Optimized Component Selection for Onboard Electronics under Supply Constraints and Technical Compatibility Requirements using Genetic Algorithms

Lali Gugunishvili
SPbSUT
St. Petersburg, Russia
gugunishvililali02@gmail.com

Mark Demakov
Solanteq
St. Petersburg, Russia
markdemakov@mail.ru

Dmitry Levshun
SPC RAS
St. Petersburg, Russia
gugunishvililali02@gmail.com

***Abstract*—In the context of growing supply chain instability and stringent reliability requirements for embedded electronics, conventional manual component selection methods prove inefficient and error-prone. This paper introduces a systematic methodology based on genetic algorithms (GAs) to automate the selection of microcontrollers, sensors, and peripheral modules while enforcing technical compatibility constraints—such as voltage levels, power budgets, and communication interfaces—and incorporating real-time component availability from established global suppliers. The algorithm maintains a population of candidate parametric profiles, each encoding abstract requirements for functional component slots (MCU, sensor, power supply). These profiles are evolved through selection, crossover, and mutation until the highest-scoring configuration converges; the winning profile is then mapped to the closest real-world components available in the database. The approach is validated through physical implementation and laboratory testing of a functional prototype. Experimental results demonstrate strong correlation between the algorithm's predicted performance metrics and the actual characteristics of the assembled system, confirming the robustness and practical applicability of the proposed methodology for complex embedded system design under dynamic market conditions.**

I. INTRODUCTION

The design and deployment of reliable embedded electronic systems have grown increasingly complex in recent years, driven by demanding performance specifications, stringent size and power constraints, and evolving challenges in global component sourcing and supply chain resilience [1], [2]. International supply chain disruptions have severely limited access to previously standard electronic components, compelling engineers to seek functional alternatives under strict compatibility, reliability, and availability constraints. Traditional component selection methodologies—largely manual, experience-driven, and reliant on static datasheet comparisons—are no longer sufficient to address this multidimensional optimization challenge. The absence of systematic, automated tools for component substitution often leads to suboptimal designs, integration errors, project delays, and increased development costs [3].

In this context, the need arises for intelligent, algorithm-driven approaches capable of navigating the combinatorial space of available components while simultaneously satisfying heterogeneous engineering criteria—including electrical compatibility (voltage levels, current consumption, signal integrity), interface support (UART, I^2C, SPI, etc.), functional equivalence, logistical availability, and cost-efficiency. Moreover, such approaches must be adaptable to rapidly evolving restrictions and component obsolescence.

A. Genetic Algorithms as a Solution for Constrained Component Selection

Genetic algorithms (GAs), as a class of evolutionary optimization techniques, offer a promising solution to this problem. Their ability to explore vast, non-linear, and constrained search spaces without requiring gradient information makes them uniquely suited for component selection tasks, where the solution landscape is discrete, highly combinatorial, and subject to hard compatibility constraints. Unlike deterministic or rule-based systems, GAs can discover non-intuitive yet viable configurations—for instance, selecting a domestically produced sensor with non-standard interface logic, compensated by software adaptation or level-shifting circuitry—thereby enabling functional substitution even in the absence of pin-to-pin compatibility [5]–[7].

It is precisely this combinatorial intractability of exhaustive search that motivates the use of genetic algorithms as the optimization engine of this work. Exhaustive search evaluates every possible configuration exactly once, requiring $\mathcal{O}(\prod |C_j|)$ fitness evaluations—a number that grows multiplicatively with each additional component slot or database expansion. For onboard systems with four or more functional blocks and databases of hundreds of components per category, this renders brute-force search computationally intractable. The genetic algorithm, by contrast, operates on a population of fixed size and converges reliably within a bounded number of generations—typically 25 to 50 in the experiments presented here—regardless of the total search space size, achieving a solution quality comparable to exhaustive search at a fraction of the computational cost.

B. Scope and Validation Approach of This Study

This paper presents a GA-based methodology for the automated, multi-criteria single-objective selection of electronic components for onboard systems. The problem is multi-criteria by nature—involving voltage compatibility, supply-chain availability, performance, and interface compatibility—but single-objective by implementation: all criteria are combined into a single weighted composite fitness function whose weights sum to one. This design choice avoids the complexity of Pareto-front navigation while preserving full configurability of design priorities.

