when the density is less. Thanks to this, active seed point by using the information from active trial points moves much faster outside the local extrema than inside them. It is the movement of seed point which is similar to the eye tracking.

Generally, sample points are moving slower than the seed points. Sample points are responsible for the exploration of solution space, and the seed points for the exploitation of local extreme – close correlation is between these elements. The reduction of seed points demonstrates the identification of local extrema as well as helps in the exploration of the environment. The reduction of sample points is the indicator of local extrema exploitation as well as it indicates the exploration of solution space. Sample points perform some kind of approximation of the environment function. Seed points initiate the process of optimization in the location of these points. This cooperation significantly reduces the impact of the environment to the algorithm operation. Stop criterion is based on the monitoring of solutions generated by the algorithm in connection with information about the reduction parameters.

4. The construction of metaphors

The processing method determines the behaviour of two groups of particles: one group is called sample points with function of exploration and the second one is called seed points with function of exploitation over the search space.

This method has been implemented as a metaphor for the immune system and the PSO system. Due to the fact that metaphors are not implemented as a canon of these algorithms, they will hereafter be referred to as Semi-Immune and Semi-PSO.

The exception from the standard approach to the immune algorithm is that the antigen is represented by the seed points, not by the environment. Furthermore, antigens change their position under the influence of antibodies, which are represented by the sample points. A similar situation happens in the nature when viruses mutate to defend themselves against the immune system. However, the mechanism of autoimmunity is the basis for the removal of antibodies. This mechanism results in autoaggression to its own cells. On the other hand, the removal of the weakest of the antibodies that are grouped and surrounded by antigens is an interpretation of the mechanism of their reduction.

The Semi-PSO algorithm implements a strategy of round-up – it seems to be a kind of predator-prey strategy – as the cooperation of two particles systems: predators represented by sample points and prey – which are the seed points. Prey is encirclemented by a group of predators. Predators move in the direction of their group leader and prey. The group leader tries to cut off the escape route of prey. Prey, on the basis of both the leader and the weakest predator observations, runs towards the safe place – a local extremum. Encirclement of more than one prey results in the elimination of the weakest of them. However, excessive approch of predators causes the elimination of the weakest of them. These mechanisms seem to be the natural elements of a struggle for survival. The principle of predator-prey algorithms described so far in other papers was different. The adaptation of weighting coefficients that create the behavioral scenarios seems to be the violation of these algorithms canon. A detailed description of the Semi-PSO algorithm is to be presented in later chapters. This algorithm has a high efficiency of both exploration and exploitation of the solution space.

5. Semi-PSO – round-up algorithm

Lyrics in a playful way show the strategy of the algorithm:

It is not the point

to catch the bunny

But to chase him ...

Zieliński A., Osiecka A., Bunny, Skaldowie, Muza 1969

(in Polish, translation by author).

The strategy of the algorithm supports the view that the most enjoyable thing about playing is not catching the "bunny", but the process of tracking it down. It demonstrates the practical importance of this strategy.

The proposed algorithm shows cooperation of two systems of particles: predators and their prey. Predators use the strategy of round-up – they outnumber the prey. Encirclemented by the group of predators, prey moves faster than they are. The prey by escaping from one encirclement falls into another one. Prey, based on the observation of the leader of predators and the weakest of them, is run towards the safe place that is to the local extreme. In this algorithm, neighbourhood of predators and prey must be determined in each iteration. Predators move in the direction towards the prey and the group leader who tries to cut off escape route of prey. Predators and prey are concentrated in areas of local extrema. Encirclement of more than one prey causes the elimination of the weakest of them. On the other hand, the excessive rapprochement of predators causes an elimination of the weakest of them. These mechanisms seem to be the natural elements of a struggle for survival. Randomly created individuals replace the eliminated ones. The discussed algorithm can be described as follows:

1. Random creation of predators ($E_i = [e_{i1}, e_{i2}, ..., e_{id}]$) and prey ($Z_i = [z_{i1}, z_{i2}, ..., z_{id}]$) – cooperating system.
2. Activation of the prey – random choice of a particle.
3. Prey elimination – searching whether the given neighbourhood of the active prey for other individuals from their group are present; worse adapted

ones are replaced by randomly created new ones; prey that is left becomes the active one.

4. Activation of the predators – predators in the neighbourhood of the prey become active.
5. Predators elimination – the predators that are the weakest and are too close to the strong ones are eliminated and replaced by non-active randomly created ones.
6. Processing – replacement of active elements, as follows:

   Assessing the value of fitness function for the active predators $(f(E_i))$ and the selection of the best $(E_{i,best})$ and the worst $(E_{i,worst})$ of them. Calculation of the velocity vectors for predators (V_i^E) and prey (V_i^Z) using the following equations:

$$\begin{aligned} V_i^E(t+1) &= w^E V_i^E(t) + c_1^E \varphi_1^E (E_{i,best} - E_i) \\ &\quad + c_2^E \varphi_2^E (Z_i - E_i), \end{aligned} \tag{1}$$

$$\begin{aligned} V_i^Z(t+1) &= w^Z V_i^Z(t) + c_1^Z \varphi_1^Z (E_{i,best} - Z_i) \\ &\quad + c_2^Z \varphi_2^Z (Z_i - E_{i,worst}). \end{aligned} \tag{2}$$

   where: φ_1^E, φ_2^E, φ_1^Z, φ_2^Z – are the random values from the range $[0,1]$; c_1^E, c_2^E, c_1^Z, c_2^Z – are the learning rates; w^E, w^Z – are the particle's inertia weights.

