Improved Multi-objective Genetic Algorithm Used to Optimizing Power Consumption of an Integrated System for Flexible Manufacturing

Marius-Adrian PĂUN[2,4]*, Henri-George COANDĂ[3], Eugenia MINCĂ[3], Sergiu Stelian ILIESCU[1,2], Octavian Gabriel DUCA[4], Grigore STAMATESCU[2]

[1] Technical Sciences Academy of Romania - MO-ASTR, 26 Dacia Avenue, 030167 Bucharest, Romania
iliescu.shiva@gmail.com

[2] Faculty of Automatic Control and Computer Science, National University of Science and Technology Politehnica Bucharest, 313 Splaiul Independenţei, 060042, Bucharest, Romania
paun_marius_2009@yahoo.com (*Corresponding author*), grigore.stamatescu@upb.ro

[3] Faculty of Electrical Engineering, Electronics, and Information Technology, Valahia University of Targoviste, 13 Aleea Sinaia St., 130004, Targoviste, Romania
coanda_henri@yahoo.com, eugenia.minca@gmail.com

[4] The Scientific and Technological Multidisciplinary Research Institute, Valahia University of Targoviste, 13 Aleea Sinaia St., 130004, Targoviste, Romania
octavian_duca@yahoo.com

Abstract: The efficient management of energy consumption is an essential concern in the manufacturing industry, with far-reaching implications for both financial management and environmental sustainability. This paper proposes a new approach regarding the conceptualization and implementation of techniques for optimizing energy consumption in a production flow. In the first stage, an energy consumption monitoring system was developed, which is capable of collecting the energy consumption data for different production scenarios. The collected data represented the basis for evaluating and testing an optimization algorithm called the improved genetic algorithm (IGA), which is conceptually subordinated to the structure of the standard genetic algorithm (GA). The improved multi-objective genetic algorithm featured a high performance in terms of execution time and optimization of energy consumption. Thus, in the framework of the IGA algorithm, a layered approach to the optimization process was proposed by successively employing two genetic algorithms in the MATLAB programming environment. The first genetic algorithm identified the value of the minimum energy consumption, and the second GA adjusted the parameters of the first GA iteratively, in order to obtain the minimum consumption. Comparing the results obtained by employing the IGA algorithm with those obtained for the Non-dominated Sorting Genetic Algorithm (NSGA-II) in terms of real-time execution time, it can be noticed that a significant improvement was achieved from 25.7 s for the standard NSGA-II to 0.0527 for the IGA without any change in the performance of IGA with regard to the minimization of power consumption. By harnessing the inherent capabilities of the GA, the aim of this paper is to increase the energy efficiency for the analysed production system, thereby contributing both to cost savings and to reducing the environmental impact of manufacturing processes.

Keywords: Power consumption optimization, Power monitoring, Genetic algorithm (GA), Industrial production line.

1. Introduction

The concept of Industry 5.0 signifies a major shift in perspective, as it encompasses multiple dimensions beyond production efficiency. This paradigm recognizes the importance of integrating various goals, such as efficiency, sustainability, mitigating the environmental impacts of industrial processes, and enhancing the well-being of employees. By embracing these diverse directions, Industry 5.0 aims to foster a holistic and responsible approach towards industrial practices with a focus on optimizing the overall quality of life (Adel, 2022).

Within this context, the optimization of energy consumption in manufacturing systems necessitates the implementation of suitable optimization algorithms, the integration of advanced technologies for energy storage and recovery, the utilization of devices compatible with Industrial Internet of Things (IIoT), and the deployment of systems for data collection and analysis (Nicolae, Necula & Carutasiu, 2023)). These technological advancements enable the real-time gathering of data from manufacturing lines, which can subsequently be analysed to identify and improve energy-inefficient processes (Alexandru et al., 2023). Furthermore, based on the amassed data, predictive models can be developed with exceptional precision to anticipate the energy requirements accurately (Knežević, Blagojević & Ranković, 2023).

The optimization of production processes, particularly in cases where parallel production is feasible, represents a complex challenge, particularly in tackling the optimization issues pertaining to operations (job shop). This complexity arises due to the involvement of equipment with varying cycle times and constraints in these systems. Furthermore, the

https://doi.org/10.24846/v33i1y202403

optimization problem becomes more intricate when considering the presence of server-based processing (cloud computing). Addressing these challenges requires a comprehensive and systematic approach that incorporates advanced techniques and methodologies to optimize production processes and achieve efficient resource allocation while considering the dynamics of parallel operations, variable equipment constraints, and the incorporation of cloud computing resources (Hassan, et al. 2015).

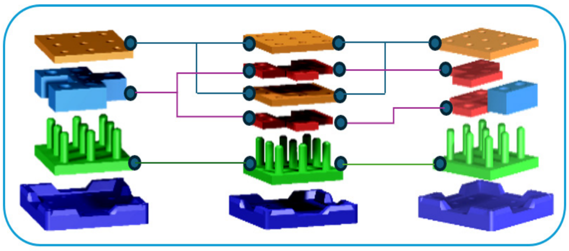
This paper focuses on optimizing the energy consumption of a manufacturing line through the application of the IGA on mathematical models representing the variation in energy consumption of manufacturing stations. To ensure a clear and logical progression of ideas, this paper is structured as follows. Section 2 presents the industrial production system in which the optimization will be achieved. Section 3 sets forth the GA-based multi-objective approach. Section 4 presents the proposed IGA-based approach by describing the analysed production scenario and also the structure of the implemented procedure for optimizing energy consumption. Section 5 presents the results obtained for applying IGA to optimize energy consumption on the manufacturing line. Finally, Section 6 summarizes the main findings of the research, reiterating the significance of energy consumption optimization in manufacturing lines.

