Journal of Automation, Electronics and Electrical Engineering

PARTICLE SWARM METHODS FOR TRANSPORT AND LOGISTICS OPTIMIZATION PROCESSES

Józef LISOWSKI

Gdynia Maritime University, Faculty of Electrical Engineering, Poland, j.lisowski@we.umg.edu.pl

DOI: https://doi.org/10.24136/jaeee.2023.002

Abstract – This article describes heuristic approaches to static optimization methods, taking into account the nature of the motion of a swarm of particles representing, for example, birds, ants, bees, fireflies, bats, krill, cuckoos, cuttlefish, cockroaches, or the pollination process of flowers. In the given descriptions of the methods, the features of swarm intelligence are detailed, the optimization quality indicators are formulated, and flow charts of the computational algorithms are provided.

Key words – transport, logistics, optimization, computer science

INTRODUCTION

Most transport and logistics processes have various possibilities for their practical implementation, from which the optimal solution can be selected as that which satisfies one or many criteria at the same time, such as minimum transport and storage costs, minimum time, minimum risk of failure or collision, and/or maximum reliability.

In addition to deterministic optimization methods, an important role in the design of transport and logistics systems, both land and sea, can be played by heuristic static optimization methods, which are a group of algorithms for searching for solutions to computational problems in a pseudo-random manner (Fig. 1).

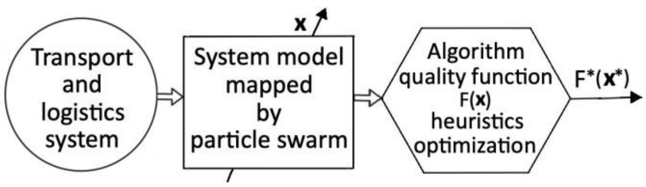

Fig. 1. Functional diagram of transport and logistics system optimization with its mapping by particle swarm

An example may be mapping the movement of ships during steering in collision situations in the form of a swarm of ants with domains assigned to them, the size of which is proportional to the safe passing distance.

Heuristics (Greek. Heurisko; "finding") denotes a set of specific rules that can help to determine the best solution to a problem. Metaheuristics, as intelligent heuristics, are based on analogies to real-world physical and biological processes which can be interpreted in terms of optimization. The most commonly used heuristic methods for static optimization are iterative optimization clustering, Monte Carlo randomization, search, harmony, simulated annealing, evolutionary, and particle swarm algorithms.[1]

1 CHARACTERISTICS OF A PARTICLE SWARM

A swarm of particles can be defined in terms of a group of individuals of a specific natural species, living in a certain place and connected by a common organization of their lives. In order to facilitate the processes of reproduction and obtaining food, individuals inevitably combine into groups.

In [5], Kennedy and Eberhart presented a general form of the Particle Swarm Optimization (PSO) algorithm, which mathematically reflects the process of imitating the behavior of an entire population of living beings, such as a group of birds, ants, bees, fireflies, bats, cockroaches, or krill (Fig. 2).

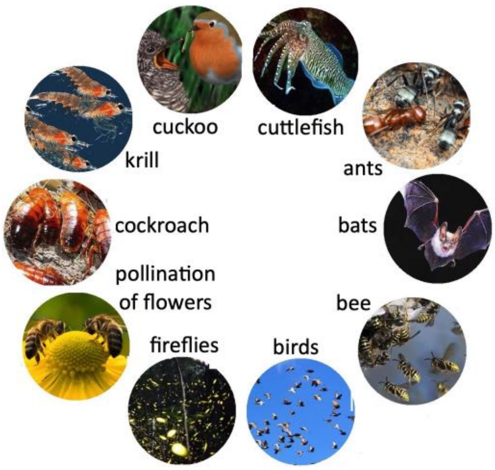

Fig. 2. Examples of individuals used in particle swarm optimization

The population does not have a central management mechanism; instead, it is characterized by the intelligence to act in a changing environment and to react collectively. In the mathematical description of an assembly of individuals, the swarm is denoted as the population and the particle as an individual.

