ADVANCED MULTI-OBJECTIVE ALGORITHM USED TO OPTIMIZE CONSUMPTION OF AN INTEGRATED SYSTEM FOR FLEXIBLE MANUFACTURING

Marius-Adrian PĂUN[1,3], Henri-George COANDĂ[2,] Eugenia MINCĂ[2], Octavian Gabriel DUCA[3], Ioana FĂGĂRĂȘAN[1], Florin DRAGOMIR[2,4], Sergiu Stelian ILIESCU[1]

The optimization of power consumption of manufacturing lines (ML) is a relevant topic in the context of increasing electricity prices. This paper presents the topic of optimizing the energy consumption of a production system, through optimization technique based on the multi-objective PSO (Particle Swarm Optimization) algorithm. The novelty of the approach lies in the development of an Advanced PSO algorithm (APSO), which brings superior results in optimizing energy consumption. The research was carried out in successive stages, starting with data collection, and ending with the implementation of the results on the production line, everything being subordinated to the criterion of optimizing energy consumption. APSO's advanced optimization technique was implemented on a flexible manufacturing line (FML) composed of five linearly interconnected workstations. The APSO optimization algorithm, following implementation and testing on FML, provides a solution for controlling the conveyor speeds of each station that guarantees in the overall approach to the process a minimum energy consumption, but also a minimum execution time of the algorithm.

Keywords: flexible manufacturing line (FML), power consumption monitoring, multi-objective optimization algorithm, PSO algorithm applied to optimize energy consumption

1. Introduction

In the current context of rising electricity prices and the growing need to implement environmental protection policies proposed mainly through the principles formulated by Industry 5.0, it is noted that factories must use increasingly

[1] National University of Science and Technology POLITEHNICA of Bucharest. Automation and Industrial Informatics Department, e-mails: paun_marius_2009@yahoo.com, grigore.stamatescu@upb.ro, iliescu.shiva@gmail.com, ioana.i.fagarasan@gmail.com.

[2,] Faculty of Electrical Engineering, Electronics, and Information Technology Valahia University of Targoviste, Romania, e-mails: coanda_henri@yahoo.com, eugenia.minca@gmail.com, drg_florin@yahoo.com;

[3] Institute of Multidisciplinary Research for Science and Technology, Valahia University of Targoviste, Romania, e-mail: octavian_duca@yahoo.com.

[4] Academy of Romanian Scientists, 54, Splaiul Independentei, 050044 Bucharest, Romania

high-performance equipment with low electricity consumption [2,5]. Thus, to improve consumption parameters and productivity, computational modelling is mainly used for efficient planning, control, and management of workloads in different production scenarios. Emphasis is placed on streamlining the consumption parameters of already processed and integrated production equipment [3].

In order to make the power consumption of a manufacturing line more efficient, the following steps must be applied: monitoring, analysis and management. At each stage, it is necessary to use a hardware and software infrastructure adapted to the type of implemented production system [4]. The analysis phase has the greatest complexity in terms of the fact that to make electricity consumption more efficient, optimization algorithms appropriate to the existing problem can be applied [7].

Achieving a high-performance optimization requires the use of algorithms such as: PID algorithms, fuzzy algorithms, metaheuristics algorithms, reinforcement learning algorithms, neural network algorithms, multiagent algorithms, predictive algorithms, etc [6]. Amongst the algorithms listed, it can be appreciated that metaheuristic optimization algorithms show good efficiency compared to other algorithms [14]. These algorithms propose iterative processes and concepts derived from artificial intelligence to easily identify the optimal solution or a value close to the optimal solution [1]. Among the metaheuristic algorithms for optimization proposed, in this paper, the PSO algorithm was chosen.

The present paper addresses the problem of optimizing power consumption by applying a multi-objective optimization algorithm on consumption data taken from a FML through six chapters: Introduction, Comprehensive overview of the FML and the power consumption monitoring system (PCMS), Modelling of power consumption through linear regression (LR) and statistical analysis of results, Advanced PSO optimization algorithm applied to optimize energy consumption, Advanced PSO algorithm versus PSO approach Discussion and Conclusions.

2. Comprehensive overview of the FML and the power consumption monitoring system (PCMS)

The manufacturing system on which the research will be carried out is an educational manufacturing line consisting of seven manufacturing stations, each station performing specific operations on the working product (Fig. 1). Among these seven stations, six are equipped with conveyors for transporting parts inside and between stations, thereby enabling the manufacturing process to proceed in a continuous flow. The system can achieve flexible manufacturing from the perspective of the type of product as well as from the perspective of the assembled configuration adapted to market requirements [15].