2. Description of the Industrial Production System

This article addresses the analysis and improvement of manufacturing processes by leveraging theoretical optimization concepts applied to equipment within a manufacturing system. The specific objective of this article is to implement an improved multi-objective GA on a flexible manufacturing system with the aim of optimizing energy consumption in a manufacturing scenario. This optimization aims to streamline the production flow for making a complex product on the production line, considering the unique requirements and constraints associated with the production scenario.

2.1 Flexible Manufacturing Line (FML)

The flexible manufacturing system incorporates a network of workstations, wherein each workstation possesses distinct characteristics tailored to the specific nature of the tasks performed on the workpieces. Within the manufacturing system, the products can be categorized based on their architecture into three main types: simple products, complex products, and hybrid products (Figure 1). These product variations are assembled on the production line and can be further configured in numerous combinations to meet customer demands. Consequently, in addition to the conventional sequential production aspect, the workstations also integrate a flexible component, which necessitates the utilization of equipment capable of producing a wide range of configurations (Abdullah, Humaidi & Shahrom, 2020).



a) b) c)

Figure 1. The structure of a simple product (a), complex product (b), and hybrid product (c)

The incorporation of workstations with unique task-specific characteristics, along with the ability to handle different product architectures and configurations, highlights the adaptability and versatility of the flexible manufacturing system (Figure 2). By acknowledging the presence of both sequential and flexible components within the manufacturing system, the significance of implementing equipment capable of producing various product configurations becomes obvious. Consequently, this approach serves as a base for further exploring ways to optimize the implementation of production flows and enhance the overall efficiency and effectiveness of the flexible manufacturing system (Filipescu et al., 2020).

To ensure the comprehensive completion of a product, irrespective of its architecture, a specific manufacturing flow is employed within the system. For the manufacturing of a simple product two possible flows are considered as follows: WS1 → WS2 → WS3 → WS4 → WS5 → WS4 → WS5 → WS6 or WS3 → SCARA Robot (SR) → WS6.

The operations involved in first flow presented include releasing the transport tray onto the conveyor, assembling the basal part, assembling small parts, assembling the upper part, returning the part to WS3, assembling an additional layer of small parts, adding another top part, and finally compacting the entire product (Duca et al. 2022).

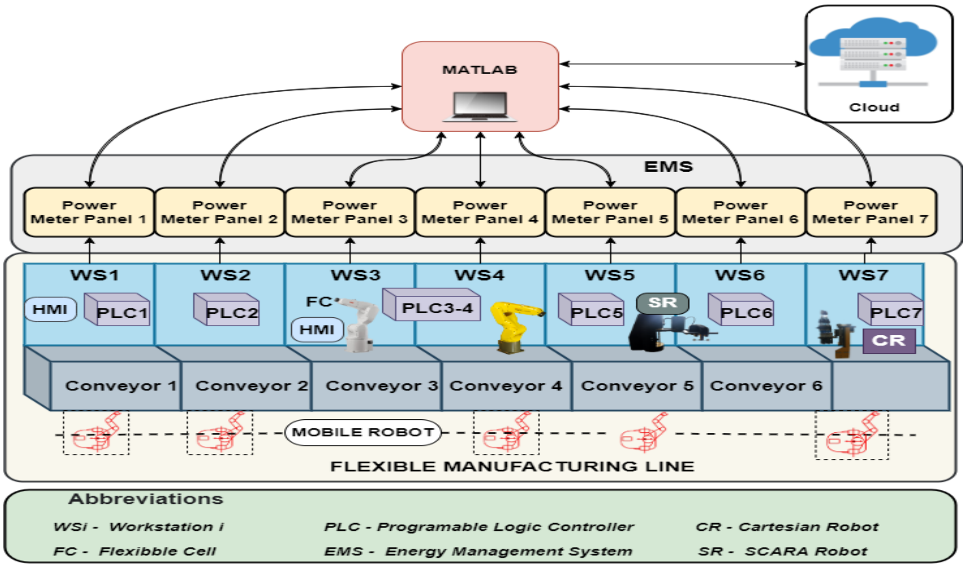


Figure 2. The modular hierarchical structure of the FML equipped with the centralized real-time energy consumption monitoring system (EMS), and with the subsystems Power Meter Panel(i), i = 1,..7, connected to WS(i), i = 1, ..7, respectively

The manufacturing line implements manufacturing flows of significant complexity, allowing to produce products in both random order and according to a predefined algorithm. Within this context, the potential for process optimization utilizing multi-objective GA is noted, specifically regarding the reduction of energy consumption through the optimization and control of production processes. The operator defines the quantity and types of products to be manufactured. Subsequently, an optimized production algorithm is employed to establish the optimal order for manufacturing various products.

The energy performance of the production system is heavily influenced by the strategies employed for optimizing energy consumption and reducing the waiting time. Waiting time, both from the perspectives of productivity and energy consumption, results in losses.

The flexible production system under investigation in this study currently lacks an energy optimization algorithm. Consequently, it becomes imperative to assess the system's suitability for implementing such an energy optimization algorithm.

2.2 The Developed EMS Implemented on FML

The integration of an EMS (Figure 3) on the assembly and disassembly production line offers a significant potential to optimize energy usage, lower costs, and enhance overall energy efficiency.