The proposed approach maintains a population of candidate parametric profiles—abstract specifications of what the ideal component for each functional slot should look like—and evolves them toward the highest-scoring composite configuration. Each profile is encoded as a chromosome whose genes specify target parameter values (e.g., desired voltage, required interface, minimum performance, availability zone) for each functional slot (MCU, sensor, power supply). Upon convergence, the best-scoring chromosome is mapped to the real-world component database: for each slot, the system selects the commercially available part whose measured parameters most closely match the evolved gene values.

The primary objective of this work is to demonstrate that a genetic algorithm can bypass exhaustive brute-force search—which becomes computationally infeasible as the number of component combinations grows combinatorially with database size—and instead converge to a high-quality system configuration in polynomial time. The GA approximates the optimal solution without exhaustive enumeration, making automated component selection tractable for real engineering workflows even as the component database scales to thousands of entries [8].

II. PROBLEM STATEMENT

Let $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$ denote the set of all available electronic components stored in the database, where each component c_i is characterized by a parameter vector $p_i = (v_i, a_i, \pi_i, \gamma_i)$ representing operating voltage, supply-chain availability, performance score, and interface compatibility, respectively.

A candidate system configuration is represented as a chromosome $X = (g_1, g_2, \ldots, g_m)$, where each gene g_j encodes the parametric requirements for functional slot j (e.g., target operating voltage, required interface type, minimum performance threshold, availability constraint). Here m is the number of required component slots (e.g., MCU, sensor, power regulator). The chromosome thus defines an abstract ideal parametric profile—not a specific commercial part—which the GA evolves toward optimality.

The feasible configuration space $\mathcal{X}$ contains only those chromosomes whose parametric profiles satisfy mandatory hard constraints—such as radiation hardness certification requirements—enforced as gatekeeping filters prior to fitness evaluation. Any chromosome failing these constraints is excluded regardless of its other scores.

The fitness of a configuration X is evaluated by a weighted composite function:

$$F(X) = \sum_i w_i \cdot s_i(X) - \sum_j p_j \cdot c_j(X)$$

where $s_i(X) \in [0, 1]$ are normalized scores for beneficial criteria (voltage compatibility, performance, supply-chain availability), $c_j(X) \in [0, 1]$ are penalty factors (interface mismatches, regulatory restrictions), and w_i, p_j are user-defined weights satisfying $\sum w_i + \sum p_j = 1$.

The optimization problem is formally stated as:

$$X^* = \arg\max_{X \in \mathcal{X}} F(X)$$

The cardinality of the exhaustive search space is $|\mathcal{X}| = \prod_{j=1}^{m} |C_j|$, where $C_j \subseteq \mathcal{C}$ is the pool of eligible components for slot j. This grows combinatorially with database size, rendering brute-force enumeration computationally infeasible. The genetic algorithm is employed to approximate X^* in polynomial time—bounded by population size × generation count—bypassing exhaustive search entirely.

III. METHODOLOGY

The proposed genetic algorithm operates on a structured dataset of real-world electronic components, including microcontrollers, sensors, communication modules, and peripheral devices. Initially, the system filters this dataset according to predefined engineering requirements—such as operating voltage, current consumption, interface compatibility, measurement accuracy, physical dimensions, and import/export availability.

This dataset is formally organized in a machine-readable format (e.g., JSON file), where each component is represented as a structured record with uniquely identified parameters. For scalability and performance, this structure is designed to be easily convertible into a relational or key-value database [13], [14], enabling efficient querying and dynamic constraint evaluation during the evolutionary process.

From this filtered component pool, the algorithm generates an initial population of artificial parametric profiles—chromosomes encoding abstract component requirements rather than specific commercial parts. Each chromosome specifies, for every functional slot (e.g., MCU, sensor, power regulator), a set of target parameter values: desired operating voltage, required interface type, minimum performance threshold, and availability zone. Gene values are drawn from the parameter ranges observed across the real component database, ensuring that the evolved profiles remain grounded in physically realizable specifications.

Through iterative application of selection, crossover [15], [16], and mutation operators, the algorithm evolves the population toward increasingly optimal parametric profiles. Upon convergence, the highest-ranked chromosome represents the algorithm's optimal abstract specification.

This optimal parametric profile is then mapped back to the component database via a similarity search: for each functional slot, the system identifies the real-world commercial component whose measured parameters most closely match the evolved gene values. This mapping step is critical—it bridges the gap between the abstract optimum found by the GA and the physically available, commercially sourced hardware that will be assembled into the final device.