The assembly process implemented on the manufacturing line is aligned with market demand and, by extension, customer preferences. The products made are educational in nature, they are assembled for demonstration purposes. The assembly of products requires the sequential passage of the working part through a defined number of stations in a predetermined order, that corresponds to the specific assembly requirements. Consequently, depending on the type of product, a certain manufacturing flow will be used [9].

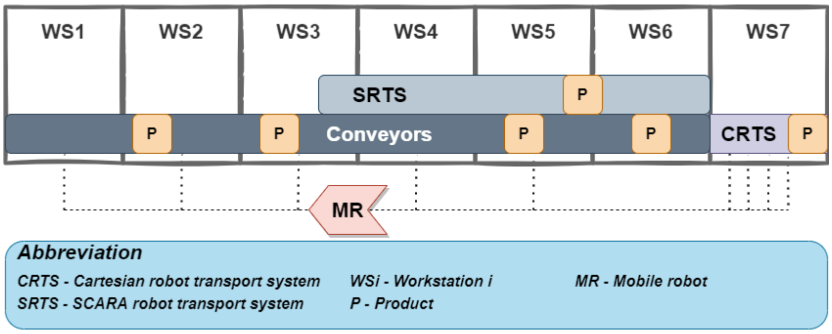


Fig. 1. Flexible manufacturing line (FML) equipped with manipulators and mobile robots, for assembly-disassembly operations.

The assembly process encompasses three the production of three distinct product types: type A product, type B product and type H product (Fig. 2) where:

- Type A product, characterized as a straightforward product, comprises the subsequent components arranged in the specified sequence: transport tray, basal part, small parts layer, and top part;
- Type B product, classified as a complex product, comprises the subsequent components arranged in the specified sequence: transport tray, basal part, small parts, top part, small parts, and top part;
- Type H product is a hybrid product that has the characteristics of previous products with the difference that it can have small parts of different sizes.

All three pieces can be performed both sequentially and in a pseudo-parallel manner.[8]

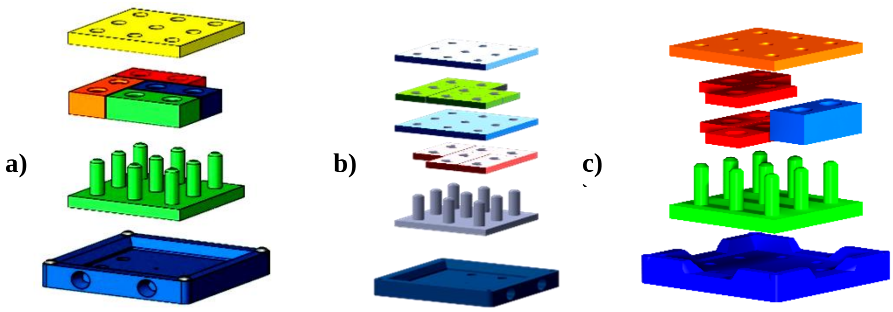


Fig. 2. Product types assembled/disassembled on FML: a) product A – simple product; b) product B complex product; c) product H - hybrid typology.

The FML allows the implementation of the disassembly process in two separate workstations (WS) in WS3 and WS7. Both WS are equipped with robots adapted to the tasks they perform. These robots have dedicated clamping systems with which they can carry out every step of the disassembly process. Given the system's architecture, which allows to produce distinct product types, it is necessary that the disassembly process be carried out in accordance with the structure of the product. Consequently, for disassembling a product a number of disassembly tasks equal to the number of assembly tasks must be used.

For advanced analysis of consumption behaviour, FML was equipped with power meters on each WS. The architecture of the consumption data acquisition system (Fig. 3) was adapted to the architecture of the production system. In this approach, the PCMS allows the analysis of WS behaviour in an integrated manner. The PCMS (Fig. 3) comprises several components, including seven measuring meters with communication via Modbus RTU protocol, control panel, Modbus RTU – USB converter and embedded Raspberry Pi 3B+ system. The meters are connected to a Modbus RTU bus.

