Transactions on Soft Computing

www.tsc.reapress.com

Trans. Soft. Comput. Vol. 1, No. 2 (2025) 99–108.

Paper Type: Original Article

Solving Mixed-Model Assembly Line Balancing Problem Using a Modified Artificial Fish Swarm Algorithm

Fahimeh Tanhaie*

Department of Industrial Engineering, Kosar university of Bojnord, Bojnord, Iran; Fahimeh.tanhaie@kub.ac.ir.

Citation:

Received: 05 June 2024 *Revised: 14 July 2024* *Accepted: 29 November 2024*	Tanhaie, F. (2025). Solving mixed-model assembly line balancing problem using a modified artificial fish swarm algorithm. *Transactions on soft computing, 1*(2), 99-108.

Abstract

The Mixed-Model Assembly Line Balancing Problem (MALBP) is a critical optimization challenge in manufacturing systems, where multiple product models with similar production processes are assembled on the same line. Efficiently assigning tasks to workstations while respecting precedence constraints and cycle time limitations is essential to minimize idle time and maximize productivity. In this paper, we propose an improved Artificial Fish Swarm Algorithm (AFSA) enhanced with group escaping behavior; a natural reaction observed in fish swarms when sensing danger. This novel hybridization improves both convergence speed and global search ability of the algorithm. A detailed pseudocode of the proposed method is provided, and its performance is evaluated on several benchmark instances of MALBP. Experimental results indicate that the proposed algorithm outperforms the standard AFSA in terms of solution quality and stability, making it a promising approach for real-world mixed-model assembly line optimization.

Keywords: Mixed-model assembly line, Balancing problem, Artificial fish swarm algorithm, Group escaping behavior.

1 | Introduction

The Mixed-Model Assembly Line Balancing Problem (MALBP) is a well-known combinatorial optimization challenge in production planning and manufacturing systems. In this problem, various product models with similar processing requirements are produced on the same assembly line in an intermixed sequence. The main objective is to assign tasks to workstations in such a way that production constraints, such as precedence relations and cycle time limitations, are satisfied while optimizing performance indicators like line efficiency,

Corresponding Author: Fahimeh.tanhaie@kub.ac.ir
https://doi.org/10.48314/tsc.v1i2.44

workload balance, and idle time minimization. Due to its NP-hard nature, classical mathematical programming approaches often struggle to find optimal solutions within reasonable computational time, especially for large-scale instances [1], [2]. Therefore, metaheuristic algorithms have become increasingly popular for tackling real-world MALBP cases [3], [4].

Artificial Fish Swarm Algorithm (AFSA) is a population-based metaheuristic inspired by the collective behaviors of fish swarms in natural environments. In aquatic ecosystems, fish exhibit self-organized movement patterns without centralized control. They tend to remain close to the group and are typically distributed around areas rich in food resources. Research has identified three fundamental behaviors commonly observed in fish schooling: Swarming: where individuals move collectively toward the center of the group while avoiding overcrowding,

Following: where fish follow their nearest neighbors, and Preying: where fish move toward regions with higher food concentration. AFSA, first introduced by Li [5], simulates these basic behaviors to guide Artificial Fish (AF) toward optimal solutions in the search space. The algorithm offers several advantages, including robustness, simplicity, and insensitivity to initial conditions. It has been successfully applied to various optimization domains such as data mining [6], clustering [7], image registration [8], and PID control [9]. However, one of the major drawbacks of the standard AFSA is its relatively slow convergence speed and tendency to get trapped in local optima.

An important behavior frequently observed in natural fish schools but rarely incorporated into AFSA is group escaping behavior. This behavior occurs when a part of the school detects a threat or external stimulus, prompting the entire swarm to rapidly change direction to avoid danger. As illustrated in *Fig. 1*, this reaction enhances the group's responsiveness and exploration capability in dynamic environments. Motivated by this natural mechanism, we propose a novel hybrid version of AFSA that integrates group escaping behavior to improve both convergence speed and global search ability. The proposed method is applied to solve the MALP, demonstrating superior performance compared to the conventional AFSA.

Fig. 1. Group escape behavior of fish schools.

