Computational Intelligence Paradigms in Product Design Engineering: A Comprehensive Review

Atuha Digina Simon
Mbarara University of Science and Technology, Mbarara, Uganda.
dihinasimson3217@hotmail.com

Correspondence should be addressed to Atuha Digina Simon : dihinasimson3217@hotmail.com

Article Info
Journal of Computational Intelligence in Materials Science (https://anapub.co.ke/journals/jcims/jcims.html)
Doi: https://doi.org/10.53759/832X/JCIMS202402005.
Received 24 October 2023; Revised from 25 January 2024; Accepted 27 February 2024.
Available online 16 March 2024.

Abstract – Computational Intelligence (CI) models are established from biological paradigms and purpose to address complex challenges. Probabilistic methodologies and soft computing, which incorporate various models of CI, are typically employed in the domain of CI. Ontologies are fundamental in product design engineering process since they provide a common basis for incorporating various information sources. This research reviews five major models of CI: artificial intelligence, artificial neural networks, artificial immune systems, swarm intelligence, and evolutionary computation. The paper discusses the origins and applications of every paradigm, as well as its application in product design engineering. The study also reviews the functions of ontologies in the incorporation of information sources and facilitation of smart algorithms and techniques in the domain of product design engineering. In addition, it assesses the application of data mining, case-based reasoning, decision-making algorithms, hybrid techniques, qualitative reasoning, and process modeling in product design engineering. This article ends with an examination of modification, differentiation, customization, development, and building of process models within the field of product design engineering. It also reviews how CI approaches may be employed in addressing unique process challenges.

Keywords – Swarm Intelligence, Computational Intelligence, Fuzzy Systems, Artificial Neural Networks, Evolutionary Computing, Artificial Immune Systems.

I. INTRODUCTION

Product design (PD) refers to the process of creating an item based on design concepts, such as models, drawings, prototypes, or sketches. This process encompasses the whole lifecycle of the thing, including manufacturing, logistics, and marketing. The design process of a product has many stages, including detail design, product planning, product styling, product development, and concept design. Millward and Lewis [1] posits that product design places more emphasis on the act of creation and production, as well as on aesthetic and cultural evaluations, compared to the conventional focus of (mechanical) engineering. However, several publications fail to acknowledge the distinct contribution of industrial design within the broader framework of product design, sometimes seeing it as only one step in a complex, multi-step process. According to Kim, Joines, and Feng [2], industrial design (ID) is a part of product advancement process. They argue that engineers face more complex technical problems in their design activities, which require greater advancement effort compared to the problems addressed by industrial designers. Consequently, PD is often included as a part of the mechanical engineering (ME) program or taught separately at design colleges as a distinct kind of ID. Nevertheless, ID may have a greater influence on the process of PD, but this would need educational innovation and transformation.

Product design engineering is a multifaceted field that relies on and contributes to other disciplines, but lacks a well-established formal structure. The multidisciplinary nature of PD engineering has led to the development of several computational methods that have been documented in the exposition. The objective of this study is to examine the latest advancements in Computational Intelligence (CI) as they relate to PD, and to organize these CI methodologies into a comprehensive and cohesive framework. It is impossible to fully cover such a wide-ranging subject in a survey. An endeavor has been undertaken to achieve equilibrium between the level of specificity and the accessibility of literary references. The disparity in coverage is mostly attributed to the limited accessibility of information, rather than the significance of the subject matter.

The field of Product Development Engineering (PDE) aims to merge the formerly separate disciplines of ME and ID. Its goal is to cultivate individuals known as “integralists” who possess comprehensive knowledge and expertise in all aspects of the product advancement process. The origins of this emerging field may be traced back to Glasgow, Scotland in the late

1980s. The development of the manufacturing industry at that moment required a novel type of engineering professional who has expertise in both the fields of engineering and design. The introductory PDE course was established in the late 1980s as a result of the partnerships between art & design, and mechanical engineering departments at the Glasgow University. The principal objectives of the course were to connect two fields and put much emphasis of critical analysis of the product design process.

Many product engineers have conducted studies, established complex designs, and effectively created different goods, which are crucial for our daily routine. Product engineering expertise is in high demand among businesses across industries. This paper delineates the many tasks of product design engineers and provides guidance for anyone interested in pursuing a career in this field. Experienced manufacturers and analysts use computer-aided design (CAD) software to design a variety of applications and industrial systems. From the early stages of idea generation to the final stages of manufacturing the finished product, these experienced people play a role in every step of the manufacturing process and, moreover, their design must conform to all relevant industry standards and product specifications, while also meeting customer needs.

This article reviews CI models and their applications in engineering. Several CI models such as artificial immune systems, fuzzy systems, swarm intelligence, evolutionary computation, and artificial neural networks are the subject of this article This addresses the complex design and explores the application of probabilistic methods further than these models problems. The article also highlights the importance of ontologies in product development and technology design, as they facilitate the creation of intelligent algorithms and provide a convenient framework for integrating data sources and then displaying these insights use to create a value chain and have a more informed selection.

The rest of the article is arranged as follows: Fuzzy systems, swarm intelligence, evolutionary computation, innate immunity, and artificial neural network research are among the CI models presented in Section II. Section III focuses on the classification of CI tools, methods, and algorithms. In this section, ontologies, evolutionary computation, data mining, and decision-making, hybrid algorithms, qualitative reasoning, and case-based reasoning concepts are critically reviewed. Section IV focuses on the process perspective of product design engineering, focusing on typological characteristics of processes, selection of CI approaches, and evolutionary computations and process perspective. Lastly, Section V draws a conclusion to the research on the paradigms of CI within the field of product design engineering.

II. COMPUTATIONAL INTELLIGENCE PARADIGMS

This section examines five primary CI paradigms: NN, EC, SI, AIS, and FS. **Fig 1** provides a concise overview of the objective of the book. Probabilistic methodologies are often used with CI paradigms, as seen in the image. Soft computing, a phrase introduced by Hızıroğlu [3], refers to a distinct combination of paradigms, often including several computational intelligence paradigms and probabilistic approaches. The arrows signify the potential for merging approaches from several paradigms to create hybrid systems. The roots of each CI paradigm may be traced back to biological systems.