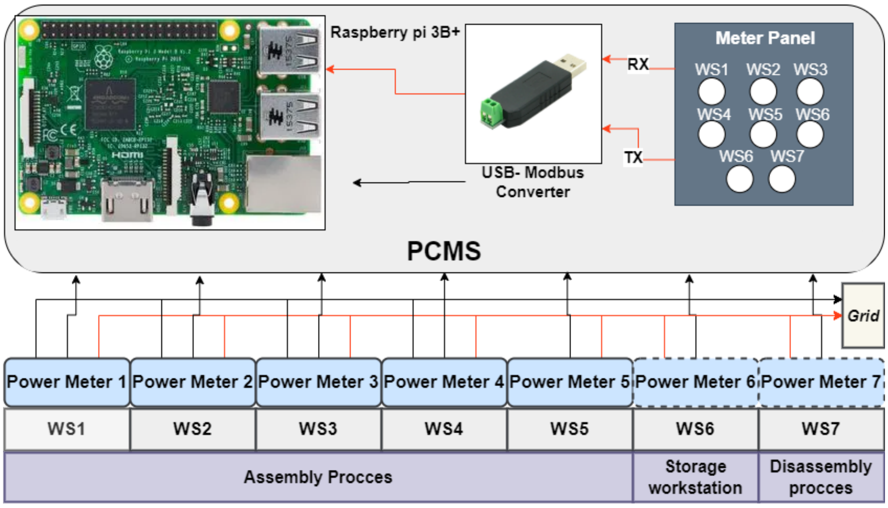


Fig. 3. Architecture of the PCMS.

The utilization of this system offers several advantages. It allows the use of a programming language such as Python that uses data analysis libraries. Moreover, it is running at high code execution speed, and offers the possibility of creating graphical interfaces. Additionally, it exhibits low power consumption, contributing to energy efficiency. Nevertheless, if the system is used in a sealed and ventilated automation enclosure, the system can deliver high performance at a cost-effective price point.

Tabel 1.

Average acquisition times of energy consumption data for each WS

WS	WS1	WS2	WS3	WS4	WS5	WS6	WS7
Data acquisition time (s)	0.279	0.279	0.23	0.279	0.318	0.28	0.318

Table 1 presents the performance metrics related to data acquisition from meters via the Modbus protocol. It is important to highlight that the Modbus protocol does not support parallel querying of connected meters. The data transmission rate of the counters operates at 9600 bits per second, which causes a correlation between the amount of data transferred and the time.

In the context of the meters installed on workstations, the approximate transfer-data time is 0.32s. However, in the case of the manufacturing process of a product with straightforward architecture, six power meters were analysed, this results in an acquisition time of approximately 1.92 seconds for each individual meter. This acquisition time is not convenient because it does not permit a comprehensive analysis of the station's consumption behaviour. The solution found and to mitigate this time constraint involved activating each meter only when its corresponding station is in production. Thus, a single request can be achieved observing the consumption behaviour for each station in a time of 0.32s. However, it's important to acknowledge that in production scenarios, there may arise instances where two, three, four, five, or even all six meters are concurrently in use.

3. Modelling of power consumption through linear regression (LR) and statistical analysis of results

For each WS, energy consumption data at different working speeds were collected by the developed monitoring system. Subsequently, the collected data underwent a thorough analysis during which mathematical models of energy variation depending on the conveyor's working speed were generated by applying LR. In order to change the conveyor speed, a telegram transmitted from the programmable automatic controller (PLC) to the driver/converter in the WS is used.

The LR formula is represented by means of the relationship between a dependent variable (often denoted as "Y"), in this case total power consumption, and one or more independent variables, as appropriate, (often denoted as "X"), in this case the equivalent velocity [10]. The LR formula (1) for independent single-variable is:

$$Y = \beta_0 + \beta_1 * X + \varepsilon \tag{1}$$

Where: Y - dependent variable, x - independent variable, β_i - regression coefficient and ε - error term.

By utilizing the LR formula to the data collected for each WS the production system, mathematical models were created. The mathematical model obtained for each WS are presented in formula 2.

$$Z=\begin{pmatrix} (-0.0008*v_1+8.4175)*(v_1*0.0114+69.304)+tsi_1*43.9193 \\ (-0.0035*v_2+20.209)*(v_2*0.0481+70.804)+tsi_2*98.39177 \\ (-0.0052*v_3+21.350)*(v_3*0.0126+115.57)+tsi_3*178.834 \\ (-0.0025*v_4+85.454)*(v_4*0.0021+189.36)+tsi_4*131.9418 \\ (-0.0034*v_5+24.056)*(v_5*0.0080+30.561)+tsi_5*25.11458 \end{pmatrix} \quad (2)$$

Where: v_i - conveyor equivalent speed, and tsi_i - the stationary time caused by the operation of the next station.