The process of searching for an optimal solution (i.e., the adaptation of the swarm to the environmental conditions) takes place through the phased movement of the particles to the next position.

[1] **Article financed from Research Project Ordered of UMG No. PBZ_WE/2023/PZ/02. „Control algorithms synthesis of autonomous objects".**

The obtained best fit to the environment, representing the optimization quality index, is stored as the optimal point and passed on to the entire population. On the basis of such information, individual particles can change their position, aiming at the point with the best properties.

The intelligence of a swarm of particles can be represented by the following properties:

- proximity—the ability to calculate time and distance,
- quality—the ability to assess the impact of the environment,
- differentiation of reactions—the ability to create different reactions,
- stability—the ability to create behavior with a moderate reaction to changes in the environment,
- adaptation—the ability to change behavior, if forced by the environment.
- In this context, the following notation are introduced:
- the swarm population is $S = \{x_1, x_2, ..., x_M\}$,
- each particle x_i belongs to the N-dimensional search space D,
- the position of the particle is $x_i = \{x_{i1}, x_{i2}, ..., x_{iN}\}$,
- the particle's displacement is $v_i = \{v_{i1}, v_{i2}, ..., v_{iN}\}$,
- the swarm memory is P, comprising the position of each particle with the best value with respect to the objective function F: $P = \{p_{i1}, p_{i2}, ..., p_{iN}\}$,
- the index g, denoting the index of the particle with the best value for the objective function at a given moment.

The movement of a swarm of particles in the k^{th} iteration can be described by the following relationships:

$$\begin{aligned} v_{ij}(k) = v_{ij}(k-1) + c_1 R_{1j}\left[p_{ij}(k-1) - x_{ij}(k-1)\right] + \\ + c_2 R_{2j}\left[p_{gj}(k-1) - x_{ij}(k-1)\right] \end{aligned} \tag{1}$$

$$x_{ij}(k) = x_{ij}(k-1) + v_{ij}(k) \tag{2}$$

Where: R_{1j} and R_{2j} and random numbers in the range [0,1], calculated for each component separately; and c_1 and c_2 are coefficients describing the sensitivity to changes in the optimization objective function F(k).

The operating principle of the particle swarm algorithm is depicted in Figure 3.

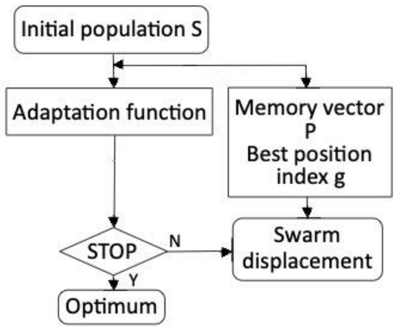

Fig. 3. Flow chart for the particle swarm algorithm

2 PARTICLE SWARM METHODS

2.1 BIRD ALGORITHM

The bird PSO (Particle Swarm Optimization) algorithm imitates the flocking behavior of birds; which, by communicating with and observing each other, can exchange information, thus improving the identification of optimal points in the field [5]. This PSO algorithm follows Equations (1) and (2).

2.2 ANT ALGORITHM

Bonabeau et al. in [2] introduced the Ant Colony Optimization (ACO) algorithm, which is a probabilistic problem solving technique by looking for good paths in graphs, inspired by the behavior of ants searching for food for their colony. An ant colony is a decentralized problem solving system composed of many relatively simple interacting units characterized by:
- self-organization,
- flexibility,
- robustness.

The flexibility of the colony allows it to adapt to changing environments, and resilience means the functioning of the colony even when some individuals fail to fulfill their tasks. Ant algorithms draw inspiration from observing the behavior of ant colonies. It has been discovered that ants communicate with each other and with the environment through the chemicals they secrete. These substances are called pheromones. Ants use the pheromone trail they leave on the substrate to transmit information to other ants.

