Silesian J. Pure Appl. Math.
vol. 7, is. 1 (2017), 83–92

Marek BALCER

Institute of Mathematics, Silesian University of Technology, Gliwice, Poland

MATHEMATICAL BACKGROUND FOR PARTICLE SWARM AND HEURISTIC ALGORITHMS

Abstract. The article seeks to clarify some concepts and principles that are used in constructing algorithms that utilize particle swarm as a tool for searching extremes of target functions, including heuristic algorithms. The author also draws attention to some philosophical aspects of creating metaphors by ordering basic ways of constructing the transition vectors.

1. Introduction

This paper aims at describing the principles of creating optimization algorithms using the notion of particle swarm, as well as presenting an integrated mathematical feasible description. It is known that most of them were created by the authors' inspiration with the observations of nature and social behaviour of the subjects [1,7,10]. The methods used for their creation are called methaheuristic [9]. Despite the fact that they originate from various observations and concern behaviour of various communities, there are some common features that might be described in the form of a common mathematical model. This model shows the behaviour of a particle swarm in time and defines the basic rules of construction of the following set based on the available information accumulated throughout

2010 Mathematics Subject Classification: 90C53, 68W40.
Keywords: PSO, heuristics, algorithms, optimization.
Corresponding author: M. Balcer (Marek.Balcer@polsl.pl).
Received: 10.07.2017.

the history of the whole process. At each stage of iteration, the optimized value that is described as the predetermined function of the target is observed. It is the main value that is approximated to a given function in a particular searching area. In this case, the seeking value is at each stage of a particular particle that implements the value. It is also interesting to investigate the behaviour of a whole set of particles in the history of its successive generations, exempli gratia whether it is spread uniformly in the D area or if it begins to accumulate over time around certain selected points in the research area (absorbency, subset submission, see also [6]). The following section of this article attempts to provide answers to the questions presented above.

The basic problem

Let $R^n, (n \in N)$, be an Euclidean n-space and F-objective function, that

$$F : D \to R, \text{ where } D \subset R^n.$$

How to find the maximum (optionally minimum) value of function **F** in a domain D and point $P^* \in D$ (point of realization) such, that for any other point $P \in D$ $F(P) \leqslant F(P^*)$ (optionally $F(P) \geqslant F(P^*)$).

PSO algorithms and Heuristic algorithms use particles for describing computational process [4].

2. What is a particle?

The particle P is any object recognized by determining its position within the contractual time specified as a discrete variable of a natural value $0, 1, 2, \ldots, t, \ldots$ indicating the number of the next iteration.

Sequence

$$P = (P^0, P^1, \ldots, P^t, \ldots),$$

where $P^0, P^1, \ldots, P^t, \ldots$ are from the domain D, determines the particle P by the indication of its position in subsequent iterations.

A vector of transition – changing the position of the particle P from t to $t+1$ $\overrightarrow{P^t P^{t+1}}$ denoted as v^{t+1} and that

$$P^{t+1} = P^t + v^{t+1}.$$

Above equation is the main equation of particle movement.

Basic rules for constructing the transition vector.

1. **Random walk**
$$v^{t+1} = \alpha_1 randomR^n,$$

2. **Leaving the previous direction of change**
$$v^{t+1} = \alpha_2 v^t;$$

3. **The transition towards another point generated in the iteration process**
$$v^{t+1} = \alpha_3(P^* - P^t),$$
where $P^* \in D$ is any generated point, $\alpha_1, \alpha_2, \alpha_3$ – controlling coefficients.

In practice, we can use the combination of all

$$v^{t+1} = \beta_1 randomR^n + \beta_2 v^t + \beta_3(P^* - P^t),$$

where $\beta_1, \beta_2, \beta_3$ – the parameters depending of the used method.

3. Particle swarm

Let M be natural. The sequence

$$(P_1, P_2, \ldots, P_M)$$

is M-particle swarm of particles from the domain D.

For i-particle ($i = 1, 2, \ldots, M$):

$$P_i = (P_i^0, P_i^1, \ldots, P_i^t, \ldots)$$

movement equations are

$$P_i^{t+1} = P_i^t + v_i^{t+1}.$$

If algorithm design is based on the particle swarm and the transition vectors then it is **the particle swarm algorithm** [8]. Also **Heuristic algorithms** are based on particle swarm, but heuristic inspirations gives some new rules for constructing the movement of swarms [11].

4. The history

In order to understand further points of discussion, it is necessary to introduce some concepts related to the history of the process that are subjected to the swarm of particles in the course of the algorithm.