In the formulation of these models, consideration was given to the incorporation of lower and upper boundaries. The upper boundary, named as the maximum equivalent workstation (WS) speed, was represented by the value 2000, while the lower boundary, named the minimum equivalent WS speed, was represented by the value 1000. The LR model was used because the conveyor driver over the speed range has a linear behavior. Thus, the LR model best shows the behavior of the conveyor belt speed – energy consumption in the declared speed range.

The total power consumption change equation for a process implemented on manufacturing lines will be the sum of the power consumption corresponding to each workstation:

$$y=\sum_{i}^{1} z(i) \quad i=1, 2, \dots 5 \quad (3)$$

4. Advanced PSO optimization algorithm applied to optimize energy consumption

PSO is an optimization algorithm inspired by the collective behaviour of birds or fish. This algorithm finds extensive applications in addressing single-objective optimization problems and multi-objective optimization problems across diverse fields of endeavour. PSO provides an effective approach to exploring the solutions space and finding optimal solutions or in cases where exact optimality is challenging, it excels in finding solutions that are close to optimal [12].

The PSO algorithm aims to find an optimal solution to the problem in the following form:

$$f(a_1, a_2, a_3, a_4, ..., a_n) = f(A) \quad (4)$$

Where: a_n - variable in state space and $f(A)$ - global minimum/maximum optimization value.

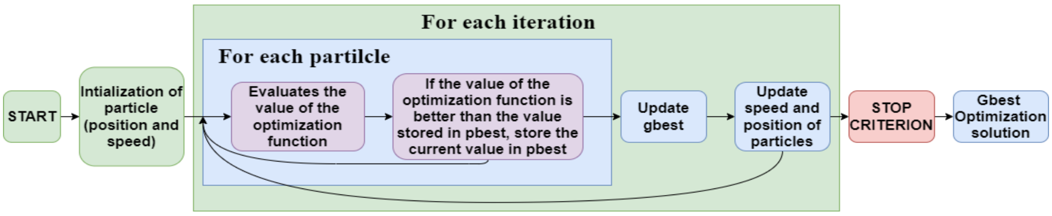


Fig. 4. Schematic representation of the PSO iterative algorithm

The fundamental concept underlying PSO entails creating a population of particles that move through state space to find the best solution. Each particle represents a candidate solution with its own position and speed in the search space. The particle's position indicates a potential solution, while its velocity governs both the direction and the extent of its movement. The subsequent position of the particle is determined by the following relationship:

$$a_i^{k+1} = a_i^k + v_i^{k+1} \tag{5}$$

Where: v_i^k - the velocity vector specific to each particle i, it reflects both social behavior and personal learning behavior and k -current iteration value.

The objective is to iteratively fine-tune particle positions and velocities to progressively converge towards the optimal solution (Fig. 4).

The motion of particles in PSO algorithm is influenced by their best position (P_{best}) but also by their optimal overall position (G_{best}). Particles adjust their speeds by considering these positions and their interactions with one another, with the objective of advancing towards superior solutions. This interaction is governed by social and cognitive components, which control the balance between exploration and exploitation [11]. During each iteration, particles update their velocities and positions based on the following equations:

$$V_{i,j}^{k+1} = \omega * V_{i,j}^k + c_1 * r_{1,j}^k * \left[Pbest - a_{i,j}^k\right] + c_2 * r_{2,j}^k * \left[Gbest - a_{i,j}^k\right] \tag{6}$$

In these equations, V^k and A^k represent the velocity and position of a particle at time t, respectively. The parameter "ω" is the weight of inertia controlling the impact of the particle's previous velocity and controls the balance between exploration and exploitation, c_1 and c_2 determine the influence of the particle's best-known position (P_{best}) and the best-known position across all particles (G_{best}). The "P_{best}" signifies the best value achieved by a particle, while "G_{best}" signifies the best value obtained by any particle in the entire swarm. Furthermore, "r_1^k"and "r_2^k" generate a random number ranging between 0 and 1.