Thus, the genome encoding strategy ensures that:

- All candidate solutions are grounded in real component specifications from the database.
- Compatibility and availability constraints are inherently respected during evolution.
- The final output—after mapping the optimal parametric profile to the component database—is a manufacturable, commercially sourced system configuration.

IV. RELATED WORK

Several approaches have been proposed for automated electronic component selection.

A common industrial approach is parametric search, implemented in electronic component databases and sourcing platforms [17]. Engineers specify parameters such as voltage, package type, tolerance, or interface type, and the system filters the component database accordingly. Although this method is computationally efficient, it does not consider system-level compatibility between multiple components and often requires manual evaluation of the filtered results.

Another approach uses clustering methods from unsupervised machine learning. Components are grouped according to similarity of their technical parameters (e.g., voltage, performance, or interface support). Components within the same cluster can be treated as potential alternatives. While clustering helps reduce the search space and identify similar components, it does not directly optimize complete system configurations [18], [19].

Heuristic optimization methods, such as Tabu Search, treat component selection as a combinatorial optimization problem and iteratively explore possible component combinations. These algorithms use memory structures to avoid revisiting previously explored solutions and help escape local optima. However, their effectiveness decreases as the number of possible combinations grows [20].

In contrast, the proposed approach employs a genetic algorithm to optimize parametric profiles of system components under multiple constraints. The algorithm evolves candidate profiles describing desired component parameters and evaluates them using a composite fitness function that incorporates electrical compatibility, performance, and supply-chain availability. After convergence, the optimal profile is mapped to the closest real components in the database, producing a practical system configuration.

V. FITNESS FUNCTION

The fitness function quantitatively assesses candidate parametric profiles—that is, chromosomes representing abstract component specifications—by evaluating how well their encoded parameter values satisfy a multidimensional set of engineering, logistical, and operational constraints. The problem is inherently multi-criteria: voltage compatibility, supply-chain availability, measurement performance, and interface compatibility must all be considered simultaneously. However, the implementation is deliberately single-objective: all criteria are aggregated into one composite scalar score through a weighted summation, where the weights are user-defined and constrained to sum to one. This design preserves the full richness of multi-criteria engineering judgment while keeping the optimization problem tractable and computationally efficient. The composite metric is expressed as follows:

$$F(x) = \sum_{i=1}^{n} w_i \cdot s_i(x) - \sum_{j=1}^{m} p_j \cdot c_j(x) \tag{1}$$

where x is a candidate configuration, $s_i(x) \in [0,1]$ are normalized scores for beneficial criteria (voltage compatibility, performance, supply-chain feasibility), $c_j(x) \in [0,1]$ are penalty factors (interface mismatches, regulatory restrictions), and w_i, p_j are user-defined weights ($\sum w_i + \sum p_j = 1$). Hard constraints—such as mandatory radiation tolerance—are enforced as gatekeeping filters applied prior to fitness evaluation, excluding non-compliant candidates regardless of other scores. This structure enables balanced optimisation across often-conflicting objectives while remaining adaptable to changing requirements.

For instance, voltage compatibility may be scored such that parametric profiles specifying an operating voltage closer to a target value—say, 3.0 V within an acceptable range of 2.3 V to 3.0 V—receive proportionally higher scores, thereby encouraging optimal power matching across the system architecture. Similarly, import availability and geopolitical compliance are integrated by assigning preferential weighting to profiles compatible with components from non-sanctioned suppliers (e.g., Chinese or neutral-market manufacturers), while penalizing or excluding profiles that would require components from restricted sources.

Radiation tolerance is enforced not as a tunable optimization parameter, but as a mandatory gatekeeping criterion: any parametric profile specifying requirements that no radiation-hardened component can satisfy—i.e., lacking verifiable total ionizing dose (TID) tolerance or single-event effect (SEE) immunity documentation—is automatically excluded from the solution space, regardless of its scores on other criteria.

A. *Composite Fitness Metric and Weighted Scoring System*

The overall fitness value is computed as a composite metric, combining normalized scores across all active criteria—including but not limited to electrical compatibility, supply chain feasibility, performance efficiency, and environmental resilience—each scaled by user-defined weights that reflect the relative importance of the corresponding parameter in the current design context. This flexible structure allows the genetic algorithm to adapt its search strategy to evolving project requirements, making it suitable not only for the immediate task at hand but also for future applications with expanded or re-prioritized constraints.