The first problem to which ant algorithms were used was the Traveling Salesman Problem (TSP), which is a problem of path optimization. Ant algorithms use a positive feedback mechanism based on an analogy to the behavior of leaving a trail on the substrate and following it observed in ant colonies. For this purpose, the so-called virtual pheromone trail. With this mechanism, good solutions are kept in memory so that they can be used to obtain even better solutions.

However, one must be careful not to reinforce good, but not very good solutions, so as not to lead to premature convergence of the algorithm to a local minimum, i.e. to the so-called stagnation. To prevent this, a kind of negative feedback is used, which consists in the evaporation of the pheromone trace (pheromone evaporation).

The use of the ACO ant algorithm to determine a safe ship trajectory in a collision situation was developed by Lazarowska [6]. The calculation of the safe trajectory of the own vessel using the ant algorithm consists of the following three stages:
- data initialization,
- constructing solutions,
- updating pheromone traces (Fig. 4).

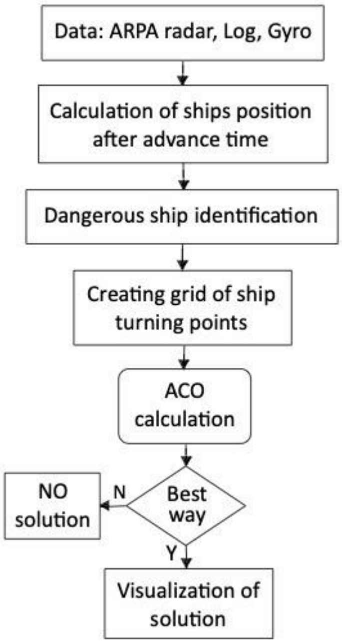

Fig. 4. Flowchart detailing the use of the ant colony algorithm to determine a safe trajectory for a ship

The entire possible course of ship's route, from point wp_0 to the target point wp_e, consists of k stages. While sailing, the ship must not infringe upon the hexagonal domains of other objects it passes (Fig. 5).

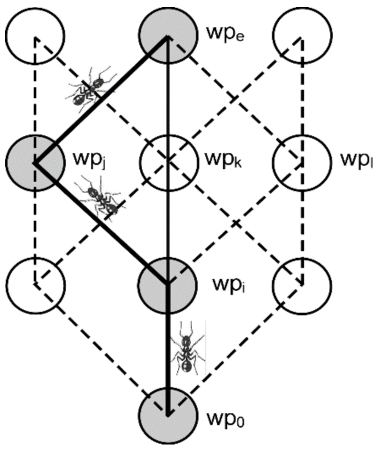

Fig. 5. Sample route chosen by an ant [6]

An ant moves to the next vertex with probability:

$$p_{wp_{ij}}(t) = \frac{\left[\tau_{wp_j}(t)\right]^{\alpha}\left(\eta_{wp_{ij}}\right)^{\beta}}{\sum_{l \in wp_i}\left[\tau_{wp_l}(t)\right]^{\alpha}\left(\eta_{wp_{il}}\right)^{\beta}} \tag{3}$$

where: $\tau_{wpj}(t)$ is the size of the pheromone trace at apex j; and η_{wpij} is the visibility, which is the inverse of the distance between vertices i and j.

If the coefficient $\alpha = 0$, then the choice of the nearest vertex is most likely, while if $\beta = 0$, then only pheromone reinforcement works, both of which are expected to lead to an unsatisfactory solution. Once the route is mapped, the pheromone trail is updated by:

- pheromone evaporation by a constant value ρ, and
- pheromone deposition, in the form of increased pheromone trace value.

2.3 BEE ALGORITHM

The Base Bees Algorithm (BBA) population search approach, developed by Pham [8], mimics the swarming and foraging behavior of honeybees.

All species of bees live in societies, establishing large colonies which, depending on the species, number from a few to 20,000–80,000 individuals. The life of families is coordinated through pheromones, including the division of social roles.