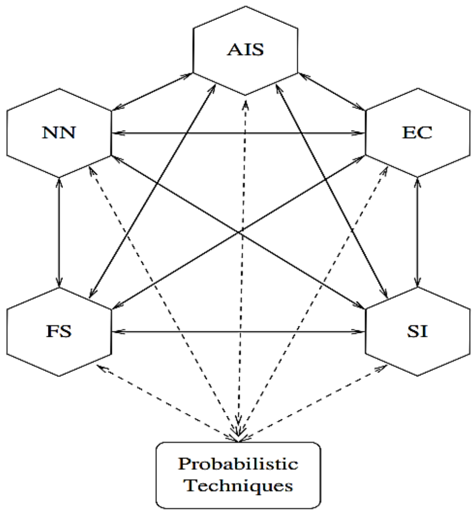

Fig 1. CI Paradigms

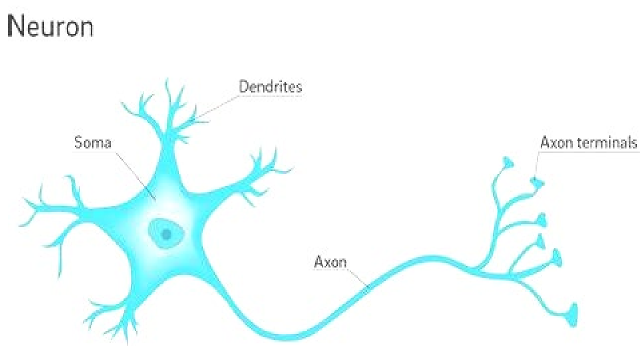

Fig 2. A Biological Neuron

NNs simulate the functioning of biological neural systems, EC simulates the process of natural evolution, including FS first appeared from studies on how creatures interact with their habitat, behavioral and genetic changes over time, AIS simulates the functioning of the human immune system, and SI simulates the organisms living in colonies or swarms social behavior.

Artificial Neural Networks

The brain functions as an intricate, non-linear, and parallel computational system. Neural systems can accomplish tasks like pattern recognition, vision, and motor control at a far quicker rate than computers, despite events occurring in the milliseconds for neural systems and nanosecond range for computers. Furthermore, the capacity to acquire knowledge, retain information, and make generalizations has led to investigations into biological brain systems computational simulations of biological systems of brain, referred to as ANN. The human cortex is believed to have between 10 billion to 500 billion neurons, and 50 trillion interconnection. Neurons are arranged into around 1000 primary sections, each containing around 500 NN. Will it then be feasible to accurately simulate the human brain? Currently unavailable. Present achievements in neural modeling mostly focus on tiny artificial neural networks designed to address particular tasks.

Solving problems with a single aim may be very straightforward by using neural networks of modest size, taking into account the limitations imposed by current computational storage and power capacity. The brain has the capacity to concurrently tackle several issues by using various regions of the brain in a dispersed manner. We have yet to make significant progress... Neurons are the fundamental components of biological neural systems. **Fig 2** demonstrates that a neuron is composed of an axon, dendrites, and a cell body. Neurons have extensive interconnectivity, with interconnections occurring between the dendrite of one neural and the axon of another neural. This link is often known as a connection. Indications go from the dendrites, via the cell structure, to axon, and then spread to all linked dendrites. An electrical impulse is sent to the axon of a neural just whenever the cell “fires”. A neuron has the ability to either suppress or stimulate a signal.

Artificial neuron (AN) is a computational representation of a BN. Every AN takes indications from the habitat or other ANs, collects and processes these inputs, and when activated, sends an output signal (OS) to all interconnected ANs. **Fig 3** depicts AN. The input signals (IS) are modulated either by inhibitory or excitatory effects, which are determined by positive and negative numerical weights assigned to each link to the AN. The regulation of an AN`s firing and the intensity of the OS are governed by a mathematical operation known as the activation function (AF). The AN gathers and processes all arriving inputs, calculating a net intake indication based on the corresponding weights. The net IS is used as the input to the AF, which computes the OS of the artificial neuron.

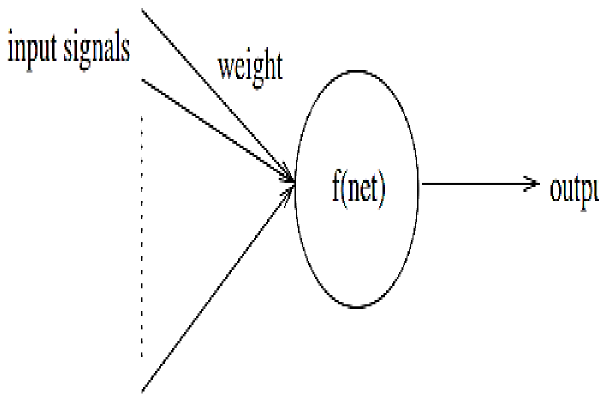

Fig 3. A diagram of AN

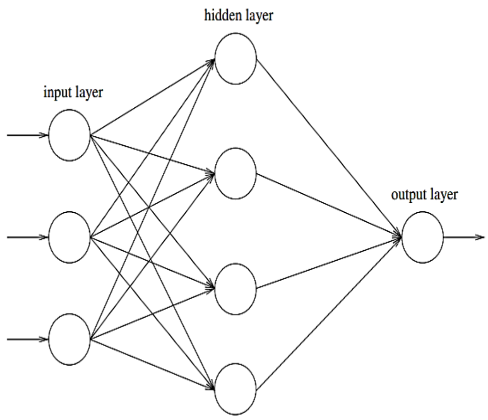

Fig 4. A diagram of ANN

An ANN is a hierarchical network composed of artificial neurons (ANs). A neural network (NN) typically has an input layer, single or multiple hidden strata, and an input stratum. The ANs in one stratum are interconnected, either completely or partly, with the ANs in the subsequent strata. It is also feasible to have feedback links to prior levels. A conventional neural network architecture is shown in **Fig 4**.

Evolutionary Computation

Evolutionary computing (EC) aims to replicate natural evolutionary processes, with a primary focus on the survival of the fittest, meaning that the weaker individuals are eliminated. Survival in natural evolution is attained via the process of reproduction. Offspring, derived from the genetic material of two or more parents, inherit a combination of traits from each parent, ideally including the most favorable attributes. Individuals that inherit unfavorable traits exhibit weakness and ultimately succumb in the struggle for survival. This phenomenon is seen in several avian species, when a single hatchling successfully acquires a greater quantity of nourishment, hence enhancing its physical prowess, ultimately resulting in the expulsion and subsequent death of its fellow siblings from the nest. Algorithms that evolve use a group of people, with each person being known as a chromosome. A chromosome determines the traits of people within a community. Every individual trait is denoted as a gene.

An allele is the term used to describe the value of a gene. Each generation involves people engaging in competition to create children. Individuals with superior survival qualities have a higher likelihood of reproducing. Crossover is the process by which offspring are produced by the combination of parental components. Every member of the population has the potential to experience mutation, which modifies certain alleles of the chromosome. An individual's survival strength is assessed by a fitness function that accurately represents the goals and limitations of the challenge at hand. Following each generation, people may experience culling, which involves the removal of some individuals, or they may survive and progress to the next generation, a phenomenon known as elitism. In addition, phenotypes, which include behavioral features, may have effect on the evolutionary process via two mechanisms: phenotypes can impact behavioral properties and genetic alterations can develop independently.