B. Practical Demonstration: Fitness Calculation Examples

To demonstrate the operation of the proposed fitness function, the following section presents fitness calculation examples for two alternative parametric profiles—that is, two candidate chromosomes evolved by the GA. Each profile specifies abstract parameter values for the sensor slot (voltage target, availability zone, performance level, interface compatibility). The fitness score is computed for the profile as a whole, representing how closely this abstract specification matches the engineering requirements. After the GA converges, the winning profile is mapped to its closest real-world component match. The following parameters were defined for the fitness function:

1. **Voltage** (V_s) — The closer the operating voltage is to 3.0 V, the higher the score.
2. **Availability** (A_s) — Components from Chinese or logistically accessible suppliers receive a bonus; components from sanctioned manufacturers are penalized.
3. **Performance** (P_s) — Evaluated based on key functional metrics, such as sensor accuracy or power efficiency.
4. **Compatibility Penalty** (C_p) — Applied if incompatibilities exist (e.g., logic level mismatch, unsupported interface, or required level-shifting circuitry), see Table I.

The fitness value for Configuration 1 (as a complete system) is computed as follows:

$$\begin{aligned}\text{Fitness} &= w_1 V_s + w_2 A_s + w_3 P_s - w_4 C_p \\ &= 0.4 \times 0.7 + 0.3 \times 0.2 + 0.2 \times 0.9 - 0.1 \times 0.0 \\ &= 0.28 + 0.06 + 0.18 - 0.00 = 0.52\end{aligned}$$

Where: $w_1 = 0.4$ (voltage compatibility), $w_2 = 0.3$ (import availability and geopolitical compliance), $w_3 = 0.2$ (performance), $w_4 = 0.1$ (compatibility penalty); $V_s = 0.7$ (3.3 V, slightly above ideal), $A_s = 0.2$ (low availability), $P_s = 0.9$ (high performance), $C_p = 0.0$ (no penalty).

In contrast, Table II presents Configuration 2—an alternative parametric profile which, while specifying lower measurement precision, demonstrates superior logistical accessibility and voltage compatibility.

The fitness value for Configuration 2 is computed as follows:

$$\begin{aligned}\text{Fitness} &= 0.4 \times 1.0 + 0.3 \times 1.0 + 0.2 \times 0.6 - 0.1 \times 0.3 \\ &= 0.4 + 0.3 + 0.12 - 0.03 = 0.79\end{aligned}$$

Notwithstanding its superior accuracy specification, Configuration 1—whose required component falls within a sanctioned-manufacturer availability zone—yielded a lower composite fitness score than Configuration 2. Configuration 2, with its optimal voltage compatibility and favorable geopolitical sourcing profile, represents the highest-scoring parametric profile under prevailing operational and regulatory conditions. After GA convergence, this winning profile is mapped to the real-world component database, and the commercially available part most closely matching its gene values—the JMCU-1464 sensor module—is selected as the final hardware recommendation [21].

TABLE I. CONFIGURATION 1 (CANDIDATE PARAMETRIC PROFILE)

Parameter	Value	Score (0–1)	Comment
Supply Voltage	1.8–3.6 V → selected	0.7	3.3 V is slightly above 3.0 V—acceptable, but not optimal
Availability	Sanctioned brand	0.2	Penalty due to import restrictions
Performance	Accuracy ±0.5 °C	0.9	High-precision sensor
Compatibility	Works with 3.3 V MCU—no conflicts	0.0	No compatibility penalties applied

TABLE II. CONFIGURATION 2 (CANDIDATE PARAMETRIC PROFILE)

Parameter	Value	Score (0–1)	Comment
Supply Voltage	2.4–3.0 V → selected	1.0	Perfectly matches target 3.0 V
Availability	Chinese origin, easy to import	1.0	Maximum bonus for supply chain accessibility
Performance	Accuracy ±1.0 °C	0.6	Lower accuracy compared to BME280
Compatibility	Operates with 3.0 V MCU but requires 3.3 V logic level → level shifter needed	0.3	Penalty applied due to interface mismatch requiring additional hardware

VI. SETUP OF THE EVOLUTIONARY OPTIMIZATION PROCESS

Experimental trials were conducted with population sizes ranging from 1,000 to 100,000 individuals. Each individual represents a candidate parametric profile—an abstract specification of component requirements—encoded as an ordered chromosome where each gene corresponds to one functional slot (e.g., sensor, ADC, voltage regulator) and encodes target parameter values for that slot (desired voltage, required interface, minimum performance, availability zone).