   The values of these coefficients are adapted in order to obtain the relevant scenarios of behaviour. The maximum values of the weighting coefficients are only given. Weighting coefficients have strong influence on the behaviour of particles, so their adaptation has a significant impact on the efficiency of the algorithm work. New positions of the predators and position of the prey are determined according to the following equations:

$$E_i(t+1) = E_i(t) + V_i^E(t+1), \tag{3}$$

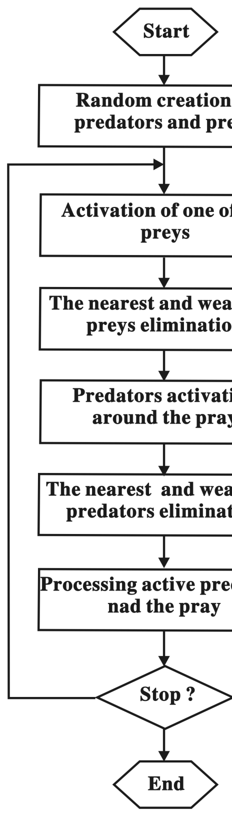

Figure 1: The algorithm diagram.

$$Z_i(t+1) = Z_i(t) + V_i^Z(t+1). \tag{4}$$

7. Iteration from step no. 2 until termination criterion is reached.

The movement of predators depends on the position of the best predator and prey location. However, the movement of the prey depends on the position of the best and worst of predators – the movement is carried out in the direction of the best predator and in the direction opposite to the worst one. This processing allows to detect the local extrema. Predators are responsible for the exploration of solution space, and the prey is responsible for exploitation of local extrema – there is strong interdependence between them. Elimination of prey demonstrates the identification of local extremum; it is also an indicator of the environment exploration. The reduction of predators is an indicator of local extrema exploitation and the mechanism of the solution space exploration (cluster identification). The weighting coefficients are responsible for creating scenarios of behaviour that are dependent on the mutual position of active elements. Predator being in close proximity to the prey tries to cut off its escape route (prey escapes in the direction of local extreme). The direction in which it moved so far has much less influence on its behaviour. Also, predator movement in the direction of prey may facilitate its escape. Therefore, in this case the weighting coefficients w^E and c_2^E should have less influence on predator behaviour. The c_1^E weighting coefficient responsible for the movement in the direction of the leader of predators should have the dominant influence on the behaviour of the predator. In this particular situation weighting coefficients w^E and c_2^E have dominant influence on the movement of the leader of predators. In the case of long distance, predator moves fast in the direction of prey. Therefore, weighting coefficients w^E and c_2^E have the dominant influence on his behaviour. In this case the weighting coefficient c_1^E would be less important in the behaviour of the predator. Scenarios of prey behaviour would also be dependent on its position in the relation to predators. In the case of bad position of prey, that is a short distance from the weakest of the predators, the weighting coefficients w^Z and c_2^Z

determining the rapid escape from this location would have a significant impact on its behaviour. However, in the case of small distance from leader of predators to prey, providing also of the approach to a local extreme, weighting coefficient c_1^Z would have a significant impact on its behaviour. Examples of carrying out the functions of the weighting coefficients are presented below.

Example I:

$$c_1^E = \frac{\|E_{i,best} - E_i\|}{\|E_{i,best} - E_i\| + \|Z_i - E_i\|} \cdot c_{1,max}^E, \tag{5}$$

$$c_2^E = \frac{\|Z_i - E_i\|}{\|E_{i,best} - E_i\| + \|Z_i - E_i\|} \cdot c_{2,max}^E, \tag{6}$$

$$w^E = c_2^E \tag{7}$$

and

$$c_1^Z = \frac{\|E_{i,best} - Z_i\|}{\|E_{i,best} - Z_i\| + \|Z_i - E_{i,worst}\|} \cdot c_{1,max}^Z, \tag{8}$$

$$c_2^Z = \frac{\|Z_i - E_{i,worst}\|}{\|E_{i,best} - Z_i\| + \|Z_i - E_{i,worst}\|} \cdot c_{2,max}^Z. \tag{9}$$

$$w^Z = c_2^Z \tag{10}$$

Example II:

$$c_1^E = \frac{\min\left\{\|E_{i,best} - E_i\|, \|Z_i - E_i\|\right\}}{\|E_{i,best} - E_i\|} \cdot c_{1,max}^E, \tag{11}$$

$$c_2^E = \frac{\min\left\{\|E_{i,best} - E_i\|, \|Z_i - E_i\|\right\}}{\|Z_i - E_i\|} \cdot c_{2,max}^E, \tag{12}$$

$$w^E = \frac{\min\left\{\|E_{i,best} - E_i\|, \|Z_i - E_i\|\right\}}{\|Z_i - E_i\|} \cdot w_{max}^E, \tag{13}$$

or

$$w^E = \left(1 - \frac{\min\left\{\|E_{i,best} - E_i\|, \|Z_i - E_i\|\right\}}{\|E_{i,best} - E_i\|}\right) \cdot w^E_{max}, \tag{14}$$

where w^E can be 0,

and

$$c^Z_1 = \frac{\min\left\{\|E_{i,best} - Z_i\|, \|Z_i - E_{i,worst}\|\right\}}{\|E_{i,best} - Z_i\|} \cdot c^Z_{1,max}, \tag{15}$$

$$c^Z_2 = \frac{\min\left\{\|E_{i,best} - Z_i\|, \|Z_i - E_{i,worst}\|\right\}}{\|Z_i - E_{i,worst}\|} \cdot c^Z_{2,max}, \tag{16}$$

$$w^Z = \frac{\min\left\{\|E_{i,best} - Z_i\|, \|Z_i - E_{i,worst}\|\right\}}{\|Z_i - E_{i,worst}\|} \cdot w^Z_{max}, \tag{17}$$

or

$$w^Z = \left(1 - \frac{\min\left\{\|E_{i,best} - Z_i\|, \|Z_i - E_{i,worst}\|\right\}}{\|E_{i,best} - Z_i\|}\right) \cdot w^Z_{max}, \tag{18}$$

where w^Z can be 0.