Swarm Intelligence

The SI emerged from the examination of swarms, or colonies, of social beings. Research on the interpersonal conduct of creatures in swarms has led to the improvement of very effective grouping and methods for enhancement. Research on simulations into the coordinated but unexpected movements of birds and animals have contributed to the development of the particle swarm optimization (PSO). Similarly, research on the foraging practices of ants has led to the creation of algorithms for ant colony enhancement. PSO is a probabilistic enhancement method that mimics the collective interpersonal of flocks of birds. Particle Swarm Optimization (PSO) is an algorithmic technique that involves a population-based search approach. In this techniques, the individuals, identified as particles, are arranged into a classification known as a "swarm." Every individual in the swarm symbolizes a potential optimization's solution issue.

Within a PSO, every fragment is navigated through a search area with several dimensions. The particle's location in the search space is modified based on its own experience as well as the experiences of nearby particles. Consequently, a particle utilizes both its own best position and that of its neighbors to move towards an optimal solution. The particles exhibit a behavior where they move towards an optimal solution while simultaneously exploring a broad region around the presently the best option. The evaluation of each particle's performance, namely its proximity to the world minimum, is determined by a pre-established fitness function that is directly linked to the issue being addressed. PSO has several applications such as approximating functions, clustering data, optimizing mechanical constructions, and solving systems of equations.

Innate Immune Systems

The innate immune system (IIS) has a remarkable capacity for pattern recognition, enabling it to differentiate between foreign cells (known as antigens or non-self) and the body's own cells (known as self). When the IIS comes into contact with an antigen, it demonstrates its adaptive characteristics by storing the antigen's structure. This allows for a quicker response to the antigen in the future. There are four models of the IIS that may be found in IIS research (see **Table 1**).

Table 1. Models of the Innate Immune System

Literature	Model	Description
Sattler [4]	Classical perspective of the immune system	The conventional perspective on the immune system is that it differentiates between the cells in the body and other foreign entities by using lymphocytes generated in the lymphoid organs. These cells acquire the ability to attach to antigens via a process of learning.
Ada and Nossal [5]	Clonal selection theory	The clonal selection hypothesis posits that active B-cells generate antibodies via a process of replication. The resulting clones undergo genetic mutations.
Aickelin and Cayzer [6]	Danger theory	The danger hypothesis posits that the IS has the capability to differentiate between antigens that are hazardous and those that are not hazardous.
Nk [7]	Network theory	Network theory assumes that B-Cells form a network. Upon encountering an antigen, a B-Cell undergoes activation and then triggers the activation of all other interconnected B-Cells in the network.

Fuzzy Systems

In traditional set theory, items are classified as either belonging to a set or not. Similarly, in reasoning with binary values, the parameters must have values that are either 0 or 1. The conclusion of an inferencing process also has similar restrictions. Human logic, however, is often not this precise. Typically, our reasoning and observations include a degree of uncertainty. For instance, people has the ability to comprehend the statement: "Certain Computer Science students have the capacity to code in the majority of programming languages." However, how can a machine accurately depict and engage in logical thinking with this particular piece of information? Fuzzy sets and fuzzy logic enable the process of approximation reasoning. Fuzzy sets give a degree of confidence to the belonging to a collection as an element. Fuzzy logic enables the process of drawing conclusions from ambiguous information, resulting in the derivation of new information, where each piece of information is assigned a specific level of confidence. Fuzzy sets and logic enable the representation of intuitive reasoning.

III. TAXONOMY OF CI TOOLS, ALGORITHMS, AND METHODS

Consensus among experts indicates that CI will make a significant contribution to design automation. Algorithms of machine learning integrate historical design data that is dispersed in time and place, resulting in a unified and comprehensible body of design knowledge. The only obstacle is in the uniform depiction of this data. The possible applications of CI in engineering for product design may be categorized into seven primary types. The classes are recognized and associated with the three categories shown in **Fig 5** of **Table 2**. Category 1 in **Fig 5** has been omitted from **Table 2** due to the computer code-advancement technique necessitating a comprehensive coverage that could not be included in this work. The selection of a CI technique for a certain category is determined by the extent to which the literature covers it. If a substantial gathering of studies was discovered, then a "×" was inserted in the corresponding item of **Table 2**. The next sections cover the texts related to each of the seven techniques listed in **Table 2**.

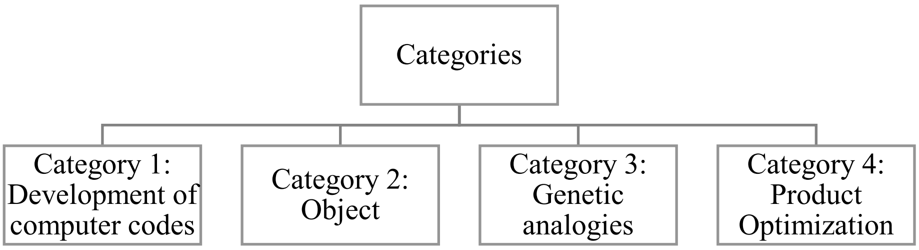

Fig 5. Categorization of designing techniques in engineering design of products

Table 2. CI techniques and investigation strategy groupings in **Fig 5**

	Category 2; Design Objects	Category 3; Genetic Analogy	Category 4; Optimization
Hybrid Approaches	x	x	x
Qualitative Reasoning			x
Case-Based Reasoning	x		x
Decision Making			x
Evolutionary Computation		x	
Data Mining	x		
Ontologies	x		

Ontologies

An ontology is a consensus-based collection of concepts and definitions that allows different parties to exchange heterogeneous information using a shared language. "Intelligent" algorithms and approaches use knowledge structured inside ontologies. The representation and nature of ontologies are crucial in the field of PD engineering. Ontologies serve as the formal foundations for techniques that characterize knowledge as intricate arrangements. They are essential for category 2 perspectives, as shown in **Table 2**. Ruíz and Hilera [8] used engineering feature ontologies to create software that verifies whether product configurations meet both organizational and physical limitations. Yoshioka et al. [9] created a model for knowledge-based engineering that heavily relies on ontologies of physical ideas. These ontologies serve as "pluggable" domain model inside the scheme. These ontologies of physical concepts serve as the shared foundation for integrating disparate knowledge sources.