Let $P_1, P_2, \ldots, P_M$ is M-particle swarm, such that for

$$i \in N(1 \leqslant i \leqslant M) \text{ and } t \in NP_i = (P_i^0, P_i^1, \ldots, P_i^t, \ldots).$$

Definition 1. *History of particle i to t iteration is a set*

$$H_i^t = \{P_i^0, P_i^1, \ldots, P_i^t\}.$$

Definition 2. *History of t-iteration is a set*

$$H^t = \{P_1^t, P_2^t, \ldots, P_M^t\}.$$

Definition 3. *Global history of swarm to t iteration is a set*

$$H_{glob}^t = \bigcup_{i=1\ldots M} H_i^t.$$

5. Process of creation heuristic algorithm

The most of heuristic algorithms work on partial or some assumptions of the basics of the Particle Swarm Algorithms of Optimization. A randomly spaced set of particles is used in the algorithms to search for possible solutions for the function. Particles that are randomly distributed at the beginning of the algorithm constitute the initial population. During subsequent iterations (epochs) performed according to these rules, subjects belonging to this population will undergo evolution. The algorithm allows you to modify the characteristics of the initial population to select these individuals that best match the given criterion. The goal is to seek the optimum of the function to be tested. Therefore, individuals of the initial population will evolve (move) in successive epochs to fit the given optimization criterion best. What do we observe in individuals? Biological behaviour will not be particularly described in this paper. See [1, 11] for further theoretical description of this issue. First of all, as for the algorithms in the swarm, the process of creating a history acts as a focal point of discussion and in consequence the focus is put on the following behaviour

– way of moving particles, i.e. construction of transition vectors;

– communication between individuals;

– principles of objective function optimization.

The idealized natural behaviour constitutes the inspirations for the construction of subsequent histories (stages) of the swarms' movement in time. These inspirations are described in a few simple rules assigned to the behaviour of particular swarms. They are also called metaphors. It seems appropriate to use concept of meta inspiration since it accepts a very simplified scheme of natural behaviour which can additionally be interpreted differently in the form of so-called pseudo code. Meta inspiration is the term incorporating the basis for the computational intelligence of algorithms provided in the form of a pseudo-code. However, it should be noted, that in constructing we can deal with ambiguity resulting from different interpretations of natural behaviour.

I Metaphore (meta-inspiration)
Some observed natural rules (behavior)

II Algorithm in pseudocode
Construction particle swarm algorithm based on meta-inspiration

Generally we observe two main groups of construction movements of particle swarms.

A. A classic transition with conditions

$$P_i^{t+1} = P_i^t + \begin{cases} Q_1, & \text{if } W_1, \\ & \vdots \\ Q_l, & \text{if } W_l, \end{cases}$$

where t is a number of iteration, Q_i some transition vectors and W_i conditions (relations) based on history of the process.

Example 4.

$$P_i^{t+1} = \begin{cases} \alpha \quad random(R^n), & \text{if } F(P_i^t + v_i^{t+1}) < F(P_i^t), \\ & \vdots \\ P_i^t + v_i^{t+1}, & \text{if } F(P_i^t + v_i^{t+1}) \geqslant F(P_i^t). \end{cases}$$

When a new position is "better" in terms of the criterion function from the previous one – go to it, otherwise select a random new position

Example 5.

$$P_i^{t+1} = \begin{cases} P_i^t, & \text{if } F(P_i^t + v_i^{t+1}) < F(P_i^t), \\ & \vdots \\ P_i^t + v_i^{t+1}, & \text{if } F(P_i^t + v_i^{t+1}) \geqslant F(P_i^t). \end{cases}$$

When a new position is "better" in terms of the criterion function from the previous one – go to it, otherwise leave the present position.

B. Global conditions

The operations performed throughout the swarm during the transition from one epoch to the next.

Example 6. Condition (p, q).

Replace $p\%$ of the worse solutions to the other (for example: new random particles), leave the best $q\%(p + q = 100\%)$.

6. General definition of the particle swarm algorithm in the set theory

Symbolically algorithm in the area D for one of its implementation can be described as a sequence of histories (R-realization):

$$(H^0 \to H^1 \to H^2 \to \ldots \to H^t \to H^{t+1} \to H^s)_R, \text{ where}$$
$$H^t \subset D \text{ for } t = 0, 1, \ldots, s$$
$$\text{or}$$
$$(0, H_R^0) \to (1, H_R^1) \to (2, H_R^2) \to \ldots \to (t, H_R^t) \to (t+1, H_R^{t+1}) \to \ldots \to (s, H_R^s).$$

A simple transition that can be described

$$(H^t \to H^{t+1})_R$$

or

$$((t, H^t) \to (t+1, H^{t+1}))_R.$$

We can treat that the concrete realization R as the relation between successive pairs (time, history):

$$((t, H^t), (t+1, H^{t+1})) \in R.$$

So we can interpret them as elements of the set

$$(T \times 2^D) \times (T \times 2^D),$$

where T is the set of values of the time variable, 2^D is the set of all subsets of the set D. As the swarm algorithms are mostly not deterministic algorithms due to the use of random variables, hence the two different realizations usually get different results. Eg. for the transition $H^t \to H^{t+1}$ we get

$$((t, H), (t+1, H^1))_{R_1} \text{ in realization } R_1,$$
$$((t, H), (t+1, H^2))_{R_2} \text{ in realization } R_2,$$

such, that

$$H^1 \neq H^2.$$

Definition 7. *Relation A_F is a swarm algorithm in the area D if*

1. $A_F \subset (T \times 2^D) \times (T \times 2^D)$,
   and

2. *for any $t \in T$ and for any $H \subset D$,*
   if $t+1 \in T$ then is a set $H' \subset D$, such that
   $((t, H), (t+1, H')) \in A_F$.