2 | Literature Review

Literature review showed that researchers have made some attempts to improve the performance of AFSA. In order to increase the diversification of the artificial fishes based on their parents' characteristics, Gao et al. [10] presented AF Swarm optimization algorithm with crossover, CAFAC. In this algorithm, the crossover operator is first explored. Numerical results demonstrated that the proposed method can outperform the original AFSA. Jiang and Yuan [11] increased the search quality of AFSA by discussing the possibility of parallelization of AFSA and making the analysis of parallel AFSA.

Some researchers have tried to combine different algorithm with AFSA and improve the performance of AFSA. Huadong Chen et al. [12] presented a hybrid algorithm to train forward neural network using a hybrid of particle swarm optimization and AFSA. The proposed method was more effective than AFSA and PSO. The hybrid in the proposed algorithm not only has the AF behaviors of swarm and follow, but also takes advantage of the information of the particle.

Belacel et al. proposed an AFSA with adaptive visual to improve the performance of fuzzy clustering [13]. In this paper AFSA enhances the performance of the Fuzzy C-Means (FCM) algorithm. A numerical result showed the advantages of the proposed algorithm. Also, Chen et al. [14] reported a hybrid algorithm by using the characteristics of AFSA and Chaos Optimization Algorithm. This method is an efficient global optimization algorithm for solving global optimization problem. In this approach, adding chaos is suitable for updating the velocities of AF and improving the convergence rate and the accuracy.

Fernandes et al. [15] presented a new method with a set of movements, closely related to the random, the searching and the leaping fish behaviors. This algorithm tested on a set of seven benchmark problems. Yazdani et al. [16], a new algorithm is presented for optimization in static and continuous environments by hybridizing AFSA and cellular learning automata. Experimental results showed that the proposed algorithm has an acceptable performance.

In order to improve the optimization performance of the AFSA, many researchers have presented some improved algorithm. Although all the proposed algorithms can improve the performance of the original algorithm, but it still cannot get satisfactory results for some problems. In this paper, we proposed a new method based on the group escape behavior of fish that is ignored in most of the researches. There are some researches on observing group escape behavior of fish school biologically [17], but there is no research on applying on AFSA.

When the predator comes very close to a fish, the fish chooses the group escape behavior and if there is no predator in the visible area of the fish, the fish moves using its schooling behavior.

3 | Proposed AFSA with Group Escape Behavior

In this section an AFSA based on group escape behavior is proposed which in that both ability of local search and global search to standard AFSA has been increased and improved. Improved AFSA is a random search algorithm based on simulating fish swarm behaviors which contains group escape behavior, preying behavior, swarming behavior and following behavior. It constructs the simple behaviors of AF firstly, and then makes the global optimum appear finally based on animal individuals' local searching behaviors. Before explaining the improved AFSA, we introduce some definitions [5], [18] and [19]:

Assuming in an n-dimensional searching space, there is a group composed of n articles of AF. Situati6on of each individual AF can be expressed as vector $x = (x_1, x_2, x_3, \dots, x_n)$ is denoted the current state of AF, where X_n is control variable.

$f(x)$ is the fitness or objective function of X, which can represent food concentration of AF in the current position.

$Dist_{ij} = \|X_i - X_j\|$ is denoted the Euclidean distance between fishes.

Visual and Step are denoted respectively the visual distance of AF and the distance that AF can move for each step.

X_v is the visual position at some moment. If the state at the visual position is better than the current state, it goes forward ad step in this direction, and arrives the X_{next} state, otherwise, continues an inspecting tour in the vision.

Try-number is attempt times in the behavior of prey.

δ is the condition of jamming $(0 < \delta < 1)$.

Supposed X_v is the visual position at some moment and X_{next} is the new position. The movement process is represented as

$$X_v = X_i + \text{Visual} \times \text{Rand}(0,1),$$
$$X_{next} = X + \frac{X_v - X}{\|X_v - X\|} \times \text{Step} \times \text{Rand}(0,1),$$

where rand () produces random numbers between 0 and 1. This behavior is shown in *Fig. 2.*

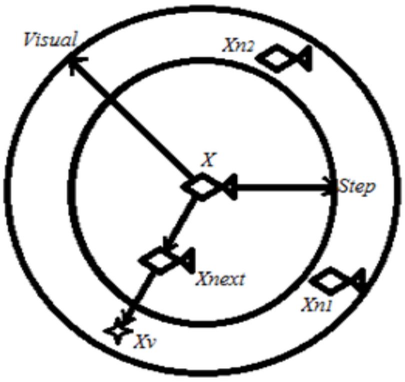

Fig. 2. The movement process of fish.