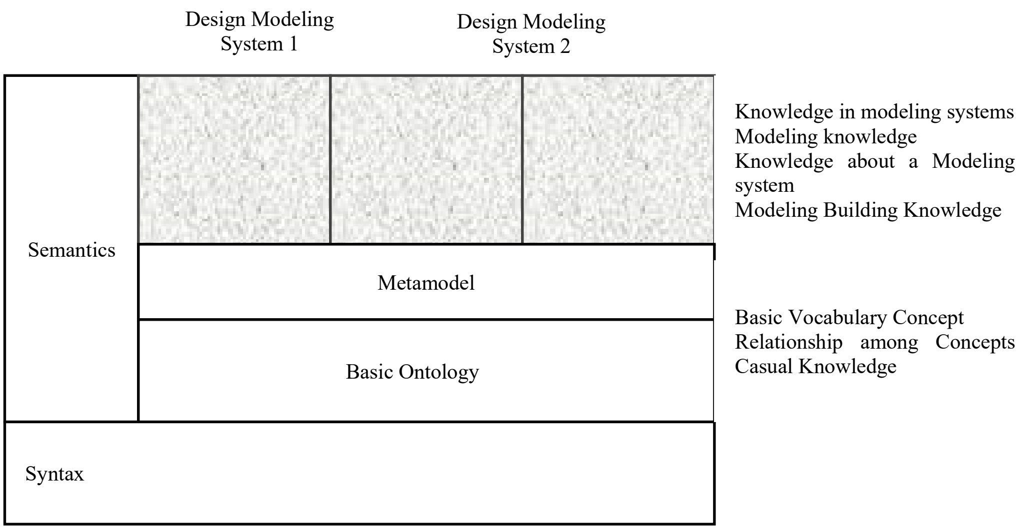

Fig 6. The engineering knowledge structure model

The engineering knowledge structure model proposed by Wielinga, Schreiber, and Breuker [10] was built based on our extensive research efforts. This framework assumes a hierarchical model seen in **Fig 6**. The bottommost layer delineates the syntax for representing knowledge. The higher levels include the semantics of knowledge, where both the integration of knowledge at the conceptual level and the integration of data at the operational level occur. The strata are partitioned into three sub-layers. The first foundational ontology layer offers a shared language (or set of terms) for representing essential physical concepts. These notions exhibit linkages, such as super-sub hierarchy and causality. The fundamental principles covered in this foundational layer are used to depict connections among design modeling structures. Integration of data level relies on the alignment of data across several design modeling tools. In order to do this, a model of concept of the design item is constructed at the foundational layer.

The design object is represented as a network of ideas in a conceptual model known as a 'metamodel'. The second intermediate layer encompasses knowledge pertaining to system modeling and the construction of models. This knowledge includes instructions on using various modeling systems, as well as information on the output and input requirements of these systems. This kind of information is referred to as “modeling knowledge.” The third higher layer pertains to the representation of system-specific information that is included inside each system of modeling. Moon et al. [11] used ontologies to construct and effectively implement a scheme for capturing and reusing understanding about item functionalities in a major electric firm. The authors said that a crucial aspect of their structure was its capacity to articulate the information that creators would normally rely on tacitly, and to facilitate the dissemination of this knowledge among team members. Kakabadse, Kouzmin, and Kakabadse [12] used ideologies to establish a Web-based repository that facilitates the decentralized creation of automobile factors. This was achieved by using regular Web technology and standards. Utilizing the Internet for these objectives enables people to contribute and explore material utilizing widely available and reliable mechanisms that are already established in other sectors.

Ding et al. [13] documented the creation of ontologies to enhance the exploration of design spaces using the semantic matrix notion of the Web. The study being discussed is a component of the Geodiseproject, which aims to develop a comprehensive web-based knowledge-based system (KBS) for the purpose of designing and optimizing processes related to fluid dynamics. Huang, Trappey, and Yao [14] created agent-based systems that use ontologies to facilitate processes of collaborative design. These systems do this by presenting users with a unified and integrated view of heterogeneous and scattered sources of information. The University of Toronto's company Intelligence Laboratory has created a comprehensive ontology-based system to model all aspects of a company, such as quality management, supply chain management, requirements, and more. Jurišica, Mylopoulos, and Yu [15] used ontologies to examine needs in the field of software for managing networks.

Data Mining

The amount of “legacy data” accumulated by the sector is increasing at an unusual pace. Processing and analyzing a substantial amount of data may be challenging, but it has the potential to provide significant insights. Data mining (DM), also known as knowledge discovery, offers techniques for efficiently exploring and condensing the existing data into a format that is easy to use. DM may be used with other methodologies to cultivate sophisticated systems. It is classified as category 2 (see **Table 2**) because it transforms create elements from the usually disorganized old information. Wang et al. [16] introduced a DM approach that utilizes past design data to forecast product cost. The decision-making information was extracted using a rough-set theory technique. Bansal and Priya [17] used data mining techniques to get information from design processes that included a computer-aided design (CAD) system. A technique known as prolonged dynamic programming was devised to retrieve the information.

Goncharenko, Kryssanov, and Tamaki [18] developed a design approach that utilizes information obtained from data related to product lifecycle. Fahmi, Kashyzadeh, and Ghorbani [19] assembly data and analyzed manufacturing of rotors for gas turbines to determine and measure the connections between vibration and balance data. This analysis led to enhancements in the design of component tolerances. Delen et al. [20] used a decision-tree method to facilitate product creation by examining top-level information, like culture of the manufacturing company, market position, philosophy, and strategy, as well as consumer behavior. The acquired information was intended to be incorporated into the product development process. La Rocca [21] used data mining techniques to extract expertise from disciplines associated with the design processes of microtechnological devices. Liou and Chen [22] use data mining methods to construct based on TRIZ, semantic portals that facilitate the reengineering of metal components into plastic.

Evolutionary Computation

Evolutionary computing (EC) refers to a category of global optimization approaches, which are inspired by the principles of natural growth. This technique begins by fostering the formation of a collective of individuals who actively address a certain issue. The initial population may be generated randomly using an algorithm. Individuals get a health assessment, and the efficacy of their approach is measured to determine how well they treat the issue. Furthermore, some operators, such as convergence, mutation, and replication, are subsequently applied to people who are impacted by natural evolution. A novel population is generated by considering the fitness values of recently evolved individuals. In order to preserve the population number in the ecosystem, some entities are eliminated. This approach is repeated until the termination requirements are met.

Several evolutionary computational approaches have been devised, such as evolutionary programming, genetic algorithms (GAs), genetic programming (GP), and evolutionary strategies (ES). The various methods are further upon in **Table 3**.

Table 3. Evolutionary computational approaches

Literature	Approaches	Description
Holland [23]	Genetic Algorithms	An approximate representation of the 30,000 genes that make up a human body may be shown as a vector consisting of the 4 letters (e.g., A, C, G, and T). If one chooses the binary form, they may also be represented ones and zeros in succession. Regardless of their phenotypic characteristics (appearance), any two persons only vary in a tiny proportion of their "genetic vectors."
Koza [24]	Genetic Programming	The GP develops a computer software using the plan programming linguistic as the answer. The authors examined the correlation between the efficiency and intricacy of the developed structures. Using strategies for statistical search, the author presented a unique strategy to GP by fusing a local parameter tweaking mechanism with a GP-based search with adaptation of tree topologies.
Juste, Kita, Tanaka, and Hasegawa [25]	Evolutionary Programming (EP)	EP, commonly referred to as evolutionary algorithms (EA), integrates the principles of the strongest survive and natural selection. EA manages structures` population, which are originally generated randomly. These structures develop by genetic operators, including selection, recombination, mutation, and survival.