In these expressions $c^E_{1,max}$, $c^E_{2,max}$, w^E_{max}, $c^Z_{1,max}$, $c^Z_{2,max}$, w^Z_{max} are the maximum values of weighting coefficients. These coefficients have the same meaning as coefficients of a typical PSO algorithm. However, the functions described above are responsible for adaptation. Consequently, small changes in the values of these coefficients will not influence visibly on the behavior of the algorithm. The principles to determine their values is the same as in PSO algorithm [4].

The algorithm can take into account the situation in which the predators are on neighbouring hunting areas, while prey can move throughout all the space. This algorithm, as the PSO algorithm, belongs to the group of stochastic algorithms. The present algorithm in a very effective way combines stochastic search of solution space with the search described by behaviour of the particles. This algorithm also has a property of observation similar to behaviour of the eye.

6. The analysis of Semi-PSO algorithm

The evaluation of the algorithm is performed by comparing the effectiveness of the designation of global extreme of test functions in relation to certain algorithms of optimization. Test functions describe the stationary environment. Acronyms formed for compared algorithms are as follows: S-PSO – Semi Particle Swarm Optimization (discussed algorithm), PSO – Particle Swarm Optimization [40], RCGA – Real-Coding Genetic Algorithm [41], CGA – Continuous Genetic Algorithm [42], CHA – Continuous Hybrid Algorithm [43], ECTS – Enhanced Continuous Taboo Search [44], CTSS – Continuous Taboo Simplex Search [45], SEA – Simplex and Evolution Algorithm [46]. Acronyms formed for test functions are as follows: EM – Easom, FR – Fichier, SH – Shubert, G&P – Goldstein and Price, ZV – Zakharov (for 2, 5 and 10 Dimensions), RK – Rosenbrock (for 2, 5 and 10 Dimensions), BN RC – Branin RCOS, DJ – De Joung, $S_{4,n}$ – Shekel (for $n = 7, 10$).

The tolerance radius of the preys for functions: DJ , $S_{4,7}$ and $S_{4,10}$ is twice times higher and for functions: ZV and RK for 10D it is five times higher than the others. The result of tolerance area increase is the decrease in precision of extreme determining. Predators' tolerance areas set in the algorithm are the same except functions of the 10D spaces. For these functions, tolerance areas of predators and preys are the same. For functions DJ, $S_{4,7}$ and $S_{4,10}$ the tolerance radius of the preys is twice smaller than the tolerance radius of predators, while in the remaining cases it is four times smaller. For the data presented in tables and figures 2 to 5 the number of predators amounts to 100 and the number of preys – 10, whereas for figures 7 to 9 the number of predators amounts to 20 and the number of preys – 6. This allows to compare behavior of the algorithm with the same settings in different test environments. The consequence of this approach is suboptimal efficiency of the algorithm – despite it, the efficiency of the algorithm is satisfactory, as shown by the results contained in table 2. Operating parameters of the algorithm for these functions were set at the same value and it influenced on the results obtained. But, it allows the identification of interesting properties illustrated by the results collected in table 4.

Table 1 summarizes the number of iterations needed to achieve 100% success rate. Only for SEA algorithm and test function Easom, Zakharov and Rosenbrock succes rate is 97%. The presented algorithm is compared with seven optimization algorithms for fourteen test functions. Such type of analysis is the basis for the majority of articles and it allows the comparison of the algorithm effectiveness, yet its behavior is not shown. The mechanism of behavior seems to be important when choosing an algorithm for the problem. In an indirect way such information is obtained using various test functions. Knowledge of the behavior of the algorithm may be important in solving untypical problems. The success rate determines the ability to solve the problem. The number of iterations of the algorithm allowing to achieve success determines its cost.

The study (table 1) shows that the S-PSO is better than PSO, RCG and CGA, but worse than SHI, CTS and SEA PSO. The proposed algorithm is better realized for the functions: DJ, $S_{4,7}$, $S_{4,10}$, ZV and RK for 5D and 10D.

In order to illustrate the algorithm better, table 1 can be supplemented by data contained in table 2. Figures in table 2 show a large variation – it results from the algorithm operation. The initial distribution will decide of the number of iterations required to achieve the success.

Figure 2 shows the history of global extreme designation for the G&P function. The tolerance radius is lower by 5% in the case of characteristic b. These characteristics show the distinct effect of the tolerance area on the precision of extreme designation. The extreme designation with less precision is significantly faster as well. It is obvious because the time and precision of global extreme designation are conflicting goals. The tolerance area allows to determine which of these goals will be realised.

The analogous graphs are presented for function RK for 2D (figure 3) – they are the typical graphs illustrating the convergence of the algorithm.

The history of changes in the standard deviation of global extreme designation is shown in figure 4. This graph shows that for all tests the error of local extreme designation decreases in time. However, this increase in precision of ex-

Table 1: The comparison of the number of iterations of the algorithm (S-PSO) with other algorithms for the selected test function.

Algorithm Test fun.	S-PSO*	PSO	RCG	CGA	ECTS	CHI	CTS	SEA
EM	653	740	642	1504	1284	952	325	197**
FR	307	580	–	430	–	132	98	258
SH	721	800	946	575	370	345	283	420
G&P	425	480	270	410	231	259	119	–
ZV 2D	613	380	437	620	195	215	78	90**
RK 2D	497	1660	596	960	480	459	369	266**
BN RC	433	740	490	620	254	295	125	272
DJ	358	500	449	750	338	371	155	–
$S_{4,7}$	621	29180	1143	680	910	620	590	–
$S_{4,10}$	768	30160	1235	650	898	635	555	–
ZV 5D	1066	1530	1115	1350	2254	950	–	–
ZV 10D	2764	7440	2190	6991	4630	100	–	–
RK 5D	2423	33100	4150	3990	2142	3290	–	–
RK 10D	6340	106960	8100	21563	15720	14563	–	–

* – *proposed algorithm*, ** – *success rate reaches 97%.*

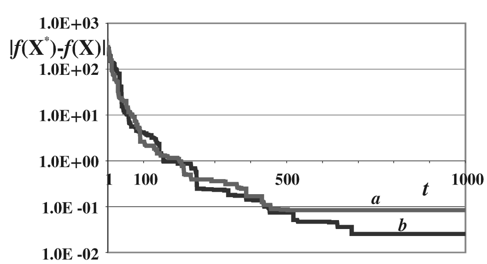

Figure 2: The history of global extreme designation for the G&P function.