The calculation P_{best} assumes the following formula:

$$P_{best,i}^{k}=\begin{cases}P_{best,i}^{k} & if\,(\,f\,(a_i^{k+1}>P_{best,i}^{k})\\ a_i^{k+1} & if\,(\,f\,(a_i^{k+1}<P_{best,i}^{k})\end{cases} \tag{7}$$

The calculation G_{best} involves the use of the following formula:

$$G_{best,i}=\min\left\{P_{best,i}^{k}\right\};\ i=1,2\ldots,n \text{ and } n>1 \tag{8}$$

Another aspect that should be mentioned, is that in the algorithm was applied with dynamic inertia weight, more precisely the parameter ω dynamically changes its value according to the equation:

$$\omega^{k}=\omega_{\max}-\left(\frac{\omega_{\max}-\omega_{\min}}{k_{\max}}\right)*t \qquad \omega_{\max}>\omega_{\min} \tag{9}$$

Where: $\omega_{\max},\omega_{\min}$ - interval of change of inertia weights and $k_{\max}$ - maximum number of iterations.

Additionally, the condition proposed by Van den Bergh and Engelbrecht, Trelea [13] was applied to reduce the chance of divergent conduct, formulated as follows:

$$\omega>\frac{c_1+c_2}{2}-1 \tag{10}$$

The process continues until an algorithm break condition is met, such as reaching a maximum number of iterations or obtaining a satisfactory solution. The best solution identified during iterations is considered the optimal or near-optimal solution to the optimization problem.

5. Advanced PSO algorithm versus PSO approach. Discussion

The PSO optimization algorithms were tested on an HP workstation featuring the following hardware configuration: dual Intel Xeon E5-2650 processors with a frequency of 2.0 GHz and eight cores, 128 GB of RAM, an Nvidia Quadro K4200 graphics card with 4 GB of memory, and a storage capacity of 1 TB.

The optimization problem proposes adjusting the equivalent conveyor speeds with the aim of identifying the most advantageous speeds in terms of both the overall energy consumption of the manufacturing system and the reduced execution time of the PSO optimization algorithm. This approach was designed to facilitate rapid optimization responses. Due to the multiple assembly capabilities, the scenario in which the system assembles a product in simple configuration was considered, i.e. the product passes through the first five workstations (Fig. 5).

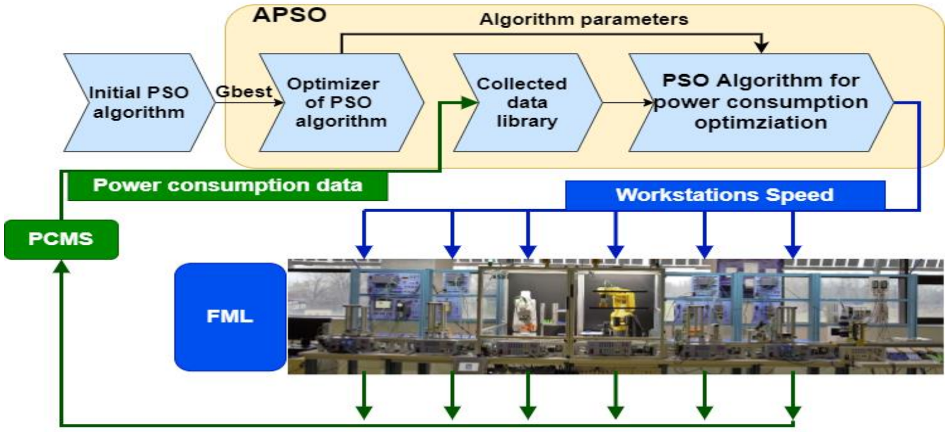


Fig. 5. Applied multi-objective algorithm - APSO implemented for optimization of assembly/disassembly process of FML.

The procedure addresses the issue of optimizing energy consumption as follows: starting from the data collected with the monitoring system, mathematical models related to the energy consumption of workstations are made by applying LR on them, and finally the PSO algorithm will be applied on these mathematical models. Thus, for each WS, cycle times, variation of cycle time depending on conveyors speed, local delay time in the station and variation of station energy consumption were considered. Furthermore, constraints are imposed on the station's delay time, which must always be greater than or equal to zero, as well as on the state space.

Optimizing energy consumption solves the problems related to obtaining minimum energy consumption, but also effectively eliminates the risk of bottleneck production processes. To expedite the optimization process, another PSO algorithm has been integrated to optimize the parameters of the energy consumption optimization algorithm. Through this optimization procedure, it becomes possible to achieve minimal power consumption optimization time while still ensuring the reduction of power consumption to its minimum level.