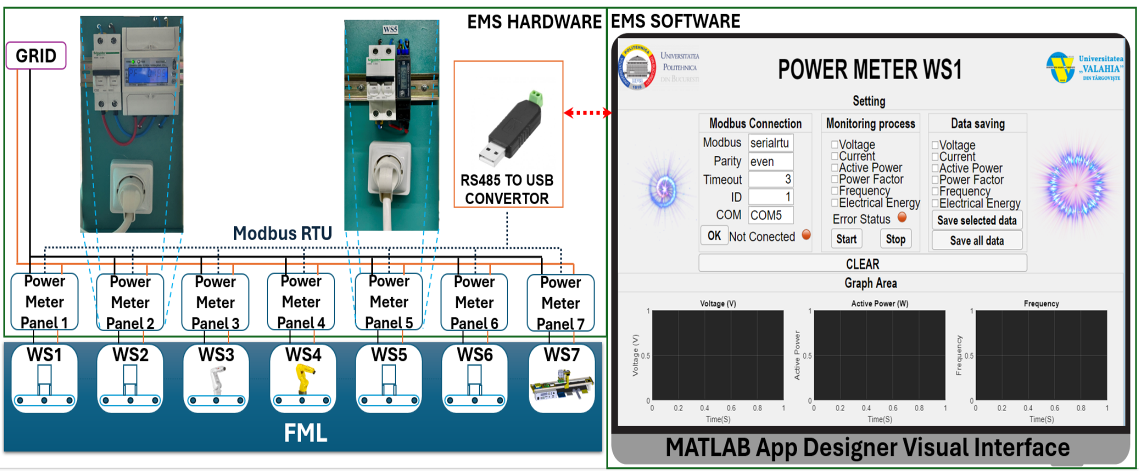


Figure 3. The real-time MATLAB Interface Application FML integrated in an EMS architecture for a flexible manufacturing line

This can be accomplished through the installation of sensors and measuring equipment that record real-time energy consumption data, coupled with the implementation of software capable of analysing and interpreting this data (Limpraptono et al., 2021).

An EMS can help identify sources of inefficient energy consumption and provide recommendations for optimization measures. For example, it can show which equipment consumes the most energy resources and suggest changes or optimizations to improve its efficiency (Sun, Lin & Meng, 2022).

An important aspect in the process of optimizing the energy consumption of manufacturing systems is the introduction of an EMS adapted to the physical architecture of the system.

The EMS system proposed in this article for monitoring energy consumption consists of a hardware part and a software part (Figure 2). The hardware consists of seven smart energy meters mounted on a Modbus RTU bus connected to a laptop (Figure 3). On the other hand, the software application developed in MATLAB allows the representation and saving of real-time energy consumption data from the manufacturing line. The monitoring system consists of the following equipment: seven measuring meters with communication via Modbus RTU protocol, control panel, Modbus RTU – USB converter and a computer that allows data acquisition. To achieve this, it was necessary to design a specialized program in the MATLAB programming environment as well as an interface through which to display all data.

In this regard, an interface was created in MATLAB App Designer, which allows monitoring workstations and making real-time graphs. This function is especially practical when a large amount of data needs to be analysed. This advanced functionality enables individual monitoring of the energy consumption at each workstation, providing valuable insights into the energy usage patterns (Figure 3). This individual workstation monitoring offers a comprehensive view of the energy consumption distribution within the manufacturing line.

3. GA Multi-objective Approach

GA is an optimization method used for solving problems that require an optimal solution with or without constraints (Serban & Carp, 2017). By simulating the process of natural selection, GA explores potential solutions through its iterative processes and steadily refines them over successive generations. This optimization approach by implementing the genetic variation, crossover, and mutation can generate new candidate solutions and iteratively converge towards optimal or near-optimal solutions.

GA commonly employs three types of rules in each iteration to generate new populations:

1. *selection rules:* selection rules in GA involve a stochastic process wherein individuals are chosen as potential parents based on their individual scores or fitness values;

2. *crossover rules:* crossover rules govern the combination of genetic material from two selected parents to generate offspring, known as children. Through crossover operations, genetic information is exchanged and recombined, leading to new individuals that inherit traits from both parents;

3. *mutation rules:* mutation rules introduce variability and diversity into the population by randomly altering the genetic material of selected individuals, typically parents, to generate new genetic variations for future generations.

3.1 Structure of Multi Objective Problems

Multi-objective optimization algorithms are meant to solve problems that involve finding at least two or more goals (minimum/maximum) associated with optimization functions. These optimization functions usually have different constraints of either equality or inequality. Therefore, multi-objective optimisation problems can be formulated as follows:

$$\begin{array}{ll} f_n(x) & n = 1,2,3,.....,N, \\ g_t(x) \le 0 & t = 1,2,3,......,T, \\ h_k = & k = 1,2,3,......,K, \\ x_l \le & l = 1,2,3,.......,L, \end{array} \tag{1}$$

where $f_n(x)$ represents objective functions, $g_t(x)$ inequality constraint functions, h_k equality constraints and x_l inequality constraints.

The solution for implementing a multi-objective algorithm is of the form $x \in R^l$ where l is the vector of decision variables $x = (x_1,x_2,x_3,...,x_l)^T$. Solutions that satisfy boundary constraints constitute the space of feasible states $S \subset R^l$. Thus, the optimization problem has an N dimension hereinafter referred to as the lens space in which $Z \subset R^N$. For each solution x in the decision variable space there is an associated point $z = (z_1,z_2,z_3,...,z_N)^T$. Thus, a solution is both a vector and a point corresponding to the objective vector. Optimal solutions in multi-objective optimization are dominant terms.