It is well known that most of fish schools have a group behavior: group escaping. When predator is discovered by one or more fishes in the school, the whole school changes the direction very quickly. In the proposed algorithm, when an AF could not find better situation in food prey behavior or in swarm behavior with doing a freely movement, this situation saves in problem space. So, other fishes sense this situation as an enemy and change their direction with high speed and moving away from it. We see this group escape behavior as follows:

When the fish discover a water area with more food, they will go quickly toward the area. Consider the state of AF is X_i, Select a state X_j within its sensing range randomly. If $f(X_i) > f(X_j)$, the new situation is not better than current situation and this fish has to informs other fish, so we save this situation as forbidden zone in problem space to prevent other fishes going there.

Forbidden zone:

$$\{X_j = X_i + \text{Visual} \times \text{Rand}(-1,1), \quad \text{if } f(X_i) > f(X_j), \text{for each i and j}\}.$$

With this introduction, the basic behaviors of improved AFSA are defined as follows (the three most common behaviors of fishes like preying, swarming and following are from the standard AFSA [5].

3.1| Prey Behavior

In general, when the fish discover a water area with more food, they will go quickly toward the area. Consider the state of AF is X_I, Select a state X_j within its sensing range randomly. If $f(X_i) \leq f(X_j)$ and X_j is not in forbidden zone, then fish move to X_j. On the contrary, select randomly state X_j that is not in forbidden zone and determine whether to meet the forward conditions, repeat several time, if still not satisfied forward conditions, then move one step randomly.

$$X_j = X_i + \text{Visual} \times \text{Rand}(-1,1).$$

If $f(X_i) \leq f(X_j)$ and X_j is not in forbidden zone, it goes forward a step as follows.

$$\overline{X}_i(t+1) = \overline{X}_i(t) + \frac{\overline{X}_j - \overline{X}_i(t)}{Dist_{i,j}} \times Step \times Rand(0,1).$$

If $f(X_i) > f(X_j)$, we add this situation to forbidden zone in problem space to prevent other fishes going there.

Updating forbidden zone:

$$\{X_j = X_i + Visual \times Rand(-1,1), \quad \text{if } f(X_i) > f(X_j), \text{for each i and j}\}.$$

Swarm behavior: in the process of swimming, the fish will swarm in order to share the food of the swarm. Supposed the current state of AF is X_i, number of AF is n and number of neighbors around the AF X_{center} is n_c, if $\delta > (n_c/n)$ indicates that the partners have more food and less crowded, if $f(X_{center}) \geq f(X_i)$ and X_{center} is not in forbidden zone, then go forward toward the center of the direction of the partnership, otherwise prey behavior.

$$X_{center} = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

If $f(X_{center}) \geq f(X_i)$ and X_{center} is not in forbidden zone, it goes forward a step as follows.

$$X_i(t+1) = X_i(t) + \frac{X_{center} - X_i(t)}{\|X_{center} - X_i(t)\|} \times Step \times Rand(0,1).$$

If $f(X_{center}) < f(X_i)$, we add this situation to forbidden zone in problem space to prevent other fishes going there.

Updating Forbidden Zone:

$$\left\{X_j = X_{center} = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad \text{if } f(X_{center}) < f(X_i), \text{for each i}\right\}.$$

Follow behavior: when one fish of the fish swarm discovers more food, the other fish will share with it. Supposed the state of AF is X_i , explore its optimal state X_n from Visual neighbors , the number of partners of X_n is n_n and the number of AF is n, If $\delta > (n_c/n)$indicates that near distance have more food and not too crowded, further move to the front of X_n position; otherwise perform foraging behavior.

$$X_i(t+1) = X_i(t) + \frac{X_n - X_i(t)}{\|X_n - X_i(t)\|} \times Step \times Rand(0,1).$$

Taking into account the above mentioned behaviors, the pseudocode of improved AFSA can be written as follows:

Begin: for each AF $i \in [1 \dots N]$. Initialize X_i such as step, visual, the number of exploratory try number.