The initial PSO algorithm configuration includes the following specifications: it involves six input variables, uses for each variable two vectors with upper limit (value 2000) and lower limit (value 1000), an initial population comprising 50 individuals and allows a maximum number of 1000 iterations. Thus, the PSO algorithm applied to the working speeds of stations aims to find the global minimum:

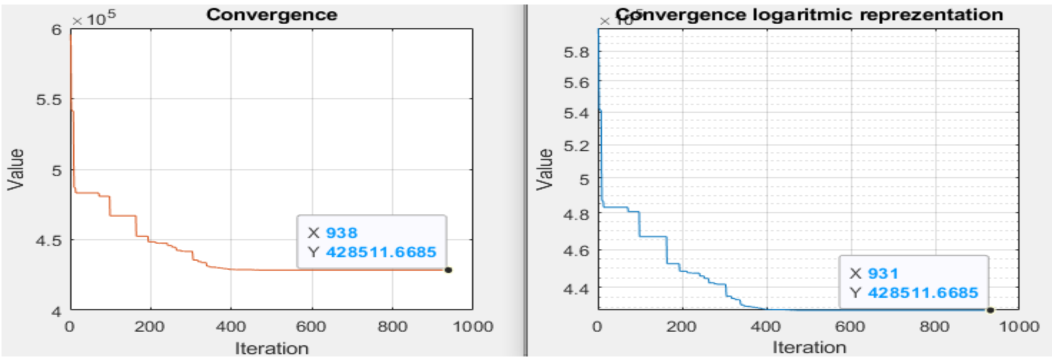
$$f(v_1, v_2, v_3, v_4, v_5) = f(V) \tag{11}$$

Where: v_1, v_2, v_3, v_4, v_5 - values of speeds for which the minimum power consumption is achieved and $f(V)$ - minimum value of power consumption. The constraints of the algorithm refer to the following formula:

$$1000 < v_l < 2000 \quad l = 1,2,3,4,5; \tag{12}$$

In addition, conditions have been applied to the delay times for each assembly station as follows:

$$tsi_l = 0 \quad if\,(tsi_l < 0) \quad l = 1,2,3,4,5; \tag{13}$$



a) b)

Fig. 6. a) PSO algorithm convergence chart on cartesian scale and b) PSO algorithm convergence chart on logarithmic scale.

The algorithm successfully converges towards the optimal solution, which corresponds to a minimum power consumption value (Fig. 6) of 0.11903 kW*h (428511.66 W*s). The speed values that allow achieving this energy performance are: v_1=1000, v_2=2000, v_3=1564, v_4=2000, v_5=1000 and v_6=2000.The execution time of the algorithm in MATLAB is 3.755 s. The delay times at assembly stations corresponding to these speed values are ti_1= 6.0722s, ti_2= 0s, ti_3 = 67.2370s, ti_4 = 0s, and ti_5 = 0s.

Substantial enhancements were introduced to the initial PSO algorithm. These modifications include: an algorithm stop criterion largely dependent on changes to the global minimum, an optimization algorithm was applied to the initial algorithm, hereinafter referred to as optimizer, and state space constraints were applied by introducing upper limits and lower limits.

The advanced PSO (APSO) is a PSO-based algorithm that adjusts various parameters, including the number of particles, the number of iterations, algorithm inertia coefficients, degrees of individual and social confidence, and the number of loop repetitions until optimization function's value remains unchanged.

The optimizer assumes that Go_{best} is:

$$Go_{best} = g(n, k, \omega_{Max}, \omega_{Min}, c_1, c_2, rep)\,|_{\min\ \ execution\ \ time} = g(B) \tag{14}$$

Where: Go_{best} - the overall minimum execution time of the algorithm; n - number of particle; $g(B)$ - the overall minimum/maximum value of the algorithm's execution time; $\omega_{Max}, \omega_{Min}$, - algorithm inertia coefficients; - c_1, c_2, degrees of individual and social confidence, and - the rep number of repetitions of the overall minimum value for the maximum algorithm that stops the execution of the algorithm.