3.2 Multi-objective Algorithms

Multi-objective optimization algorithms are designed to solve problems with multiple objectives (Djebbar & Boudia, 2022). Unlike single-objective optimization, where only one optimal solution is sought, multi-objective optimization aims to find a set of solutions that represent a compromise between different objectives (Delorme, Battaïa & Dolgui, 2014).

Non-dominated genetic sorting algorithm (NSGA-II) is a popular evolutionary algorithm for multicriteria optimization. It uses a combination of genetic operators such as selection, crossover, and mutation to evolve a population of candidate solutions (Verma, Pant & Snasel, 2021). The algorithm applies non-dominated sorting to determine the Pareto dominance for solutions, which classifies them according to their non-dominated status. NSGA-II offers a diverse set of solutions covering the pareto-optimal front, allowing decision-makers to choose the most suitable solution according to their preferences (Ma et al., 2023).

Strength Pareto Evolutionary Algorithm (SPEA2) is another evolutionary algorithm widely used in multi-objective optimization. It uses a fitness function assignment scheme that combines both goal-based fitness and density estimation to promote diversity among solutions. SPEA2 maintains an external archive of non-dominated solutions and uses a selection mechanism to guide the evolution process. It aims to find a well-distributed and representative set of optimal Pareto solutions.

Multi-objective Particle Swarm Optimization (MOPSO) is a variant of the particle swarm optimization algorithm adapted for multi-goal optimization. It incorporates the concept of Pareto domination and guides the movement of particles in search space to converge towards the Pareto-optimal front. The MOPSO maintains the best personal position for each particle and updates it based on the dominance relationship with the most well-known positions found up to that point. The algorithm provides a good balance between exploration and exploitation, enabling the discovery of diverse and high-quality solutions (Yunus & Alsoufi 2020).

4. Improved Genetic Algorithm

The improvement of GA can be achieved through two approaches: modifying the algorithm itself or adjusting its parameters to align it with the specific addressed problem. This article focuses on a scenario where the GA adapts to the given problem by utilizing another optimization algorithm to fine-tune the parameters and achieve optimal results in the shortest possible time.

To accomplish these improvements, the initial implementation of the GA must be established. The GA functions by generating an initial population which progressively evolves across successive iterations thereby enabling the discovery of optimal solutions. In each iteration, a suitable selection of parent agents is made from the population, ensuring the production of offspring for future generations. The children generated in each generation contribute to the progressive convergence towards the optimal solution. A graphical representation of the algorithm's structure implemented on the production line is depicted in Figure 4.

This approach enables the GA to leverage the optimization capabilities of an additional algorithm, resulting in an improved performance and efficiency. By incorporating parameter tuning mechanisms, the GA can adapt its parameters iteratively to enhance the quality and speed of convergence, leading to the identification of optimal solutions in a shorter timeframe.

The solution takes the following structure:

$$X_{f_1} = (x_1, x_2, x_3, ..., x_7) \tag{2}$$

The candidate solutions in the optimization process represent the specific values of the working speeds for the station conveyors within the production system. The energy consumption for each production station is determined based on consumption change equations presented in vector Z. The fitness function assesses the overall energy consumption by summing all elements of vector Z.

By iteratively applying the genetic operators within the algorithm and evaluating the fitness function, the speeds that produce the minimum energy consumption values for the entire production system are determined.

Through this iterative optimization process, the GA effectively identifies the optimal working speeds for the station conveyors, thus minimizing energy consumption within the production system.

This approach fosters sustainable and efficient operations, reducing energy costs and improving the overall environmental performance of the manufacturing process.

To assess candidate solutions, an objective function is employed, taking the candidate solution as input data, and generating an output that represents its quality. By comparing the values obtained for each candidate solution, their quality can be determined.

Following the definition of the fitness function, various criteria have been established to halt the algorithm. These criteria imply a maximum number of iterations to prevent excessive computation and impose a time limit on the algorithm's execution.

In addition, the implemented GA algorithm incorporates the genetic selection operator known as Roulette Wheel Selection. This operator assigns a value to each parent on the wheel based on their fitness values derived from the fitness function. Parents with higher fitness values are more likely to be chosen as parents for the next generation.

The selection probability is calculated using equation (3):

$$P_i = \frac{f_i}{\sum_{j=1}^{N} f_i} \quad i = 1,2,3,4....7; \tag{3}$$

where f_i is the value of the fitness function for iteration i and $\sum_{j=1}^{N} f_i$ is the value of the sum of all fitness functions.

The functions tailored for optimization in this paper are:

$$f_1(v_1, v_2, v_3, v_4, v_5) = \sum_{t=1}^{7} z(t), \tag{4}$$

$$\begin{gathered} f_2(Max\_Iter, Max\_Pop, \beta, \gamma, \mu, \nu) = \\ time(GA_{f1}(Max\_Iter, Max\_Pop, \beta, \gamma, \mu, \nu)), \end{gathered} \tag{5}$$

where Max_Iter is the maximum number of iterations allowed for the GA algorithm, Max_Pop is the maximum permitted population, β is the coefficient of selection, γ the coefficient of the number of chromosomes, μ is the mutation coefficient and v is the importance of mutation.