Decision Making

Product engineering is a crucial component of makings choice. Sensible decision-making tools are particularly valuable in PD engineering because to the intricate nature of the choices involved and the potential hazards connected with making incorrect selections. The decision-making algorithms aim to maximize different design results, and as a result, they are classified under category 4 in **Table 2**. Griffin, Winfield, and Douglas [26] created a Knowledge-Based System (KBS) specifically designed for the purpose of selecting rolling element bearings. They used heuristic knowledge, with the assistance of a manufacturer's collection, to formulate a solution. Van Der Gaag et al. [27] presented a CI methodology known as signposting, which aids in decision-making throughout the design process. Signposting enhances one's understanding by emphasizing the interconnections between design factors, hence facilitating both strategic problem-solving and inference knowledge abilities.

To aid in decision-making during collaborative design, Gal et al. [28] created a decision-making network built on agents. Within a setting that encourages goal-based bargaining, a software agent represents each designer. Bousquet and Page [29] used agents to simulate the interactions across system design utilized by different teams working on intricate and extensive design challenges. The negotiation process between agents was automated using an evolutionary technique, which made it easier to share design solutions between different systems. Aliahmadi, Sadjadi, and Jafari-Eskandari [30] introduced a technique for constructing decision-making systems based on decision theory and use the value of data. This strategy considers the imprecision of data obtained from tests required to confirm important hypotheses for task completion, a common occurrence in PD engineering scenarios.

Case-Based Reasoning

Case-based reasoning (CBR) is a methodology that seeks to replicate the human ability to adjust and repurpose answers from familiar issues to unfamiliar ones. It presupposes that comparable issues can be resolved using comparable methods. A case is a concise representation of an issue together with its corresponding solution. Novel issues are examined and contrasted with established instances till an optimal correlation is identified. The answer of the complement case is used (and sometimes modified) to address the new challenge. The performance of CBR is optimized when the library of known instances meets two criteria: 1) each example accurately represents a specific class of frequent issues, and 2) each case exhibits some degree of resemblance to a few other instances in the collection. The primary functioning components of CBR consist of collecting and examining instances, generating a "similarity measure" for novel situations, and adjusting existing answers to address new challenges. CBR is classified as a plan object (category 2, **Table 2**) method based on the way the examples are organized, and it falls into category 4 (optimization) due to its spatial search features. CBR has been utilized in the development of software with proven solutions.

Case-Based reasoning (CBR) may be used in various contexts depending on how similarity is defined. Bartsch-Spörl, Lenz, and Hübner [31] used CBR in conjunction with grammars devoid of context to simulate human thinking in evaluating product requirements. Aamodt and Plaza [32] used CBR to simulate the deterioration of civil engineering structures, specifically focusing on infrastructure. They employed a substantial amount of data, consisting of a huge number of cases, to analyze the degradation and strength of these architectures. Bichindaritz [33] created a CBR approach specifically for the conceptual layout of structures. The system facilitates the ranking breakdown of design instances, provides several perspectives, and encapsulates the result of the design. There are several techniques for retrieving multiple cases, and the modification of cases is achieved by a "replay" approach of existing procedures. It is important to understand that adaptation often involves a parametric procedure that necessitates a model with parameters for the thing being developed. Intelligent systems might consider parametric models as their main focus.

Qualitative Reasoning

Qualitative reasoning enables the construction of models in situations where the connections between variables and parameters are not clearly defined. These approaches aim to find optimal solutions for simplified or abstracted scenarios, placing them in category 4 (see **Table 2**). Although they lack the ability to process intricate and specific "real life" data, they may provide basic guidance to design engineers. The qualitative reasoning technique is very compatible with the information that may be derived from the data sets. Bond graphs are very suitable for integrating process modeling components. The following methods are used to clearly define the behavior of components.

1) Utilization of a restricted quantity of adaptable universal phrases and symbols to create a logical visual framework that portrays the existence and interplay of influences affecting the dynamic functionality of the system.
2) Facilitating the easy creation and modification of the structure, which is crucial in the design of a creative system.
3) Utilize the model structure to systematically formulate a comprehensive and rational set of equations that are suited for computer`s system simulation.

Lamontagne and Plaza [34] outlined the use of bond diagrams inside systems of engineering, using a consistent set of ideal components. They also presented established methods for converting these elements into a simulation model. In their work, Itkonen, Ekman, and Kojo [35] examined the concept of bidirectional thinking as it relates to the field of engineering for product design. Further research in qualitative logic, like Frank's work [36], has focused on the use of engineering's use of qualitative physics. Analogical reasoning is one of the qualitative reasoning theories that have been studied in relation to product design engineering in these research (see [37]). Wałęga, Zawidzki, and Lechowski [38] created a non-numerical model of rigid-body physics using qualitative reasoning.

Hybrid Approaches

As various methodologies and procedures have developed fully, there has been a growing interest in integrating them. The combination of many methodologies has resulted in hybrid methods that may alleviate the limitations of the individual methods. Hybrid techniques include categories 2–4 in **Table 2**, since they integrate elements from all the main methods. Sriram [39] devised a hybrid approach that incorporates procedural, programming with objects, and production rules techniques to articulate heuristics in engineering in a blackboard knowledge-based system (KBS) for the purpose of constructing liquid retention structures. The system may provide guidance in both initial design and subsequent design phases. Abdullah et al. [40] devised a novel framework for processing of knowledge in mold design by integrating elements of modelling of product, neural networks, frame-based knowledge-based systems (KBSs), and case-based reasoning (CBR). It was observed that there was a significant improvement in design efficiency.

Tor, Britton, and Zhang [41] created a system that combines blackboard architecture with CBR to facilitate the planning of stamping process in developing die design. The system's value lies in its ability to use both prior data and alternative reasoning approaches in conjunction with Case-Based Reasoning (CBR). Several hybrid techniques belong to the realm of soft computing methods and are extensively studied in various conferences and publications. The fuzzy set theory is a significant factor contributing to the advancement of soft computing, as shown in [42] and [43]. Dam and Saraf [44] described a noteworthy use of a genetic algorithm (GA) for the purpose of designing neural network architectures. The integration of GA with neural networks has been shown to decrease the manufacturing applications and design complexity due to computing. The use of programming in evolution was employed to modify preexisting design solutions inside a CBR. It is stated that these "knowledge-based" methodologies have a wider range of applications compared to traditional case-based design approaches. Li and Huang [45] used the process of analytical hierarchy technique in conjunction with neural networks, fuzzy systems, and expert systems to create a tool for decision support for the design of adaptable manufacturing systems.

IV. PROCESS PERSPECTIVE OF PDE

The product advancement procedures may be seen as objects that share similarities with the goods they create. This section presents a classification of plan procedures based on the comparability between the operation features and the result. The classification in this segment pertains to the three primary groupings of **Table 4**.