This representation enables the genetic operators—crossover and mutation—to efficiently explore diverse parametric combinations while ensuring that every evolved profile remains grounded in physically realizable parameter ranges drawn from the component database.

Three distinct implementations of the genetic algorithm were evaluated: NSGA-II (Non-dominated Sorting Genetic Algorithm II), SPEA2 (Strength Pareto Evolutionary Algorithm 2), and MOEA/D (Multi-objective Evolutionary Algorithm based on Decomposition). Although these algorithms are traditionally used for multi-objective optimization, in this work they are applied to a single-objective problem—each of their internal ranking mechanisms is mapped to the single composite fitness score $F(X)$. Analysis of the results revealed statistically insignificant differences between the algorithmic variants. Notably, increasing the population size did not enhance the accuracy of component selection; on the contrary, a gradual decline in selection precision was observed, likely attributable to overexploration of the parameter space and dilution of high-fitness profiles [22].

To maintain diversity in the population and avoid premature convergence to local optima, mutations were introduced: random changes to one or more gene values within a chromosome (e.g., shifting the target voltage, switching the required interface type, or adjusting the availability zone). The mutation probability was set to 5–10%, allowing the algorithm to explore novel parametric combinations while preserving overall stability.

Crossover (recombination) was also applied—combining gene values of two “parent” profiles to generate new “offspring” specifications. A single-point crossover was used: each parent chromosome was split at a random position, and the segments were exchanged. This enabled the inheritance of favorable parameter targets from high-performing profiles and accelerated convergence toward optimal configurations.

The algorithm demonstrated consistent convergence to high-quality parametric profiles by the 25th generation, attributable to a carefully balanced trade-off between exploitation—the refinement of promising profiles through crossover—and exploration—the introduction of novel parameter combinations via mutation. The stability of the approach was further validated through ten independent experimental runs.

In genetic algorithms, an excessively large population size may lead to reduced accuracy due to slower convergence and dilution of high-fitness individuals. However, this effect can be mitigated by increasing crossover and mutation probabilities, which introduce beneficial diversity and facilitate escape from local optima, provided these rates remain within reasonable bounds. Achieving high accuracy therefore does not lie in maximizing individual parameters, but in their balanced, task-specific tuning.

VII. EMPIRICAL VALIDATION: FROM ALGORITHMIC DESIGN TO HARDWARE IMPLEMENTATION

The true value of any design automation tool lies not in theoretical optimization, but in its ability to produce practical, real-world solutions that meet specified engineering criteria—particularly under constrained conditions such as import restrictions, interface mismatches, or limited component availability. In this work, the genetic algorithm was employed not merely as a computational experiment, but as a decision-support system for selecting an optimal configuration of electronic components, where “optimal” is defined not only by performance parameters but also by logistical feasibility and system-level compatibility.

This section presents the empirical validation of the algorithm’s output. The highest-scoring parametric profile generated by the GA was mapped to the real-world component database, and the closest-matching commercially available parts were identified for each functional slot. The resulting component configuration was then physically assembled into a functional prototype without modification or manual override. The prototype, designed as a compact onboard module for temperature and atmospheric pressure monitoring, serves as a representative subsystem of a spacecraft, chosen for its clear functional boundaries and well-defined interface requirements.

The prototype’s actual performance metrics (power consumption, measurement accuracy, stability) were measured and compared against the algorithm’s predictions: not to confirm numerical precision alone, but to verify whether the algorithmic model of compatibility, availability, and integration complexity accurately reflects real-world engineering constraints. This step transforms the GA from a theoretical optimizer into a validated engineering instrument capable of guiding component selection in environments where trial-and-error is prohibitively expensive or logistically impossible.

A. *Component Selection and Prototype Assembly*

In accordance with the highest-ranked parametric profile generated by the genetic algorithm, the following real-world components were identified through database mapping and assembled into a working prototype:

Microcontroller Unit (MCU): XD32FV103CBT6—a RISC-V based microcontroller, identified as the closest database match to the evolved MCU parametric profile (3.0 V nominal operating voltage, low power consumption, I^2C and SPI support, confirmed availability under current supply chain constraints).