Crossover, performed as a part of the optimization algorithm, involves generating new generations by combining the chromosomes of parents. The specific type of crossover employed in the algorithm is uniform crossover. This method allows for the exchange of genetic material between parents by uniformly selecting gene segments from each parent.

In addition to crossover, the optimization algorithm incorporates genetic mutation. This operation introduces random changes to chromosomes, typically affecting one or more genes. It is important to note that the applied algorithm maintains a low probability of mutation occurrence, ensuring that changes are introduced gradually and selectively.

To further enhance the optimization process, an additional optimization algorithm, referred to as the GA optimizer, is applied. The GA optimizer operates on the premise of optimizing parameters that impact the performance of the primary optimization algorithm. Essentially, it is another iteration of the GA that aims to fine-tune parameters to improve the execution time of the optimization algorithm. Figure 4 illustrates the overall structure of this optimizer. The algorithm generates candidate solutions that are represented in a specific form, capturing the desired variables or components needed to address the optimization problem at hand. The candidate solutions can be expressed as:

$$X_{f_2} = (Max\_Iter, Max\_Pop, \beta, \gamma, \mu, \nu) \tag{6}$$

The fitness function not only quantifies the execution time of the algorithm, but it also aims to minimize energy consumption to a level that satisfies the user's preferences. As a result, the algorithm's stop criterion considers the user's desired outcome alongside the requirement to achieve minimal energy consumption.

By accounting for user preferences in the fitness function, a comprehensive optimization approach is established. This user-centric perspective allows the algorithm to strike a balance between the execution time and energy consumption, ensuring that the resulting solution meets both the desired level

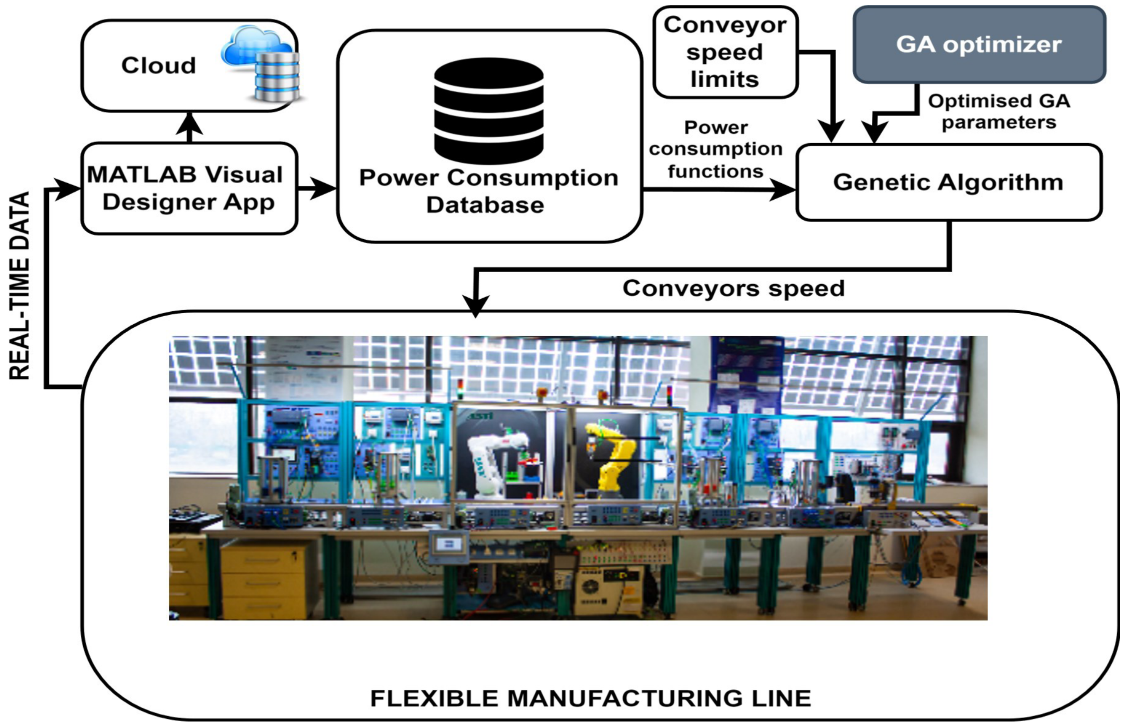


Figure 4. FML's real-time control structure dedicated to optimizing the energy consumption of workstations, adaptable to product manufacturing scenarios, from simple to complex typologies

of energy efficiency and the user's satisfaction. Striking an optimal balance between energy consumption and execution time is important in addressing the optimization problems effectively.

The use of IGA to optimize the manufacturing flow for assembling the complex product involves in the first phase the development of the fitness function associated with this process. This article, from among all production scenarios, considered the scenario in which the complex product is assembled through the following manufacturing flow: WS1 → WS2 → WS3 → WS4 → WS5 → SR → WS4 → WS5.