The aforementioned features will comprise the collection of constructions put out in this study. This will serve as a crucial component of a cyber infrastructure for engineering product design.

Selection of CI Approaches

The typology described in Section IV-A might be used as a reference for choosing CI strategies. Let us investigate the following hypothetical business as an example: The organization is very experienced, with substantial corporate design expertise kept in many traditional databases, such as the CAD database. The organization aims to achieve incremental, platform-based modularity by using standardized interfaces, alternative design methodologies, and mass customization to solve the pointed out process challenges. The solutions will rely on modifications to steps aspects due to the exchange of procedural knowledge across groups via personnel moves. In regard to this, one may contemplate other CI approaches by examining the relationship between process features and the previously mentioned categories. This decision posits that the CI approaches pointed out in this manner would be most suitable for use by the firm.

Typological Process Features

Table 4. Typological features of the plan process

Characteristics	Description
Modularity	The primary and most evident distinction is that between platform-oriented and modular procedures, similar to product advancement. Modular procedures consist of pre-existing components that may be combined to form a complete process. The modules' functional nature suggests that optimization methods, namely those categorized as 4 in **Table 2**, are likely to be utilized. Platform-oriented procedures use shared foundations that are adapted to meet particular requirements.
Platform Orientation	Similar to the classification of product platforms outlined in [46], processes may be built using standardized components, fundamental components, shared architecture, and standardized interfaces. Procedures that rely on standard constituents are constructed by combining individual constituent procedures. The overall architecture of these operations is determined whenever a new procedure is created.
Differentiation	The standardized processes, referred to as such in this context, are derived from widely-used process models designed to achieve certain objectives, such as implementing best systems. Alternatively, processes might be either delayed differentiation operations or early differentiation alterations. This categorization focuses mostly on optimization, namely the identification of an ideal process that aligns with corporate and other limitations. As a result, it falls under category 4 of **Table 2**.
Modification	Operations may be usually adjusted for three primary motives. Alterations may be made to tailor the steps for usage in a different context. Steps may be modified for advancement, either gradually via constant progressively or through more drastic reengineering methods.
Customization	Processes may be classified based on the extent to which they are reused. Distinct procedures that are not intended to be reused, in contrast to the steps created following the concepts of mass customisation. Regarding the latter, our intention is to identify certain process characteristics in advance as variables that are likely to vary depending on how the procedure is implemented. Considering procedure components is similar to the design items belonging to category 2 (**Table 2**).
Construction	Processes may be distinguished based on the manner in which they are developed, similar to the recognized types of product design: redesign, variant design, or inventive design. All the strategies are classified as the fourth category (see **Table 2**) because the creation of the new procedure is mostly focused on achieving the desired goals. In addition, reverse and variant-engineered approaches may also be classified as category 3 if they rely on transformative similarities.
Evolution	Operations may also be characterized based on their temporal evolution. Both steps components and data elements may be considered as design objects, which is a defining attribute of the second category (see **Table 2**). If the particular changes throughout time use the transformative analogy, then these procedures fall under category 3.

Evolutionary Computations and Perspectives of Processes

Process modeling (PM) encompasses two concepts: 1) Parallel to the horizon or perpendicular to the vertical axis. 2) Perpendicular to the horizon or parallel to the y-axis. The development of a process model often involves both vertical and horizontal expansion, rather than being created at a single level. The topmost node in the hierarchy symbolizes the comprehensive procedure that is divided into more specialized parts. A network of activities is often found in the most comprehensive form (the horizontal notion). The use of notions from evolutionary computation will be used to support the horizontal concept in PM. The practicality of using evolutionary computing, namely Genetic Programming (GP), is shown in **Fig 7**.

The model shown in **Fig 7(c)** is the result of applying the crossover operator (CO) as in **Fig 7(a)** using the submodel as in **Fig 7(b)**. One of the numerous operators stated in GP that may be applied to engineering for product design is the CO shown in **Fig 7** (see, for instance, the operators given in [47]). The principles of evolutionary computation may be used to facilitate the horizontal idea of modeling process. The computational evolution usage in modeling of horizontal processes is shown via the utilization of three specific activity operators. 1) Focus on a certain area or field of expertise. 2) Formulate a general statement or principle that encompasses several instances or situations. 3) Undergo genetic mutation. To illustrate these operators, let's examine the structure shown in **Fig 8 (a)**.

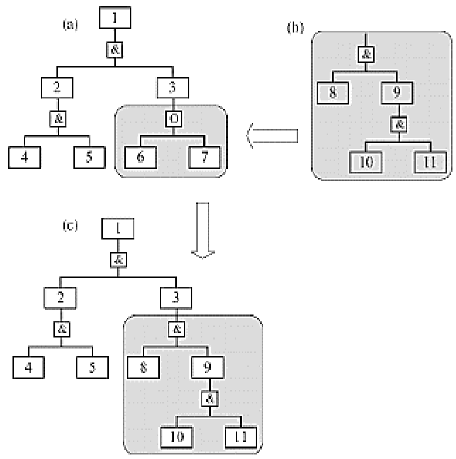

Fig 7. Crossover operator illustration in a model of procedure

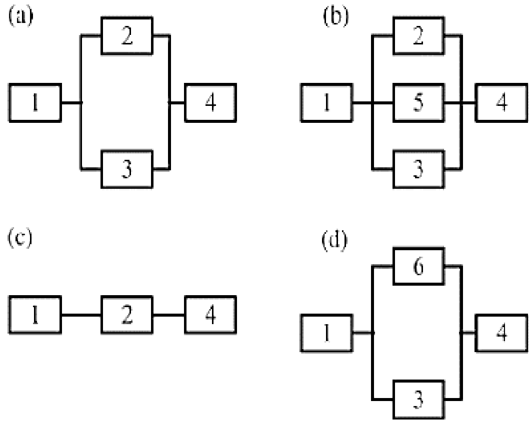

Fig 8. Operators in the model of activity. (a) Model of reference. (b) Generalization. (c) Specialization. (d) Mutation

Generalization operators modify the structure shown in **Fig 8 (a)** to create the structure shown in **Fig 8 (b)** by adding activity 5. Furthermore, the processes of mutation and specialization are shown in **Fig 8 (c)** and **(d)**. Furthermore, algebra may be used to define mechanisms, inputs, controls, outputs, and logical connections, in summation to the activity operators. A connection or an EXCLUSIVE would become an OR connector if the generalize operator was applied to it. Data mining is a technology that may enhance the autonomy of process models. As an example, a decision made by an algorithm of data mining may choose the route {1–2–3} in the model shown in **Fig 8 (a)**, relying on real-time data.