Environmental Sensor: JMCU-1464—a multi-parameter sensor module (CO, NH_3, NO_2 detection + temperature and humidity), identified as the closest match to the evolved sensor profile (3.0 V operating voltage, Chinese-market origin, acceptable accuracy ±1.0 °C temperature, ±3 % RH humidity). As correctly anticipated by the GA, this component requires minor interface adaptation.

Level Shifter (as penalized in the fitness function): **TXB0104**—a bidirectional voltage-level translator, inserted between the XD32FV103 (3.0 V logic) and the JMCU-1464’s I^2C interface (which operates at 3.3 V internally). The need for this auxiliary component was correctly identified and penalized by the algorithm’s compatibility module ($C_p = 0.3$), validating the fitness function's ability to anticipate real-world integration requirements.

Power Supply: XC6206P302—a low-dropout regulator (LDO) providing a stable 3.0 V supply from a 3.7 V Li-Po battery, ensuring voltage compatibility across all components.

B. System Architecture and Functional Verification

The logical interconnection of the selected components is illustrated in Fig. 1.

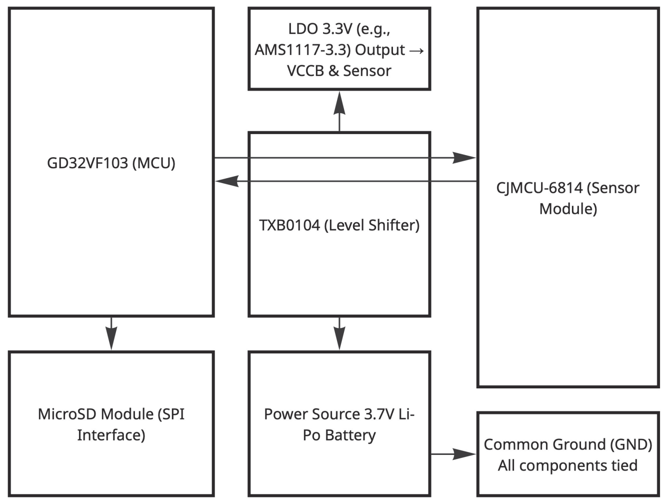

Fig. 1. Logical architecture of the assembled unit

The successful physical assembly of the prototype according to the genetic algorithm's top-ranked parametric profile—and the subsequent database mapping that identified the XD32FV103, JMCU-1464, and TXB0104—provides strong empirical validation of the methodology's practical utility. All selected components were confirmed to be commercially available, logistically accessible, and functionally compatible when integrated into a working system.

Crucially, the algorithm correctly anticipated and accounted for the interface mismatch between the 3.0 V logic of the MCU and the 3.3 V internal logic of the sensor, applying an appropriate compatibility penalty ($C_p = 0.3$) and, through the database mapping step, recommending the necessary level-shifting circuitry. This demonstrates that the fitness function effectively models real-world integration constraints, not merely idealized component specifications.

The prototype was assembled using a modular breadboard approach, confirming that the recommended configuration does not rely on custom PCB design or proprietary tooling—a critical advantage in resource-constrained environments. The system successfully established stable I^{2}C communication, executed basic sensor readout routines, and demonstrated correct data flow to the SD logging module via SPI, thereby verifying the algorithm's ability to generate functionally coherent and implementable solutions [23]–[26].

VIII. CONCLUSION

This study has demonstrated that a genetic algorithm can serve as a computationally tractable alternative to exhaustive brute-force search for the automated selection of electronic components in onboard systems. The core contribution is practical: where brute-force enumeration becomes infeasible as the component search space grows combinatorially, the GA converges to a high-quality configuration within a bounded number of generations—consistently by the 25th generation in the experiments presented here—regardless of database size.

The methodology operates in two stages. First, the GA evolves a population of abstract parametric profiles—chromosomes specifying target parameter values for each functional slot (MCU, sensor, power supply)—toward the highest composite fitness score. The problem is multi-criteria by nature, incorporating voltage compatibility, supply-chain availability, measurement performance, and interface compatibility, but single-objective by implementation: all criteria are aggregated into a single weighted fitness function whose weights sum to one, keeping the optimization tractable while preserving full engineering expressiveness. Second, the optimal parametric profile is mapped to the real-world component database, and the commercially available parts most closely matching the evolved gene values are selected as the final hardware configuration.

The fitness function's dynamic weighting and penalty mechanisms ensure that no single parameter dominates the selection process. Instead, the function reflects the complex trade-offs that engineers must make when ideal components are inaccessible—deliberately accepting minor performance compromises, such as reduced sensor accuracy, in exchange for guaranteed supply chain access and seamless system integration, as demonstrated in the comparative analysis of Configuration 1 and Configuration 2.