In order to provide viable results, the algorithm must have integrated a condition that excludes all values obtained for tests performed on the parameters of the PSO algorithm that do not reach at least the value G_{best} obtained after application without the optimizer. The data exclusion criterion is:

$$time = \begin{cases} 1000 & if\,(G_{best} < Pt^{t}_{best}) \\ time_{calc} & if\,(G_{best} < Pt^{t}_{best}) \end{cases} \tag{15}$$

Where: G_{best} - the minimum obtained by applying the PSO algorithm without the optimizer, Pt^{t}_{best} - the overall minimum time at iteration t, $time$ - time calculated within the Optimizer, and - $time_{calc}$ the overall minimum value of the electrical power calculated at the iteration t of the optimizer Pt^{t}_{best}

Besides, for each optimized variable, constraints were defined in the form of maximum and minimum limits of the space states.

$$10 < n < 60 \tag{16}$$

$$0.01 < \omega_{\min} < 0.29 \tag{17}$$

$$400 < Iter < 1000 \tag{18}$$

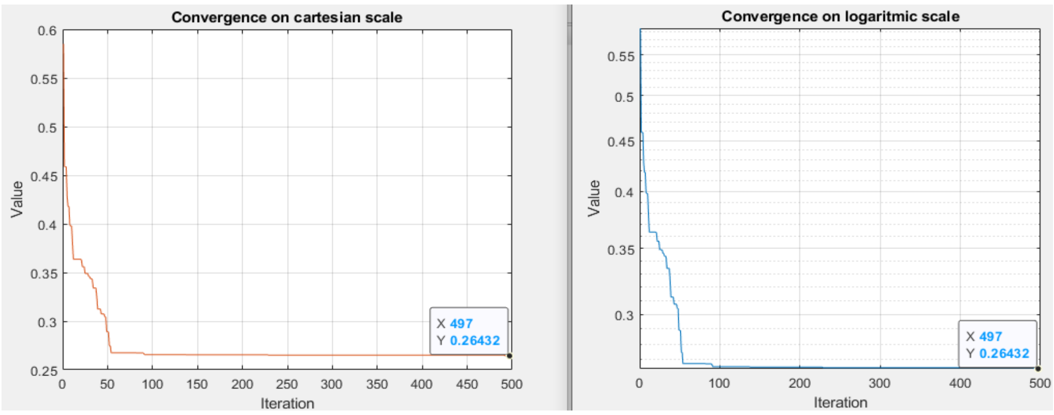
$$1 < c_1 < 3 \tag{19}$$

$$0.3 < \omega_{\max} < 0.9 \tag{20}$$

$$1 < c_2 < 3 \tag{21}$$

$$5 < rep < 10 \tag{22}$$

Consequently, after optimization, the APSO algorithm it is adapted to optimize FML power consumption. The integration of the optimizer into the initial algorithm has yielded substantial improvements in execution time. Therefore, the algorithm has been improved in terms of optimization time more than ten times, starting from an execution time of 3.755 s to an execution time of 0.262 s. Furthermore, the convergence charts demonstrate that optimization performance has not changed (Fig. 7).



a) b)

Fig. 7. Optimizer execution time with APSO. The convergence graph on cartesian scale (a) and logarithmic scale (b)

The optimal numerical values that yield the best performance in terms of execution time for the PSO algorithm are as follows: 10 particles, maximum number of iterations is 402, maximum inertia coefficient is 0.657, minimum inertia coefficient is 0.039, individual confidence is 1.031, social confidence is 2.712 and maximum number of repetitions is 6 (Table 2). These parameter values have been adjusted by optimizer to significantly enhance the algorithm's execution efficiency while maintaining its optimization efficiency.

Tabel 2.

Features of applied algorithms

Algorithm	**n**	**Iter**	ω_{min}	ω_{max}	**c1**	**c2**	**rep**	**Time (s)**
PSO	50	1000	0.9	0.2	1.9	1.9	6	3.755
APSO	10	402	0.657	0.037	1.031	2.712	6	0.262

6. Conclusions

The present work presented the results obtained for the application of a multi-objective algorithm on the energy consumption data to identify the work speeds of the FML workstations but also to identify the tuning parameters of the intensified PSO algorithm for optimal results.

In summary, an advanced APSO algorithm is conceptually presented that performs an intensified optimization, in the sense of increasing the accuracy and efficiency of calculations, versus the PSO algorithm. The objective of the research was to identify a high-performance technique for optimizing energy consumption in a flexible manufacturing process. The APSO algorithm was implemented and tested on an FML system, equipped with local (workstation) and total energy consumption monitoring systems. Following the implementation and testing of APSO on FML, it resulted that energy consumption monitoring provides data at sample intervals of 0.32s, while APSO provides the iterative optimization solution, every 0.262s. The time window corresponding to an APSO interaction is subordinated to that corresponding to a data acquisition cycle, which guarantees the efficiency of the APSO.