V. CONCLUSIONS

This study examines several paradigms of Computational Intelligence (CI) and their practical implementations in the field of product design engineering. The five major models discussed include EC, AIS, SI, ANN, and FS. The models are based on biological structures and seek to mimic the performance of natural processes. The research seeks to put more emphasis on ontologies in the field of product design engineering, as they establish a common basis for the incorporation of data sources, and allowing the establishment of smart algorithms and techniques. The paper also covers data mining approaches, which are capable of effectively evaluate and summarize existing information into formats, which can be effectively used. Evolutionary computing is a critical optimization approach, which draws inspiration from the concepts of natural development. This process entails creating a group of individuals to handle certain issues and employ operations such as selection, replication, mutation, and convergence to mimic natural evolution.

The examination of decision-making approaches integrates knowledge-based models, case-based reasoning, and agent-based networks. The approaches seek to optimize design outcomes and enhance insightful decision-making at each design process phase. The examination relates to the application of hybrid approaches and qualitative reasoning as a critical approach for establishing models and incorporating diverse processes and methodologies. The methodologies enhance the efficacy and efficiency of product design engineering. The article further categorizes the processes of product development into modularity, process perspectives, and platform orientations. The relevance of modification, differentiation, customization, development, and creation of process models in accomplishing particular objectives and adjusting to different contexts is emphasized. In general, the article provides a critical review of the different approaches and paradigms within the field of computational intelligence (CI) and their practical application in product design engineering, It concentrates on the relevance of employing these approaches to increase efficacy and productivity of the design process.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding was received to assist with the preparation of this manuscript.

Competing Interests

There are no competing interests.

References

[1]. H. Millward and A. Lewis, "Barriers to successful new product development within small manufacturing companies," Journal of Small Business and Enterprise Development, vol. 12, no. 3, pp. 379–394, Sep. 2005, doi: 10.1108/14626000510612295.
[2]. B. Kim, S. Joines, and J. Feng, "Technology-driven design process: teaching and mentoring technology-driven design process in industrial design education," International Journal of Technology and Design Education, vol. 33, no. 2, pp. 521–555, Mar. 2022, doi: 10.1007/s10798-022-09739-0.
[3]. A. Hızıroğlu, "Soft computing applications in customer segmentation: State-of-art review and critique," Expert Systems With Applications, vol. 40, no. 16, pp. 6491–6507, Nov. 2013, doi: 10.1016/j.eswa.2013.05.052.
[4]. S. Sattler, "The role of the immune system beyond the fight against infection," in Advances in Experimental Medicine and Biology, 2017, pp. 3–14. doi: 10.1007/978-3-319-57613-8_1.
[5]. G. L. Ada and G. J. V. Nossal, "The Clonal-Selection Theory," Scientific American, vol. 257, no. 2, pp. 62–69, Aug. 1987, doi: 10.1038/scientificamerican0887-62.
[6]. U. Aickelin and S. Cayzer, "The danger theory and its application to artificial immune systems," arXiv (Cornell University), Jan. 2002, [Online]. Available: https://arxiv.org/pdf/0801.3549
[7]. J. Nk, "Towards a network theory of the immune system.," PubMed, vol. 125C, no. 1–2, pp. 373–89, Jan. 1974, [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/4142565
[8]. F. Ruíz and J. R. Hilera, "Using ontologies in software engineering and technology," in Springer eBooks, 2006, pp. 49–102. doi: 10.1007/3-540-34518-3_2.
[9]. M. Yoshioka, Y. Umeda, H. Takeda, Y. Shimomura, Y. Nomaguchi, and T. Tomiyama, "Physical concept ontology for the knowledge intensive engineering framework," Advanced Engineering Informatics, vol. 18, no. 2, pp. 95–113, Apr. 2004, doi: 10.1016/j.aei.2004.09.004.
[10]. B. J. Wielinga, A. T. Schreiber, and J. Breuker, "KADS: a modelling approach to knowledge engineering," Knowledge Acquisition, vol. 4, no. 1, pp. 5–53, Mar. 1992, doi: 10.1016/1042-8143(92)90013-q.
[11]. S. K. Moon, T. W. Simpson, J. Shu, and S. Kumara, "Service representation for capturing and reusing design knowledge in product and service families using object-oriented concepts and an ontology," Journal of Engineering Design, vol. 20, no. 4, pp. 413–431, Aug. 2009, doi: 10.1080/09544820903151723.
[12]. N. K. Kakabadse, A. Kouzmin, and A. Kakabadse, "From tacit knowledge to knowledge management: leveraging invisible assets," Knowledge and Process Management, vol. 8, no. 3, pp. 137–154, Jan. 2001, doi: 10.1002/kpm.120.
[13]. L. Ding, P. Kolari, Z. Ding, and S. Avancha, "Using Ontologies in the Semantic Web: A survey," in Springer eBooks, 2007, pp. 79–113. doi: 10.1007/978-0-387-37022-4_4.
[14]. C. Huang, A. J. C. Trappey, and Y. Yao, "Developing an agent-based workflow management system for collaborative product design," Industrial Management and Data Systems, vol. 106, no. 5, pp. 680–699, Jun. 2006, doi: 10.1108/02635570610666449.
[15]. I. Jurišica, J. Mylopoulos, and E. Yu, "Ontologies for Knowledge Management: An Information Systems Perspective," Knowledge and Information Systems, vol. 6, no. 4, pp. 380–401, Jul. 2004, doi: 10.1007/s10115-003-0135-4.
[16]. J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," Journal of Manufacturing Systems, vol. 48, pp. 144–156, Jul. 2018, doi: 10.1016/j.jmsy.2018.01.003.
[17]. M. Bansal and Priya, "Machine Learning perspective in VLSI Computer-Aided Design at different abstraction levels," in Lecture notes on data engineering and communications technologies, 2021, pp. 95–112. doi: 10.1007/978-981-16-1866-6_6.
[18]. I. Goncharenko, V. V. Kryssanov, and H. Tamaki, "An agent-based approach for collecting and utilizing design information throughout the product life cycle," 1999 7th IEEE International Conference on Emerging Technologies and Factory Automation. Proceedings ETFA '99 (Cat. No.99TH8467), Jan. 2003, doi: 10.1109/etfa.1999.815353.
[19]. A.-T. W. K. Fahmi, K. R. Kashyzadeh, and S. Ghorbani, "A comprehensive review on mechanical failures cause vibration in the gas turbine of combined cycle power plants," Engineering Failure Analysis, vol. 134, p. 106094, Apr. 2022, doi: 10.1016/j.engfailanal.2022.106094.
[20]. D. Delen, H. Zaim, C. Kuzey, and S. Zaim, "A comparative analysis of machine learning systems for measuring the impact of knowledge management practices," Decision Support Systems, vol. 54, no. 2, pp. 1150–1160, Jan. 2013, doi: 10.1016/j.dss.2012.10.040.
[21]. G. La Rocca, "Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design," Advanced Engineering Informatics, vol. 26, no. 2, pp. 159–179, Apr. 2012, doi: 10.1016/j.aei.2012.02.002.
[22]. Y. Liou and M. Chen, "USING COLLABORATIVE TECHNOLOGY FOR TRIZ INNOVATION METHODOLOGY," Int. J. Electron. Bus. Manag., vol. 9, pp. 12–23, Jan. 2011, [Online]. Available: https://dblp.uni-trier.de/db/journals/ijebm/ijebm9.html#LiouC11.
[23]. J. H. Holland, "Genetic algorithms," Scientific American, vol. 267, no. 1, pp. 66–72, Jul. 1992, doi: 10.1038/scientificamerican0792-66.
[24]. J. R. Koza, "Genetic programming as a means for programming computers by natural selection," Statistics and Computing, vol. 4, no. 2, Jun. 1994, doi: 10.1007/bf00175355.
[25]. K. A. Juste, H. Kita, E. Tanaka, and J. Hasegawa, "An evolutionary programming solution to the unit commitment problem," IEEE Transactions on Power Systems, vol. 14, no. 4, pp. 1452–1459, Jan. 1999, doi: 10.1109/59.801925.
[26]. R. T. Griffin, M. J. Winfield, and S. S. Douglas, "Bearing selection using a knowledge based system," in Tribology series, 1989, pp. 489–493. doi: 10.1016/s0167-8922(08)70227-x.
[27]. A. Van Der Gaag, R. Jago, A. Gallagher, K. Stathis, M. Webster, and Z. Austin, "Artificial intelligence in Health Professions Regulation: An exploratory qualitative study of nurse regulators in three jurisdictions," Journal of Nursing Regulation, vol. 14, no. 2, pp. 10–17, Jul. 2023, doi: 10.1016/s2155-8256(23)00087-x.
[28]. Y. Gal, B. J. Grosz, S. Kraus, A. Pfeffer, and S. M. Shieber, "Agent decision-making in open mixed networks," Artificial Intelligence, vol. 174, no. 18, pp. 1460–1480, Dec. 2010, doi: 10.1016/j.artint.2010.09.002.
[29]. F. Bousquet and C. L. Page, "Multi-agent simulations and ecosystem management: a review," Ecological Modelling, vol. 176, no. 3–4, pp. 313–332, Sep. 2004, doi: 10.1016/j.ecolmodel.2004.01.011.
[30]. A. Aliahmadi, S. J. Sadjadi, and M. Jafari-Eskandari, "Design a new intelligence expert decision making using game theory and fuzzy AHP to risk management in design, construction, and operation of tunnel projects (case studies: Resalat tunnel)," The International Journal of Advanced Manufacturing Technology, vol. 53, no. 5–8, pp. 789–798, Aug. 2010, doi: 10.1007/s00170-010-2852-7.
[31]. B. Bartsch-Spörl, M. Lenz, and A. Hübner, "Case-Based Reasoning – survey and future directions," in Lecture Notes in Computer Science, 1999, pp. 67–89. doi: 10.1007/10703016_4.
[32]. A. Aamodt and E. Plaza, "Case-Based reasoning: foundational issues, methodological variations, and system approaches," Ai Communications, vol. 7, no. 1, pp. 39–59, Jan. 1994, doi: 10.3233/aic-1994-7104.
[33]. I. Bichindaritz, "Memory Structures and Organization in Case-Based Reasoning," in Studies in computational intelligence, 2008, pp. 175–194. doi: 10.1007/978-3-540-73180-1_6.
[34]. L. Lamontagne and E. Plaza, Case-Based Reasoning Research and development. 2016. doi: 10.1007/978-3-319-47096-2.
[35]. M. Itkonen, K. Ekman, and I. Kojo, "Murjottelu – interdisciplinary training campaign for industrial design and engineering students," European Journal of Engineering Education, vol. 34, no. 3, pp. 263–271, Jun. 2009, doi: 10.1080/03043790903038858.