Endfor: blackboard -$^{best}_{X_i} f(X_i)$ to record the current status of each fish and select the optimal value recorded.

Repeat: for each AF $i \in [1 \dots N]$.

Save forbidden zone based on group escaping behavior.

Perform prey behavior on $X_i(t)$ and compute $X_{i,prey}$.

Update forbidden zone based on group escaping behavior.

Perform swarm behavior on $X_{i,prey}(t)$ and compute $X_{i,swarm}$.

Update forbidden zone based on group escaping behavior.

Perform follow behavior on $X_{i,prey}(t)$ and compute $X_{i,follow}$.

if $f(X_{i,swarm}) \le f(X_{i,follow})$.
Then $X_i(t+1) = X_{i,follow}$.

Else

$X_i(t+1) = X_{i,swarm}$.

Endfor

if $f(X_{Best-AF}) \ge f(\text{Blackboard})$.

Then $\text{blackboard} = X_{Best-AF}$ Optimal value in Blackboard is updated (until stopping criterion is met).

Therefore, in the improved AFSA other fishes sense unsuitable situation as a forbidden zone and change their direction with high speed and moving away from it, so, the ability of local search and global search to standard AFSA has been increased.

4 | Experimental Results

To evaluate the performance of the proposed improved AFSA in solving the MALBP, we conducted a series of experiments on benchmark instances of this problem.

The notation applied for problem are as follows.

M	Number of models ($j, u \in \{1, 2, .., M\}$)
N	Maximum number of workstations ($i, g, s \in \{1, 2, .., N\}$)
$k, h, W \in \{1, 2, .., N\}$	Index of candidate workstations
$w \in \{1, 2, .., N\}$	Index of candidate operators
n	Number of workstations in the primary balancing
Q	Subset of tasks common in all models
d_j	Demand of model j
bigM	A high-value parameter
IP_i	Immediate predecessors of task i
IB_{ijw}	If task i of model j is assigned to workstation w
t_{ijk}	Processing time of task i of model j at workstation k
Ea_{ij} (La_{ij})	Earliest (Latest) workstation that task i of model j could be assigned

Decision variables:

$$x_{ijk} = \begin{cases} 1, & \text{If task i of model j is assigned to station k}, \\ 0, & \text{otherwise.} \end{cases}$$

T_k	Positive variable: total processing time at station k.
$\bar{T}$	Positive variable: average processing time of each station.
CT	Positive variable: stochastic cycle time.

5 | Multi Objective Modeling

Balancing problems in MMAL can be considered from different perspectives. In this study, three objectives are discussed. Now a mathematical model for the defined problem is presented:

$$Z_1 = \min \sum_{k=1}^{N} \left| T_k - \bar{T} \right|. \tag{1}$$

$$\sum_{k=Ea_{ij}}^{La_{ij}} x_{ijk} = 1, \quad \text{for all } i, j. \tag{2}$$

$$x_{ijk} \leq \sum_{k=Ea_{sj}}^{La_{sj}} x_{ijk} = 1, \quad \text{for all } i, j, \quad \text{for all } s \in IP_i, \quad \text{where} \begin{cases} Ea_{sj} \leq Ea_{ij}, \\ Ea_{ij} \leq La_{sj}. \end{cases} \tag{3}$$

$$\sum_{i \in S_{jk}} t_{ijk} * x_{ijk} \leq CT^{ub}, \quad \text{for all } j, k. \tag{4}$$

$$T_k = \sum_{i=1}^{N} \sum_{j=1}^{N} d_j * t_{ijk} * x_{ijk}, \quad \text{for all } k. \tag{5}$$

$$\bar{T} = \frac{1}{N} * \sum_{k=1}^{N} T_{kl}. \tag{6}$$

$$\sum_{i \in S_{jk}} x_{ijk} - \left\| S_{jk} \right\| * y_{jk} \leq 0, \quad \text{for all } j, k. \tag{7}$$