Table 2: Statistical data of numbers of iterations

Param.	numbers of iterations of the algorithm				
Test fun.	min.	avg.	max.	med.	std. dev.
EM	260	653	985	698	260
FR	108	307	684	263	195
SH	306	721	984	778	191
G&P	149	425	683	447	207
ZV 2D	254	613	987	582	273
RK 2D	110	497	754	640	263
BN RC	75	433	732	492	218
DJ	115	358	736	243	242
$S_{4,7}$	380	621	853	612	169
$S_{4,10}$	474	768	953	852	191
ZV 5D	746	1066	1437	1017	269
ZV 10D	1588	2764	3806	2805	899
RK 5D	1762	2423	3487	2439	562
RK 10D	4527	6340	7848	6159	1172

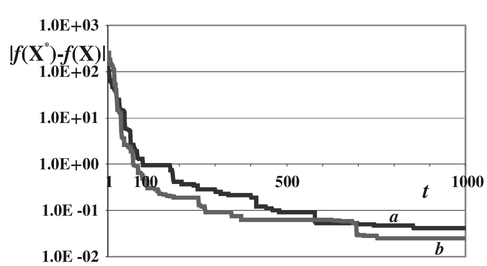

Figure 3: The history of global extreme designation for the RK 2D function.

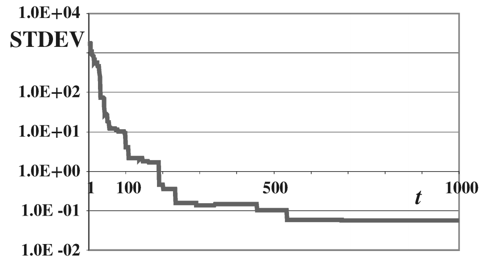

Figure 4: The history of changes in the standard deviation of global extreme designation for FR function.

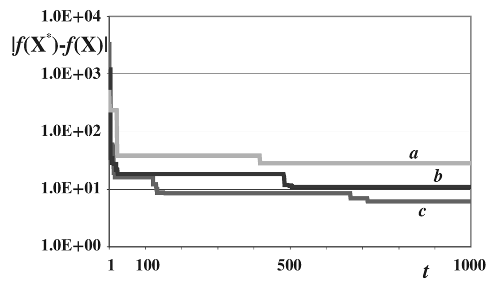

Figure 5: The history of global extreme designation for the ZV 2D function.

treme designation is limited by the existence of the tolerance area. It illustrates the convergence of the algorithm in the global extreme designation.

Figure 5 shows the history of the global extreme designation. The tolerance radius decreases by 5% for the following characteristics. Characteristics "c" has the smallest tolerance area and simultaneously – the smallest error. It is worth noticing that the extreme designation in all cases is very fast.

Because the error value of extreme designation depends on the selection of the value of tolerance area, the implementation of the adaptation mechanism will reduce the error significantly, as it was confirmed by tests carried out (table 3).

Fitness function is always designated for active predators and prey, as well as for newly created particles which replace the eliminated predators and prey. In table 4 average values of prey and predators elimination are given, as well as

Table 3: Statistical data of error of extremes designation

Param.	Error of extremes designation				
Test fun.	min.	avg.	max.	med.	std. dev.
EM	2,0E-05	2,5E-04	5,4E-04	2,1E-04	1,56E-04
FR	7,34E-03	8,29E-02	1,88E-01	8,3E-02	5,56E-02
SH	1,31E-02	6,07E-01	1,84E+00	5,01E-01	6,21E-01
G & P	1,55E-03	2,52E-02	5,17E-02	2,51E-02	1,8E-02
ZV 2D	6,0E-05	2,59E-04	7,7E-04	1,3E-04	2,56E-04
RK 2D	1,8E-04	2,54E-02	5,92E-02	1,7E-02	2,43E-02
BN RC	2,0E-04	3,7E-03	8,85E-03	1,99E-03	3,72E-03
DJ	7,0E-04	2,95E-02	5,64E-02	3,23E-02	1,63E-02
S4,7	5,04E-03	2,44E-02	4,84E-02	2,43E-02	1,65E-02
S4,10	1,88E-03	2,82E-02	6,41E-02	2,43E-02	2,0E-02
ZV 5D	2,16E-01	5,31E-01	9,38E-01	4,71E-01	2,68E-01
ZV 10D	1,0E+00	2,49E+00	3,99E+00	2,51E+00	1,07E+00
RK 5D	6,23E-01	9,95E-01	3,99E+00	9,32E-01	2,6E-01
RK 10D	8,61E+00	1,48E+01	1,95E+01	1,57E+01	3,97E+00

Table 4: Data of particles' activity.

Param. Test fun.	Prey elimination*	Predators elimination*	Active predators*
EM	0,03	6,12	4,76
FR	0,05	3,78	4,70
SH	0,06	10,02	5,31
G&P	0,03	3,06	4,24
ZV 2D	0,05	3,44	4,40
RK 2D	0,03	3,08	4,70
BN RC	0,04	3,51	4,23
DJ	0,02	1,18	3,65
$S_{4,7}$	0,01	1,03	4,26
$S_{4,10}$	0,01	1,09	4,27
ZV 5D	0,02	1,35	3,89
ZV 10D	0,02	1,14	3,73
RK 5D	0,04	3,15	6,00
RK 10D	0,004	0,47	4,21

* – *per one itreration*

the average valuess of active predators referenced to one cycle of the algorithm.

It is easy to notice that the average values of predators and prey elimination and active predators computed for one work cycle of algorithm are dependent on the parameters of the algorithm work and the environment has little effect on them. This feature gives new possibilities in the application of the algorithm.