The optimization procedure proposed and demonstrated in this article allows reducing energy consumption on the FML in the most convenient time. This methodology can be applied for any manufacturing line that asselmby products in manufacturing flow to obtain minimum power consumption.

R E F E R E N C E S

[1]. *Hantash, Neda & Khatib, Tamer & Khammash, Maher,* "An Improved Particle Swarm Optimization Algorithm for Optimal Allocation of Distributed Generation Units in Radial Power Systems", Applied Computational Intelligence and Soft Computing, 2020, pp. 1-8, doi: 10.1155/2020/8824988;

[2]. *Cristina Nichiforov; Iulia Stamatescu; Ioana Făgărăşan; Grigore Stamatescu,* "Energy consumption forecasting using ARIMA and neural network models", 5th International Symposium on Electrical and Electronics Engineering (ISEEE), 2017, pp. 1-4, doi: 10.1109/ISEEE.2017.8170657.

[3]. *Victoria Jayne Mawson, Ben Richard Hughes*, "The development of modelling tools to improve energy efficiency in manufacturing processes and system, Journal of Manufacturing Systems", 2019, vol. 51, pp. 95-105;

[4]. *Alessandro Cannata, Stamatis Karnouskos, Marco Taisch*, "Energy efficiency driven process analysis and optimization in discrete manufacturing", Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE, pp. 4449 – 4454**;**

[5]. *Kazuhiro Ohara, Masashi Tsugeno, Hiroyuki Imanari, Yasuyuki Sakiyama, Kazutoshi Kitagoh, Jun Yanagimoto*, "Process optimization for the manufacturing of sheets with estimated balance between product quality and energy consumption", Manufacturing Technology, 2014, vol. 63, pp 257-260;

[6]. *Abdul Salam Shah, Haidawati Nasir, Muhammad Fayaz, Adidah Lajis, Asadullah Shah,* "A Review on Energy Consumption Optimization Techniques in IoT Based Smart Building Environment", Information, 2019, vol. 10, pp. 108;

[7]. *Richard Opokua, George Y. Obenga, Louis K. Osei, John P. Kizito,* "Optimization of industrial energy consumption for sustainability using time-series regression and gradient descent algorithm based on historical electricity consumption data", Sustainability Analytics and Modeling, 2022, vol. 2, pp. 100004;

[8]. *Octavian Duca, Eugenia Minca, Adrian Filipescu, Daniela Cernega, Razvan Solea, Claudiu Bidica*, "Event-Based PID Control of a Flexible Manufacturing Process", Inventions, vol. 7, no. 4, 2022, pp. 86;

[9]. *Octavian Duca; Valentin Gurgu, Eugenia Minca, Adrian Filipescu; Florin Dragomir; Otilia Dragomir*, "Optimal control of the complete assembly/disassembly cycle for a mechatronics line prototype", 23rd International Conference on System Theory, Control and Computing (ICSTCC), 2019, pp. 620-625;

[10]. *G. Nicolae, H. Cucu, C. Burileanu, A. Buzo, C. Feuerbaum, G. Pelz* "Automatic design optimization of microelectronic power switches", Scientific Bulletin of University Politehnica Bucharest, 2023, vol. 85, no. 1;

[11]. *Imen Harbaoui Dridi*, *Essia Ben Alaïa*, *Pierre Borne*, *Hanen Bouchriha*, “Optimization of m-MDPDPTW Using the Continuous and Discrete PSO”, Studies in Informatics and Control,vol. 2019, 28, no. 3, pp. 289-297;
[12]. *Milić Vukojičić, Mladen Veinović,* “Optimization of Multimodal Trait Prediction Using Particle Swarm Optimization”, Studies in Informatics and Control, 2022, vol. 31, no. 4, pp. 25-34;
[13] *Andries P. Engelbrecht*, “Computational Intelligence: An Introduction” John Wiley and Sons, 2007; ch. 16, pp. 289-358.
[14]. *K.C Bhosale, P.J. Pawar*, Material Flow Optimization of Flexible Manufacturing System using Real Coded Genetic Algorithm (RCGA), Materials Today: Proceedings, vol. 5, no. 2, 2018, pp. 7160-7167;
[15]. *Keiji Kamei*, *Takafumi Arai*, Optimization for Line of Cars Manufacturing Plant using Constrained Genetic Algorithm, Journal of Robotics Networking and Artificial Life, vol. 5, no. 2, 2018, pp. 131-134;