$$\sum_{i=1}^{M} y_{jk} - M * z_k = 0, \quad \text{for all } k. \tag{8}$$

$$\text{bigM} * (1 - x_{gjk}) - \sum_{u=1}^{M} \sum_{h \neq k} x_{guh} \geq 0, \quad \text{for all } g \in Q, \text{ for all } j, k. \tag{9}$$

$$0 \leq CT \leq CT^{ub}. \tag{10}$$

$$x_{ijk}, y_{jk}, z_k \in \{0,1\}. \tag{11}$$

$$T_k \geq 0, \bar{T} \geq 0, CT \geq 0. \tag{12}$$

The first objective function minimizes the workload variation at each station. *Eq. (2)* warrants that each task needed by every model is assigned to only one station. *Constraint (3*) considers the precedence relations between tasks. *Constraint (4)* restricts the total process. *Eqs.* (5) and *(6)* calculate the process time at each station and average process time at each station on all lines for vertical balancing, respectively. *Constraints (7)* and *(8)* imply that the same station is used for each model on each line and if a station is not utilized by a model, it will not be applied by other models. *Constraint (9)* ensures that common tasks of models are assigned to the same station on each line. *Constraint (10)* limits the cycle time for each line and *Constraints (11)* and *(12)* define all decision variables.

MALBP involves assigning tasks from multiple product models to workstations in a way that minimizes idle time and balances the line while respecting precedence constraints and cycle time limitations. Due to its NP-hard nature, exact methods are often computationally prohibitive for large-scale problems, making metaheuristics like AFSA promising candidates for efficient solutions.

We applied both the standard AFSA and our improved version incorporating group escaping behavior to several well-known MALBP instances. The parameters used in the algorithm were set as follows: the population size was set to N=20 AF; the maximum number of attempts in prey behavior was set to try number=5; the visual distance was set to 4.5; and the step size was set to 0.3. The maximum number of iterations allowed was 2000. We compared the performance of the standard AFSA with the proposed improved AFSA in terms of cycle time, line efficiency, and computational time. The experimental results are summarized in *Table 1* and *Fig. 3*.

Table 1. Experimental results for MALBP instances.

Algorithm	Instance Size	Cycle Times		Line Efficiency		Time	
		Mean	Best	Mean	Best	Mean	Best
Standard AFSA	Small scale	38.48	36.72	82.41	85.30	166.77	127.59
Improved AFSA	Small scale	37.12	35.25	84.03	87.15	159.69	124.55
Standard AFSA	Medium scale	58.74	56.34	79.21	81.45	293.86	267.74
Improved AFSA	Medium scale	55.62	53.18	81.05	83.96	280.56	250.24