To evaluate the efficiency of the algorithm better, the data of algorithm operation are given in table 5. Using data from tables 1 and 4, the average cost of work per one algorithm iteration was determined and then the average cost of the designation of extreme was calculated, which is expressed as the average number of the fitness function computing.

The cost of the algorithm work is clearly dependent on the environment and

Table 5: Cost of extremes designation.

Param. / Test fun.	avg.*	min.	avg.	max.
EM	11,91	3095	7772	11727
FR	9,53	1030	2927	6521
SH	16,40	5017	11813	16133
G&P	8,33	1241	3543	5690
ZV 2D	8,89	2257	5443	8771
RK 2D	8,81	969	4378	6643
BN RC	8,77	658	3800	6420
DJ	5,86	674	2099	4312
$S_{4,7}$	6,31	2397	3915	5380
$S_{4,10}$	6,38	3023	4895	6078
ZV 5D	6,26	4672	6675	9000
ZV 10D	5,90	9361	16293	22437
RK 5D	10,19	17963	24706	35549
RK 10D	5,68	25730	36034	44605

* – *per one itreration*

Table 6: Data for comparation of extremes designation error.

Algorithm Test fun.	SPSO	Modified PSO	Uniform Design PSO
SH	6,2059E-01	5,8247E-14	2,3437E-14
BN RC	3,7222E-03	0,0E+00	0,0E+00
RK 10D	3,9692E+00	6,2436E-01	1,1308E+00

it increases with the complexity of the environment. The number of particles equals to 110, but only (on average) less than 8% is involved in the calculation of one cycle and only these particles are subject to adaptation mechanism. The remaining particles fulfill also a very important function – they contain information on the solution space. It is of great importance for the proper operation of the algorithm (see the description of the algorithm).

Discussions on precision of extreme designation can be extended of the exemplary comparison with the results obtained in [4] (table 6). The proposed algorithm compared with algorithms tuned for high precision of extreme designation obtains worse results.

In the article [47] the test environments were divided into four groups with the following test functions: Group A: Unimodal and Simple Multimodal Problems: 1) Sphere function, 2) Rosenbrock's function, Group B: Unrotated Multimodal Problems: 3) Ackley's function, 4) Griewank's function, 5) Weierstras's function, 6) Rastrigin's function 7) Non-continuous Rastrigin's function, 8) Schwefel's function. Group C: Rotated Multimodal Problems: 9) Rotated Ackley's function, 10) Rotated Griewank's function, 11) Rotated Weierstras's function, 12) Rotated Rastrigin's function, 14) Rotated Schwefel's function Group D: Composition Problems: 15) Composition function 1 (CF1) in [48]: (CF1), 16) Composition function 5 (CF5) in [48]: (CF2). By means of the abovementioned test environments the efficiency of the following algorithms group was verified : PSO with inertia weight (PSO-w) [7]; PSO with constriction factor (PSO-cf) [49]; Local version of PSO with inertia weight (PSO-w-local);

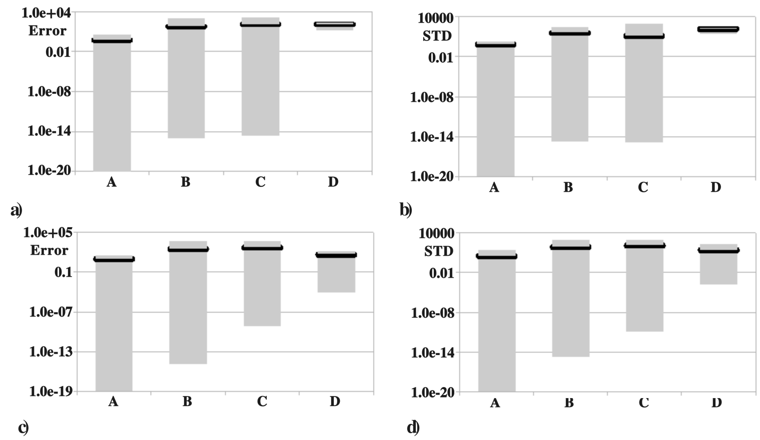

Figure 6: The global extreme designation error by S-PSO and the standard deviation for groups of test functions.

Local version of PSO with constriction factor (PSO-cflocal) [50]; Unified Particle Swarm Optimization (UPSO) [51]; Fully informed particle swarm (FIPS) [15]; Fitness Distance Ratio based on Particle Swarm Optimization (FDR-PSO) [10]; Cooperative PSO (CPSO-H) [39]; Cellular PSO (CLPSO) [47].

For the same main settings, the effectiveness efficiency of the proposed algorithm operation was referred to the range of values obtained in the particular groups of test functions. As it results from the presented graphs the obtained values are located within the ranges presented in the graphs. For a better graphic presentation ranges in the figures are limited – the minimum values for the figure 6a, b reach $9.84E-118 \pm 3.56E-117$, whereas for figure 6c, d – $1.16E-113 \pm 2.92E-113$.

Effective criterion for stopping the algorithm work is difficult to achieve. It is shown among others in the paper [6]. In the proposed algorithm the removal of preys is responsible for the extreme identifying (see the description of the algorithm). These removals are repeated in the same areas of the solution space. These eliminations combined with slight changes in fitness function are certain indicators of the extreme. The obtained values of algorithm cost comply with such criterion (table 7). These results can be compared with the average values

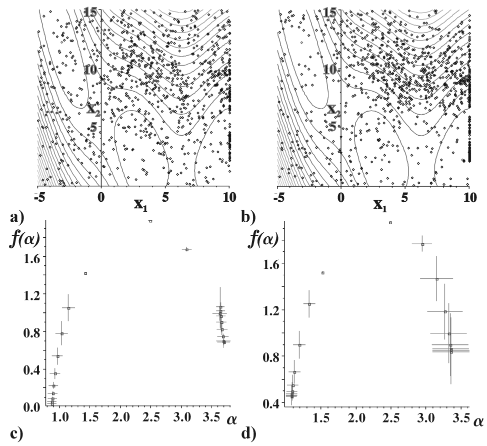

Figure 7: The distribution of predators in the environment (a, b) and the multifractal spectrum of the presented distributions(c, d).

calculated on the base of data for the various "stop criteria" and presented in the study [6]. The numbers in parentheses denote the fraction of runs that located the global extreme. On this basis, it is possible to conclude that the computation cost of the proposed algorithm is higher in the majority of cases – but it gives the certainty of global extreme designation.