The functions that make up the vector $\vec{Z}$ are obtained by applying linear regression to the data monitored with EMS:

$$\vec{Z} = \begin{pmatrix} (-0.0008*v_1+8.41)*(v_1*0.0114+69.304)+tsi_1*43.92 \\ (-0.0035*v_2+20.2)*(v_2*0.0481+70.804)+tsi_2*98.39 \\ (-0.0052*v_3+21.35)*(v_3*0.0126+115.57)+tsi_3*178.83 \\ (-0.0025*v_4+85.45)*(v_4*0.0021+189.36)+tsi_4*131.94 \\ (-0.0034*v_5+24.05)*(v_5*0.0080+30.561)+tsi_5*25.12 \\ 133.76*48.22+tsi_r*48.08 \\ (-0.0025*v_6+85.45)*(v_6*0.0021+189.36)+tsi_4*131.94 \\ (-0.0034*v_7+24.05)*(v_7*0.0080+30.561)+tsi_5*25.11 \end{pmatrix}, \quad (7)$$

where v_i, $i=\overline{1,7}$ is the conveyor speed, while tsi_k, $k=\overline{1,5}$ and tsi_r are variables that model the delay time in relation to the production time of the next workstation.

The vector $\vec{Z}$ contains the functions for varying the energy consumption in relation to the speed of the conveyors but also to the working speed of the SR. It should be noted that the equivalent speed was considered to have two command limits: an upper limit equal to 2000 and a lower limit equal to 1000. In addition to this, the delay time was considered, this time was calculated as the difference between the production time of the next station and the production time of the current station. If the difference in production times is negative, the delay time takes the value 0. The fitness function implemented in MATLAB is performed in accordance with the vector $\vec{Z} \in R^7$. Thus, to achieve this fitness function, the sum of all elements of the vector $\vec{Z}$ was calculated as follows:

$$F_f = \sum_{i=1}^{u} z(i) \quad i = 1,2,3,4,.......,u \quad (8)$$

where $z(i)$ the value of power consumed for the passage of the product through the station i, i is the variable characterizing the equations denoted with regard to the vector $\vec{Z}$ and u is the total number of functions associated with the vector $\vec{Z}$.

During the implementation of the algorithm in MATLAB, an important aspect considered was ensuring that the difference between the working time of a station, tsi_i, and the working time of the next station is always greater than or equal to zero. In cases where this difference is less than zero, it is set to zero to maintain consistency and avoid negative delays. This condition promotes an efficient workflow management within the manufacturing system.

To calculate the delay time, the difference between the production time of the current station and the production time of the next station is determined. This calculation enables the identification of potential delays in the production flow, aiding in the analysis and optimization of the system. By considering the delay time, manufacturing processes can be coordinated to minimize waiting times and optimize production efficiency (9):

$$\begin{aligned}
&tsi_1=(-0.0035*v_2+20.2)-(-0.0008*v_1+8.42);\\
&tsi_2=(-0.0052*v_3+21.35)-(-0.0035*v_2+20.21);\\
&tsi_3=(-0.0025*v_4+85.45)-(-0.0052*v_3+21.35);\\
&tsi_4=(-0.0034*v_5+24.06)-(-0.0025*v_4+85.46);\\
&tsi_5=(-0.0016*1500+7.22)-(-0.0034*v_5+24.06);\\
&tsi_r=(-0.0025*v_4+85.45);
\end{aligned} \tag{9}$$

If delay time is less than zero, then automatically the waiting time in a certain station will take the value zero.

$$\begin{aligned}
&if\,(tsi_k<0)\quad tsi_k=0,\qquad k=1,2,3,4,5,\\
&if\,(tsi_r<0)\quad tsi_r=0,
\end{aligned} \tag{10}$$

where k is the variable that allows the selection of delay time for workstations.

5. Results and Discussion

The optimization of production processes by applying optimization techniques, in the analysed scenario, has the role of reducing energy consumption and at the same time the execution time for the optimization algorithm. Thus, the two objectives can be achieved by applying the IGA algorithm to the analysed collection of consumption data, by identifying in the first phase the optimal value of energy consumption, and then implementing an optimizer on the existing algorithm to obtain the best execution time adapted to the optimization of the assembly system.

To accommodate the increased processing demands resulting from the larger number of variables, a Dell computer with a high-performance configuration was employed for this analysis. The computer utilized for this analysis exhibited the following configuration: two octa-core Intel Xeon processors, each operating at 2 GHz, provided substantial processing power, a video card equipped with 4 GB of NVIDIA KP4200 memory facilitated enhanced graphics processing capabilities and the computer was equipped with an internal solid-state drive (SSD) memory with a capacity of 1 TB.

5.1 Optimization of Power Consumption Using NSGA-II Algorithm

The originally applied GA was implemented for a population of 200 individuals with a migration interval of 20 units. The maximum number of generations used was 600. The execution time required to identify optimal values was 25,7 s. The obtained value of the fitness function was 42237.1 W*h (0.011732 W*s). The speeds that allowed this value to be obtained were (as it can be seen in Figure 5) $x(1)=1029$, $x(2)=1000$ $x(3)=2000$, $x(4)=2000$ and $x(5)=2000$.

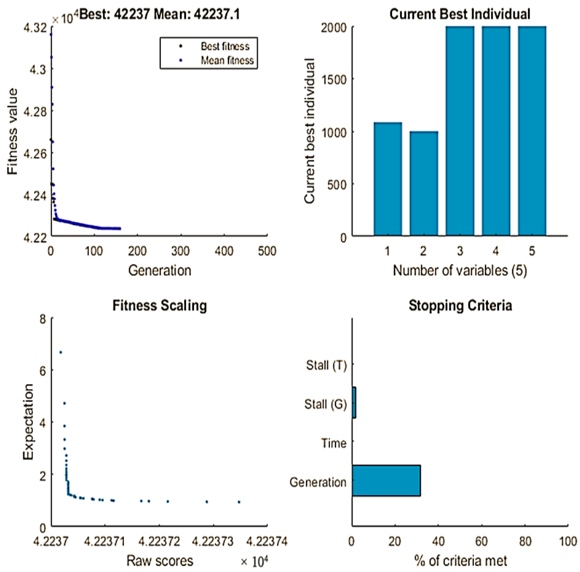


Figure 5. Application of a GA in MATLAB on the production flow for assembling the complex product