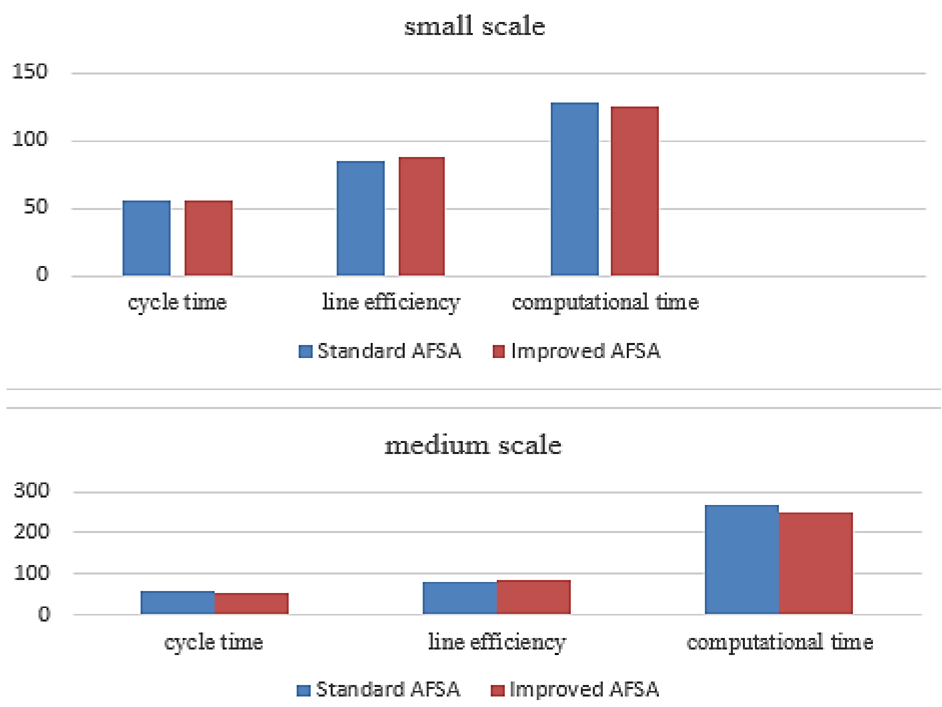

Fig. 3. Experimental results.

As shown in *Table 1*, the improved AFSA consistently outperforms the standard AFSA in both solution quality and computational efficiency. The incorporation of group escaping behavior significantly enhances the algorithm's exploration capability, leading to better convergence and higher-quality solutions. Specifically, the improved algorithm achieves lower cycle times and higher line efficiency across all tested instances, demonstrating its effectiveness in solving real-world MALBP scenarios.

5 | Conclusion

The AFSA is a promising metaheuristic method that offers several advantages, including robustness, simplicity in implementation, and insensitivity to initial parameter settings. These characteristics make it suitable for solving complex optimization problems such as the MALBP, which is known to be NP-hard and difficult to solve using exact methods within reasonable computational time. However, the standard AFSA suffers from slow convergence speed and a tendency to get trapped in local optima. Moreover, it lacks an effective mechanism to utilize the experiences of other swarm members during the search process, which limits its global exploration ability.

To address these limitations, we proposed an improved version of AFSA by incorporating group escaping behavior, a natural phenomenon observed in fish schools when sensing danger. This behavior allows the entire swarm to rapidly change direction based on the reaction of a few individuals, enhancing both the responsiveness and diversity of the search process. The hybridization of AFSA with this behavior significantly improves the algorithm's convergence speed and solution quality. Experimental results on benchmark MALBP instances demonstrated that the proposed improved AFSA outperforms the standard version in terms of cycle time reduction, line efficiency improvement, and computational performance. The enhanced algorithm shows a more stable search behavior and better global optimization capability, making it a suitable approach for real-world assembly line balancing applications. As future work, we plan to apply the proposed algorithm to other combinatorial optimization problems in manufacturing systems, such as multi-objective assembly line balancing, U-shaped lines, and robotic assembly line balancing. Additionally, hybridizing the proposed method with other local search techniques or machine learning strategies could further enhance its performance and adaptability.

Acknowledgments

The author sincerely appreciates the valuable efforts of the reviewers and editorial team for their constructive comments and professional evaluations, which contributed significantly to improving the quality, technical accuracy, and presentation of this research.

Funding

No external financial support was received for conducting this research.

Data Availability

The benchmark datasets and computational results used in this study are available from the corresponding author upon reasonable request.

References

[1] Alinaghian, M., & Naderipour, M. (2016). A novel comprehensive macroscopic model for time-dependent vehicle routing problem with multi-alternative graph to reduce fuel consumption: A case study. *Computers & industrial engineering*, *99*, 210–222. https://doi.org/10.1016/j.cie.2016.07.029

[2] Naderipour, M., & Alinaghian, M. (2016). Measurement, evaluation and minimization of CO2, NOx, and CO emissions in the open time dependent vehicle routing problem. *Measurement*, *90*, 443–452. https://doi.org/10.1016/j.measurement.2016.04.043

[3] Tanhaie, F., Rabbani, M., & Manavizadeh, N. (2021). Sequencing mixed-model assembly lines with demand management: Problem development and efficient multi-objective algorithms. *Engineering optimization*, *53*(7), 1101–1118. https://doi.org/10.1080/0305215X.2020.1759580

[4] Tanhaie, F., Rabbani, M., & Manavizadeh, N. (2020). Applying Available-To-Promise (ATP) concept in mixed-model assembly line sequencing problems in a make-to-order (MTO) environment: Problem extension, model formulation and Lagrangian relaxation algorithm. *Opsearch*, *57*(2), 320–346. https://doi.org/10.1007/s12597-019-00436-6%0A%0A