The BBA algorithm performs local searches in conjunction with random mechanisms. A honeybee colony can move long distances (over 14 kilometers) in multiple directions at once, use multiple food sources, and thrive by accumulating food. Due to the lower effort required, bees collect nectar and pollen from rich flowering grounds. Foraging for food begins with bees called colony scouts being dispatched to promising flower areas. Scout bees move randomly. In order to forage, a group of bees pursues exploration, retaining a certain percentage of individuals for scouting. After returning to the hive, the scouting bees are scored. After exceeding a certain composition of sugar, nectar, and pollen, the bees then perform a waving dance. Such a dance—as their speech—is necessary for their communication in the colony and relays the following three pieces of information: Direction of movement, distance from hive, and assessment of quality of food source. In this way, bees can be sent to the correct place for foraging. After completing their dance, the Scout Bee returns to the flower grounds with recruits from the hive. While collecting food, bees assess its level to decide the next dance after returning to the hive.

The algorithm BBA requires the following data:

- the number of scouts n,
- the number of places selected from m visited,
- the number of best areas e per m of places visited,
- the number nep of recruits for the top e areas,
- the number NSP of recruits for other selected areas,
- the initial size of ngh and neighborhood areas.

The following modifications have been made to the basic BBA algorithm:

- the ABC (Artificial Bee Colony) algorithm, developed by Karabog in 2005, in which the intelligent food-gathering behavior of a swarm of bees is simulated, where the bee colony consists of two groups: Worker bees and unemployed bees (as observers and scouts). The amount of nectar determines the quality of the solution. The number of food sources corresponds to the number of bees. The basis for a bee's choice of food source is a probability p.
- the MHBO (Marriage in Honey-Bees Optimization) algorithm, proposed by Abbas in 2001, which is based on the marriage of a queen and drones. Queens represent potential optimal solutions. Mothers have mating flights with drones, resulting in offspring, which are then improved by workers using a specific metaheuristic. The worst mothers are replaced by the best offspring.

2.4 Firefly algorithm

In 2008, Yang developed the Firefly Algorithm (FA), referring to the movement of fireflies through the use of bioluminescence [9].

The constrained optimization algorithm minimizes the cost function F(x) by calculating a value of x* for which the following relationship holds:

$$F\left(x^{*}\right)=\min_{x\in S} f\left(x\right) \tag{4}$$

The algorithm represents three principles:

- fireflies are genderless and, thus, are equally attractive to each other,
- a firefly's attractiveness is proportional to its brightness, decreasing with increasing distance between the fireflies; if the fireflies are equally attractive, then they move randomly,
- the brightness of a firefly corresponds to the value of the quality function being optimized.

This approach uses a representation of a swarm of m members (fireflies) while iteratively implementing an optimization process, wherein x_i is the solution of firefly i in iteration k and $f(x_i)$ is its cost. A firefly has characteristic β, defining its pull on other members of the swarm.

The attractiveness of firefly i decreases with its distance
$r_j = d(x_i, x_j)$ to firefly j:

$$b = b_0 e^{-g r_j^2} \tag{5}$$

where: β_0 is the maximum attractiveness, and γ is the absorption coefficient.

A swarm member has light intensity I_i, which is the inverse of the cost function $f(x_i)$. First, fireflies are placed in S, either randomly or as the result of a deterministic strategy. Firefly i iteratively moves, taking into account the attractiveness of swarm members with greater brightness $I_j > I_i$, $\forall j = 1, \ldots, m$,

$j \neq i$, decreasing with distance and with a random step u_i.

The overall operation of the algorithm can be described in three steps:

1) determining the parameters of the algorithm ($\nu, \beta_0, \gamma, \alpha$) and the number of iterations), determining the objective function f(x), generating the population of fireflies, and the light intensity of the ith firefly, corresponding to the value of the objective function f(xi);

2) until the algorithm stops:
   - if ($I_j > I_i$), then move firefly i towards firefly j,
   - select the attractiveness,
   - find a new solution and update the light intensity.
3) upon occurrence of the algorithm stop condition, identification of the firefly with the highest intensity, followed by visualization of the optimization results (Fig. 6).