The multifractal analysis may be used for evaluating the algorithm work. Obtained multifractal spectrum gives information about the efficiency of searching the solution space by predators as well as the behaviour of the prey.

Figure 7a and b shows the distribution of predators in the environment. They perform a uniform exploration of the solution space in the search of prey. It should be noted that the distribution of predators in a single iteration is not uniform, because the predators encircle the prey. A comparison of figures 7a and b indicate that the distribution of predators in figure 7a is more uniform. It is confirmed by the graph of multfractal analysis where the multifractal spectrum from the figure 7d is relevant to the distribution of predators in figure 7b. A similar analysis can be performed by observing the behavior of prey.

Figure 8 clearly shows prey grouped in local extremes. Clustering of prey is also illustrated by the multifractal spectrum. The graph of multifractal anal-

Table 7: Data of the algorithm work.

Test fun.	Param.	SPSO	LDW PSO	CENTER PSO	SIMPLE PSO	DYNAMIC PSO
	min.	3095	807	813	793	806
EM	avg.	7772	5141	5196	4245	3985
	max.	11727	17999	18205	14478	13446
	min.	5017	2512	2535	1794	1805
			(0,94)	(0,94)	(0,98)	(0,99)
SH	avg.	11813	5778	5890	3528	2940
			(0,96)	(0,96)	(0,99)	
	max.	16133	9958	10224	5838	4510
	min.	658	942	949	856	902
			(0,96)	(0,96)	(0,99)	(0,94)
BN RC	avg.	3800	4002	4043	1766	1658
			(0,97)	(0,97)		(0,98)
	max.	6420	8496	8698	3088	2664
	min.	2397	3516	3318	1613	1885
			(0,51)	(0,52)	(0,48)	(0,45)
S4,7	avg.	3915	8170	8379	2648	2380
			(0,62)	(0,63)	(0,54)	(0,54)
	max.	5380	12845	13337	5283	3703
			(0,82)	(0,81)	(0,72)	(0,73)
	min.	3023	4342	4239	1726	1768
			(0,62)	(0,63)	(0,49)	(0,45)
S4,10	avg.	4895	9107	9319	2729	2313
			(0,71)	(0,71)	(0,56)	(0,54)
	max.	6078	14685	15186	5328	3703
			(0,93)	(0,92)	(0,78)	(0,79)

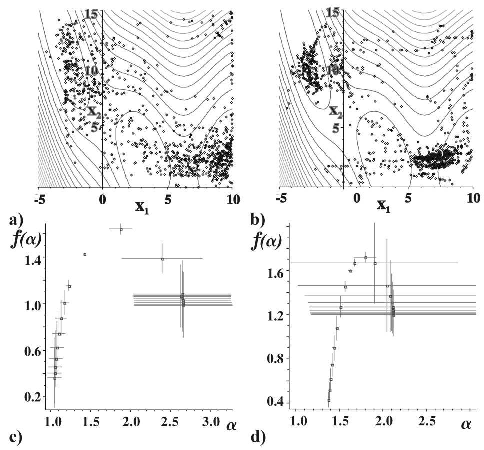

Figure 8: The distribution of prey in the environment (a, b) and multifractal spectrum of the presented distribution (c, d).

ysis 8d and distribution in figure 8b confirm the clustering. What is more, the multifractal analysis may be particularly useful in the research on multi-dimensional problems. Data from this analysis can be useful in the tuning of the algorithm and also for the automatization of algorithm tuning process while working. Since the experiments on the automatic tuning of algorithm work have not been conducted, it is suggested to start further research. Figure 9 illustrates the behavior of encircled prey.

The increase of prey speed movement caused by the increase of maximum value of the weighting coefficients is clearly visible in the following figures: 9a, b, c, d. In figure 9a the prey cannot escape from the encirclement, as evidences the strong exploitation nature of the algorithm (does not mean algorithm stagnation due to the mechanism of elimination). Whereas figures 9b, c, d illustrate the successive encirclements and escapes of prey. Figures 9b and c show a good balance between exploration and exploitation. Figure 9b shows more intensive exploitation, while the figure 9c shows more intensive exploration. Strong exploratory character of algorithm is shown in figure 9d. The behavior of prey in figure 9c is the best. It moves very fast outside the area of extreme and it follows to the next extreme. Prey moves much slower inside extreme performing

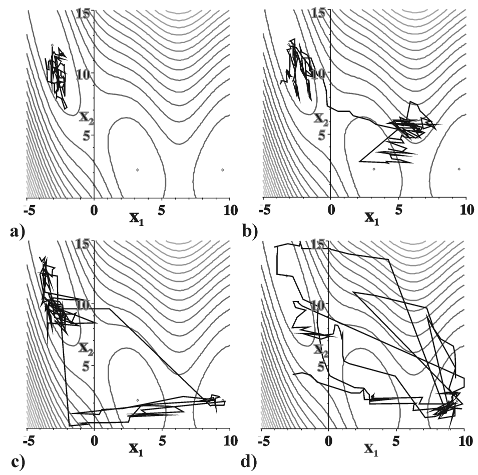

Figure 9: Sample path of encircled prey movement.

its exploitation. This behavior is also an illustration of the process of observation. To obtain information about the image, the human eye moves in a way passing through various points of the image focusing on the details. This suggests that eye movement has an exploration phase and an exploitation phase. Moving prey creates an association with this movement. For figure 9a it can be concluded that gaze is fixed on one point and for figure 9d that gaze is restless (glassy eyed). While on the figures 9b and c, gaze is moving through the image by focusing on some detail. We experience the power of the sense of sight daily. Thanks to these properties, the algorithm should use high efficiency also in the non-stationary environments.