[5] Li, X. (2003). *A new intelligent optimization method-artificial fish school algorithm* [Thesis]. Doctor thesis of zhejiang universi*ty*. https://cir.nii.ac.jp/crid/1572261549215438080

[6] Zhang, M., Shao, C., Li, M., & Sun, J. (2006). Mining classification rule with artificial fish swarm. *2006 6th world congress on intelligent control and automation*, 5877–5881. IEEE https://doi.org/10.1109/WCICA.2006.1714205

[7] Xiao, L. (2010). A clustering algorithm based on artificial fish school. *2010 2nd international conference on computer engineering and technology, (*7), V7-766. IEEE. https://doi.org/10.1109/ICCET.2010.5485745

[8] Wang, Y., Zhang, W., & Li, H. (2014). Application of artificial fish swarm algorithm in image registration. *Computer modelling & new technologies*, *18*(12B), 510–516. http://www.cmnt.lv/upload-files/ns_39brt086 CMNT1901-203 DEF-6 CMNT-73-AG-ed-VG.PDF

[9] Luo, Y., Wei, W., & xin Wang, S. (2010). Optimization of pid controller parameters based on an improved artificial fish swarm algorithm. *Third international workshop on advanced computational intelligence*, 328–332. IEEE. https://doi.org/10.1109/IWACI.2010.5585187

[10] Gao, X. Z., Wu, Y., Zenger, K., & Huang, X. (2010). A knowledge-based artificial fish-swarm algorithm. *2010 13th IEEE international conference on computational science and engineering*, 327–332. IEEE. https://doi.org/10.1109/CSE.2010.49

[11] Jiang, M., & Yuan, D. (2012). Parallel artificial fish swarm algorithm. *Advances in control and communication* (pp. 581–589). Springer. https://doi.org/10.1007/978-3-642-26007-0_72%0A%0A

[12] Chen, H., Wang, S., Li, J., & Li, Y. (2007). A hybrid of artificial fish swarm algorithm and particle swarm optimization for feedforward neural network training. *International conference on intelligent systems and knowledge engineering 2007*, 1025–1028. Atlantis press. https://doi.org/10.2991/iske.2007.174

[13] He, S., Belacel, N., Hamam, H., & Bouslimani, Y. (2009). Fuzzy clustering with improved artificial fish swarm algorithm. *2009 international joint conference on computational sciences and optimization*, (2), 317–321. IEEE. https://doi.org/10.1109/CSO.2009.367

[14] Chen, Z., & Tian, X. (2010). Notice of retraction: Artificial fish-swarm algorithm with chaos and its application. *2010 second international workshop on education technology and computer science* (Vol. 1, pp. 226–229). IEEE. https://doi.org/10.1109/ETCS.2010.253

[15] Fernandes, E., Martins, T., Maria, A., & Rocha, A. (2009). Fish swarm intelligent algorithm for bound constrained global optimization. *CMMSE 2009*, 1–3. https://www.researchgate.net/publication/228418487_Fish_swarm_intelligent_algorithm_for_bound_constrained_global_optimization

[16] Yazdani, D., Golyari, S., & Meybodi, M. R. (2010). A new hybrid algorithm for optimization based on artificial fish swarm algorithm and cellular learning automata. *2010 5th international symposium on telecommunications*, 932–937. IEEE. https://doi.org/10.1109/ISTEL.2010.5734156

[17] Parrish, J. K., Viscido, S. V, & Grunbaum, D. (2002). Self-organized fish schools: An examination of emergent properties. *The biological bulletin*, *202*(3), 296–305. https://doi.org/10.2307/1543482

[18] Wang, C. R., Zhou, C. L., & Ma, J. W. (2005). An improved artificial fish-swarm algorithm and its application in feed-forward neural networks. *2005 international conference on machine learning and cybernetics* (5), 2890–2894. IEEE. https://doi.org/10.1109/ICMLC.2005.1527436

[19] Luo, Y., Zhang, J., & Li, X. (2007). *The optimization of pid controller parameters based on artificial fish swarm algorithm*. 2007 IEEE international conference on automation and logistics, 1058–1062. IEEE. https://doi.org/10.1109/ICAL.2007.4338724