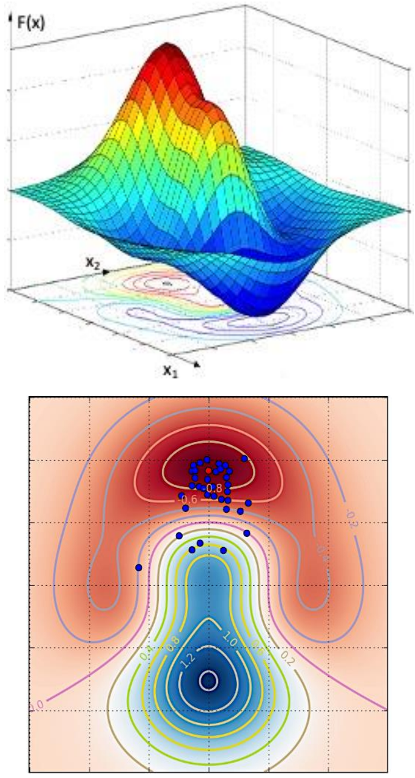

Fig. 6. Graph of the objective function F(x1,x2) (top) and visualization of the search for optimal value of quality function $\mathbf{F}^*(\mathbf{x}_1^*, \mathbf{x}_2^*)$ using the firefly algorithm (bottom)

2.5 CUCKOO ALGORITHM

In 2009, Yang and Deb presented the Cuckoo Search (CS) algorithm, which mimics the behavior of cuckoos using the nests of other birds to incubate their eggs and raise their chicks.

The principles of the CS algorithm are as follows:

- • three simplified rules:
  - A cuckoo lays one egg in a random nest,
  - Nests with high-quality eggs will be passed on to the next generation,
  - the number of nests is constant, and a tossed egg is detected with probability p_a [0, 1];
- the host bird may throw the egg out of the nest or abandon it and build new one;
- estimation of the number of solutions replaced with new random solutions with probability p_a;

- the quality or usefulness of a solution is proportional to the value of the quality function F;
- an egg in a nest represents a solution;
- a cuckoo egg is a new solution;
- a new and better solution (cuckoo's egg) will replace a weak solution in the nest;
- the algorithm can be extended to a version in which many eggs represent a set of solutions, or a simple approach, in which each nest has one egg.

The movement of many animals and insects can be described using Lévy flights—random walks in which the stride length is determined based on a long-tail distribution. Notably, Lévy here refers to the French mathematician Paul Pierre Lévy [7].

A detailed description of the CS algorithm is as follows:

- a new solution is created as $x(t+1) = x(t) + \alpha\, u$;
- where $\alpha > 0$ is the step size, related to size of the parameter domain, and
- u is a number drawn from a Lévy distribution with infinite variance and mean value $u \sim t^{-\lambda}$, $1 < \lambda \leq 3$;
- New solutions are generated on basis of Lévy flight around the best solution, thus speeding up local searches;
- significant part of new solutions should be generated in regions distant from the current best solution, which ensures that system will not become trapped in a local optimum.

Figure 7 illustrates the search for the minimum of the quality function F using the cuckoo algorithm on an example using the Michalewicz function:

$$F(x) = -\sum_{i=1}^{n} \sin x_i \left(\sin\left(\frac{i x_i^2}{\pi} \right) \right)^{2m} \qquad (6)$$
$$0 \leq x_i \leq \pi$$

where: $n \in Z_+$ is the dimension and $m \in Z+$ is a function parameter (typically m = 10).

The Michalewicz function has multimodal dependence and is often used as a tool for testing optimization algorithms. The test area is usually bounded by a hypercube $0 \leq x_i \leq \pi$; i = 1, ..., n. The function in this region has n! local minima, where n is dimension of the function. The parameter m is responsible for the angle of inclination of the valleys and edges. The difficulty of finding the minimum increases proportionally with an increase in the parameter m. For large values of m, minimizing the function can be compared to "looking for a needle in a haystack", as the values of the function are constant outside of the narrow valleys. With an increase in the parameter m, the valleys become narrower and the search for the minimum becomes more and more difficult.