7. Future work

On the base of above-presented discussion it is easy to indicate the potential modifications of algorithm that should improve its efficiency and retain its operation mechanics. The first modification can realised as a local method for a designated cluster by predators. This method can be executed after the elimination of the prey. When such a cluster determines the local extreme, another modification can be proposed – particles are discouraged to move in direction of this area. There are also possible other modifications as the adaptive change of tolerance area or changes in the behavior of scenarios. One should assume that the presented algorithm in the hybrid form meets the high requirements of efficiency. However, the properties and the possibility to apply this algorithm in non-stationary environments are currently the subject of a separate study.

8. Conclusion

The new algorithm is proposed in the article. Its principle of operation is based on the cooperation of two systems of particles. Semi-PSO algorithm describes a situation taken from life – a round-up game strategy. The presented algorithm is not described so far in the world literature and they have features similar to the mechanism of observation. They have high efficiency of random search of solutions space combined with searches resulting from the motion of particles. The first particle system has the exploration function, and the other one – exploitation function. This classification is conventional and results from analysis of the behavior of particle systems. As it has been shown, cooperation of the particles have a strong influence on the behavior of the algorithm – stronger than the impact of the environment. In the S-PSO algorithm behavioral scenarios are applied, that are implemented by functions of the weighting coefficients. Parameters of the algorithm, discussed in the article, give interesting ability to control its operation in comparison with other algorithms. This algorithm has the self-control ability between the process of exploitation and exploration of the solution space. This also applies to the stop criterion, which for this group of algorithms is determined on the basis of their behavior – the extreme is indicated by a prey elimination. The comparison S-PSO algorithm with other algorithms shows its good qualities. The present form of the algorithm has very interesting properties. However, the algorithm in its present form cannot be of practical importance. The reason for this is a rather big error of extreme designation. The future works should be focused on the modification introducing the effective exploitation method activated by the occurrence of prey elimination. Creating of such a hybrid algorithm can significantly accelerate and improve the accuracy of extreme designation. Major weakness of the proposed algorithm for the stationary environments is a repeated designation of the same extreme – it can be also eliminated.

Properties of the new algorithm makes it worthy of interest, the practical application and work on its development. This study can be also as an inspiration to search other solutions that implement co-operation or co-evolution.

References

[1] D. Wolpert, W. Macready, No free lunch theorems for optimization, IEEE Transaction on Evolutionary Computation 1 (1) (1997) 67–82.

[2] I. Gosciniak, Immune algorithm in non-stationary optimization task, in: Proceedings of the 2008 International Conference on Computational Intelligence for Modelling Control & Automation, CIMCA '08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 750–755.

[3] D. Sedighizadeh, E. Masehian, Particle swarm optimization methods, taxonomy and applications, International Journal of Computer Theory and Engineering 1 (5) (2009) 1793–8201.

[4] T. Chen, T. Chi, On the improvements of the particle swarm optimization algorithm, Advances in Engineering Software 41 (2) (2010) 229–239.

[5] H. Gao, W. Xu, Particle swarm algorithm with hybrid mutation strategy, Applied Soft Computing 11 (8) (2011) 5129–5142.

[6] I. Tsoulos, A. Stavrakoudis, Enhancing PSO methods for global optimization, Applied Mathematics and Computation 216 (10) (2010) 2988–3001.

[7] Y. Shi, R. Eberhart, A Modified Particle Swarm Optimizer, in: Proceedings of IEEE International Conference on Evolutionary Computation, IEEE Computer Society, Washington, DC, USA, 1998, pp. 69–73.

[8] H. Fan, Y. Shi, Study on vmax of particle swarm optimization, in: Proc. Workshop Particle Swarm Optimization, Indianapolis, 2001.

[9] Y. Shi, R. Eberhart, Particle swarm optimization with fuzzy adaptive inertia weight, in: Proc. Workshop Particle Swarm Optimization, Indianapolis, 2001, pp. 101–106.

[10] T. Peram, K. Veeramachaneni, C. Mohan, Fitness-distance-ratio based particle swarm optimization, in: Swarm Intelligence Symp., 2003, pp. 174–181.

[11] X. Hu, R. Eberhart, Y. Shi, Engineering optimization with particle swarm, in: IEEE Swarm Intelligence Symposium, SIS 2003, IEEE Neural Networks Society, Indianapolis, 2003, pp. 53–57.

[12] M. Lovbjerg, T. Krink, Extending particle swarm optimizers with self-organized criticality, in: Proc. Congr. Evol. Comput., Honolulu, 2002, pp. 1588–1593.

[13] X. Xie, W. Zhang, Z. Yang, Dissipative particle swarm optimization, in: Proc. Congr. Evol. Comput., 2002, pp. 1456–1461.

[14] X. Hu, R. Eberhart, Multiobjective optimization using dynamic neighborhood particle swarm optimization, in: Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02, CEC '02, IEEE Computer Society, Washington, DC, USA, 2002, pp. 1677–1681.

[15] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: Simpler, maybe better., IEEE Transaction on Evolutionary Computation 8 (3) (2004) 204–210.

[16] T. Blackwell, P. Bentley, Dont push me! collision-avoiding swarms, in: Proc. IEEE Congr. Evol. Comput., Honolulu, 2002, pp. 1691–1696.

[17] V. Miranda, N. Fonseca, New evolutionary particle swarm algorithm (epso) applied to voltage/var control, in: Proc. 14th Power

Syst. Comput. Conf., Seville, Spain, 2002, p. "[Online] Available: http://www.pscc02.org/papers/s21pos.pdf".

[18] P. Angeline, Using selection to improve particle swarm optimization, in: Proc. IEEE Congr. Evol. Comput., Anchorage, 1998, pp. 84–89.