Experimental trials with population sizes from 1,000 to 100,000 individuals revealed that larger populations did not improve accuracy; on the contrary, they led to diminished selection precision due to overexploration and dilution of high-fitness profiles. This underscores that algorithmic performance is not a function of scale alone, but of thoughtful, context-sensitive parameter tuning—specifically, a mutation rate of 5–10% and single-point crossover.

Most significantly, the physical prototyping and functional testing of the algorithm's top-ranked configuration confirmed a strong correlation between predicted and measured system behavior. The successful assembly of the prototype—comprising the XD32FV103 microcontroller, JMCU-1464 sensor, and TXB0104 level shifter, all identified through the post-GA database mapping step—demonstrated that the algorithm correctly anticipated real-world integration challenges, including the necessity of auxiliary circuitry to resolve voltage-level mismatches. The fact that the compatibility penalty ($C_p = 0.3$) was applied precisely where needed, and that the resulting configuration functioned as intended without manual intervention, validates the fitness function's ability to model not just component specifications, but system-level engineering constraints.

Future work will focus on expanding the component database to include a broader range of sensors, communication modules, and power management ICs—particularly those emerging from accessible markets. Additionally, efforts will be made to integrate real-time market availability APIs, enabling the algorithm to dynamically adjust its recommendations based on live inventory and lead time data. Further adaptations will target multi-board or distributed aerospace systems, where inter-module compatibility and power distribution add additional layers of complexity. Ultimately, the goal is to achieve seamless integration of this methodology into industry-standard CAD and PLM environments—transforming the genetic algorithm from a standalone optimization tool into an embedded, intelligent assistant within the engineer's daily workflow.

Acknowledgment

The study was supported by the grant of the Russian Science Foundation No. 24-71-10095, https://rscf.ru/en/project/24-71-10095/.

References

[1] Levshun, D., Bakhtin, Y., Chechulin, A. & Kotenko, I. Analysis of attack actions on the railway infrastructure based on the integrated model. *International Symposium On Mobile Internet Security*. pp. 145-162 (2019)

[2] Levshun, D., Kotenko, I. & Chechulin, A. The integrated model of secure cyber-physical systems for their design and verification. *International Symposium On Intelligent And Distributed Computing*. pp. 333-343 (2019)

[3] D. Levshun, A. Chechulin, and I. Kotenko, "A technique for design of secure data transfer environment: Application for I2C protocol," in *Proc. 2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, 2018, pp. 789–794. DOI: 10.1109/ICPHYS.2018.8390807.

[4] S. Parameswari K., "Microcontroller Selection for Automotive Embedded Systems" (Dissertation), 2022. [Online]. Available: https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-321759

[5] G. Mamdoohi *et al.*, "Implementation of genetic algorithm in an embedded microcontroller-based polarization control system," *Engineering Applications of Artificial Intelligence*, vol. 25, 2021, pp. 869–873. DOI: 10.1016/j.engappai.2012.01.018.

[6] R. Haq, D. Purwanto, and R. Mardianto, "Microcontroller-Based Multi-Objective Genetic Algorithm for Mobile Robots Path Planning," in *Proc. 3rd Int. Conf. Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, Yogyakarta, Indonesia, 2023, pp. 122–126. DOI: 10.1109/ICE3IS59323.2023.10335453.

[7] P. Henrique de Oliveira Santos *et al.*, "Multi-Objective Genetic Algorithm Implemented on a STM32F Microcontroller," in *Proc. 2018 IEEE Congress on Evolutionary Computation (CEC)*, Rio de Janeiro, Brazil, 2018, pp. 1–7. DOI: 10.1109/CEC.2018.8477938.

[10] F. Ortega-Zamorano, J. M. Jerez, D. U. Muñoz, R. M. Luque-Baena, and L. Franco, "Efficient implementation of the backpropagation algorithm in FPGAs and microcontrollers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1840–1850, 2016. DOI: 10.1109/TNNLS.2015.2460991.

[8] S. Medeiros D. R., M. F. Torquato, and M. A. C. Fernandes, "Embedded genetic algorithm for low-power, low-cost, and low-size-memory devices," *Engineering Reports*, 2020. DOI: 10.1002/eng2.12231.