In two-dimensional space (n = 2), the Michalewicz function takes the form:

$$F(x_1, x_2) = -\sin x_1 \left(\sin\left(\frac{x_1^2}{\pi} \right) \right)^{2m} - \sin x_2 \left(\sin\left(\frac{2 x_2^2}{\pi} \right) \right)^{2m} \qquad (7)$$
$$0 \leq x_1, x_2 \leq \pi$$

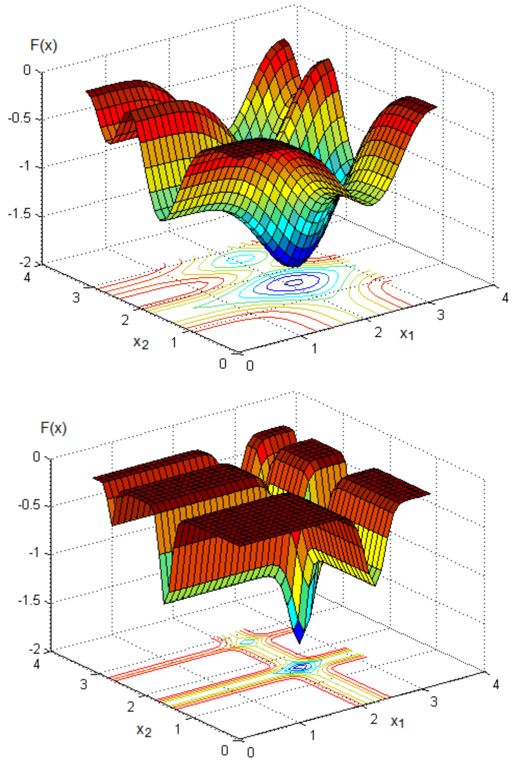

Fig. 7. Graph of the Michalewicz function for m = 1 (top), with minimum determined by the cuckoo algorithm as $\mathbf{F^* = -1.84}$; $\mathbf{x_1^* = 2.07; x_2^* = 1.57}$; and, for m = 10 (bottom), $\mathbf{F^* = -1.80}$; $\mathbf{x_1^* = 2.20; x_2^* = 1.57.}$

2.6 BAT ALGORITHM

Bats are the only mammals capable of flight. Living in flocks and leading a nocturnal lifestyle, they use echolocation by sending ultrasounds which are reflected back to them. This is the basis of the BA (Bat Algorithm) described by Yang in 2010. The bat population is represented by individuals in positions x_i, moving with velocities v_i, and generating impulses with frequencies f_i. The bats fly randomly, changing the wavelength and volume of their echolocations to find prey, and the rate of impulse emission can be customized to better determine the distance from their objective [11].

2.7 COCKROACH ALGORITHM

The CSO (Cockroach Swarm Optimization) algorithm, presented by Cheng et al. in 2011 [1], maps the swarming, scattering, and ruthless behaviors of cockroaches [1].

In the first behavioral aspect, each cockroach searches among the other cockroaches that it can see, in order to determine one whose solution is better than its own. If it finds one, it changes its solution by taking a step towards a cockroach with a better solution with a length randomly selected from the range [0, step], where step is the maximum step length (a parameter set at the beginning of the algorithm). If, among all the cockroaches visible to it, the individual does not find a better solution, then it takes an analogous step towards the current global best solution. In the second behavioral aspect, each cockroach takes a random step in the solution space, thus introducing an element of randomness into the algorithm. In the third behavioral aspect, a stronger individual may eat a weaker one which, in the algorithm, is manifested by a random cockroach from the population adopting the value of the current global optimum.