5.2 Optimization of Power Consumption Using IGA Algorithm

The optimization of the manufacturing process by means of the IGA algorithm involved using the obtained value to apply the algorithm with a single GA objective to energy consumption data and achieving a minimum satisfactory value for the user. The minimum value was later used for meeting the feasibility criterion of the GA optimizer, so that the optimizer could change the parameters of the GA algorithm and the performance of minimizing the value of power consumption for this algorithm remained unchanged while obtaining a minimum execution time for this algorithm.

The GA optimizer implemented upper and lower limits for each parameter of the GA algorithm. This ensured that the parameters of the GA algorithm remained within specified boundaries. Thus, the parameters of the GA algorithm have the following limits:

$$\begin{aligned} &100 =< Max\_Iter <= 1000, \\ &50 =< Max\_Pop <= 1000, \\ &0 <= \beta <= 1, \qquad 0 <= \gamma <= 1, \\ &0 <= \mu <= 1, \qquad 0 <= \nu <= 1 \end{aligned} \tag{11}$$

The GA optimizer exhibits specific characteristics that govern its operation within the GA framework. These characteristics include the number of analyzed variables, that is 7, the maximum number of iterations, that is 100, a population size of 1000 individuals, the selection coefficient with the value 1, the coefficient corresponding to the number of analyzed chromosomes with the value 1, the crossover coefficient with the value 0.1, the mutation coefficient with the value 0.1, and the mutation importance coefficient, with the value 0.1.

The outcomes achieved through the implementation of the enhanced GA were illustrated in Figure 6, showing notable improvements in execution time. These enhancements empower the algorithm for real-time data analysis derived from the power meters, enabling the adjustment of conveyor speeds based on changes in production times, all without requiring manual intervention from operators.

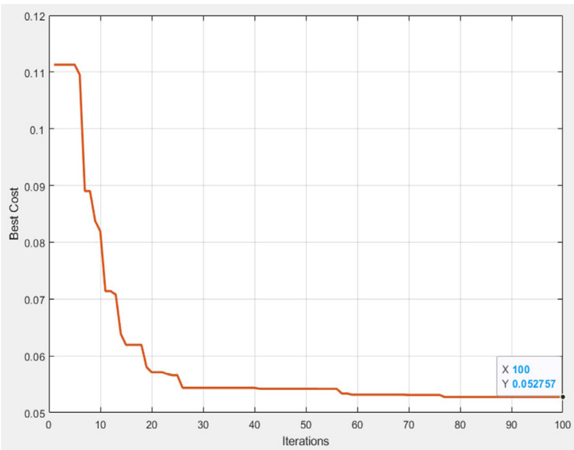


Figure 6. Representation of IGA convergence on a Cartesian scale

The original algorithm took 25.7 s to complete the optimization process, whereas the implementation of the IGA achieved a remarkable optimization time of just 0.0527 seconds (Figure 7). This significant improvement in execution time underscores the efficiency and effectiveness of the enhanced algorithm. However, it is important to note that achieving this level of improvement necessitated considerable effort in fine-tuning the algorithm. The process of refinement and optimization took nine hours. The best values obtained for the optimizer are: $(118.3, 51.1, 0.045, 0.9, 0.6, 0.46, 0.46)$. By means of these parameters, the velocity vector was identified: $(1000,\ 1000,\ 2000,\ 2000,\ 2000)$.

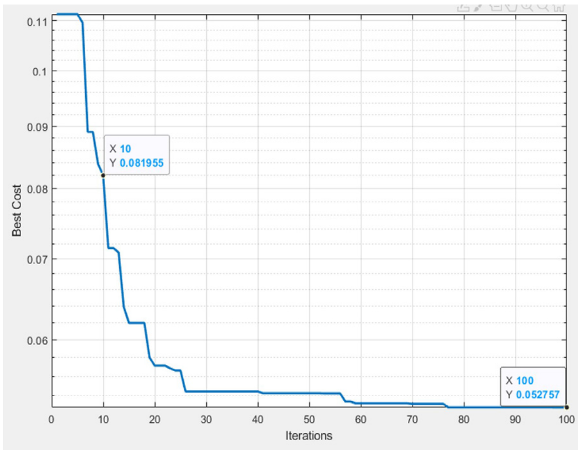


Figure 7. Representation of IGA convergence on a logarithmic scale

6. Conclusion

This research addresses the optimization of energy consumption in a manufacturing flow executed on a flexible manufacturing line. The utilization of a GA aims to optimize the fitness function's value, which represents the energy consumption throughout the entire manufacturing process. By applying the proposed GA optimization procedure, the minimum processing time for the standard GA single- objective function was found. The optimization procedure has been extended to a multi-objective GA approach, namely to the optimization of two variables: energy consumption and algorithm processing time.

In this paper, the IGA algorithm is approached as a two-layered optimization algorithm, in which the optimization of energy consumption is prioritized over the execution time for the algorithm. Thus, the energy consumption for the manufacturing flow for manufacturing a complex product is minimized to the value of 42237.1 W*h with an execution time of 0.0527 s, a very good time compared to the execution time for the standard NSGA-II algorithm, which is 25.7 s.