[9] J. Sawulski and M. Ławryńczuk, "Real-time optimisation of control strategy for a low fuel consumption vehicle engine using STM32 microcontroller: preliminary results," in *Proc. 18th European Control Conference (ECC)*, 2019, pp. 4246–4251. DOI: 10.23919/ECC.2019.8796299.

[11] M. Dendaluce *et al.*, "Microcontroller Implementation of a Multi Objective Genetic Algorithm for Real-Time Intelligent Control," in *Proc. Int. Joint Conf. SOCO'13-CISIS'13-ICEUTE'13*, Advances in Intelligent Systems and Computing, vol. 239, Springer, Cham, pp. 71–80. DOI: 10.1007/978-3-319-01854-6_8.

[12] H. Dong, Y. Liu, C. Yang, and M. Ao, "Coherent accumulation by using sequence shifting and genetic algorithm," *Optik*, vol. 138, pp. 137–144, 2017. DOI: 10.1016/j.ijleo.2017.03.080.

[13] Texas Instruments, "Microcontrollers (MCUs) & Processors." [Online]. Available: https://www.ti.com/microcontrollers-mcus-processors/products.html

[14] STMicroelectronics, "STM32 Microcontroller Selector." [Online]. Available: https://www.st.com/content/st_com/en/stm32-mcu-product-selector.html

[15] Y. Alghouass, B. Doerr, M. Krejca, and M. Lagmah, "Proven Approximation Guarantees in Multi-Objective Optimization: SPEA2 Beats NSGA-II," 2025, pp. 1–8. DOI: 10.48550/arXiv.2505.01323.

[16] L. Yuri, T. Abe, K. Yuta, and A. Claus, "MOEA/D with Adaptive Number of Weight Vectors," 2021, pp. 85–96. DOI: 10.1007/978-3-030-90425-8_7.

[17] Balueva, A., Desnitsky, V. & Ushakov, I. Approach to detection of denial-of-sleep attacks in wireless sensor networks on the base of machine learning. *International Symposium On Intelligent And Distributed Computing*. pp. 350-355 (2019)

[18] H. S. D. and S. S. Wenzel, "A literature review on machine learning in supply chain management," in *Artificial Intelligence and Digital Transformation in Supply Chain Management*, Proc. Hamburg International Conference of Logistics (HICL), vol. 27, pp. 413–441, 2019.

[19] S. B. Z. I. and P. P. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Emerging Artificial Intelligence Applications in Computer Engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[20] T.-K. Dao, T.-T. Nguyen, T.-D. Nguyen, T.-X.-H. Nguyen, and T.-T. Nguyen, "A review of the Tabu Search Algorithm for industrial applications," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 15, no. 3, pp. 179–186, 2024. DOI: 10.5890/JIHMSP.2024.240802.

[21] C. García-Martínez, F. Rodriguez, and M. Lozano, *Genetic Algorithms*. Springer, 2025, pp. 1–36. DOI: 10.1007/978-3-319-07153-4_28-2.

[22] L. Gugunishvili, M. Demakov, and D. Levshun, "Multi-Objective Optimization in Microcontroller Selection: A Comparative Analysis of Genetic Algorithms," in *Proc. Int. Russian Automation Conference (RusAutoCon 2025)*, 2025, pp. 290–295. DOI: 10.1109/RusAutoCon65989.2025.11177302.

[23] A. Akkaya and C. Közkurt, "Genetic Algorithm Research: A Bibliometric Analysis," 2024, pp. 1–23. Pioneer and Innovative Studies in Engineering.

[24] L. Guezouli, S. Ameghchouche, and H. Menzer, "Enhancing Green Vehicle Routing Efficiency Through Machine Learning-Optimized Genetic Algorithms," *Int. J. Netw. Distrib. Comput.*, vol. 13, no. 21, pp. 1–14, 2025. DOI: 10.1007/s44227-025-00062-3.

[25] D. Naaman, B. Ahmed, and I. Ibrahim, "Optimization by Nature: A Review of Genetic Algorithm Techniques," *Indonesian Journal of Computer Science*, vol. 14, pp. 268–284, 2025. DOI: 10.33022/ijcs.v14i1.4596.

[26] R. Abulail, "Enhanced Fitness Proportionate Selection Algorithm for Parent Selection in Genetic Algorithms," *J. Internet Serv. Inf. Secur.*, vol. 15, no. 1, pp. 257–270, 2025. DOI: 10.58346/JISIS.2025.I1.016.