2.8 FLOWER POLLINATION ALGORITHM

The FPA algorithm (Flower Pollination Algorithm), developed by Yang in 2012 [12], takes inspiration from phenomenon of pollination of flowering plants, according to the following rules:

- biotic and cross-pollination is carried out by pollen carried by pollinators characterized by Lévy flight;
- abiotic pollination and self-pollination is local pollination;
- flower persistence—that is, the probability of reproduction—is proportional to similarity of two flowers involved;
- local and global pollination are controlled by a switching probability p ∈ [0, 1], due to physical proximity of factors such as wind, and local pollination can contribute significantly to pollination activity.

The above rules lead to the following relations:

$$\begin{aligned} x_i^{t+1} &= x_i^t + L\left(x_i^t - g*\right) \\ x_i^{t+1} &= x_i^t + \varepsilon\left(x_i^t - x_k^t\right) \end{aligned} \tag{8}$$

where: $x_i^t \in x_i^t$ is the solution vector and g* is the current best solution.

Flipping between the two equations takes place with probability p. The quantity ε is a random number with normal distribution, and L is the step length resulting from a Lévy distribution. Lévy flights enable global and local searches for optimal solutions. Unlike standard algorithms, Lévy flights represent long jumps, allowing the algorithm to climb out of local valleys. Lévy steps can be represented by the following approximation:

$$L @ \frac{1}{s^{1+b}} \tag{9}$$

Where: β is the Lévy exponent.

It can be difficult to figure out correct Lévy steps; as such, a simple way to generate Lévy flights is to transform two normal distributions u and v as follows:

$$s = \frac{u}{|v|^{1+b}} \tag{10}$$

where: $u \cong N(0, \sigma^2)$, $v \cong N(0,1)$, and σ is a function β.

2.9 KRILL ALGORITHM

The Krill Herd (KH) algorithm, described in 2012 by Gandomi and Alavi [4], imitates the life of a krill herd.

The minimum distance of the krill from food and from the highest stocking density is taken as the quality function for optimization. The position of a krill individual over time is influenced by the following factors:

- the movement of other individuals;
- feeding activity;
- random diffusion.

The KH algorithm uses a Lagrangian model, as follows:

$$\frac{dX_i}{dt} = N_i + F_i + D_i \tag{11}$$

where: N_i denotes movement caused by other krill individuals, F_i denotes foraging movement, D_i denotes the physical dispersal of krill individuals.

2.10 CUTTLEFISH ALGORITHM

Ees et al. [3] introduced the Cuttlefish Algorithm (CFA) in 2013, which mimics the environmental change in skin color of cuttlefish. It includes two global and two local searches, taking into account reflection and visibility, which serve as the basis for finding a global solution to the optimization problem.

The calculation of a new solution (newP) using reflection and visibility proceeds is as follows: *New P = reflection + visibility*. Figure 8 illustrates three structures of matte skin—namely, chromatophores, iridophores, and leucophores—in two examples (a, b) and three different traces of radiation (1,2,3), showing the possibility of color change due to the reflection of light. The algorithm maps the work of these cells by changing the order of the six possibilities.

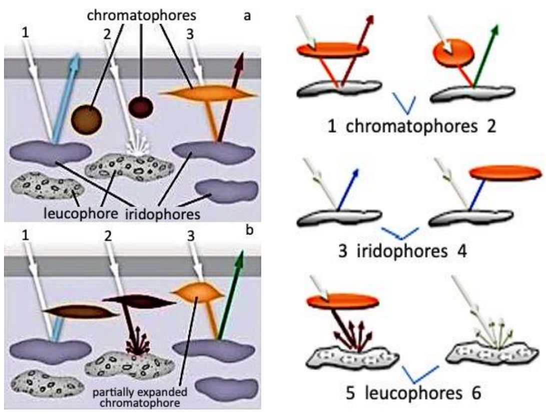

Fig. 8. Phases of cuttlefish color change (left) and change of order of six cuttlefish color events (right)

In the CFA algorithm, the population is divided into four groups (G1, G2, G3, and G4). In group G1, cases 1 and 2 (interactions between chromatophores and iridophores) are used to create new global solutions; in group G2, cases 3 (iridophorite reflection) and 4 (interaction between iridophores and chromatophores) are used to create new local solutions; in group G3, the interaction between leucophores and chromatophores (case 5) is used to find optimal local solutions; and, in group G4, case 6 (leucophore reflection) is used for global searching.