Hint.

1. In the case where exists exactly one set H' the algorithm is a deterministic algorithm.

2. Any realization R is a subset of the relation A_F.

7. When the swarm algorithm is optimization algorithm?

The basic task of a good algorithm is to achieve or at least approximate the desired value of the tested function. Some works on this issue are discussed in [2, 5, 6, 12]. Therefore, it can be specified in the following way: point of the extremum of objective function must be reachable in algorithm process and this process must be convergence.

I Condition

Convergence
Observed value $F(P^*(H^t_{glob}))$

Definition 8. *Let H be any no empty subset of the set D. $P^*(H)$ is the best position in H for function F where $P^*(H) \in H$ and for any $P \in H F(P) \leqslant F(P^*(H))$.*

Our task is to find $P^*(D)$ searching a given particle swarm of the D area. The observed value, which is to bring us closer to the desired solution should be a sequence $P^*(H^t_{glob})$ for $t = 0, 1, 2, \ldots$.

Main problems

1. $F(P^*(H^t_{glob}))$ is convergence to $F(P^*(D))$?
2. $P^*(H^t_{glob})$ is convergence to $P^*(D)$?

It is easy to notice that

$$H^t_{glob} = H^0 \cup H^1 \cup H^2 \cup \ldots \cup H^t,$$

and so

$$F(P^*(H^t_{glob})) \leqslant F(P^*(H^{t+1}_{glob})),$$

because $H^t_{glob} \subset H^{t+1}_{glob}$.

Sequence $F(P^*(H^t_{glob}))$ for $t = 0, 1, 2, \ldots$ is monotonic (non-decreasing), and bounded by $F(P^*(D))$, so it is a convergent sequence. Unfortunately, we do not know whether it converges to the expected solution.

The Target for algorithm design is convergence to $F(P^*(D))$

$$\lim_{t\to\infty} F(P^*(H^t_{glob})) = F(P^*(D))$$

The next target for algorithm design is convergence

$$\lim_{t\to\infty} F(P^*(H^t)) = F(P^*(D))$$

It is possible only if

$$P^*(H^t_{glob}) \in H^t \text{ for } t = 1, 2, \ldots$$

II Condition

Reachability
Observed position in domain D $P^*(D)$

Definition 9. *Point $P \in D$ is reachable in the algorithm A_F if and only if for any set $H^0 \subset D$ there are $k \in T$ and a sequence of subsets $H^1, H^2, \ldots, H^k$ such that*

1. *$((t, H^t), (t+1, H^{t+1})) \in A_F$ for any $t : (0 \leqslant t \leqslant k-1)$,*
2. *$P \in H^k$.*

There exist the realization R of algorithm A_F that

$$(H^0 \to H^1 \to H^2 \to \ldots \to H^t \to H^{t+1} \to \ldots \to H^k)_R$$

such that $P \in H^k$.

The Reachability is a necessary condition for optimization
The point $P^*(D)$ must be reachable by algorithm

I hope that presented mathematical background of particles swarm algorithms can be useful to

– test existing algorithms,

– search for new solutions,

– not only experimental but also theoretical comparison of the quality of the constructed algorithms.

References

1. Baonabeau E., Dorigo M., Theraulaz G.: *Swarm Intelligence: From Natural to Artificials Systems*. Oxford University Press, Oxford 1999.
2. Cleghorn C.W., Engelbrecht A.P.: *Particle swarm convergence: standardized analysis and topological influence*. In: Dorigo et al. (eds.): ANTS 2014. LNCS 8667, Springer, Cham 2014, 134–145.
3. Deb K.: *Optimization for Engineering Design: Algorithms and Examples*. Prentice-Hall, New Delhi 1995.
4. Engelbrecht A.P.: *Fundamentals of Computational Swarm Intelligence*. Wiley, Hoboken 2005.
5. Fowler A.C.: *Mathematical Models in the Applied Science*. Cambridge University Press, Cambridge 1997.
6. Gazi V., Passino K.M.: *Swarm Stability and Optimization*. Springer, Berlin 2011.
7. Judea P.: *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, Boston 1984.
8. Kennedy J., Eberhart R., Shi Y.: *Swarm Intelligence*. Academic Press, San Francisco 2001.
9. Talbi E.G.: *Metaheuristics: From Design to Implementation*. Wiley, Hoboken 2005.
10. Yang X.S.: *Biology-derived algorithms in engineering optimization*. In: Olariu S., Zomaya A. (eds.): Handbook of Bioinspired Algorithms. Chapman Hall/CRC, Boca Raton 2005, 589–600.
11. Yang X.S.: *Nature Inspired Metaheuristic Algorithms*. Luniver Press, London 2010.
12. Yudong Z., Shuihua W., Genlin J.: *A Comprehensive survey on particle swarm optimization algorithm and its applications*. Math. Problems Eng. **2015** (2015), article ID 931256.