[36]. P. M. Frank, “Analytical and qualitative model-based fault diagnosis – a survey and some new results,” European Journal of Control, vol. 2, no. 1, pp. 6–28, Jan. 1996, doi: 10.1016/s0947-3580(96)70024-9.

[37]. H. Cheong, G. M. Hallihan, and L. H. Shu, “Understanding Analogical Reasoning in Biomimetic Design: an Inductive approach,” in Springer eBooks, 2014, pp. 21–39. doi: 10.1007/978-94-017-9112-0_2.

[38]. P. A. Wałęga, M. Zawidzki, and T. Lechowski, “Qualitative Physics in <italic>Angry Birds</italic>,” IEEE Transactions on Computational Intelligence and AI in Games, vol. 8, no. 2, pp. 152–165, Jun. 2016, doi: 10.1109/tciaig.2016.2561080.

[39]. R. D. Sriram, Intelligent Systems for Engineering: a knowledge-based approach. 2012. [Online]. Available: http://ci.nii.ac.jp/ncid/BA32026755

[40]. M. S. Abdullah, C. Kimble, R. F. Paige, I. Benest, and A. Evans, “Developing a UML profile for modelling Knowledge-Based Systems,” in Lecture Notes in Computer Science, 2005, pp. 220–233. doi: 10.1007/11538097_15.

[41]. S. B. Tor, G. A. Britton, and W. Zhang, “Development of an object-oriented blackboard model for stamping process planning in progressive die design,” Journal of Intelligent Manufacturing, vol. 16, no. 4–5, pp. 499–513, Oct. 2005, doi: 10.1007/s10845-005-1660-7.

[42]. S. Datta and P. Chattopadhyay, “Soft computing techniques in advancement of structural metals,” International Materials Reviews, vol. 58, no. 8, pp. 475–504, Nov. 2013, doi: 10.1179/1743280413y.0000000021.

[43]. A. Dineva et al., “Review of soft computing models in design and control of rotating electrical machines,” Energies, vol. 12, no. 6, p. 1049, Mar. 2019, doi: 10.3390/en12061049.

[44]. M. Dam and D. N. Saraf, “Design of neural networks using genetic algorithm for on-line property estimation of crude fractionator products,” Computers & Chemical Engineering, vol. 30, no. 4, pp. 722–729, Feb. 2006, doi: 10.1016/j.compchemeng.2005.12.001.

[45]. T.-S. Li and H.-H. Huang, “RETRACTED: Applying TRIZ and Fuzzy AHP to develop innovative design for automated manufacturing systems,” Expert Systems With Applications, vol. 36, no. 4, pp. 8302–8312, May 2009, doi: 10.1016/j.eswa.2008.10.025.

[46]. T. W. Simpson, “Methods for optimizing product platforms and product families,” in Springer eBooks, 2006, pp. 133–156. doi: 10.1007/0-387-29197-0_8.

[47]. A. Kusiak and F. A. Salustri, “Computational intelligence in Product Design Engineering: Review and Trends,” IEEE Transactions on Systems, Man and Cybernetics, vol. 37, no. 5, pp. 766–778, Sep. 2007, doi: 10.1109/tsmcc.2007.900669.