3 CONCLUSIONS

Heuristic algorithms are characterized by their significant ability to solve difficult optimization problems, being intuitively inspired by biological mechanisms, and the possibility of combining them with classical optimization methods.

The significant possibilities related to the use of particle swarm algorithms for optimization in engineering practice define their high potential as a subject of further scientific research related to the solution of complex optimization problems occurring in the transport and logistics fields, especially regarding land and maritime applications.

Funding: The project was financed in the framework of the program by the Ministry of Education and Science called "Regionalna Inicjatywa Doskonałości" in the years 2019-2023, project number: 006/RID/2018/19, the sum of financing 11 870 000 PLN.

WIELOKRYTERIALNA OPTYMALIZACJA GRY WIELOETAPOWEJ

W artykule przedstawiono heurystyczne metody optymalizacji statycznej wykorzystujące naturalne zasady ruchu roju cząstek – ptaków, mrówek, pszczół, świetlików, nietoperzy, kryli, kukułek, mątw, karaluchów i zapylania kwiatów. Dla każdej z metod opisano cechę inteligencji roju, przedstawiono postać funkcji celu oraz podano zasadę działania algorytmu optymalizacji.

Słowa kluczowe: transport, logistyka, optymalizacja, informatyka

BIBLIOGRAPHY

[1] Cheng L., Wang Z.B., Song Y.H., Guo A.H. (2011) Cockroach swarm optimization algorithm for TSP. Advanced Engineering Forum, vol 1, 226-229, https://doi.org/10.3390/e19050213.

[2] Bonabeau E., Dorigo M., Theraulaz G. (1999) Swarm intelligence: from natural to artificial systems. Oxford, Oxford University Press, ISBN: 0-19-513158-4.

[3] Eesa A.S., Brifcani A.N.A., Orman, Z. (2013) Cuttlefish algorithm – a novel bio-inspired optimization algorithm. International Journal of Scientific & Engineering Research, vol 4(9), 1978-1986, ISSN: 2229-5518.

[4] Gandomi A.H., Alavi A.H. (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simulat, vol 17, 4831-4845, https://doi.org/ 10.1016/j.cnsns.2012.05.010.

[5] Kennedy J., Eberhart R.C. (1999) The particle swarm: social adaptation in information-processing systems. In D. Corne M. Dorigo and F. Glover (eds.), New ideas in optimization. London, McGraw-Hill, ID: 59759192.

[6] Lazarowska A. (2022) Safe Trajectory Planning for Maritime Surface Ships. Springer, Berlin/Heidelberg, Germany, vol 13, 1–185, https://doi.org/10.1007/978-3-030-97715-3.
[7] Lisowski J. (2022) Metody optymalizacji. Pulishing House of Gdynia Maritime Uniwersity, 133-165 (in polish), ISBN: 978-83-7421-407-0.
[8] Pham D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S., Zaidi M. (2005) The bees algorithm. Technical Report: MEC 0501, Manufacturing Engineering Centre, Cardiff University.
[9] Yng X.S. (2008) Nature-inspired metaheuristic algorithms. Luniver Press, ISBN: 978-1-905986-28-6.
[10] Yang X.S., Deb S. (2009) Cuckoo search via Lévy flights. World Congress on Nature & Biologically Inspired Computing, NaBIC 2009, IEEE Publications, 210-214, https://doi.org/10.1109/NABIC.2009.5393690.
[11] Yang X.S. (2010) A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization, NISCO 2010. Studies in Computational Intelligence, vol 284, 65-74, https://doi.org/10.1007/978-3-642-12538-6_6.
[12] Yang X.S. (2012) Flower pollination algorithm for global optimization. Lecture Notes in Computer Science, vol 7445, 240-249, https://doi.org/10.1007/978-3-642-32894-7_27.