[19] A. Leontitsis, D. Kontogiorgos, J. Pange, Repel the swarm to the optimum, Applied Mathematics and Computation 173 (1) (2006) 265–272.

[20] R. Brits, F. Engelbrecht, A.P. andvan den Bergh, Solving systems of unconstrained equations using particle swarm optimization, in: Proc. 2002 IEEE Conf. Syst., Man, Cybern., 2002, pp. 102–107.

[21] R. Brits, A. Engelbrecht, F. van den Bergh, A niching particle swarm optimizer, in: Proc. 4th Asia-Pacific Conf. Simulated Evolution and Learning, 2002, pp. 692–696.

[22] S. Bird, X. Li, Adaptively choosing niching parameters in a pso, in: Proc. 2006 Genetic Evol. Comput. Conf., 2006, pp. 3–10.

[23] X. Li, Adaptively choosing neighborhood bests using species in a particle swarm optimizer for multimodal function optimization, in: Proc. 2004 Genetic Evol. Comput. Conf., 2004, pp. 105–116.

[24] D. Parrott, X. Li, A particle swarm model for tracking multiple peaks in a dynamic environment using speciation, in: Proc. 2004 Congr. Evol. Comput., 2004, pp. 98–103.

[25] D. Parrott, X. Li, Locating and tracking multiple dynamic optima by a particle swarm model using speciation, in: IEEE Trans. on Evol. Comput., Vol. 10, 2006, pp. 440–458.

[26] S. Bird, X. Li, Using regression to improve local convergence, in: Proc. 2007 IEEE Congr. Evol. Comput., 2007, pp. 592–599.

[27] J. Kennedy, Stereotyping: Improving particle swarm performance with cluster analysis, in: Proc. 2000 Congr. Evol. Comput., 2000, pp. 1507–1512.

[28] A. Passaroand, A. Starita, Particle swarm optimization for multimodal functions: A clustering approach, Journal of Artificial Evolution and Applications 2008 (Article ID 482032) (2008) 15.

[29] R. Lung, D. Dumitrescu, A collaborative model for tracking optima in dynamic environments, in: Proc. 2007 Congr. Evol. Comput., 2007, pp. 564–567.

[30] R. Thomsen, Multimodal optimization using crowding-based differential evolution, in: Proc. 2004 Congr. Evol. Comput., 2004, pp. 1382–1389.

[31] C. Li, S. Yang, A clustering particle swarm optimizer for dynamic optimization, in: Proc. 2009 Congr. Evol. Comput., 2009, pp. 439–446.

[32] S. Yang, C. Li, A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments, in: IEEE Trans. on Evol. Comput., Vol. 14, 2010, pp. 959–974.

[33] Q. He, L. Wang, An Effective Co-evolutionary Particle Swarm Optimization For Constrained Engineering Design Problems, Engineering Applications of Artificial Intelligence 20 (1) (2007) 89–99.

[34] J. Arquilla, D. Ronfeldt, Swarming and the Future of Conflict, RAND National Defense Research Institute, Santa Monica, CA, U. S., 2000.

[35] W. Jang, H. Kang, B. Lee, K. Kim, D. Shin, S. Kim, Optimized fuzzy clustering by predator prey particle swarm optimization, in: IEEE Congress on Evolutionary Computation, CEC2007, 2007, pp. 3232–3238.

[36] M. Higashitani, A. Ishigame, K. Yasuda, Pursuit-escape particle swarm optimization, IEEJ Transactions on Electrical and Electronic Engineering 3 (1) (2008) 136–142.

[37] K. Trojanowski, S. Wierzchoń, Immune-based algorithms for dynamic optimization, Information Sciences 179 (10) (2009) 1495–1515.

[38] M. Higashitani, A. Ishigame, K. Yasuda, Particle swarm optimization considering the concept of predator-prey behavior, in: 2006 IEEE Congress on Evolutionary Computation, 2006, pp. 434–437.

[39] F. van den Bergh, A. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Transactions on Evolutionary Computation 8 (2004) 225–239.

[40] H. Kuo, J. Chang, C. Liu, Particle swarm optimization for global optimization problems, Journal of Marine Science and Technology 14 (3) (2006) 170–181.

[41] M. Bessaou, P. Siarry, A genetic algorithm with real-value coding to optimize multimodal continuous functions, Structural and Multidiscipline Optimization 23 (2001) 63–74.

[42] R. Chelouah, P. Siarry, A continuous genetic algorithm designed for the global optimization of multimodal functions, Journal of Heuristics 6 (2) (2000) 191–213.

[43] R. Chelouah, P. Siarry, Genetic and nelder-mead algorithms hybridized for a more accurate global optimization of continuous multiminima function, European Journal of Operational Research 148 (2) (2003) 335–348.

[44] R. Chelouah, P. Siarry, Tabu search applied to global optimization, European Journal of Operational Research 123 (2000) 256–270.

[45] R. Chelouah, P. Siarry, A hybrid method combining continuous taboo search and nelder-mead simplex algorithms for the global optimization of multiminima functions, European Journal of Operational Research 161 (2005) 636–654.

[46] H. Kuo, J. Chang, K. Shyu, A hybrid algorithm of evolution and simplex methods applied to global optimization, Journal of Marine Science and Technology 12 (4) (2004) 280–289.

[47] J. Liang, A. Qin, P. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Transactions on Evolutionary Computation 10 (3) (2006) 281–295.

[48] J. Liang, P. Suganthan, K. Deb, Novel composition test functions for numerical global optimization, in: Proc. Swarm Intell. Symp., 2005, pp. "[Online], Available: http://www.ntu.edu.sg/home/EP–NSugan".

[49] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6 (1) (2002) 58–73.

[50] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: IEEE Congr. Evol. Comput., 2002, pp. 1671–1676.

[51] K. Parsopoulos, M. Vrahatis, Upsoa unified particle swarm optimization scheme, Lect. Ser. on Computational Sciences (2004) 868–873.