The use of the IGA algorithm made it possible to reach optimal speed values in a time-efficient manner, which were subsequently implemented in the production flow for assembling a complex product on the FML. As a result of the optimization time being much longer in the case of the NSGA-II algorithm in comparison with the IGA algorithm, the latter has been applied for real-time control of conveyor speeds. Its architecture, also highlighted in Figure 4, allowed to increase the speed of production, thus reducing energy consumption and the risk of blockages while also minimizing electricity costs.

This research contributes to the field of energy optimization in production by demonstrating the effectiveness of the IGA algorithm in obtaining significant energy reductions and the efficiency of the layered approach of optimization algorithms in obtaining optimal values in the shortest possible time. Further research will consider applying this approach to other optimization algorithms.

REFERENCES

Abdullah, Q. A., Humaidi, H. & Shahrom, M. (2020) Industry revolution 4.0: the readiness of graduates of higher education institutions for fulfilling job demands. *Revista Română de Informatică și Automatică (Romanian Journal of Information Technology and Automatic Control)*. 30(2), 15-26. doi: 10.33436/v30i2y202002.

Adel, A. (2022) Future of industry 5.0 in society: human-centric solutions, challenges and prospective research areas. *Journal of Cloud Computing*. 11(1), 40-55. doi: 10.1186/s13677-022-00314-5.

Alexandru, T. G., Alexandru, A., Popescu, F.D. & Andraș, A. (2023) The Development of an Energy Efficient Temperature Controller for Residential Use and Its Generalization Based on LSTM. Sensors. 23(1), 453. doi: 10.3390/s23010453.

Delorme, X., Battaïa, O. & Dolgui, A. (2014) Multi-objective Approaches for Design of Assembly Lines. In: Benyoucef, L., Hennet, J.-C. & Tiwari, M. K. (eds.) *Applications of Multi-Criteria and Game Theory Approaches*. London, UK, Springer-Verlag, pp. 31-56. doi: 10.1007/978-1-4471-5295-8_2.

Djebbar, A. M. & Boudia, C. (2022) A comparison of Artificial Bee Colony algorithm and the Genetic Algorithm with the purpose of minimizing the total distance for the Vehicle Routing Problem. *Revista Română de Informatică și Automatică (Romanian Journal of Information Technology and Automatic Control)*. 32(2), 51-64. doi: 10.33436/v32i3y202204.

Duca, O. G., Bidică, C., Mincă, E., Gurgu, V., Păun, M. A. & Dragomir, F. (2022) Optimization of Production Planning for a Flexible Assembly Technology on a Mechatronics Line. *Studies in Informatics and Control*. 30(1), 53-66. doi: 10.24846/v30i1y202105.

Filipescu, A., Mincă E., Filipescu, A. & Coandă, H. G. (2020) Manufacturing Technology on a Mechatronics Line Assisted by Autonomous Robotic Systems, Robotic Manipulators and Visual Servoing Systems. *Actuators*. 9(4),127. doi: 10.3390/act9040127.

Hassan, M. A., Kacem, I., Martin, S. & Osman, I. M. (2015) Genetic Algorithms for Job Scheduling in Cloud Computing. *Studies in Informatics and Control*. 24(4), 387-400. doi: 10.24846/v24i4y201503.

Knežević, D., Blagojević, M. & Ranković, A. (2023) Electricity Consumption Prediction Model for Improving Energy Efficiency Based on Artificial Neural Networks. *Studies in Informatics and Control*. 32(1), 69-79. doi: 10.24846/v32i1y202307.

Limpraptono, F. Y., Nurcahyo, E., Ashari, M. I., Yandri, E. & Yahya, J. (2021) Design of Power Monitoring and Electrical Control Systems to Support Energy Conservation. *Proceedings of the Pakistan Academy of Sciences*. 58(S), 1-7. doi: 10.53560/PPASA(58-sp1)726.

Ma, H., Zhang, Y., Sun, S., Liu, T. & Shan, Y. (2023) A comprehensive survey on NSGA-II for multi-objective optimization and applications. *Artificial Intelligence Review*. 56(12), 15217-15270. doi: 10.1007/s10462-023-10526-z.

Nicolae, A. E., Necula, H. & Carutasiu, M. B. (2023) Optimization of energy rehabilitation processes of existing buildings. *Scientific Bulletin Series C: Electrical Engineering and Computer Science*. 85(1), 327-338.

Serban, C. & Carp, D. (2017) A Genetic Algorithm for Solving a Container Storage Problem Using a Residence Time Strategy. *Studies in Informatics and Control. 26*(1), 59-66. doi: 10.24846/v26i1y201707.

Sun, J., Lin, J. & Meng, Z. (2022) Design of a Monitoring System for Power-consumption Devices Based on Modbus Protocol. *Journal of Physics: Conference Series*. 2405(1), 012022. doi: 10.1088/1742-6596/2405/1/012022.

Verma, S., Pant, M. & Snasel, V. (2021) A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems. *IEEE Access*. 9, 57757-57791. doi: 10.1109/ACCESS.2021.3070634.

Yunus, M. & Alsoufi, M. S. (2020) Multi-objective optimizations of mechanical characteristics of objects in computer aided SIS manufacturing process using empirical PSO algorithm. *Revista Română de Informatică și Automatică (Romanian Journal of Information Technology and Automatic Control)*. 30(3), 23-36. doi: 10.33436/v30i3y202002.